OPEN LITHICS: APPLYING OPEN SOURCE TECHNOLOGIES TO PROBLEMS IN LITHIC USE WEAR EXPERIMENTATION AND ANALYSIS

by

NICHOLAS HANS WABER

BA, University of British Columbia, 2006

MA, University of Victoria, 2012

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Anthropology)

The University of British Columbia

(Vancouver)

February 2020

© Nicholas Hans Waber, 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, a thesis entitled:

Open lithics: applying open source technologies to problems in lithic use wear experimentation and analysis

submitted by Nicholas Hans Waber in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

in Anthropology

Examining Committee:

David Pokotylo, Anthropology
Supervisor

Michael Blake, Anthropology
Supervisory Committee Member

Zhichun Jing, Anthropology
Supervisory Committee Member

Susan Rowley, Anthropology
University Examiner

Kevin Fisher, Classical, Near Eastern, and Religious Studies
University Examiner

J. Mark Kenoyer
Additional Examiner

Abstract

This dissertation presents and examines open-source methods and technologies for recording and quantifying aspects of gesture in lithic use experiments, as well as measuring and mapping edge damage on stone tools. It has three interrelated parts. The first presents two tools for quantifying components of gesture in lithic experiments: 1) OpenHaft, an electronic handle designed to measure and record the load exerted on a stone tool's edge during use, and 2) cumulative stroke distance (CSD), to calculate the total volume of tool use during a task. An experiment using CSD and OpenHaft shows both distance and load to be major factors affecting wear on stone tools.

The second part applies a suite of GIS (geographic information system) tools to precisely quantify and map "macrowear"—the chipping and flaking damage from tool use. 3D digital photogrammetry was used to produce digital surface models (DSM) of experimental microblades, that were compared using custom scripts for QGIS, yielding a precise record of the volume and location of material wear. This allows researchers to observe edge attrition throughout a tool's use life. This is relevant to archaeological questions around tool design, curation, maintenance, and discard.

The third part examines the viability of the QGIS edge wear analysis method on archaeological lithics based on an adaptation of Kuhn's (1990) geometric index of unifacial reduction flaking (GIURF). To apply GIURF to archaeological lithics, it is necessary to virtually reconstruct the artifact in its pristine state, which this study attempts to do. Based on the experiment, GIS-automated GIURF does not yet virtually reconstruct worn lithics with enough accuracy for precise wear quantification, but successful reconstruction of some sections of worn microblades indicates promise for further development.

OpenHaft/CSD and the QGIS wear quantification modules help lithic analysts to perform experiments with robust control over gesture variables, better defining what is a "stroke", and precisely measure the effects of stroke variation on stone tools. The project applies the same GIS-based wear measurement methods to archaeological samples and offers a roadmap for future development. These open source techniques and technologies provide researchers with an

accessible toolkit for better lithic experimentation and objective macrowear quantification and analysis.

Lay Summary

When stone tools are used for various tasks, those tools are subjected to various forms of damage or wear. By studying wear on stone tools, archaeologists gain insight into the daily activities of past peoples. I apply free, open-source methods and technologies—both software and hardware—to the archaeological study of wear on stone tools. First, I present a digital handle that measures how much force is exerted on a stone tool's edge during use, and a method of recording how long a tool is used in experiments. These record the actions that result in edge wear. Second, I use digital mapping software to locate and measure material that is worn from experimental stone tools during use. Finally, I test the digital mapping method in simulated archaeological contexts where the un-worn state of the artifact is unknown. These tools and techniques help improve archaeological experimental methods and stone tool analysis.

Preface

This dissertation is original, independent work by the author, Nicholas Waber. This work is unpublished as of the date of submission. Chapters 2, 3, and 4 are all conceived of, and presented as inter-related but independent papers. Chapter 2, "OpenHaft and Cumulative Stroke Distance: Towards Gesture Quantification in Lithic Use Wear Experiments", and Chapter 3, "Open Source 3D Digital Photogrammetry and GIS for Quantifying Edge Damage on Experimental Stone Tools", have been submitted for publication. Any errors are my own.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This dissertation is the culmination of years of encouragement and support from many people.

Thank you to my dissertation committee, whose guidance and encouragement was crucial to this project. To Dr. David Pokotylo, my doctoral supervisor and committee chair, for tremendous support and mentorship, along with tireless editing (and endless patience!), throughout my doctoral studies. To Dr. Michael Blake for pushing me to view my work in the "big picture" context. To Dr. Zhichun Jing, whose insightful critiques and probing questions strengthened my research. Thank you also to the external examiners, Dr. Sue Rowley, Dr. Kevin Fisher, and Dr. Mark Kenoyer, whose fresh perspectives and helpful comments improved the final product.

Thank you to my cohort members and friends in the Department of Anthropology, whose humour and enthusiasm buoyed me on many occasions. Special thanks go to Huma for ceaseless encouragement (and promise of hiking once work got done), and Megan, who demonstrated the research potential of OpenHaft. Thank you also to Morgan Ritchie, who kept archaeology fun when my motivation was at its lowest.

Thank you to the Open Source community whose principles of actively sharing knowledge, skills, and enthusiasm were central to enabling and inspiring my research. Also to Shana, Dan, Kevin, and Terra Archaeology, for being extraordinarily accommodating during the writing process.

Most of all, thank you to my family for support and encouragement extending far beyond my studies. To my wife, Christine, who inspired me, kept me going, and endured years of being an impromptu editor, sounding board, and "thesis widow". To my mother, who delivered much-needed and timely cheerleading (and the occasional kick in the ass). To my father, who inspired and fostered so much of my curiosity about the past, and who I wish could have seen the final product. To my brother, Alex, for getting me away from my studies every now and then. Thanks also to my grandmother, uncles, and aunts, and mother- and father-in-law for all their encouragement throughout the process.

Formal financial support for my research came from the Social Sciences and Humanities Research Council (Doctoral Fellowship), and the University of British Columbia (Four Year Fellowship; Charles and Alice Borden Fellowship for Archaeology).

Dedication

To my wife, Christine, and my parents, Dietmar and Angela Waber.

Chapter 1 Introduction: Towards a Quantitative Description of Lithic Use-life Histories

1.1 Lithic Use Wear, Tool Function, and the Problem of Quantification

Lithic use wear analysis is a classic example of Middle Range Theory in archaeology: experimentally generated wear on lithic replicas serves as an analogue and reference collection for the identification and characterization of wear on archaeological lithic artifacts (Akoshima 1987; Clark and Woods 2014; Lin et al. 2017). While this methodological model has provided useful resources for the broad categorization of modes of ancient tool use (e.g., Akoshima 1987; Akoshima and Kanomata 2015; Kay and Mainfort 2014; Semenov 1957; Stevens et al. 2010), robust and consistent identification of specific task applications on the basis of lithic use wear—especially specific contact materials processed during an activity—remains an elusive target (Clark and Woods 2014; Odell and Odell-Vereecken 1980). This identification of lithic uses has traditionally been the realm of the specialist tasked with peering through a microscope and making qualitative pronouncements based on personal experience and expertise with experimental lithics (Newcomer et al. 1986; Odell and Odell-Vereecken 1980). The logical response to the traditionally subjective nature of most qualitative assessments of lithic use wear has been a growing interest in attempts to objectively quantify wear on stone tools. This, along with the rapid development and spread of sophisticated imaging and surface modelling technologies in recent years has resulted in a resurgence of lithic use-wear studies. A variety of 3D microscopy techniques permit researchers to record high-precision digital models of lithic surfaces in order to quantify use wear with previously impossible levels of accuracy and objective replicability (e.g., Evans et al. 2013; Evans and Donahue 2008; Macdonald 2013; Stemp 2013). Researchers can now precisely measure and describe the often-microscopic changes that a lithic surface undergoes in the course of an experimental task, and they can directly apply robust and objective metrics to the analysis of archaeological lithic artifacts.

However, the recent spate of high-tech use wear studies has not proven to be a panacea for lithic analysts. While the micron-scale precision of 3D digital models and ISO (International

Organization for Standards) surface roughness quantification methods adopted from engineering metrology lend a scientific lustre to the new techniques, they have also resulted in inconsistent experimental protocols and reporting standards. This latter problem is compounded by the limited availability of, and access to, hyper-specialized equipment that in turn compromises our ability to replicate many of the new studies' results, and to adopt the methods in our daily analytical practices. In the case of experimental protocols, with few exceptions (e.g., Iovita et al. 2017; Key 2013; Key et al. 2014; Pfleging et al. 2015; Stemp et al. 2015), the specifics of gesture—i.e., how tools are held and angled during use—have been all but neglected in use-wear studies. These few gesture-centric studies advance the science of experimental lithic analysis considerably, focusing on aspects of lithic tool use such as grip and prehension (Key and Lycett 2014, 2011) and force (Iovita et al. 2017; Stemp et al. 2015; Key 2013; Pfleging et al. 2018). Unfortunately they suffer from a few technical shortcomings: those using accessible load quantification technology (i.e., Key and Lycett 2015, 2011; Stemp et al. 2015) force tool users into unnatural motion due to the restrictions inherent to using an electronic cutting board, while the others (Iovita et al. 2017; Pfleging et al. 2015, 2018) use technology that, while extremely impressive and yielding terrific results, is exorbitantly expensive and inaccessible outside an exclusive robotics laboratory setting—factors that severely restrict its utility as a practical method.

Beyond the handful of researchers concerned with gesture as a component of lithic use wear studies, most experimenters consistently use a simple stroke count (e.g., Giusca et al. 2012; Macdonald 2013) or task duration (e.g., Dickie 2015) as a proxy for gestural control and recording. While these are essential actions to record, the general neglect of gestural specificity during tool use has led us to neglect the important questions of what constitutes a "stroke" and how strokes vary respectively between tasks and users? In an experimental setting, without gesture definition and quantification—either through control or recording—robust wear quantification is compromised by an imperfect understanding of tool use life history. The gestures of tool use must thus be understood as experimental variables and controls just like lithic raw material and contact material. Therefore, it is only logical that gesture should be recorded in detail, and efforts be made to maintain an optimal level of consistency. In terms of

using experimental data for archaeological analogy, it is risky to assume that an ancient tool user applied the same force in the same way as their modern counterpart conducting an experiment, and a detailed record of the modern gesture provides a known starting point and is thus vital for experimentally generated wear patterns to be useful references for those found on archaeological artifacts. The challenge is in balancing the need for gestural control and recording and the realities of human tool use. Simple stroke count experiments do not achieve adequate control over gesture. At the other extreme, mechanical experiments such as Pfleging et al.'s (2018) robotic arm method demonstrate ideal gestural consistency and control, but lose the human dimension of tool use; the reflexive relationship between the human user, the technology, the material, and the task (Ingold 2000). On this basis it may be necessary to sacrifice ultimate mechanical precision for adequate control but observable genuine dynamic human behaviour.

A second problematic aspect of many current lithic use-wear experiments is that the quantification of use-wear requires access to highly specialized and very expensive equipment. This requirement drastically constrains the pool of potential researchers, effectively creating an exclusive club of people with the funding and institutional resources to carry out analyses of this type. Thus, data generation and interpretation rest with this small group of specialists. Data scrutiny is similarly constrained, as efforts to replicate studies or apply a new methodology to one's own data are restricted by access to the equipment; a situation contrary to the foundational ideals of the scientific method, where the ability to replicate methods and results is of paramount importance (Marwick 2015). Also, as the vast majority of current archaeological work carried out in Canada (and many other countries) is performed in a cultural resource management (CRM) context, many of the innovative use wear methods developed in the last few years are unlikely to be broadly applied to most of the lithic assemblages excavated every year, given contractual conditions and budgets. Nevertheless, inexpensive tools and open methods, usable by non-specialists are more likely to be adopted in CRM archaeology, and may thus expand analysis to assemblages that might otherwise receive far less attention. Furthermore, in North America (and indeed any environment of colonial archaeology) the restriction of data production and analysis is particularly problematic as it is very often the Indigenous descent communities that generated the local archaeological record, but are the least likely to have ready access to

high-cost scientific laboratory equipment and dedicated trained technicians. This (re)produces the classic power imbalance where the narrative of the past is dictated and disseminated by (generally) non-Indigenous academics, with limited input from descendants. The principles of Open Science—clearly explained methods coupled with open source software and hardware that is either non-specialized (i.e., a camera) or open source and genuinely low cost (i.e., Arduino)—are a powerful counter to the hegemony of institutionalized research. One may argue that a relatively minor sacrifice in precision (confocal laser scanning microscopy, for example, will always be much more precise than home photogrammetry) is outweighed by the benefit of accessible methods in wider practice. More people doing science is better.

1.2. Aims and Scope of Research

This dissertation presents and evaluates innovative tools and methods to address the issues of studying gestural variance in lithic experiments and inaccessibility of analytical tools and techniques in use wear analysis. The first part of this study presents and evaluates cumulative stroke distance (CSD) and "OpenHaft" as a method and a low-budget analytic tool for precisely recording gesture in lithic experiments. CSD is a method of measuring the overall use life of an experimental lithic, and OpenHaft is an electronic handle onto which one may haft stone tool components permitting continuous recording of the force applied by a user during a task. These data are wirelessly transmitted to a computer where they are saved and made available to be analyzed in order to quantify if and how edge loading affects the accrual of use wear, the degradation of tools, and ultimately task performance by tool users. CSD and OpenHaft effectively record distance and load—two core components of what constitutes a "stroke". The OpenHaft device uses commercially available electronic components and the Arduino open source microcontroller system to ensure that users can access the tool as well as customize, expand, and further develop its capabilities.

The second part of this project is a suite of scripts for QGIS, a free and open-source GIS package, that permits users to quickly and easily identify and quantify edge wear on experimental lithic artifacts, and compare that wear at scales ranging from individual artifacts to

entire assemblages (or even regional assemblage collections). Edge wear quantification is valuable in understanding and interpreting tool use-life histories, including the nature of tool function, hafting methods, as well as raw material economies and tool curation practices in antiquity. The lithic digital data is acquired using free and open-source 3D photogrammetry software, and the QGIS modules are designed to be readily (and intuitively) usable by non-specialists, while permitting more advanced users to fine-tune settings and further develop the software to suit their own research goals.

The third and final component of this study applies a QGIS adaptation of Kuhn's (1990) Geometric Index of Unifacial Retouch Flaking (GIURF) to a set of experimental microblades in order to test if the unworn blade surfaces can be digitally reconstructed using only data from worn microblades. The quality of the reconstruction is quantified using the QGIS wear quantification scripts. Like OpenHaft and the experimental edge wear quantification algorithms, the QGIS edge reconstruction method, "MicroGIURF", uses free and open-source software and provides users with the QGIS scripts, upholding the principles of open source science.

1.2.1. Where is This Going?

While originally designed for the study of microblade wear, the methods and resources presented in this dissertation—both the OpenHaft and the QGIS experimental lithic analysis suite—are ultimately applicable to a broad spectrum of technologies, ranging from expedient flake tools to groundstone implements, bone or antler tools, to the surfaces in contact with the tools such as abraders or even petroglyphs. Wherever a relatively simple edge or surface (complex surfaces with multiple flake scars introduce additional confounding variables) is subject to wear as a result of human tool use, the OpenHaft and QGIS lithic analysis suite should be relevant and applicable. Questions of gesture and technique, contact material and use-wear accrual, and tool design and function are universal, and an accessible analytical toolset designed to address these lines of inquiry is needed. Other researchers have presented methods that address some aspects of quantifying use wear (e.g., Evans et al. 2013; Macdonald 2013),

tracking aspects of gesture (Key et al. 2014; Pfleging et al. 2015; Stemp et al. 2015), applying GIS to lithic surfaces (Bird et al. 2007; Davis et al. 2015), but those methods are consistently dependent on access to expensive and specialized equipment or software. The financial inaccessibility is directly contrary to the principles of the scientific method, which is predicated on the replicability of the methods and results of one's study. Replicability depends on repeating a researcher's methodological protocols, which includes the analytical environment: research equipment, software packages, etc. If that environment is inaccessible, then replicability suffers. Financial constraints are a major hurdle in terms of hardware and software accessibility, and for that reason archaeological research carried out with super-expensive tools results in generally inaccessible methods, and has low potential for replicability outside a handful of institutional laboratories (Marwick 2015). Lowering the financial barriers to participation in research also diversifies the community of researchers and enables (and invites) the presentation of more perspectives, opinions, and ideas, ultimately enriching archaeological narratives.

The issue of accessibility is central to this dissertation, and all of the methods presented here use free, open-source (and generally cross-platform) software, and OpenHaft uses inexpensive, readily-available consumer-grade electronic components and the well-known Arduino microcontroller platform. The standard build instructions deliver a basic load-sensing device that may be fixed to any manner of handle or haft. The components may be swapped between handles to test different tool types: one can measure load on a stone knife edge in the morning and then study force in atlatl throwing in the afternoon using the same OpenHaft system. The QGIS lithic analysis scripts are designed to be easy to use with a minimum of input at their most basic level—little more is required than selecting the appropriate data file and clicking "OK". While both OpenHaft and QGIS are designed to be useful "out of the box", the open source nature of the projects means that "power-users" with a greater understanding of data set quality, specific analytical needs, or just curiosity about the internal workings of the processes may easily manipulate fine-grained settings and variables in order to customize the lithics suite to meet their own requirements. Both tools use software and programming languages with large, active, and enthusiastic (even welcoming!) online communities, and both are released under open licenses to encourage free use, distribution, alteration, and development.

1.2.2. Where is This Coming From?

One driving motivation behind this research is its applicability to the study of microblade technology. Microblades are small lithic flakes produced using a prepared core knapping technique, with parallel lateral margins, a length to width ratio of at least 2.5:1 in an unmodified state, and a maximum width of 11mm (Andrefsky 2005; Odell 2003; Whittaker 1994). Microblades are a global technology, having been recovered from archaeological sites on every inhabited continent, and from temporal contexts ranging over 35,000 years from the earliest known instances in Siberia at Ust-Karakol 1 (35,100 BP) (Kuzmin, Yaroslav V. 2007) to the 19th Century AD—after the introduction of steel tools and firearms in Northwestern North America (Magne and Fedje 2007). Related prepared-core blade technologies extend the time range even further back into the Middle Paleolithic, 500,000 years ago (Wilkins and Chazan 2012). Across this spread of time and space, microblades occupy an interesting position within lithic technology in that despite having very uniform morphology, because of their nature as composite tool components they may be applied to a relatively wide variety of functions, essentially to any task where a sharp edge is useful. Applications range from single-blade end-hafted knives (Croes 1995; Flenniken 1981) and side-hafted knives (Owen 1987) to armatures on large mammal hunting weapons (Giria and Pitul'ko 1994; Lee 2007; Christensen and Stafford 2005; Waber 2011; Hare et al. 2004). However, the flexibility of composite tool components also means that identifying specific functions has hitherto generally required the archaeological recovery of intact haft components, which are made from perishable materials. As a result, microblade hafts are unknown from many places where microblades form part of the lithic toolkit, and thus the actual function of microblade tools is unknown, or at best inferred from neighbouring regions with better data.

The gap in knowledge regarding microblade hafting and function is particularly salient on the British Columbia Plateau (the author's region of focus). Here, microblades are known from archaeological sites dated to the early Holocene (Stryd and Rousseau 1996) to the proto-colonial Fur Trade period (Magne and Fedje 2007), spanning a major shift in lifeways in the mid-Holocene, during which people transitioned from a culture of small communities engaged in a

highly mobile foraging economy to a semi-sedentary, socially complex society practicing logistically organized collecting (Pokotylo and Mitchell 1998; Prentiss and Kuijt 2004). Other technologies change over this time, from housing and food storage to weaponry and projectile points, but microblades persisted throughout, and have been recovered from sites across the region, in a variety of biogeoclimatic zones, and in a variety of site types (Magne and Fedje 2007; Pokotylo and Mitchell 1998). A more complete understanding of the temporal, spatial, functional, and cultural practices of microblade hafting and use would allow archaeologists to better interpret the niche(s) of microblade technology in Plateau technological strategies, which in turn has relevance to microblade use in other complex hunter-gatherer societies around the world, as well as to notions of technological evolution.

A significant obstacle for microblade function identification is the absence of any haft elements in the archaeological record. Made from organic material, such as bone, antler, or wood, preserved microblade tool hafts are rare, thus in most cases the means to identify a microblade's hafting method depends on the spatial patterning of wear (and wear of different types and intensities) on different sections of the artifact (Rots 2003, 2004, 2010). The edge of the blade that sits within the haft (i.e., the edge opposite the cutting edge in a side-haft configuration) is protected from the dynamic wear forces, but may be in static contact with the haft itself or any mastic used to fix the microblade in place. The result should be that the operational edge and the hafted edge each accrue and exhibit different wear properties. On an assemblage scale, spatial patterns in wear on different parts of a microblade should indicate which hafting practice was used at that place, and thus regional patterns of composite tool designs may be identified on the basis of microblades.

Investigating the unresolved nature of microblades calls for a specific suite of tools and techniques. Use-wear is the primary line of evidence for identifying microblade function, and thus demands the greatest attention. However, the two aspects of use-wear in question, intensity and location, require different methodological approaches. Quantifying the intensity of lithic tool use allows insight into how technological strategies prioritize tool effectiveness, tool-type functionality as it relates to specific tasks, the nature of tool maintenance and curation, and raw

material optimization practices. Assemblage-scale wear location patterns (and thus the identification of hafting zones) are the primary proxy for reconstructing composite tool form in contexts without preserved hafts (Bird et al. 2007; Schoville 2010). Each of these factors of wear demands a different approach, but both share the common trait that they are ultimately spatial analyses of wear on stone tools. Given the shared spatial analysis aspect, GIS offers the optimal solution for microblade wear analysis: a robust and well-established spatial analysis tool with components designed specifically for surface analysis and pattern recognition. GIS is generally perceived as only applicable to landscape-scale analyses, but the principles and methods exist independent of scale. Measurement in kilometres, metres, or millimetres is ultimately inconsequential as long as the data are sound and the applied algorithm is appropriate.

Microblades are ideal for spatial wear analysis as they straddle the boundary between formal tool types with regular, (relatively) uniform morphologies and expediently produced, low-investment flake tools that are not subject to extended curation. Inexpensive both in terms of raw material use and labour costs, the tiny bladelets are eminently disposable and thus are unlikely to experience substantial modification or retouch, and the consistent shape means that patterns of wear related to hafting should be quite consistent across multiple samples. Indeed, manufacturing the hafts of composite microblade tools requires such greater investment of time, effort, and resources than the individual blades; it is highly implausible that they would be modified for individual blades (Waber 2011). We may therefore infer that microblade manufacture was guided by a standard/optimal microblade model: one that can fit in the haft and be used for the designated task. On this basis, morphological variation in microblades is relatively rigidly constrained. As a result, the spatial distribution of wear on individual microblades may be meaningfully compared to that of other microblades, and patterns may be identified and described over assemblages. This is in marked contrast to most other expedient flake tools, where highly variable flake morphology informs decisions regarding the effective edge location (and whether modification is required to make the edge effective) and the ergonomic shape of the tool. These expedient flakes do not conform to the morphological standards required of hafted tools, so the flake tools are held in ways dictated by comfort or habit. The “business end” is determined ad hoc, rather than through established design

paradigms. Thus, while microblades are made expediently (i.e., they are made as-needed, and require little/no modification prior to use), they conform closely to a morphological standard, and it is that fact that makes them eminently suitable for meaningful use-wear analyses with emphasis on habitual patterns of tool design.

Building an understanding of microblade technology is pertinent to core anthropological questions surrounding social organization, foraging optimality models in socioecological adaptations, and the nature and history of human-environmental interactions, as it addresses aspects of technological strategy among developing-, transitionary-, and complex hunter-gatherers (e.g., tool design; tool and raw material curation practices; differential access to resources due to geographic, sociopolitical, or other factors; logistical planning and organization around other resource-gathering/processing activities), and is also essential to crystalizing the fundamental culture-historical narratives of a region. However, with the paucity of preserved composite microblade tools and the wide variety of possible applications, regional microblade traditions must be studied independently, rather than simply replicate neighbouring models. A better understanding of microblade application as viewed through use-wear patterns is a good step towards a more complete picture of this interesting and important technology.

1.3. New Methods in Use-wear Quantification: OpenHaft and the QGIS Lithic Analysis Toolbox

This dissertation presents OpenHaft and a set of QGIS scripts for quantifying lithic use-wear, and uses lithic experiments to assess and demonstrate the utility of these methods. OpenHaft combined with the cumulative stroke distance provides an effective means of recording the volume and intensity of "strokes" in lithic experiments. This is a major step towards a better understanding of how gesture affects lithic use wear. The QGIS lithic wear quantification modules measure and map material loss on experimental lithic artifacts. The data from these scripts may be used to examine how edge damage develops in different use contexts,

and how that may apply to archaeological analogues. This provides archaeologists with greater clarity in regards to the life histories of tools, and how those objects articulated with cultural concepts of tool efficiency, material economy, and technological strategies in general. All of the methods presented here are built on a foundation of open technology and open data. This is an effort to open high-tech lithic analysis methodologies up to as many interested researchers as possible.

1.3.1. OpenHaft and Gesture Quantification

A critical first step in improving the objectivity and replicability of lithic use-wear studies with an experimental base is to control for, or at least record, the gesture(s) forming the active task component of the experiment. As discussed above, controlling and recording gesture in lithic experiments depends on a balance between adequate precision and maintaining genuine tool use action (as used by a human). The motions must be recorded, and must be consistent between tests, but they must also simulate the behaviours of our archaeological research subjects; mechanical motion is inadequate as it is not informed by active decision making; mechanical motion replicates a single action *ad infinitum*, but tool use must be understood as a portion of a *chaîne opératoire*, where the tool user's cognitive goal of completing a specific task guides his or her actions (Edmonds 1990; Sellet 1989; Pelegrin 1990; Boëda et al. 1990), rather than arbitrary repetition. To this end it is appropriate to consider the components of human gesture in the context of task operation order to identify what should (and can) be recorded. Two aspects of motion are fundamental to the application of action to a task: volume and intensity. Volume may be understood as how much motion occurred, and intensity is how much force was exerted during that volume of motion. Measuring gesture volume and intensity may be achieved by combining a basic "lithic odometer"–cumulative stroke distance (CSD)–with OpenHaft to record the length and load of each individual stroke in a lithic experiment. This not only records components of each stroke for analysis with wear data, but also permits researchers to directly observe previously intangible elements of gesture: how tool users reflexively adapt their action to the exigencies of the tool and the material being worked. CSD and OpenHaft simultaneously regularize varied strokes by allowing accurate measurement of stroke components, and they

highlight the unique and variable nature of strokes that might otherwise be externally perceived as uniform, by permitting observation of previously invisible details of gesture.

1.3.2. Wear Quantification

Recording and measuring use-wear experienced by stone tools used in the OpenHaft system may be achieved using a suite of free, open-source software tools. The process uses a set of QGIS Processing Toolbox scripts to analyze a set of 3D digital models of experimental stone tool surfaces recorded at different stages of a use-wear experiment. The GIS compares the lithic surfaces at successive stages of wear in order to identify and measure the difference(s) in volume. As material is worn away during use, that area of the lithic surface registers a difference from the preceding iteration of the tool with less wear; the GIS measures that difference and maps it onto the digital model. The result is a detailed record of how much wear accrued on each individual tool, expressible as the absolute and relative volume of material loss, as well as the location and extent of edge wear, including the proportion of wear relative to the total potential usable tool edge. The results may be collated for entire experimental assemblages, and edge wear patterns may be compared for variables such as lithic raw material, contact material, haft form, volume of use, and (as made possible by OpenHaft) intensity of use. 3D lithic surface models made at multiple stages of an experiment provide researchers with detailed data not only about absolute wear on a discarded lithic tool, but about how different variables affect the gradual development of wear over the course of a tool's use life. Applied to archaeological analogs, this facilitates inferences about the specific use-life patterns and stages observed in different assemblages. For example, understanding the use-lives of discarded microblades at sites representing different points in the seasonal foraging round would provide insight into if and how toolkits may have been adapted to specific sets of tasks, the (non-)immediate availability of viable tool stone, or logistical demands made on mobile task groups.

1.3.3. Testing GIS-based GIURF (Geometric Index of Unifacial Reduction Flaking)

One notable problem exists in the broad application of the automated GIS-based edge wear quantification scripts to archaeological assemblages: the QGIS wear quantification algorithm requires a "fresh edge" starting point for comparison to the worn edge. In experimental analyses the fresh edge is exactly that—a 3D digital record of each lithic tool in its original unworn state. Since worn archaeological lithics presumably underwent their edge alteration in the depths of antiquity, prior to being discarded, the opportunity to record the fresh edge has passed, and it is necessary to "virtually" reconstruct the edge. The virtual edge can thereby be directly compared to the 3D digital model of the worn lithic, and the difference between the two digital models marks the wear that took place between initial tool production and ultimate artifact recovery. In the specific context of microblade use wear, the "simple" edge (a feature shared with other lithic technologies such as un-retouched expedient flake tools) is ideal for virtual edge reconstruction. In an idealized and simplified form, the convergent dorsal and ventral surfaces, in pristine states, produce a triangular prism. When compared with the 3D digital model of the actual artifact, deviations from the idealized prism may be understood to represent modification—edge damage. On this basis, digital automated quantification of edge damage requires that prism, which is the virtual reconstruction of the tool's edge.

Restoring "virtual flakes" to a stone tool is not a new concept. Kuhn (1990) proposed a method of calculating unifacial tool retouch based on a geometric index of unifacial reduction flaking (GIURF). GIURF essentially uses a cross-section of an artifact and extends the un-altered dorsal and ventral surfaces of the stone tool beyond the worked edge. Those extended surfaces form a triangle with the flaked surface, the area of which represents the GIURF value. Greater retouch results in a greater area, and thus a higher GIURF (Eren and Sampson 2009). This method was updated and adapted to use 3D digital models of unifacial stone tools, but essentially followed the same triangulation protocol for edge extrapolation, and cross-section samples taken at certain intervals (Morales et al. 2013).

A logical extension of the 3D digital GIURF technique is to automate the triangulation process with GIS, and to calculate the exact volume of wear for the entire artifact rather than the imprecise volume estimate based on extrapolated cross-section wear areas. The triangulation process is straight forward. A simple QGIS Processing Script may be applied to sample the lithic surface at a desired distance from the tool's edge at set intervals and the triangulate the intersection of the surface and a "Z=0" plane that represents the original edge. Unfortunately, the experimental microblades were not successfully reconstructed using the QGIS GIURF. However, edge point extrapolation and surface reconstruction were achieved in some specific instances, and accurate wear quantification was possible in these cases. The poor results of the general GIS-based GIURF use-wear experiment force one to cast a critical eye on the viability of the original GIURF, and to question how reliable conclusions drawn from the application of that method may be. While the GIS-based automated edge extrapolation experiments were generally unsuccessful, they will hopefully provide some impetus for further inquiries into virtual reconstruction of worn lithic artifacts

1.4. Chapter Sequence

Chapter 2: OpenHaft and Cumulative Stroke Distance: Towards a Quantification of Tool Use Gesture

Here I present and evaluate OpenHaft and Cumulative Stroke Distance (CSD) as tools for recording gestural components of strokes in lithic use wear experiments. The design, programming and function of OpenHaft is described in detail, with two case studies designed to test and demonstrate the OpenHaft/CSD system. The first experiment examines OpenHaft and CSD to demonstrate how variability in human operator action is expressed in load application that results in differences in material wear. Graphite rods are used as a constant 'lithic-proxy' material to better isolate gesture as a variable. The graphite is measured between experimental sets to track wear during the experiment. CSD is used to control the gestural stroke-length variable, and OpenHaft to record variation in load during the test. The second experiment applies the principles of the first test to experimental lithics: a set of experimental microblades were hafted to OpenHaft and used to cut wood, performing a consistent gesture (incising a

straight line in the wood) and volume (50 strokes, each 10cm long, for 500cm CSD). The microblades were recorded using 3D digital photogrammetry before and after the first 500cm CSD, and after the second 500cm CSD. Use wear volume was derived from comparison of the 3D models, and compared with OpenHaft load data. The results reveal that actions in seemingly simple tasks are actually considerably more complex that one might expect. The entire technological system—operator, tool, and contact material—directly affects gesture, which may be observed in the load data. Furthermore, lithic use wear does not necessarily accrue in a predictable, linear fashion, but rather in fits and starts as different sections of a tool's edge fail under load. Both the graphite and microblade tests support the viability of OpenHaft as a low-cost open source tool for recording load in lithic use-wear experiments.

Chapter 3: Opensource 3D Digital Photogrammetry and GIS for Quantifying Edge Damage on Experimental Lithic Artifacts

This chapter presents the QGIS Lithic Edge Wear module suite. The programming and function of each component module is described, and a test case illustrates the effectiveness and potential of the digital toolkit. The QGIS modules are a set of Python scripts for the QGIS 2.X Processing Toolbox that interpolate and prepare a raster surface from a point cloud model of an experimental lithic tool derived using 3D digital photogrammetry, and compare pre- and post-use surfaces for edge damage or surface alteration. That data is recorded and mapped for further analysis and comparisons across and between assemblages. The experiment uses a set of microblades used to incise straight lines in wood and tests the assumption that lithic tools with uniform morphology, hafted the same way, and used for a consistent task (500cm CSD) should exhibit similar patterns of use-related edge damage. The volume of wear should be relatively similar and should accrue at roughly the same rate, and given the single hafting method, wear should be concentrated on the same section(s) of each tool. The experiment shows that edge damage manifests differently depending on blade morphology and hafting formation: some microblades exhibited minor edge wear, while others suffered catastrophic breakage and material loss. More importantly, the experiment demonstrates that the QGIS lithic wear quantification

modules are effective tools for precisely quantifying and mapping edge wear on experimental lithics.

Chapter 4: GIURF and GIS: Testing GIS Automation of Kuhn's Geometric Index of Unifacial Retouch Flaking for Digital Edge Reconstruction and Wear Quantification

In this chapter, I describe the development of a QGIS script workflow for applying Kuhn's Geometric Index of Unifacial Reduction Flaking (GIURF) (Kuhn 1990; Eren and Sampson 2009) to digital 3D artifact surfaces, and thus automating the virtual reconstruction of worn lithic artifacts. The algorithm uses the digital surface model (DSM) and slope data derived from a digital artifact model to extrapolate the original tool edge. The test uses experimental microblades recorded using 3D digital photogrammetry before and after use wear experiments, and attempts to digitally reconstruct the original intact edges based on the 3D models. The projected edges are compared with original blade edges to assess the accuracy of the algorithm. The tests indicate that while the method is viable under optimal conditions, any variability in the artifact surface results in unreliable edge projections, and much more work is needed before GIS-based "MicroGIURF" (GIURF applied on the scale of microflakes) can be confidently applied to digitally reconstructing archaeological artifacts for the purpose of quantifying wear or retouch.

Chapter 5: Conclusion and Future Directions

Chapter 5 synthesizes the research in the previous chapters, and describes how the methodological tools may contribute to further lithic use wear experiments. I suggest that future avenues of research follow two primary avenues: 1) investigating how individual tool users affect gestural variability, and 2) applying 3D GIS analyses to archaeological lithic artifacts. In the first case, OpenHaft and CSD may provide a better understanding of how variation in the human component of a technological system affects load application, both in lithic use wear studies and in other contexts. Factors such as user expertise, physical characteristics, body

position, and even user-adaptation to different tool forms are all hitherto-unexamined aspects of lithic tool use. User expertise is especially interesting as OpenHaft may be adapted to myriad tool applications and thus may be used to observe how users apply force at various stages of skill acquisition in different activities. This is pertinent to understanding how novice tool users might manifest in the archaeological record, as well as gaining insight into apprenticeship and learning processes—topics with archaeological and anthropological relevance, but which are also salient to other fields of research, and beyond academia.

Second, if further experimental applications of the QGIS use wear quantification and mapping scripts are to be useful, a future research priority must be to bridge the gap between experimental and archaeological lithics in automated use wear analysis. Developing an accurate, precise, replicable, and accessible method for reconstructing the un-worn state of a lithic tool is integral to quantifying use wear on archaeological samples. Following this, the QGIS use wear tools may be applied to archaeological lithic assemblages, and thus the nuances of use wear patterns may be discovered on both an individual and assemblage-wide scale, leading to inferences regarding use, curation, raw material access, task localization, and other social factors that may manifest in different intensities and forms of use wear.

Chapter 2 OpenHaft and Cumulative Stroke Distance: Towards Gesture Quantification in Lithic Use Wear Experiments

2.1 Introduction

Technologies are defined as much by the actions and gestures they involve as by the physical objects employed in a given task (Pelegrin 1990; Schlanger 1990; Tostevin 2011). This concept is a core tenet of the *chaîne opératoire* approach in archaeology. As such, it is only logical that archaeologists studying past technologies are concerned with, and attentive to, questions surrounding task gestures. While lithic artifact preservation is high, gestures cannot be excavated, so we replicate and use tools for experiments in order to directly observe tools in active contexts. However, it is difficult to assess how closely analogous our experimental actions are to ancient actions. Archaeology looks at behaviour, and lithic analysis and experimentation are tools for that: action and gesture is a part of behaviour. Still, at least in the field of lithic analysis and experimentation, gestural specifics often fall by the wayside in deference to other methodological considerations; we tend to focus on *what* tools were used for, rather than *how* tools were used. If experimental gestures are not adequately controlled or recorded, their utility in an analogous context is decidedly limited. The term "stroke", as it is commonly used in use-wear studies (Evans and Donahue 2008; Giusca et al. 2012; Macdonald 2013), is symptomatic of this inattention to a fundamental aspect of experimental studies. This paper presents a new method and an open source tool for controlling and recording gestural components that are otherwise lost in the generic term "stroke".

The key to improving methodological rigour in stroke-based lithic experiments is to better standardize the "stroke" gesture—the technical action that applies a tool to a task. This is a daunting task, as different researchers using different tools to process different materials are virtually guaranteed to use different gestures. The properties of contact materials constrain and affect the actions of the operator (Ingold 2000). Many lithic experiments focus on the effect that different uses and contact materials have on a stone tool, either as polish (i.e., Evans et al. 2014, 2013; Macdonald 2013) or microflaking (i.e., Akoshima 1987; Tringham et al. 1974). Beyond

the basic cutting vs. scraping task-based gesture definition, limited consideration has been given to how the contact material may delimit or inform aspects of gesture, such as length of stroke, angle of stroke, uni- or bi-directional application force (incising vs. sawing), and how much force is exerted by the tool user, including gesture adaptation to material hardness and homogeneity (both of contact material and tool material). For example, a short, sawing (bi-directional action) stroke with considerable force may be an appropriate gesture for cutting laterally through the beam of an antler with a robust flake tool, but suboptimal for incising a longitudinal groove in the same material. Both tasks share a common contact material and may require 1000 "strokes" for completion, but from stroke length (e.g., 5cm sawing vs. 20cm incising), to stroke direction (sawing back and forth vs. graving in one direction), to appropriate force for cutting through the beam or maintaining a controlled straight line, the "strokes" involved are considerably different and should result in different formation and patterns of edge wear. The key to producing useful gestural data across these disparate actions is to deconstruct the gesture into its base components, and to at least record—if not control—those components.

The core components of a "stroke" are the distance a surface moves in contact with another surface, and the force applied during that contact (defined here as "edge load") (Rabinowicz 1995; Stachowiak and Batchelor 2014). More movement or greater edge load results in proportionately more rapid wear. Unless both factors are recorded, it is impossible to isolate the variable responsible for variation in use wear. Conversely, if both inputs are well understood and controlled for, variation in use wear may be accurately attributed to other factors such as less obvious aspects of a human stroke (i.e., the subtle rotation and lateral rotational variability of a handheld tool in use), features of tool morphology, or contact material—the popular (and elusive) target of so many lithic experiments. In order to measure stroke distance and load during the course of a lithic experiment, two methods are needed: an odometer to track the distance a tool has travelled relative to the contact material surface, and a device capable of articulating with a stone tool and measuring the force applied to that tool's edge. In the case of stroke length, I propose the application of "cumulative stroke distance" (CSD)—a simple stroke odometer—the multiplication of stroke count by stroke length. The result is the sum value describing over how much 'distance' a given tool was applied to a contact material. Thus, as

long as the CSD values are the same, tools used to work different materials may be compared with the confidence that every tool edge has experienced an equivalent use-life volume to its experimental counterparts, even if they differ in the absolute number of strokes. Recording edge load requires a more high-tech approach. In response to this need I designed and constructed OpenHaft, a low-cost open-source hand-held electronic handle with interchangeable components capable of adapting to a variety of (hafted) lithic tool types (Figure 2.1). A simple load cell continually measures the force applied during a task, and that data is wirelessly transmitted to a nearby PC for real-time monitoring and recording of the force component of a gesture. The result is that one may precisely gauge the intensity of a tool's use in conjunction with the volume of use as measured using CSD. This permits one to isolate these elements of user input to the overall wear system, and to control and quantify how loading affects wear accrual.

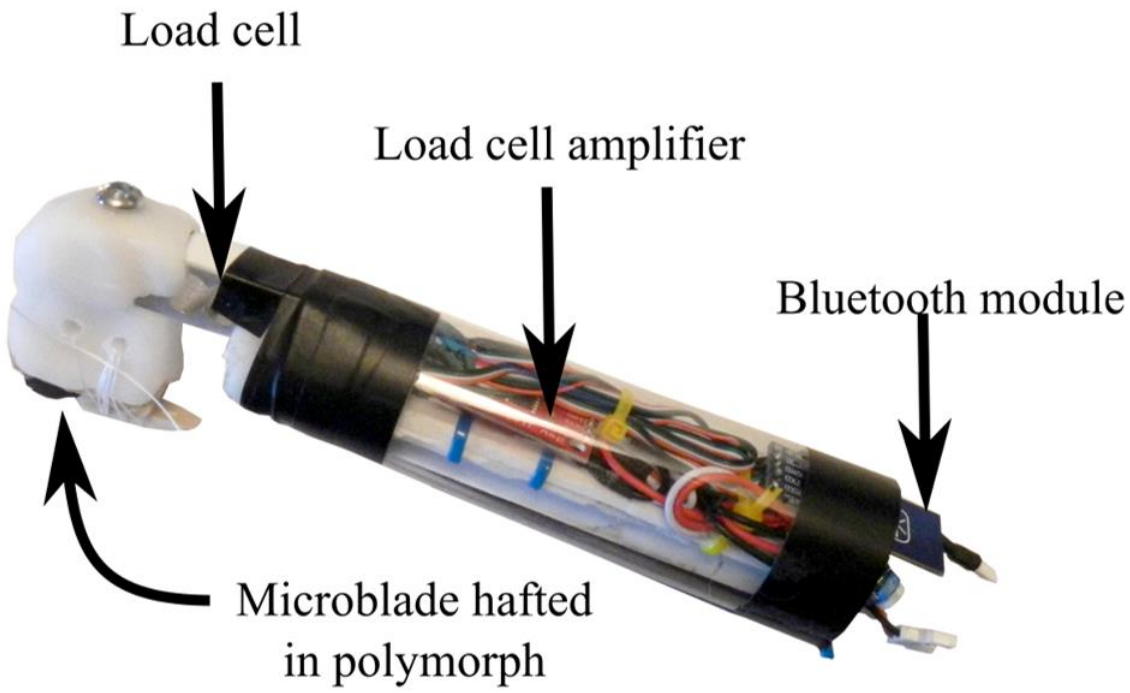

Figure 2.1: OpenHaft digital load sensing device for lithic experiments. Overall length is approximately 20cm.

OpenHaft is not the first effort towards quantifying gesture in lithic experiments; recent efforts to record edge load (Key 2013; Key et al. 2014; Pfleging et al. 2015) have had promising results in identifying and recording precise load values for stone tools in use tasks, and assessing

how individual user characteristics may affect load patterns. However, some of these improvements in load recording have limitations surrounding how measurement equipment constrains task gestures (Key 2013; Key et al. 2014; Stemp et al. 2015). In these experiments, operators used flake tools to cut objects on essentially a load-sensing kitchen cutting board. This permits the users to apply any tool to the cutting task, with no need for special hafting (a boon for studying unhafted flake tools!). Unfortunately, the protocol requires the contact material to be fixed on the cutting board and for the operators to cut without holding the object[1]. One cannot simply place a contact material on a bench balance and monitor load readings if, while the cutting tool is wielded in the right hand, the left hand is holding the worked object in place and introducing an additional load source onto the balance. This multi-source load input introduces too much additional load noise for the relevant tool data to be isolated. Thus, the operators must perform their cutting tasks in an unnatural, counter-intuitive one-handed manner with poor analogical value to ancient tool users (a limitation identified by Key (2013)), or else compromise their data collection by using the off-hand for material positioning, support, and stabilization. Unnatural gestures restricted to lab benches are suboptimal analogs for the vast majority of tasks carried out by ancient tool users, where either hand, both feet, and often even one's teeth would be used to stabilize the objects being worked (Key 2013), and those objects would be adaptively re-positioned throughout the task. The cutting board restriction shifts these studies towards the mechanical device end of the experimental spectrum, away from more naturalistic field tasks, but it does so without the improved replicability of using a robot operator.

In order to overcome the digital cutting board paradox one must incorporate the load sensor into the tool itself. By using a simple multi-purpose haft with an electronic load sensor, one may directly measure the force exerted on a tool's edge, without compound input from other sources such as supporting hands, and without being tethered to a laboratory bench. Pfleging et al. (2015) offer a very effective tool which does that: it measures normal load, as well as friction load, coupled with a visual marker for video monitoring and motion/tool position tracking. The load and motion data produced by this system is very impressive, with multiple strain gauges

1 Furthermore, if enough material is removed from the board-top object it is necessary to re-tare the balance for a newly reduced base weight.

providing torque and twist readings along with the same normal load as OpenHaft, but the force transducer unit (ATI Mini45) costs over $7000 USD, not including the data acquisition (DAQ) board or interface software. The price is a significant barrier to using this tool for smaller labs or independent researchers.

OpenHaft is a genuinely low cost (<$60 CAD) modular load-sensing haft made using off-the-shelf electronic components, based on the Arduino microcontroller platform (Appendix A). The microcontroller reads data from a load cell at a specified interval and outputs that data to a computer via a wireless bluetooth serial connection. The load cell precision is 10g (±0.05% of its maximum load of 20kg), which is quite adequate for tracking most cutting tasks, and is easily upgraded for more heavy-duty applications. Simple code (the Arduino language is simply C/C++) is used to calibrate the sensor and specify the measurement interval (default: 10Hz) (Appendix B). The data is transmitted by bluetooth, and may be received using a variety of serial interface applications (Appendix C). The output data may then be opened as a spreadsheet for analysis. Obviously, the device is not useful for analyzing edge load on unhafted flake tools (unless OpenHaft is adapted to hold the contact material rather than the lithic tool), but it provides an elegant and practical solution to measuring the application of force in virtually every hafted tool context, from microblade knives to long-handled hide scrapers. Because the system has wireless power and communication, the user's range of motion is not constrained; the tool may be used in virtually any position: seated or standing at a work bench, squatting on the ground, or even walking or running (as long as the runner stays within ~10m Bluetooth range of the receiver PC). Similarly, the contact material may be supported in whatever way is most appropriate to the test, whether that is dictated by ergonomic comfort, laboratory replicability, or field test conditions. For example, edge loading of a hide scraper may be tested by a user sitting on the ground, holding a hide between teeth, feet, and left hand, without concern about external load noise from those supports corrupting the data. Furthermore, the modular load cell component may be fixed to a passive mount with a tool bit over which the user moves the contact material. The compact form factor of the load cell and the accompanying electronics facilitates adaptation of the system to a wide variety of replica tool forms and material

processing systems. The OpenHaft name reflects the device's open-source nature, with both the hardware design and the operating code free for use, distribution, and alteration.

2.2 Case Studies

Two experiments were conducted to evaluate the potential of CSD and load recording as means of quantifying gesture, and to examine the utility of CSD and OpenHaft in specific regard to lithic use-wear experiments. The first case study recorded wear observed on graphite rods (2mm leads for a mechanical pencil), comparing wear patterns for six variables: graphite hardness (soft vs. hard), stroke intensity ("light" vs. "heavy"), and CSD (100m vs. 200m). This experiment showed that both CSD and loading are indeed major factors in wear accrual, and that both must be recorded for there to be meaningful inferences made regarding factors contributing to use wear. The second case study recorded wear on experimental microblades used to process wood, observing how the development of edge damage progresses over time, and how that may affect interpretations of use-wear. See Figure 2.2 for tool hafting reference image. The microblade material, contact material, stroke length, and load range were all consistent across the tests, but the microblade edge morphology changed as the blades dulled. This experiment showed that edge morphology changes over time, and that in some cases substantial amounts of a microblade may be lost over the course of a task. It also underscores that tool hafting is an important variable that must not be neglected in lithic experiments, as it contributes significantly to the reflexive relationship between user and tool, and thus directly informs the intensity of tool use. Although graphite wear due to friction is a different mechanism than the microflaking wear that is characteristic of flaked stone tools (see Appendix D for a detailed discussion of wear), both experiments demonstrate the importance of recording gestural details in experimental archaeology, and the utility of both CSD and OpenHaft for better understanding the factors contributing to technological behaviours, and thus the formation of the archaeological record.

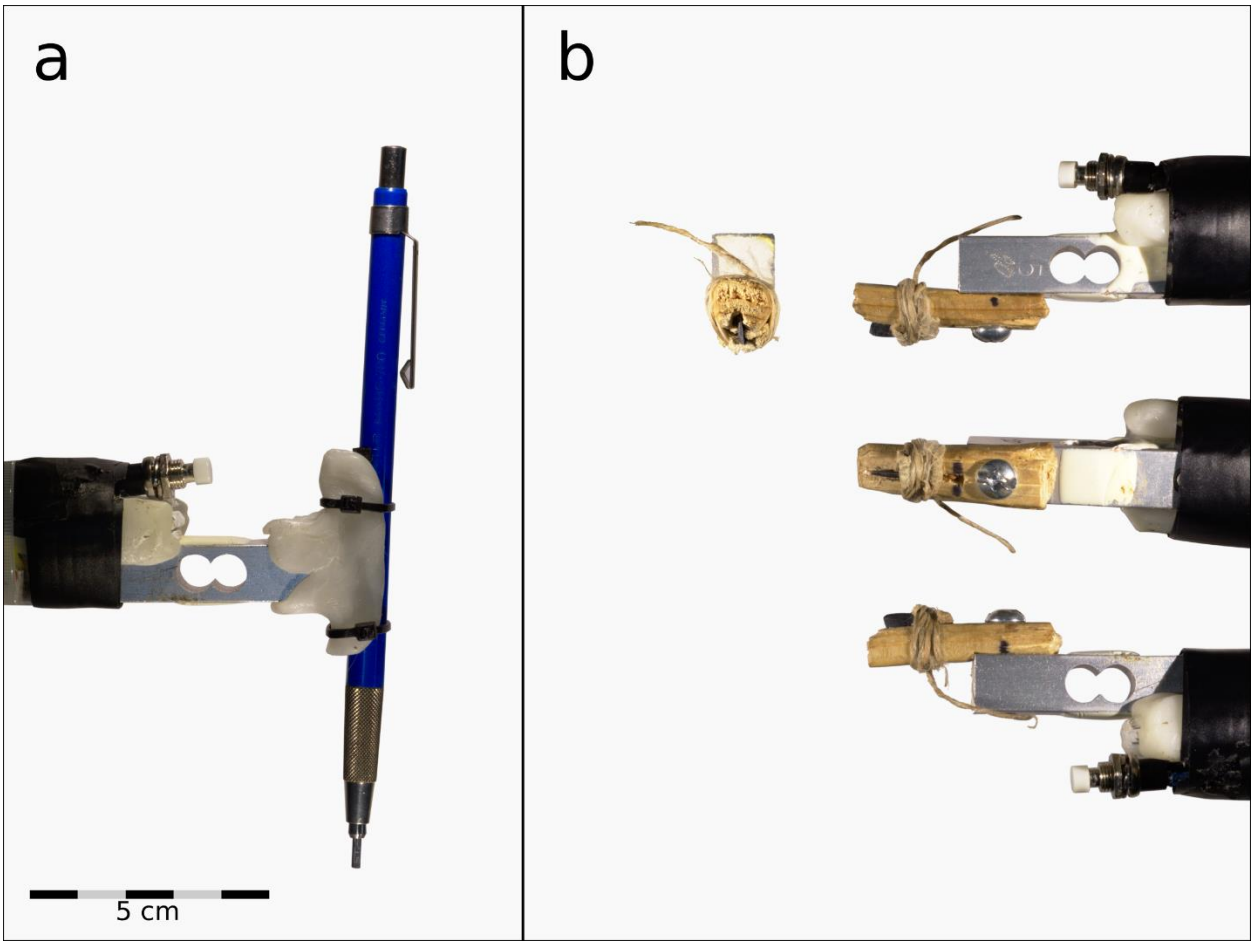

Figure 2.2: Experimental tools. a) Mechanical pencil with 2mm graphite, mounted to OpenHaft using Polymorph (heat-moldable plastic); b) experimental microblade in wooden haft. Sinew and leather binding prevents the microblade from rotating out during use.

Test Case 1: Graphite

Experiment Protocol:

A simple experiment was carried out in order to assess the utility of the CSL and the edge load device. A mechanical draughting pencil loaded with a 2mm diameter graphite lead was fixed to the end of the load cell and was used to draw a series of straight lines of the same length on a sheet of paper, with each line representing a "stroke" (Figure 2.3). At the outset of the experiment, and then after each set of 50 strokes, the length of the graphite was measured using a USB digital microscope, and the difference between pre- and post-stroke set length was used to calculate the volume of lost material. The uniform line lengths (20cm and 10cm), stroke counts, contact material (20lb A4 printer paper), and wear medium allow the effective isolation of test

variables: edge load (as perceived by the user), stroke length (10cm vs. 20cm), and graphite hardness (2B vs. HB). Perceived load was defined on a subjective basis under the broad categories “light” and “heavy”. “Light” was defined as being just enough force to draw a clear, legible line on the paper, barely more than the weight of one's hand. “Heavy” was defined as being enough force to draw as dark and solid a line as possible without breaking the lead. Lines were drawn with little or no overlap in order to avoid graphite from previous strokes acting as a lubricant and changing the friction relationship between the pencil lead and the paper. A total of eight tests were carried out using different perceived loads, stroke lengths, and grades of graphite, each for a total of 1000 strokes. The graphite was measured at 50 stroke intervals, and was re-set as needed when it wore down to the mechanical pencil's chuck. The eight tests are identified in Table 2.1.

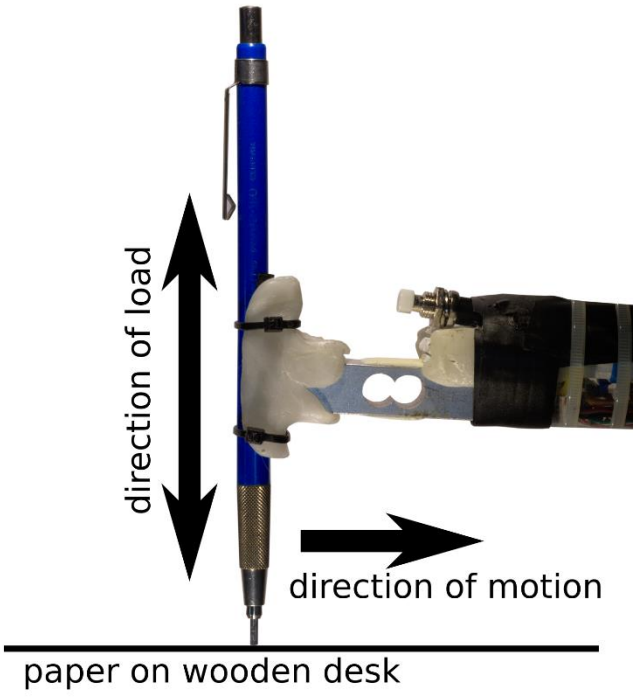

Figure 2.3: Diagram of OpenHaft with graphite indicating directions of motion and load.

Test	Perceived force	Graphite hardness	Stroke length	Stroke count	CSD
1	Light	2B	10cm	1000	100m
2	Heavy	2B	10cm	1000	100m
3	Light	2B	20cm	1000	200m
4	Heavy	2B	20cm	1000	200m
5	Light	HB	10cm	1000	100m
6	Heavy	HB	10cm	1000	100m
7	Light	HB	20cm	1000	200m
8	Heavy	HB	20cm	1000	200m

Table 2.1: Summary of experimental sets.

Graphite was selected as a control medium in this proof-of-concept study because graphite wears much more quickly than toolstone, and the cylindrical graphite rods are of uniform shape, size, and material. Thus, wear is easily measured with basic equipment, and the uniformity of the material means that variation in wear is due to variation in gesture rather than any variability in the tool. Furthermore, the tests are easily replicated by other experimenters. Graphite is obviously much softer and more friable than flakeable toolstone, and although it wears by grinding rather than through conchoidal fracture, the general principles of material wear are applicable to either material. The processes of edge damage and polish development on lithic tools are affected by the same fundamental variables of load and distance as is wear on graphite. Using hard and soft graphite mimics harder and softer stone, where wear and gesture may both be affected by the difference in material properties, and those effects are amplified in the data. Data analysis, processing, and visualization was undertaken using the R statistical package, with the "flux" and "pracma" libraries (Team 2012; Jurasinski et al. 2014; Borchers 2016). See Appendix E for a detailed description of the data processing workflow. The two primary data components used from the OpenHaft were the peak value (kg) of each stroke, and the integral of the load (area under curve for load (kg) over time (0.1 seconds)) for each set.

Results and Observations:

The graphite experiment was designed to test a core set of hypotheses:

1. "Heavy" and "light" strokes, as perceived by a single user, are two discrete categories, evident in both load and wear volume data patterns.

2. The cumulative load difference between “heavy” and “light” categories will result in a proportionately higher rate of wear in the former than in the latter.
3. All 20cm stroke length classes will exhibit approximately twice the wear of their 10cm stroke length counterparts.
4. All 20cm stroke length classes will exhibit approximately twice the cumulative load values of their 10cm stroke length counterparts.

Data analysis takes two paths: (1) comparisons of peak load and integral of applied load values, and (2) correlation analyses of wear rates depending on load, CSD, and tool material variables. Shapiro-Wilks tests of normality indicated few normal distributions (Table 2.1), therefore non-parametric statistical significance tests were used to compare all samples.

Hypothesis 1 (subjective “light” and “heavy categories are “real”) is supported. Boxplots show distinctive distributions of the “heavy” and “light” load categories, with very close grouping among the peak load values. The division between “light” and “heavy” is considerably more distinct in the load integral samples (Figure 2.4) than with peaks (Figure 2.5). Mann-Whitney tests (Table 2.2) support the boxplot patterning. In the “heavy” category of the load integral values, there is a greater similarity between the 10cm HB/2B values, and between the 20cm HB/2B values, than between the 10cm/20cm values within their respective material categories (Table 2.2). A similar pattern occurs among the “light” category samples, where 20cm HB and 2B sets exhibit no significant difference. Other force-category pairings group together visually, but there are no statistically significant pairings.

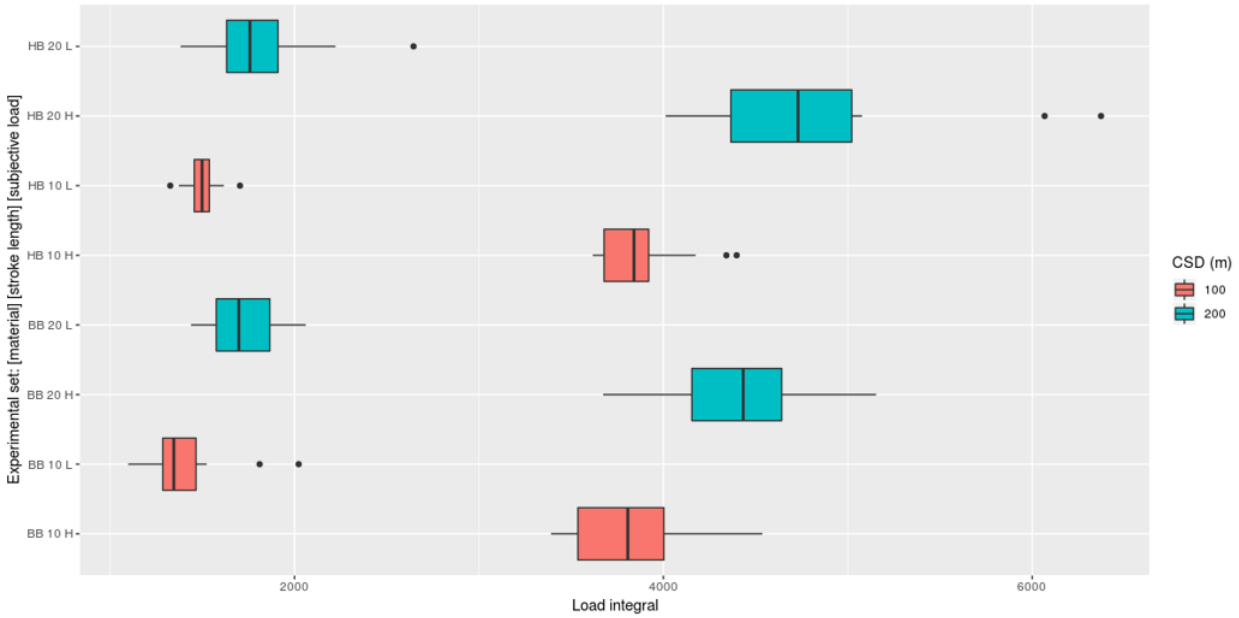

Figure 2.4: Boxplots comparing load integral values between graphite grades and perceived load categories.

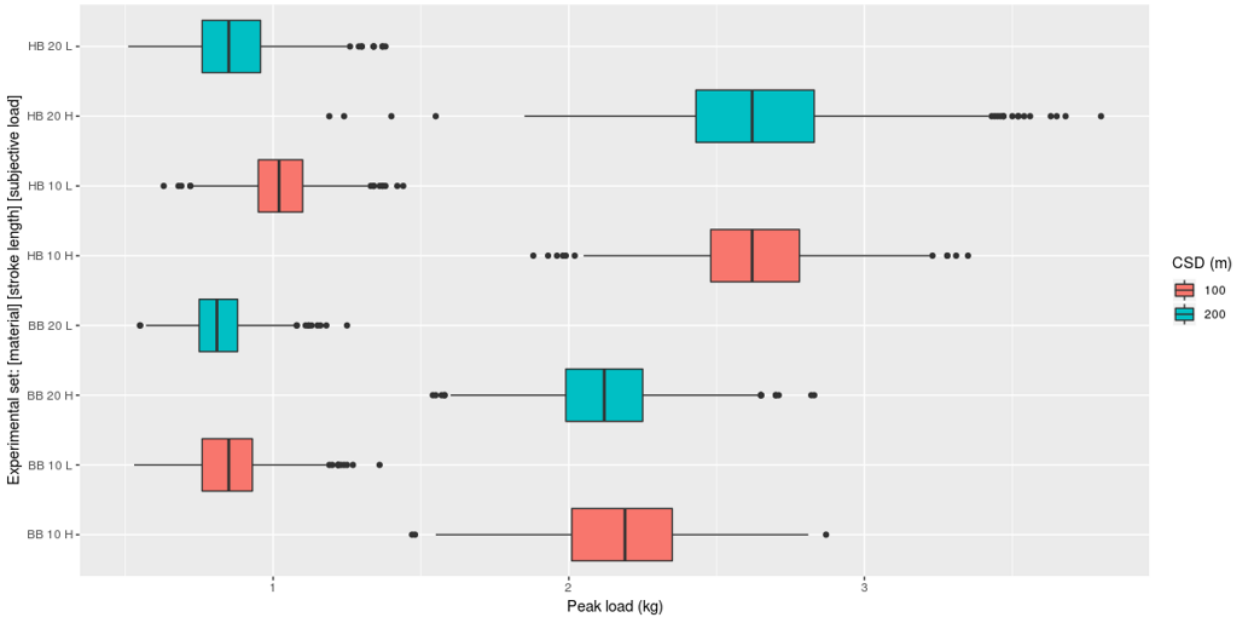

Figure 2.5: Boxplots comparing peak load values between graphite grades and perceived load categories.

Sets	Comparison	Mann-Whitney result	Null hypothesis: "light" and "heavy" subjective loads are equal.
All	"Light" vs. "Heavy" peaks	$p < 0.00001$ in all cases	Null hypothesis rejected: "light" and "heavy" peak load categories are not equal.
All	"Light" vs. "Heavy" integral	$p < 0.00001$ in all cases	Null hypothesis rejected: "light" and "heavy" load integral categories are not equal.
Integral of the load, "Heavy" category, 10cm stroke length.	10cm HB (n=20, median = 3840.834) and 10cm 2B (n=20, median = 3807.259)	U = 177, $p = 0.5468$	Null hypothesis supported: Subjective "Heavy" load integrals are equal between HB and 2B graphite grades.
Integral of the load, "Light" category, 20cm stroke length.	HB (n=20, median = 1758.131) and 2B (n=20, median = 1698.088)	U = 173, $p = 0.477$	Null hypothesis supported: Subjective "light" load integrals are equal between HB and 2B graphite grades.

Table 2.2: Mann-Whitney U test results.

In the case of "light" force application, the grade of graphite appears to be the variable that most affects the load integral sub-groupings, with stroke length being a less prevalent factor. Conversely, "heavy" force tests group according to stroke length to a much greater degree than they do according to graphite hardness, though user adaptation to tool material is still evident in the 2B-heavy sets grouping with lower load values than the HB-heavy. This is likely a response to 2B lead being more brittle and prone to crumbling under heavy load.

Hypothesis 2 ("heavy" load results in proportionately greater wear than "light" load) may be partially accepted. As expected, increasing load resulted in an increase in wear, but the relationship was not 1:1. The increase in wear between "light" and "heavy" loads within common CSD and material groups was consistently (and considerably) greater than the increase in load would predict (Table 2.3). The wear/load differences between 200m CSD HB and 2B sets are relatively similar, but the 100m CSD light had a much lower heavy/light ratio than 200m; the 100m 2B set had a much higher heavy/light ratio. This pattern may indicate that soft material amplifies the rate of wear in short stroke contexts.

Experiment	Cumulative wear (mm^3)	Percent of "light" wear	Cumulative load (N)	Percent of "light" load	Difference Index +/(-)
HB 200m Heavy	107.9	648.04	10264.4	277.35	370.69
HB 200m Light	16.65		3700.71		
HB 100m Heavy	28.21	519.52	7883.64	257.62	261.9
HB 100m Light	5.43		3060.16		
2B 200m Heavy	241.29	553.8	8561.45	242.31	331.49
2B 200m Light	43.57		3533.15		
2B 100m Heavy	119.23	825.12	8134.38	287.11	538.01
2B 100m Light	14.45		2833.21		

Table 2.3: Cumulative wear and load comparison table, paired by CSD. The difference index shows the percentage point difference between the light vs. heavy load integral percentage and the light vs. heavy wear percentage. Positive values indicate a higher percentage of wear than expected for the increase in load; negative numbers indicate a lower than expected percentage of wear for an increase in load. A difference index of 0 indicates equal proportions of wear relative to load.

Hypothesis 3 (wear increases proportionately with CSD) is also supported in some cases, and the general principle of wear increasing as CSD increases holds true. While the predicted 100% difference in wear between 20cm and 10cm tests only occurred in the 2B heavy load tests, there was considerable difference between the two CSD sets across all load and material categories (Figure 2.6 c-d, Table 2.4). Once again, material hardness (and thus material-reflexive load variation) appears to be a substantial factor in determining how much CSD affected the rate of wear. The 20cm tests across all material and load categories resulted in more than the expected double wear volume of that measured in the 10cm tests, ranging from marginally more (202%) to substantially more (382%) (Table 2.3). Based on a CSD increase of exactly 100%, but wear increases greater than 100%, variation in wear percentages between stroke length sets must be due to a combination of material hardness and load. Specifically, the softer lead appears to be more sensitive to wear at lower loads than does the harder lead, which exhibits more rapid wear over longer distance at higher loads. Again, the soft material amplifies the effects of gesture variables.

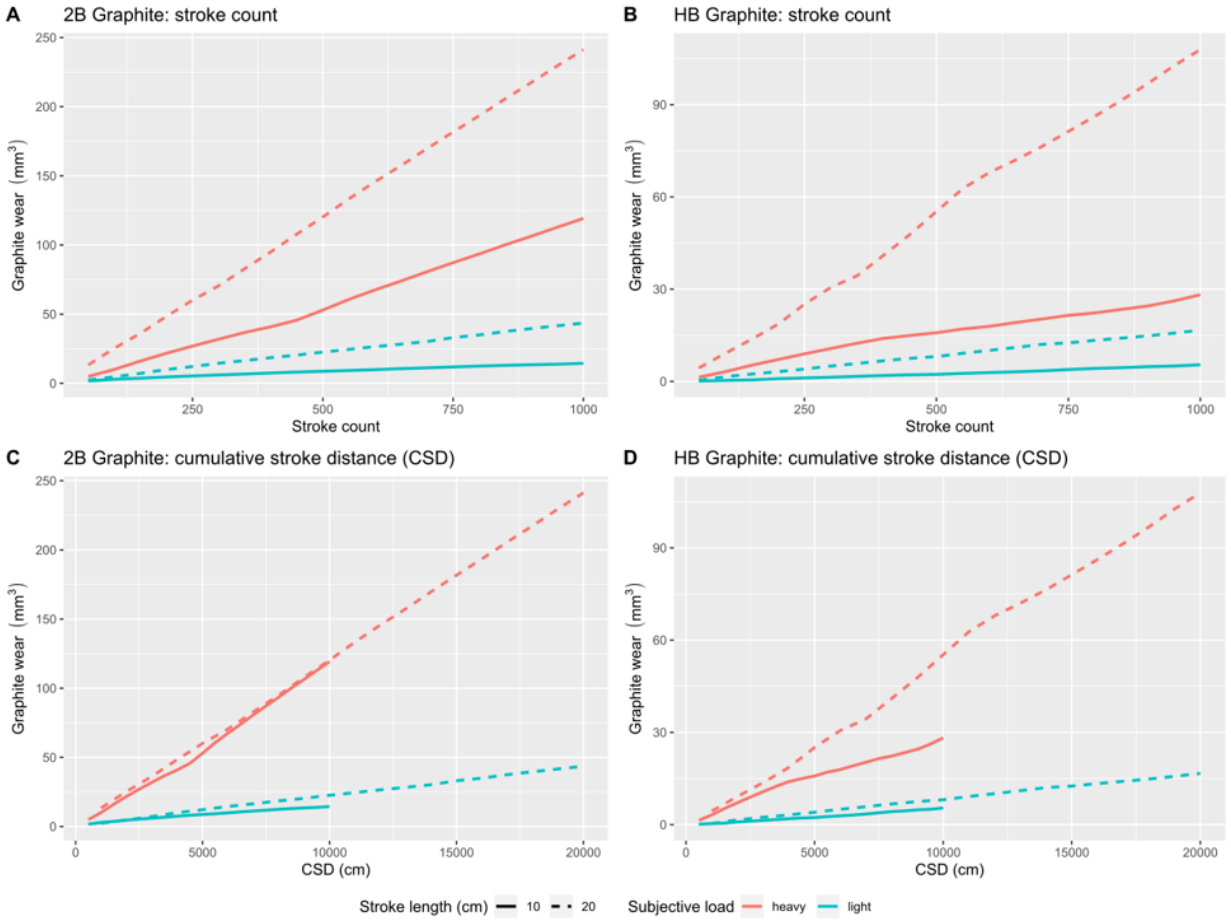

Figure 2.6: Plots comparing wear volumes between graphite grades, perceived load categories, and stroke count/CSD.

Experiment	Cumulative wear (mm^3)	Percent of 100m CSD wear	CSD (m)	Percent of 100m CSD	Difference Index +/(-)
HB 200m Heavy	107.9	382.48	200	200	182.48
HB 100m Heavy	28.21	100	100	100	
HB 200m Light	16.65	306.62	200	200	106.62
HB 100m Light	5.43	100	100	100	
2B 200m Heavy	241.29	202.37	200	200	2.37
2B 100m Heavy	119.23	100	100	100	
2B 200m Light	43.57	301.52	200	200	101.52
2B 100m Light	14.45	100	100	100	

Table 2.4: Cumulative wear and CSD comparison table, paired by subjective load. The difference index shows the percentage point difference between the 100m CSD and 200m CSD percentage and the corresponding wear percentage. Positive values indicate a higher percentage of wear than expected for the increase in load; negative numbers indicate a lower

than expected percentage of wear for an increase in load. A difference index of 0 indicates equal proportions of wear relative to load.

The expected 100% increase in cumulative load (CL), between 10cm and 20cm sets, proposed in hypothesis 4, was not observed in any of the cases (Table 2.4). There are much lower CL relationships across all groups, despite much higher cumulative wear relationships than the expected 100% increase (Table 2.4). The minor increases in CL (<35%) and major increases in wear (>100%) over the exact 100% increase in CSD suggests that CSD is a greater factor in wear accrual than is load. This is to some extent logical, as 100% greater CSD (over a contact material with identical texture) results in contact with 100% more surface asperities—the mechanism of abrasive wear (see Appendix D), while increasing load may only result in deeper "bite" by the asperities, or higher friction between materials, but is still limited by the surface area in contact between two materials—a factor that is unaffected by load under all but the most extreme circumstances (Stachowiak and Batchelor 2014). Nevertheless, the proportions of wear increase mirror the proportions of load increase at least to some extent, with the lowest percentile increase in wear corresponding with the lowest increase in load, etc. Thus, load cannot be entirely discounted from wear quantification studies, although area under curve may not be the optimal measure for expressing load over the course of an experiment.

One further observation: following the principles of the coefficient of wear (Rabinowicz 1995; Stachowiak and Batchelor 2014) (Appendix D), an index of CSD and load (integral) is promising for understanding how gestural differences affect wear development. When grouped by material, linear regression analysis shows a very strong correlation between wear volume and distance-load index (Figure 2.7). Spearman's rank tests between volume and distance-load index for HB and 2B graphite returned significant results, supporting the strong relationship between the indexed variables and wear (HB: $S = 10577$, $p < 0.0001$, rho = 0.8760366 ; 2B: $S = 13371$, $p < 0.0001$, rho = 0.8432861). In contrast, Spearman's rank tests of individual experimental sets returned significant correlations for only 2B graphite: 10cm "heavy" ($S = 372$, $p < 0.001$, rho = 0.72) and 20cm "light" ($S = 569.28$, $p = 0.008$, rho = 0.5719). This shows that while the minor variations in load experienced within single subjective force categories do not necessarily relate directly to wear volume, the general principle of greater wear from higher load

holds true. Subjective load classes and CSD are both discernible in the data. The individual sets cluster together, albeit with higher variability in the longer heavy strokes. More importantly, the distance-load index demonstrates how light long and heavy short strokes return very similar index values, but the heavier loads consistently resulted in greater wear.

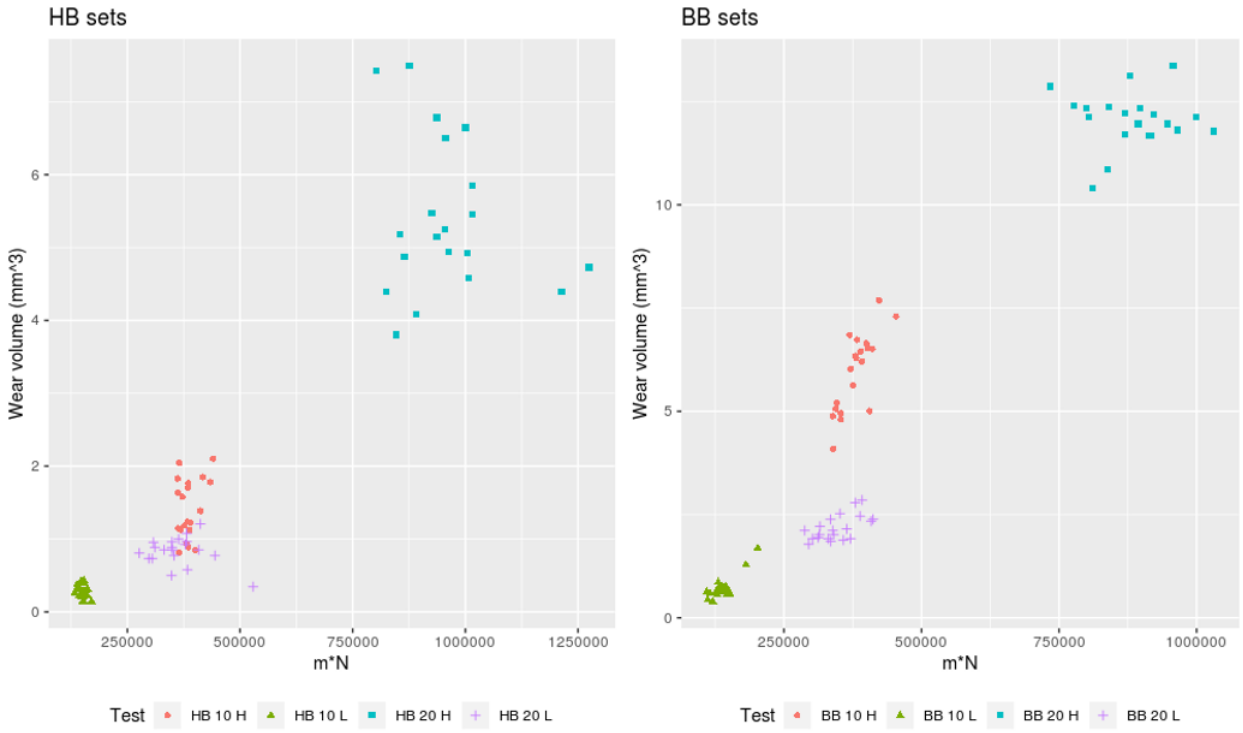

Figure 2.7: Wear volume over distance-load index.

Test Case 2: Dacite Microblades

Experiment Protocol:

A second case study using experimental dacite microblades to incise wood (Douglas fir; *Pseudotsuga menziesii*) provides a useful demonstration of the OpenHaft's utility in an experimental archaeological context. A set of 10 microblades were individually set in a wooden side-slotted haft fixed to the OpenHaft (Figure 2.8), and were used to cut a series of incisions into green, peeled, Douglas fir, parallel with the grain of the wood. Individual stroke lengths were 10cm long, and the microblades were recorded after two sets of 50 stokes each (500cm CSD and 1000cm CSD). Microblade wear was recorded using 3D digital photogrammetry and

measured with QGIS processing scripts for lithic use wear quantification (Waber 2015). It is important to note that the wear recorded for this experiment is not "microwear"—the striae and polish that is often identified as use wear—but rather "macrowear": damage to a tool's edge manifesting in microflake scars and crushed edge.

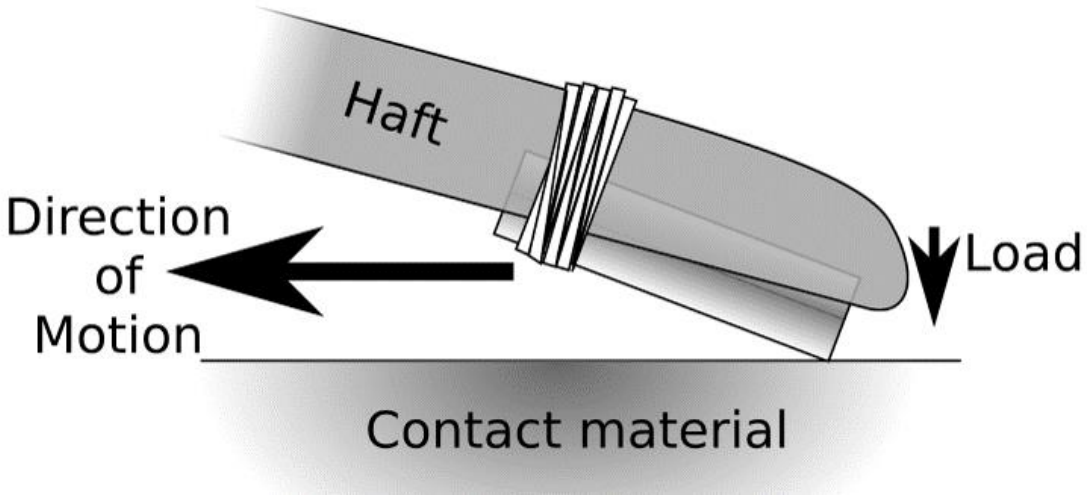

Figure 2.8: Diagram of hafted microblade used in incising task.

The primary question addressed in this experiment is whether load has a consistent and ongoing effect on use-wear accrual on experimental microblades. To that end, the following hypotheses were tested:

1. Edge load will be generally consistent between experimental sets. Without a change in the task, contact material, or tool morphology, there will be no substantial change in individual load patterns between the first and second sets of strokes.
2. With equivalent stroke sets (50 strokes at 10cm each for 500cm CSD) with the same lithic tool design on the same contact material, the total load, expressed as the intregral of load over time, will be nearly identical between experimental microblades.
3. The rate of microblade edge wear will be rapid for the first set, but will taper as the blade dulls and thus becomes more robust. The first set will exhibit more wear on each microblade than the second set.

Results and Observations:

Hypothesis 1 is supported based on a basic comparison of individual peak values; the maximum stroke load patterns were roughly equal between the first and second round of tests (Figure 2.9). This broadly reflects the common variables of the microblade tool, the task (longitudinal incision), and the contact material (soft wood). A Wilcoxon rank sum test comparing set 1 and set 2 peak heights for all experimental microblades indicates that the null hypothesis (the peak height distributions are statistically equal) may not be rejected ($W = 117220$, $p = 0.1113$). However, a closer examination shows some variation between the first and second rounds of tests. Comparing set 1 and set 2 peak load values for each individual microblade yielded only two samples with equal distributions (MB18 and MB22). A visual examination of smoothed line plots (Figure 2.10) supports this, but also provides insight into a general trend that is not immediately apparent in the basic data: peak load tends to increase as an experiment progresses. In almost every case, the early strokes started with between 2 and 3kg of load on the tool's edge, and then crept up, finishing higher than 3.5kg, and often over 4kg.

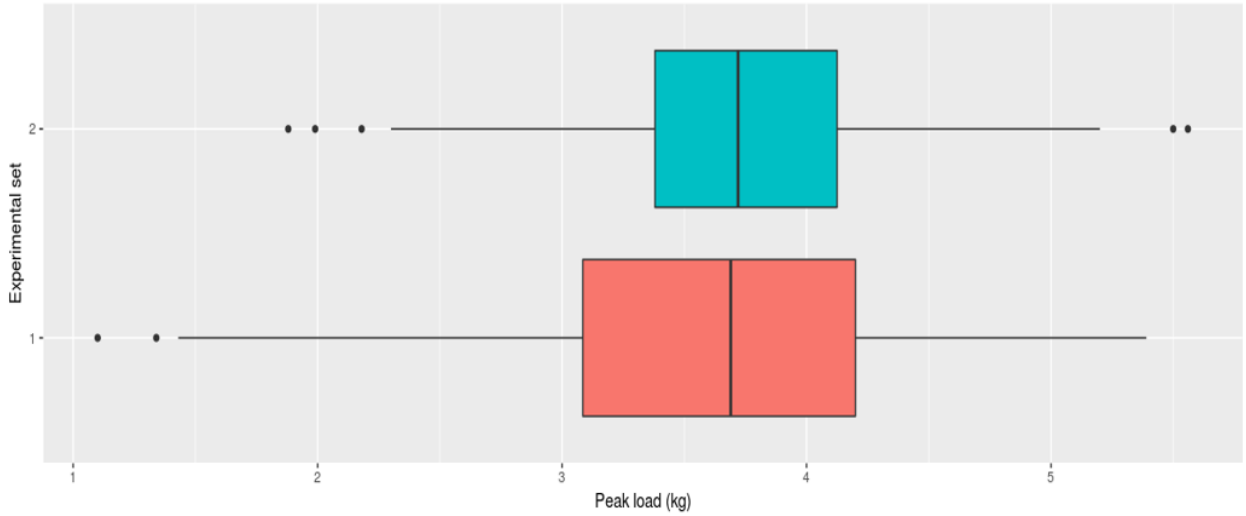

Figure 2.9: Peak load comparison between experimental sets (all microblades).

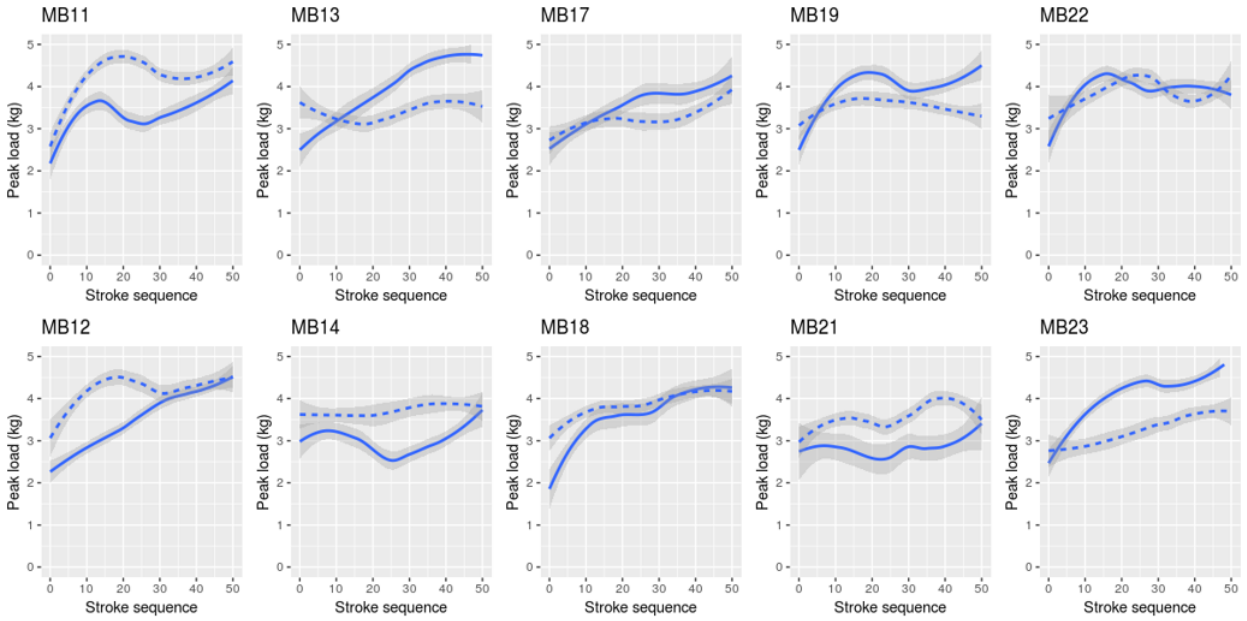

Figure 2.10: Peak load progression over experimental sequences for experimental microblades. Solid lines indicate set 1 (0-500cm CSD); dashed lines indicate set 2 (500-1000cm CSD).

This reveals a gestural pattern of initial cutting with more tentative strokes, often while relatively substantial early edge flaking occurs, including end-spalling and total blade destruction. As the user came to understand the blade's durability and optimal angle of use, the peak load would generally increase and remain relatively consistent throughout the rest of the test. This is visible in the violin plot of peak loads (Figure 2.11), where the peak data for each microblade test set quite consistently clump in the upper range, with fewer lower values. Like the variation in load between harder and softer graphite, this variation in stroke strength within tests shows the reflexive relationship between tool, contact material, and user in a lithic context. The tool user performs a gesture as dictated by the task (incising straight lines of consistent length in soft wood demands a series of generally consistent short-duration linear strokes) and within that parameter adjusts his or her gestural force depending on the feedback from the material (as the lithic edge stabilizes and the microblade's setting in the haft is shows to be firm, force increases and plateaus at a level that is sustainable over the course of the task.

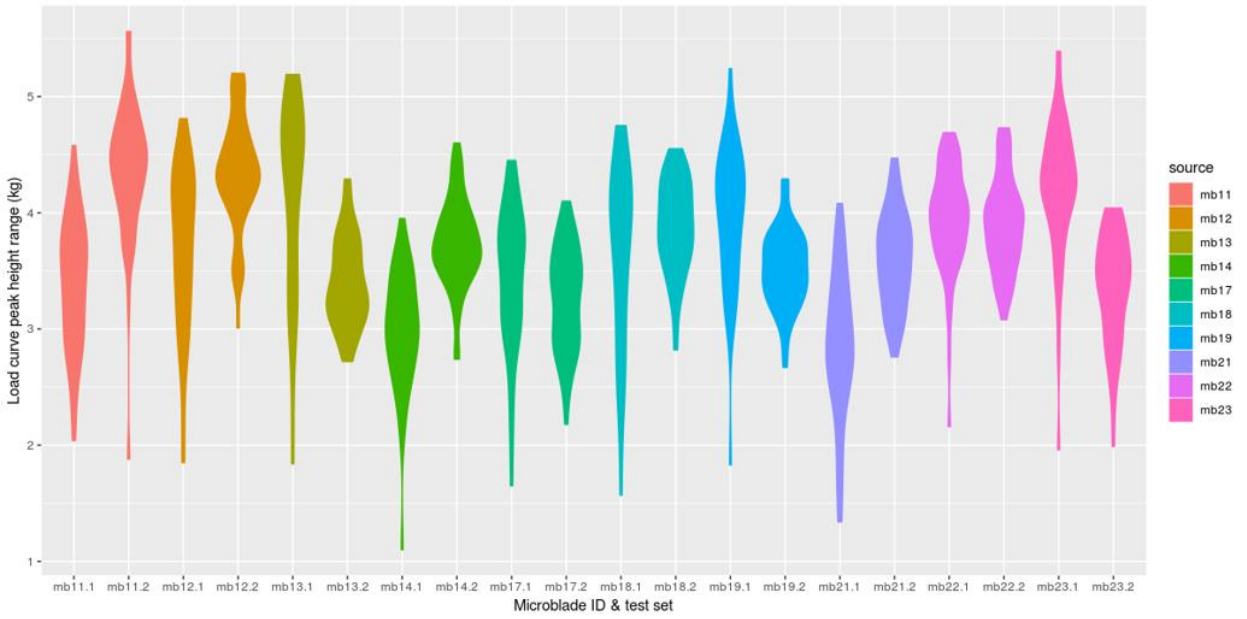

Figure 2.11: Violin plot of peak load distributions within experimental sets. Note the general clumping towards the upper load range for most microblades.

Notably, cases where the microblade became dislodged in the haft are visible as dramatic dips in the line trajectory (i.e., MB11; MB14 set 1). In these cases, the experimental set was paused, the microblade re-set and re-bound, and then the experiment continued. Evidently, the re-setting of the blade in the haft also appears to have re-set the tool user's confidence in the tool, and the maximum stroke loads are similar to those at the outset of the task rather than the most recent successful stroke.

While the variation between peak loads for individual microblades is notable, the assemblage-scale consistency shows that maximum force applied during a set of strokes is consistent as long as the task, tool, and contact material are also consistent. On this basis, Hypothesis 1 can be supported. The variation between experimental sets for individual microblades generally represents the specific conditions of the relationship between the tool user, the tool, and the contact material; strokes conducted with firmly hafted microblades yield load values consistent to each other, and the stroke patterns are comparable. Similarly, loose or recently hafted microblades result in loads consistent with each other, and inconsistent with firmly hafted microblades. Thus, we may see how a proven, sturdy tool permits the user to settle into a firm and decisive action, while uncertainty in the tool results in more tentative gestures.

Hypothesis 2 is rejected. Despite the similarity in peak load values, the integral of the load values is not equal between sets 1 and 2 (Figure 2.12). This is not so much a function of the maximum load expressed during a stroke, but rather due to strokes in set 1 being of marginally longer duration than those in set 2 (Figure 2.13). A Wilcoxon rank sum test supports the identification of the stroke durations of the two sets as not equal ($W = 174830$, $p < 0.0001$). It is uncertain what factors contributed to the difference in duration between the two sets, but it appears to have been a consistent difference within the experiments; set 1's strokes were consistently longer than set 2 (Figure 2.14).

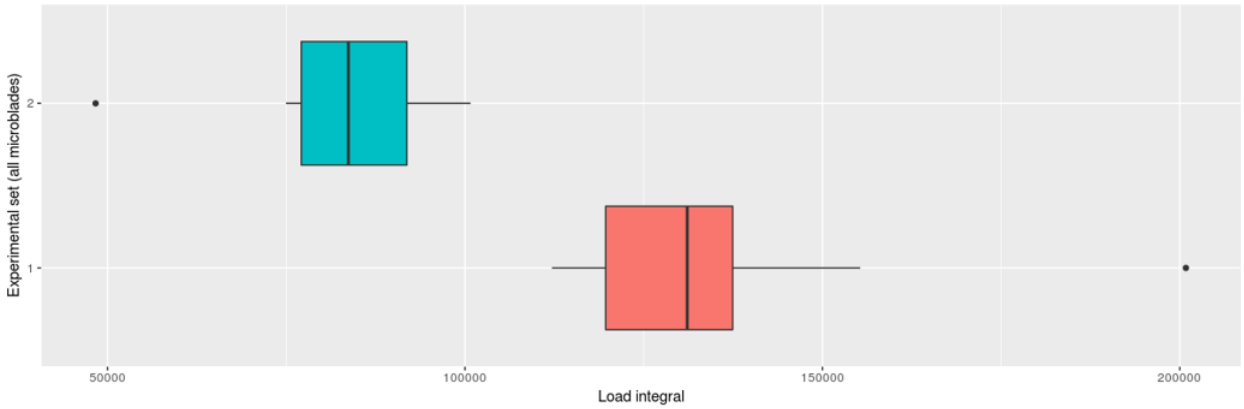

Figure 2.12: Comparison of load integral distributions for experimental sets (all microblades).

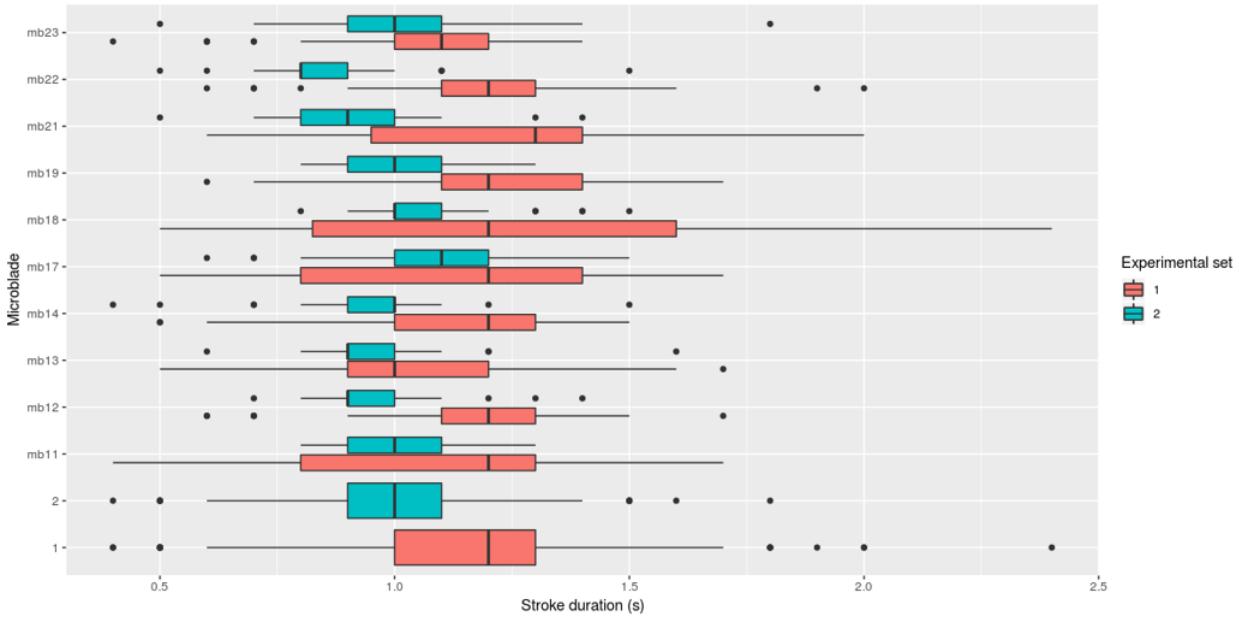

Figure 2.13: Stroke duration by experimental set. The bottom two boxes represent the entire group of microblades from sets 1 and 2, respectively.

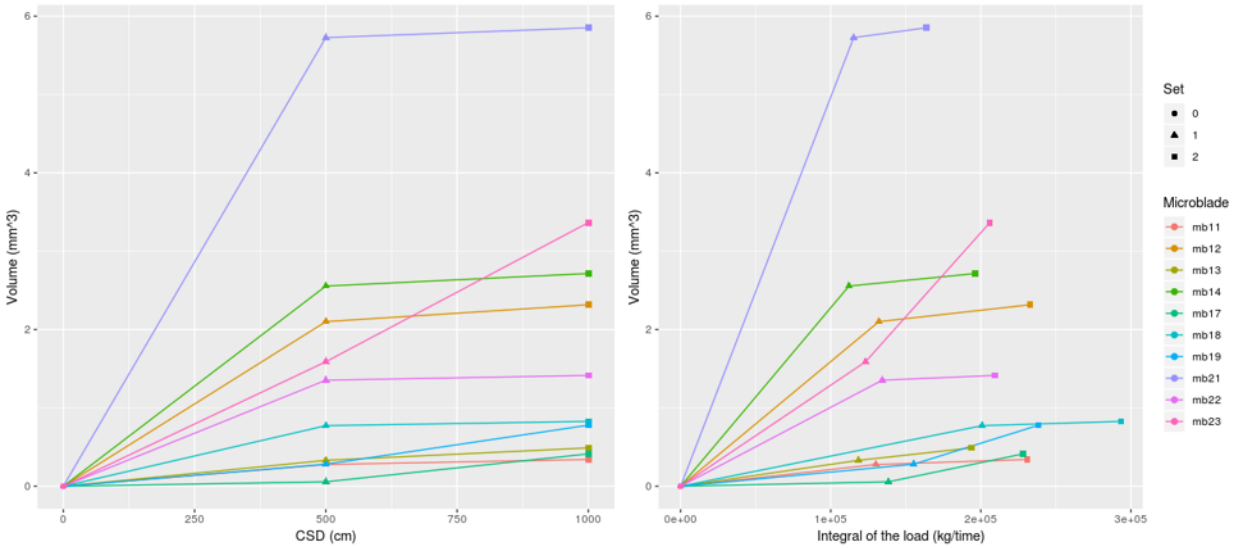

Figure 2.14: Microblade wear rate by CSD and load. Note the plateau effect in both cases. If the lighter load from set 2 was the factor behind the wear rate taper evident in the CSD-based plot, one would expect a straight line rather than a similar taper in wear volume.

Finally, hypothesis 3 holds in some cases, but not all. In most of the cases where substantial wear was recorded following set 1, the rate of wear decreased for set 2. This can be seen quite clearly in plots marking the volume of wear over CSD and the integral of the load,

respectively (Figure 2.14). It is especially evident in the former case, where at least five of the microblades demonstrate the pattern of rapid initial wear tapering to very minor wear over the second 500cm CSD. This pattern is logical, as early wear represents the sharp blade edge flaking and crushing away, leaving a more blunt, robust edge at a thicker point on the microblade's cross-section that is not as susceptible to damage (Figure 2.15). This blade wear stabilization stands in marked contrast to the graphite rods, which wore at a consistent and linear rate across multiple test sets. Another explanation is also appropriate: in at least the case of MB21, a substantial section of the microblade was spalled off during use—opposing pressure from the haft and the contact material pried off a portion of the blade, similar to flaking a burin or un-capping a beer bottle. This same effect occurred during MB23's second set.

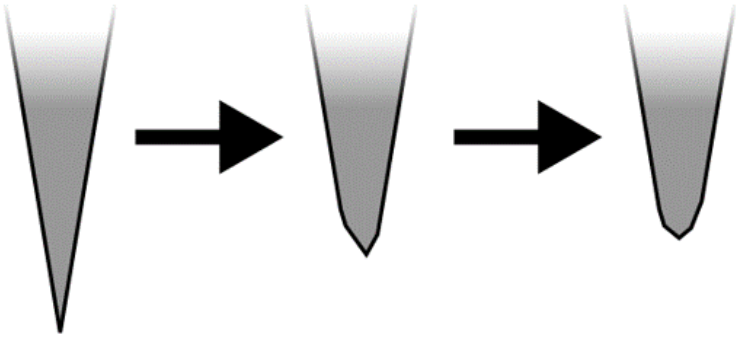

Figure 2.15: Idealized example of edge attrition relative to surface contact area.

It must be noted, however, that the pattern of a high rate of wear followed by a plateau was not observed in every case. At least four of the microblades exhibited a generally consistent rate of wear across both test sets (MB11, MB13, MB17, and MB19). These examples may represent cases where the edge started in a more robust state (i.e., was not as sharp initially), or where the microblade remained intact throughout the test and damage was restricted to the working edge rather than including spalling. Clearly, even in the case of lithic artifacts with relatively high levels of morphological consistency, such as microblades, the unique variations in shape, haft position, etc. affect the accrual of edge damage. This is relevant to the original question of whether load affects edge wear. In a general sense, load (as an integral component of

gesture) does directly affect the accrual of use wear, but it is only one of several factors. This experiment shows that the latent complexities of seemingly simple gestures informed by the reflexive relationship between user, tool, and material may result in substantial differences in edge damage. Load does affect wear, but the tool-user relationship affects load.

2.3 Conclusions and Discussion

The confirmed and partially-confirmed hypotheses generally conform to the dictates of common sense: subjective concepts of "light" and "heavy" load are identifiable in quantitative data; higher load results in more wear; greater CSD results in more wear; using the same tool for the same task on the same material results in consistent load over time; sequential wear patterns are informed by tool durability. However, the rejected hypotheses are very interesting. First, the difference in load integral values between 100m and 200m CSD sets (graphite tool) was not proportional to the difference in distance. This reflects the complexity of what one may (superficially) identify as a simple linear stroke. The stroke is not a uniform gesture, with equal force exerted throughout the action. Rather, it includes several action phases: the initial contact and loading, acceleration, deceleration, and unloading/separation from the contact material. Load during these phases is not expressed as a square wave or binary operator, but rather as a curve. Therefore, doubling the stroke distance does not double the load effect but rather it extends the middle portions of the gesture.

The results of the microblade experiment emphasize the complexities of gesture, and how those affect the development of lithic edge damage. Even with uniform CSD and a seemingly consistent task, variation in the technological system affected both gesture and wear. Perception of tool reliability appears to influence load–the growth of the tool user's confidence in the tool's edge is visible as load increases over the course of a test. Similarly, the same test performed by the same user at two different times appears to have resulted in different patterns of action in regards to stroke duration and thus total load applied over time. This demonstrates the importance of gaining an understanding of the human element of the technological system before

using use-wear to parse tasks or contact materials from archaeological material. Furthermore, this experiment shows that lithic tools do not wear uniformly even under similar conditions. While several tools exhibited regular patterns of wear (albeit with a different pattern after the initial action resulted in edge dulling and thus greater robusticity), others experienced drastic losses in material as relatively large sections were abruptly spalled away. Others still broke completely prior to completion of the first set of 50 strokes. A larger flake tool would presumably be more durable than a microblade, but the microblade experiment demonstrates how seemingly minor variations in tool morphology may result in major variations in wear and tool application. Further research is needed in this vein.

Both the graphite and the microblade tests illustrate the variability of gesture within seemingly uniform tasks, as well as demonstrating the utility of CSD and OpenHaft for gesture quantification. This is an important development for experimental use wear studies, as controlling gestural variables in lithic experiments is fundamental to wear analysis. In both case studies, the reflexive relationship between user, tool, and material is clearly shown. Whether in the form of load variation due to tool structure or material durability or in the difference in use-life volume evident in CSD values, human input variables have a significant effect on tool wear, and cannot be ignored. CSD and OpenHaft provide genuinely low-cost, accessible, and adaptable solutions to gesture quantification in lithic experiments.

Chapter 3 Open Source 3D Digital Photogrammetry and GIS for Quantifying Edge Damage on Experimental Stone Tools

3.1 Introduction

Geographic Information Systems (GIS) offer a powerful toolkit to archaeologists interested in lithic wear studies, providing a suite of resources ideally suited to detailed and nuanced analysis of surfaces at any scale. With recent advances in 3D digital recording and measurement of archaeological artifacts, such as digital photogrammetry and laser scanning, GIS principles can now be applied to the analysis of lithic surfaces (e.g. Davis et al. 2015): microlandscapes that reveal alteration and transformation related to tool use-life histories. This paper presents a method for using open-source GIS tools to quantify edge damage on experimental lithic artifacts, with a specific focus on mapping wear on microblades. The method is a synthesis of techniques, adapting surface analysis tools used in erosion and hydrology studies, and previous methods of lithic reduction quantification (Kuhn 1990; Eren and Sampson 2009; Eren et al. 2005) and edge wear mapping (Bird et al. 2007; Schoville 2010; Schoville et al. 2016), to measure the total volume of material lost from a tool's edge, and where the wear occurred. Researchers using this GIS toolkit can expediently and objectively quantify and map edge damage on stone tools, in a replicable way, so that tool use may be measured and compared between experimental tools at the individual and assemblage scale, and thus build a more robust and detailed use-wear reference collection.

GIS-based edge wear mapping is based on raster difference analysis, the same principles and method behind hydrographic erosion analysis. Basically, an experiment's lithic artifact is recorded in 3D before and after a use-wear experiment, then the worn-state digital surface model (DSM) is subtracted from the DSM of the artifact in its un-used state. Differences between the two surfaces—damage (wear, flaking, etc.) that occurred during the experiment—is precisely quantified and mapped in the output file. The 3D digital models of the lithic surfaces may be expediently (and inexpensively) recorded using 3D digital photogrammetry, so lithic tools may be recorded at intervals during an experiment and repeatedly over a tool's use life.

These methods, while theoretically applicable to microwear quantification, are discussed here in relation to "macrowear". That is, rather than examining the development of surface wear—polish, striae, etc.—this paper addresses the microflaking, chipping, and general edge damage resulting from tool use that leads to a tool becoming dull. This avenue of research has been somewhat neglected in many recent lithic use wear studies, with far more attention going to microwear, especially the incorporation of super-high resolution 3D scanning technologies (Evans and Donahue 2008; Evans et al. 2013, 2014; Macdonald 2013). However, "macro" wear (henceforth referred to as edge wear, or edge damage) offers fertile ground for developing a more nuanced understanding of cultural and economic factors around lithic technologies while they were in use. "Sharp" and "dull" are categories on a socially-defined spectrum; by identifying how archaeological cultures perceived that spectrum we may learn more about toolkit design, curation practices, uses of raw material, and other technological strategies. Furthermore, micron-scale surface wear was unlikely to have been a salient factor in emic perceptions of tool utility in antiquity, but whether a tool's edge was sharp or dull no doubt mattered.

Reference analogies through lithic experimentation are vital to understanding the relationship between observable edge wear on archaeological specimens and an archaeological tool's use-life history. Experimentation permits archaeologists to periodically record wear development, and monitor how different tools, materials, and gestures affect how wear manifests over the course of an activity. In relation to dullness, researchers can monitor how quickly a tool loses its optimal edge, and if there is a plateau effect or tipping point after which further edge degradation (and perhaps edge effectiveness) is negligible, and apply these findings to archaeological samples. Understanding what a reasonable use-life might have been for an archaeological lithic tool is important for structuring an experimental protocol meant to generate analogous wear patterns. The 1000 strokes used in recent studies (Giusca et al. 2012; Macdonald 2013) may reflect a plateau in surface microwear development (Giusca et al. 2012), but could be far beyond the practical limits of a lithic edge used for processing a hard material, such as antler or bone. Without understanding the rate of edge damage accrual on a tool, it is possible that after the first 100 strokes the blade was functionally equivalent to a spoon in terms of cutting capabilities, and thus subsequent surface microwear development is moot since the

archaeological counterpart would have been discarded or retouched immediately upon becoming dull. Identifying what constitutes a "dull" edge is therefore integral to understanding use wear. Monitoring changes in edge morphology over the course of an experiment can provide insight into dullness. By using replicable, quantitative methods, changes to edge morphology may be compared across tools and between researchers. This approach requires techniques and technologies that achieve precise and accurate results but operate at a scale larger than that of those used for microwear analysis.

GIS analysis is ideal for quantifying edge wear, especially as it develops over the course of an experiment. The raster difference analysis provides precise, objective, replicable data regarding changes to a tool's edge morphology between each sample. This permits robust comparisons between rates of wear for tools in specific applications, and used on certain contact materials. On that basis, comparisons of experimental wear characteristics should reflect realistic use lives, rather than arbitrary stroke counts. Those use-lives may be defined using GIS quantification of edge wear volume: if gesture and material are consistent over multiple samples, a taper in the rate of material volume loss represents the dullness plateau effect discussed above. Of course, tools may have been discarded or retouched prior to reaching the dullness plateau, which may be identified by comparing experimental wear quantification maps to those of archaeological samples. In the latter case, a fresh "virtual edge" must be extrapolated for DEM difference analysis. This follows Kuhn's (1990) and Eren et al.'s (2005) unifacial retouch indices, but with the added element of 3D surface interpolation for wear volume quantification. Since the difference raster maps absolute volume (which may then be calculated against the original DEM to determine percentage of material lost and wear intensity across the artifact's edge), concentrations of wear can be identified and related to different actions.

One very useful aspect of GIS-based edge wear quantification is the spatial aspect of wear. Specifically, mapping edge wear on tools indicates the position of the primary operational unit(s) (Bird et al. 2007; Schoville 2010; Schoville and Brown 2010). Mapping edge wear is especially useful in contexts where tools follow a relatively standardized morphology, but poor preservation of organic hafting elements has resulted in the recovery of only lithic tool

components. Microblades are a good example of this, where blade form varies little, but modes of hafting and use can range from sickles to knives to spear armatures. In such cases, edge wear may serve as a proxy indicator of hafting (hafts hypothetically shield tool edges from wear-related damage), and thus tool form, facilitating far more robust interpretations of tool use practices and site activities than is possible with a more fragmented understanding of how complex multi-component implements were constructed.

The GIS edge wear quantification workflow includes four basic steps: 1) recording and production of 3D digital models of the experimental tool at each stage of use; 2) alignment of 3D digital models; 3) data preparation in GIS; and 4) GIS surface comparison between the digital models. The process uses three different software packages. Model construction is achieved using 3D digital photogrammetry. The principles of digital photogrammetry have been described in detail elsewhere (see Gajski et al. 2016). In brief, angular relationships between common pixel landmarks in multiple overlapping digital photographs are calculated for each image pair, revealing the geometry of the photographed object and the relative camera positions, creating a point cloud of XYZ surface coordinates. Point clouds may be rectified—virtually shifted, rotated, and scaled—to represent real-world dimensions and locations based on control points with real-world coordinates. Digital photogrammetry is scalable, and the size of rendered surfaces are effectively only limited by the capabilities of the image capture hardware. The method described here uses high-resolution macro photography to capture detailed images of experimental microblades at different stages of wear, achieving a practical balance between image resolution and dataset size, using 18-24 images per surface rather than several hundred.

Three-dimensional digital recording of the experimental artifacts was achieved using MicMac, a free, open-source digital photogrammetry software package supported by IGN (French National Geographic Institute), designed for each step of a digital photogrammetry workflow (Duarte et al. 2016; Pierrot-deseilligny et al. 2015; Rupnik et al. 2017). MicMac was selected over the increasingly popular Agisoft Photoscan photogrammetry software for three reasons: 1) MicMac provides superior tools for masking a scene based on a single master image, and for model scaling and rectification; 2) trials with identical image sets have showed the dense

point clouds produced by MicMac to be consistently comparable and often superior in point density and surface quality to those produced by Photoscan (Chiabrando et al. 2015); 3) MicMac is free and open source while Photoscan is commercial software. It should be noted, however, that MicMac does not have a user-friendly graphical interface that permits easy "one-click" functionality. Users must understand the appropriate settings for each step and enter text commands in order to successfully produce 3D digital models. An example script is provided in Appendix F.

Digital model alignment used CloudCompare, a free, open-source program that quickly transforms and edits high-volume 3D point clouds. Cloud Compare offers two-stage point cloud alignment, with an initial "rough" alignment achieved through manually selected common control points, and a second automated "fine" alignment using root-mean-square difference. Informal trials indicate a vertical surface difference of less than 0.005mm at any point following two-stage point cloud alignment. This level of accuracy is suitable for edge damage quantification at the scale used in this process. The transformed point clouds are saved as delimited text files for access in QGIS. The Cloud Compare cross-section tool generates a perimeter profile line, which is then saved as an ESRI shapefile (SHP). Cloud Compare is also used to "clean" the point cloud, removing outlying points and smoothing data artifacts resulting from photogrammetric reconstruction of smooth, black lithic surfaces. Detailed instructions for point cloud cleaning and alignment is provided in Appendix G.

The GIS procedures for microblade edge wear quantification and mapping are split into three stages: surface modelling, DEM difference analysis, and perimeter wear mapping. All stages are carried out using QGIS 2.18[2], a powerful open-source GIS that also incorporates algorithms from other open-source GIS programs: SAGA and GRASS, accessed through the QGIS Processing Toolbox. Each element of the GIS workflow has been synthesized into a Processing script, providing a set of three easy-to-use tools for artifact edge wear analysis. The

2 QGIS version 2.18 was the "long term release" (LTR) at the time of these experiments, and the PyQgis scripting language used in QGIS 2.x Processing scripts, based on Python 2.7, is considerably more intuitive than the updated QGIS 3.x scripting language. Updated modules for QGIS 3 are planned for release in early 2020.

QGIS Processing script package is composed of three modules: “Fresh flake reconstruction”, “Worn flake reconstruction”, and “Lithic wear mapping”. The former two share functions for surface interpolation and flake curvature compensation, while the latter quantifies wear based on the reconstructed flakes.

Generating 3D surface models of microblades from photogrammetrically produced point clouds involves two steps: interpolation and curvature compensation. DEM interpolation uses point coordinates to generate a digital model of the microblade surface. With the high point density, an inverse-distance weighted interpolation algorithm produces a raster with 0.05mm resolution that has yielded high enough precision for good wear identification, yet a low enough file size for fast processing on a conventional personal computer. The default settings of the IDW algorithm and the grid resolution may be adjusted by editing the Processing script.

One challenge of adapting DEM difference calculation to quantifying lithic edge wear is that landscapes orient with a consistent vertical axis (the centre of the Earth is “down”). Changes in landscape morphology are easily measured as differences in elevation and thus raster calculation is based on measuring in one direction relative to a “flat” universal plane. However, because lithic flakes curve and twist over their lengths, it is necessary to establish a locally consistent arbitrary plane and to coerce the GIS into measuring towards that rather than along the true Z axis. To address this, the QGIS script creates a plane that stretches from edge to edge across the flake, effectively isolating the dorsal and ventral volumetric sections. Thus, dorsal and ventral wear may be independently measured, and total edge damage (where the edge material is entirely worn away) registers accurate volume loss. The local plane is realized by “flattening” the flake. Following methods used to analyze local elevation relative to river channels, flake DEMs are flattened by sampling elevation values around the flake edge perimeter, interpolating a trend surface from those elevations using SAGA’s “natural neighbour” module (Beutel et al. 2010), and subtracting that surface from the raw DEM (Figure 3.1). The resulting DEM provides the elevation for each grid relative to a virtual plane running between points on either edge of the blade. Wear is quantified relative to the centre of the flake rather than the centre of the earth.

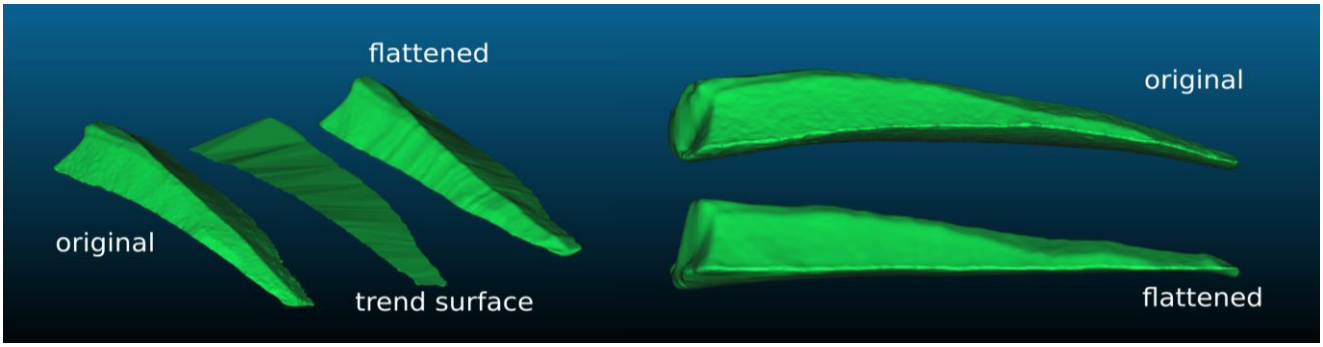

Figure 3.1: Microblade curvature compensation. The Z-axis values of the trend surface are subtracted from the curved microblade surface, resulting in the flattened microblade model.

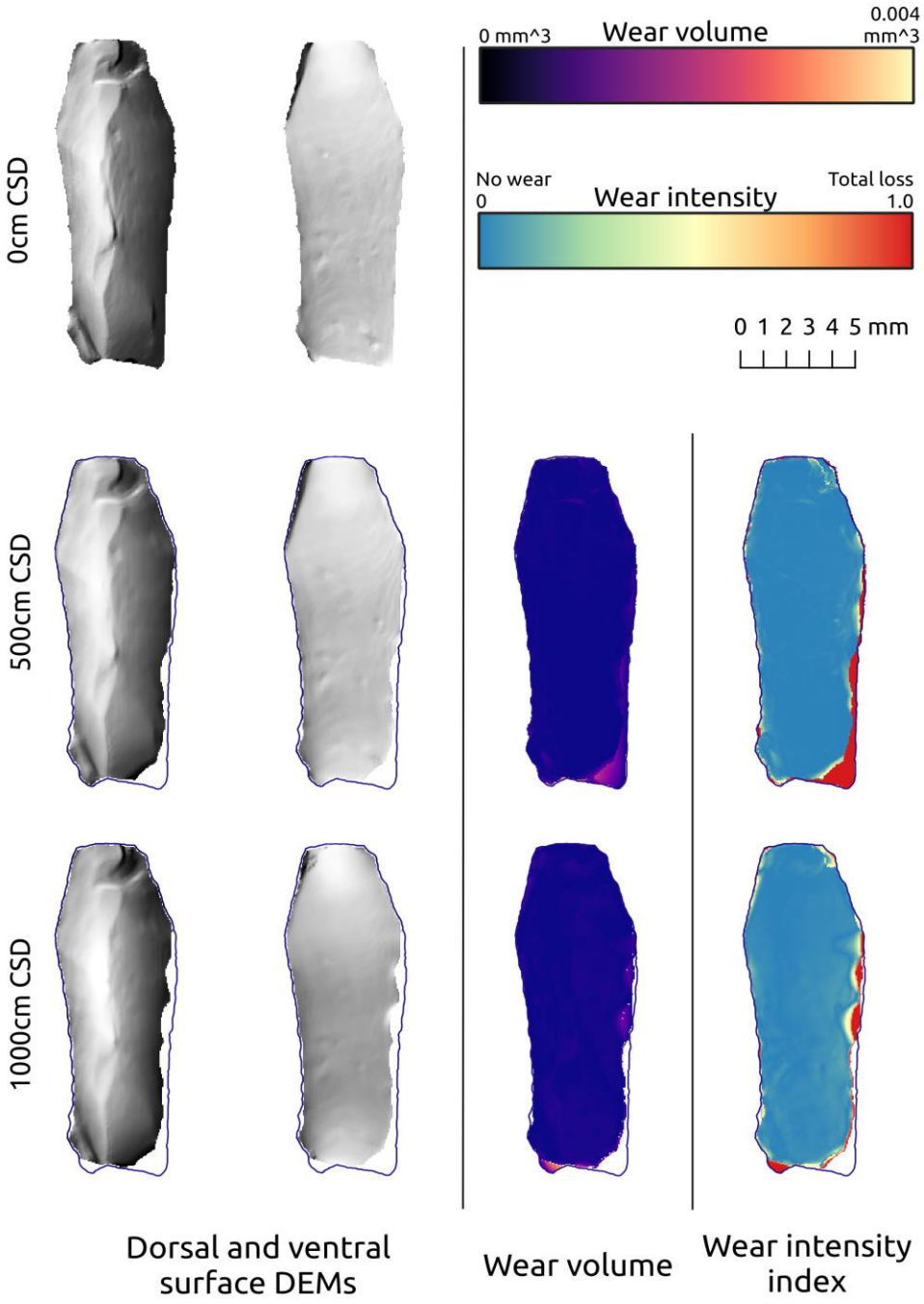

Figure 3.2: Digital surface model progression and wear rasters (volume and intensity). The dark outline marks the un-worn microblade perimeter. Note that the maximum wear volume value (0.004mm3) seems miniscule, but that is within a 0.05mm square (based on the grid resolution).

The QGIS lithic use wear quantification script produces rasters for both absolute wear volume and wear intensity, for both dorsal and ventral surfaces as well as the overall artifact. At its most basic, wear quantification subtracts the worn lithic digital elevation model (DEM) from the fresh lithic DEM, and the resulting difference raster represents material that was worn away during use. Higher grid values represent areas where more material was worn away (Figure 3.2). Dividing the difference raster by the fresh tool DEM produces and index of wear on a 0-1 scale, with 1.0 representing complete material loss where the edge has been entirely worn or chipped away, and everything less than 1.0 indicates flake scars and surface wear. The wear volume and wear index rasters may then be sampled to derive information regarding wear localization around the tool's perimeter.

The relationship of the flake curvature compensation method and the surface difference analysis is important, especially with experimental microblades. As flakes (micro- or otherwise) become detached from an edge during use, the local edge profile creeps in the direction of the flake removal (Whittaker 1994), resulting in very slight variation in the vertical positions of the worn and unworn microblade edges. The effect is slight, but noticeable in the surface difference rasters. Worn microblade DEMs with curvature compensation carried out using a trend surface based on unworn data exhibit an extremely low noise threshold (<1%) against the unworn DEM, while those with curvature compensation based on their own perimeter result in higher noise, and even "ghost flake scars" adjacent to areas of high wear. For this reason, the QGIS processing script demands unworn surface data for curvature compensation. This is easy enough in experimental contexts, but presents obvious challenges for wear quantification on archaeological samples where the unworn tool ceased to be "unworn" long before researchers could document it. Attempts to adapt flake edge reconstruction algorithms based on geometric indices of reduction (following Kuhn 1990; Eren and Sampson 2009; Hiscock and Clarkson 2005; Eren and Prendergast 2008; Eren et al. 2005) have not yielded adequate accuracy at the use-wear scale.

Two perimeter wear mapping options are available in the QGIS script: linear wear mapping and radial wear mapping. Linear mapping is used for linear artifacts such as microblades, where the edges are straight and parallel to the long axis of the tool. In these cases,

wear typically occurs on lateral margins and is positionally better understood relative to a proximal-distal spectrum than the case for flake tools with continuous margins and less linear morphology. In those cases, radial wear mapping is used, where edge damage location is expressed as a compass position relative to the centre of a stone tool (following methods initially used by Bird et al. 2007 and Schoville 2010). The QGIS method builds on Bird et al.'s (2007) and Schoville's (2010) by incorporating objectively quantified edge wear into the positional data. In the previous studies the edge wear was visually identified and subjectively classified as "light" or "severe", and was then manually registered on a presence/absence basis around the tool's perimeter on a digital image. Following individual tool mapping, the entire assemblage of lithic artifacts was synthesized, with rose diagrams of aggregate edge wear positions illustrating the parts of the tool that were consistently being used (Bird et al. 2007; Schoville and Brown 2010; Schoville et al. 2016; Schoville 2010). The QGIS wear mapping script quantifies the edge wear and records the wear (both volume and intensity) at each linear or radial position. This reduces the impact of subjective wear identification and qualitative categorization on the resulting data, and does not demand expertise in use wear analysis.

3.2 Case Study: Microblade Edge Wear

A case study was carried out to assess the viability of QGIS lithic edge wear quantification. A set of experimental microblades were used to carve a wooden object, and edge wear was measured using the lithic wear tracking scripts. The experiment was structured to test two core hypotheses regarding the development of edge damage on lithic artifacts:

- Hypothesis 1: Edge damage accrual will be rapid in the earliest stages of a tool's use-life, as sharp, friable edges are crushed and flaked away. Subsequent use should result in lower rates of edge wear because the edge left behind after the initial crushing will be less sharp and more robust than the initial tool morphology. This effect is tested by comparing recorded volumes of wear for individual microblades sampled after two rounds of a cutting task, each using identical gestures on identical contact material. The expected result is a greater volume of wear recorded following the first set than the

second; a constant—or greater—increase in the rate of wear would cause the hypothesis to be rejected.

- Hypothesis 2: On a side-hafted microblade used in an incising action, edge wear will be concentrated on the corner that forms the primary cutting component. Edge sections obstructed by haft or binding components will exhibit minimal (if any) edge damage. Wear location patterns may be assessed by mapping wear volume across the microblade surfaces and comparing the amount of wear each microblade experienced within each "zone". If the volume of recorded wear is uniform or random across all zones we must reject the hypothesis.

Both tests require precise data regarding edge wear. The first demands overall volumetric data to compare material loss between two stages of tool use. The latter requires edge wear data to be mapped in order to understand the spatial trends of wear relative to gesture and haft form. Thus, the tests are an ideal testing ground for the QGIS lithic wear scripts in an archaeologically relevant experimental context.

Materials

Experimental microblades were knapped by the author from fine-grained dacite collected from a lithic quarry locality in the Upper Hat Creek Valley, BC (Rousseau 2000, 2015). The microblades were singly hafted in a side-haft knife (Figure 3.3). The haft was made from cedar with an incised slot approximately 20mm long, 2mm deep, and 1.5mm wide. The microblades were held in place using a simulated sinew binding and a small leather pad (to prevent the blade from cutting the binding). The pad and binding was positioned on 2-3mm of the microblade's proximal (relative to the knife) edge, similar to bone hafts of the Arctic Small Tool tradition (Owen 1987). This hafting method was selected for several reasons: it sets the microblade firmly in the slot and prevents rotation during use; it is analogous to one of the two known microblade knife hafting methods used in northwestern North America (Flenniken 1981; Owen 1987); and it is fast and easy to un-mount and re-haft the blade, allowing efficient photographic recording, as opposed to using a mastic such as spruce gum, resin, or hide glue (Helwig et al. 2008; Pétillon et al. 2011; Rots 2010), which is less convenient and leaves residue on the lithic.

Figure 3.3: Detail of example side-hafted microblade. The leather pad and artificial sinew binding prevents the microblade from rotating out during high-torque applications.

Methods

The knife was mounted on an OpenHaft—an electronic tool for recording load in lithic use wear experiments. OpenHaft ensures that the load variable may be isolated for experimental sets, along with gesture type, stroke length, and raw material, thus permitting the researcher to identify which variable is most responsible for variation in wear (Waber 2016). In this experimental context, the load data is used in two ways: the live data stream is used as feedback to ensure that consistent load is applied throughout the test, and to identify wear variations relative to load fluctuation in the final volumetric data. The contact material used was green Douglas Fir (*Pseudotsuga menziesii*).

Prior to use, each microblade was photographed for 3D photogrammetric reconstruction of the unworn surfaces. Microblades were coated with a dilute solution of talc mixed in rubbing alcohol to coat the surface and provide enough surface detail for photogrammetry without significantly affecting the surface texture. The microblades were then hafted in the OpenHaft and used to incise the wooden contact material. The cutting action was unidirectional, with the

blade being drawn along the long axis of the piece of wood for 10cm, at an approximate contact angle of 25°. After 500cm of cumulative stroke distance (CSD) the microblade was unhafted and re-photographed. This was repeated until each microblade had been used for 1000cm (see Figure 3.4 for microblade reference image). The image sets were then processed using MicMac, and surface analysis carried out using the QGIS processing scripts. The output was analyzed using R (Appendix H).

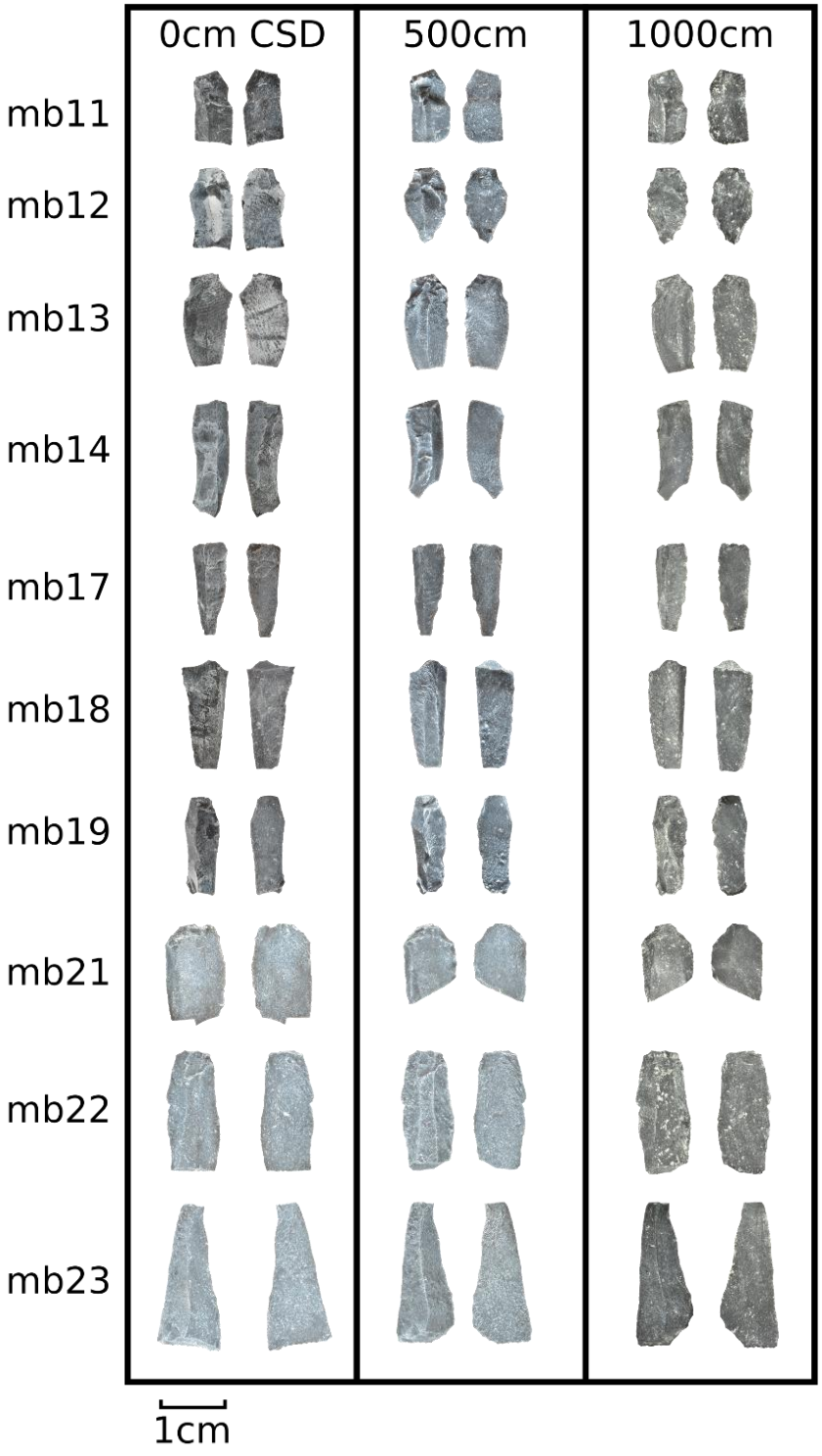

Figure 3.4: Microblade reference image.

Results

Hypothesis one may be accepted; eight of the 10 experimental microblades exhibited the predicted tapering rate of wear (Figure 3.5). In both of the other cases (MB17 and MB23), the lost volume was due to a spalling event rather than regular edge wear. This is similar to the spalling event observed on MB21 during the first round of strokes which resulted in its extremely rapid rate of initial wear. Among the other microblades, rates of wear were generally consistent with the hypothesis, with a relatively pronounced initial reduction in volume followed by much less pronounced wear as the sharp edges and less robust sections break off and wear down.

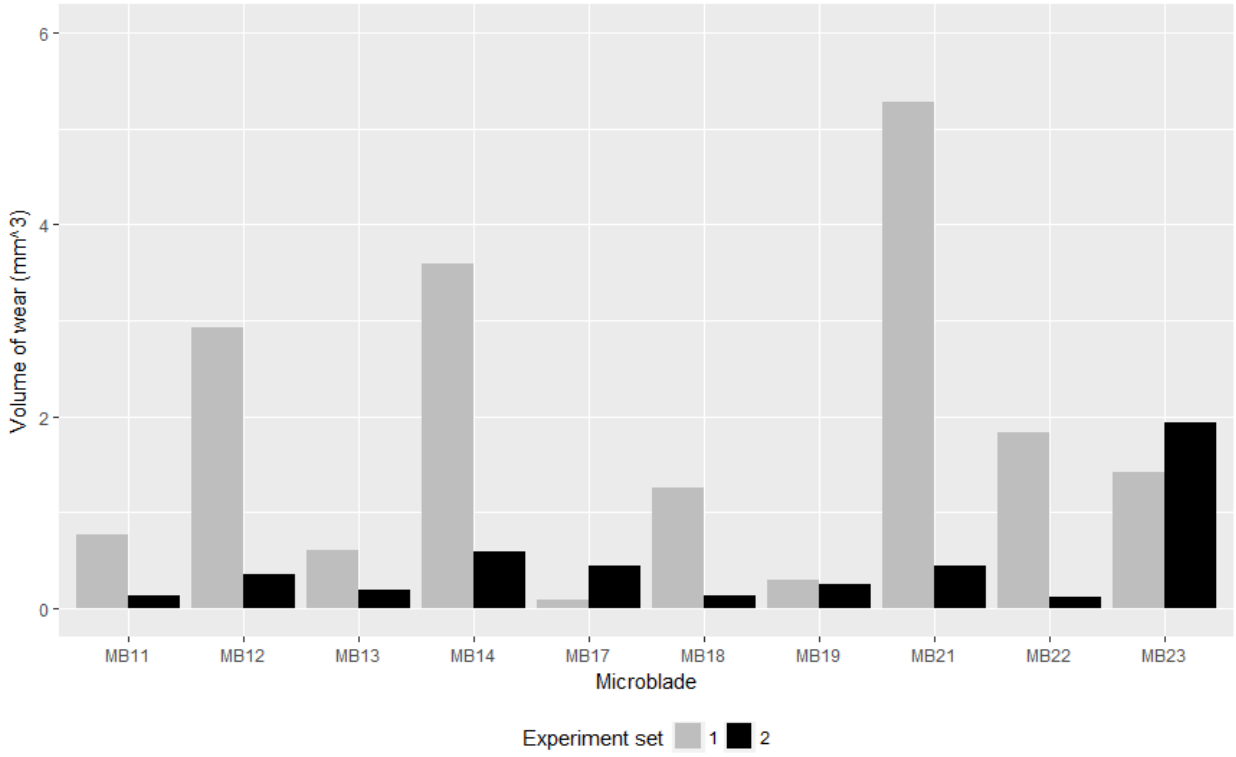

Figure 3.5: Wear volume per microblade: set 1 (500cm CSD) vs. set 2 (1000cm CSD).

Hypothesis two is supported: wear locations are very clearly identified, and the operational distal or proximal sections of each microblade consistently exhibit much higher wear than portions contained within the haft (Figure 3.6). Some microblades, such as MB11, exhibit

particularly clear patterns of edge damage concentration, with most of the wear at one position (right distal section for MB11), with minor wear on the adjacent medial section, and entirely negligible wear elsewhere on the blade. MB13 and MB19 exhibit similar patterns with wear concentrated on other sections. In contrast, instances of crushing or spalling damage, such as MB12 and MB21, respectively, show major material loss. In both cases, the microblades exhibit substantial wear—both volume and percent of volume—concentrated in sections of both lateral margins, and considerably less wear on the opposite section, which was hafted.

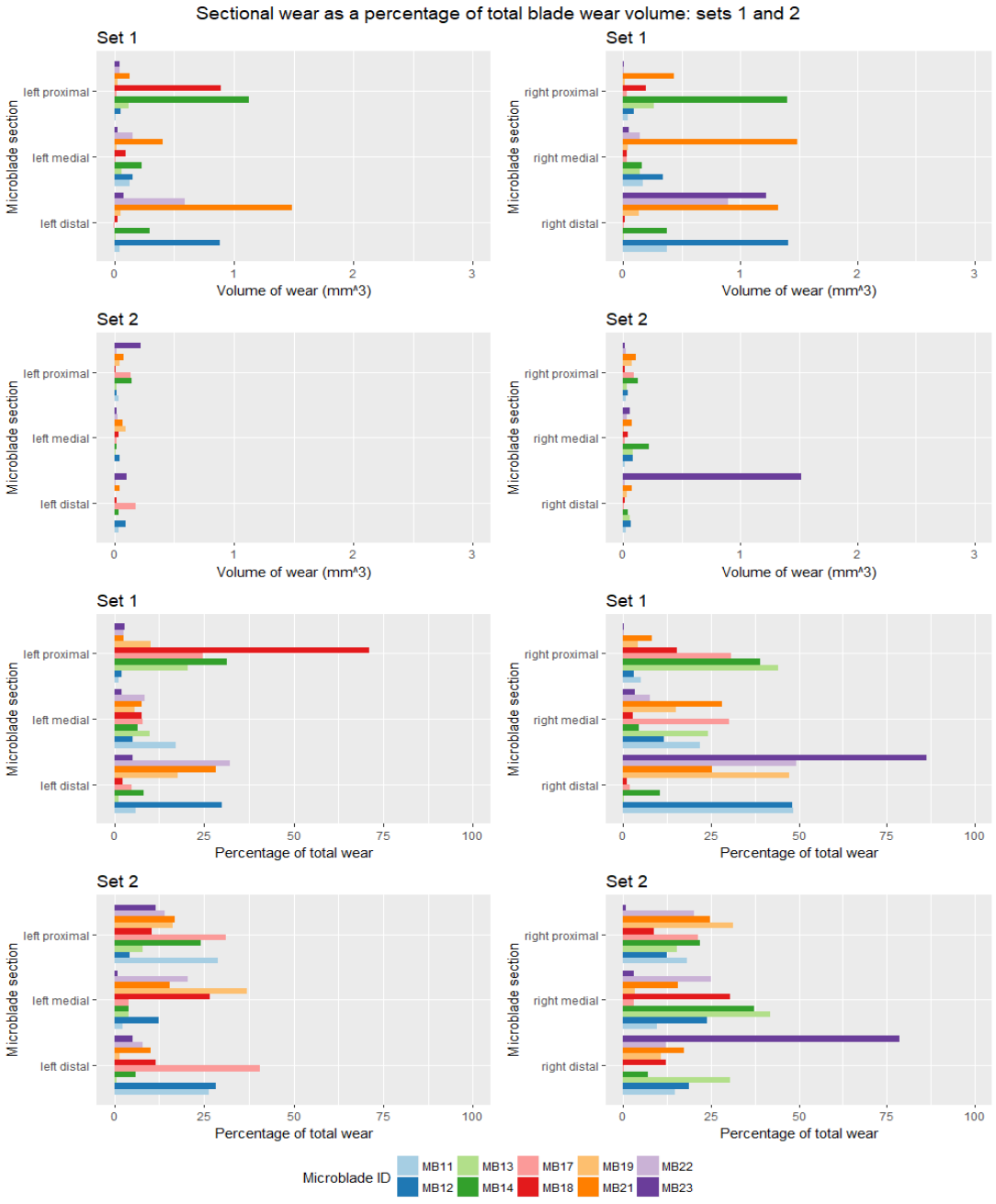

Figure 3.6: Wear volume plots. The percentages of total wear are calculated by comparing the wear volume of each segment to the total wear volume experienced by the microblade during that test session.

While the barplots provide a basic indication of the lithic material loss experienced by each microblade, the wear index rasters offer a much more detailed and nuanced perspective (Figure 3.7). Here, the specific areas of wear are clearly indicated and nature of the applications of the microblades may be seen, with high concentrations of wear on one corner of most

microblades (especially visible on MBs 11, 13, 18, 19, 22, and 23). The contrast between the first and second stages of wear is particularly interesting. Aside from MB23 (which experienced a spalling event on the distal corner), and MB17 (damaged by the haft binding), all other microblades exhibit considerably more intensive wear during the first set of strokes than the second. Some (MB12, 14, and 21) exhibit near-catastrophic spalling events, where very relatively massive sections of blade were removed. Minor edge wear following catastrophic spalling events is also visible, creating an interesting narrative regarding the use-life history of the tools where blades were re-hafted and re-used. That second stage edge wear provides an indication of hafting; MB14 evidently was worn on the left proximal portion, and MB21 exhibits wear on the right proximal, which provides orientation for the exposed sides (left and right, respectively) and the bound end (distal).

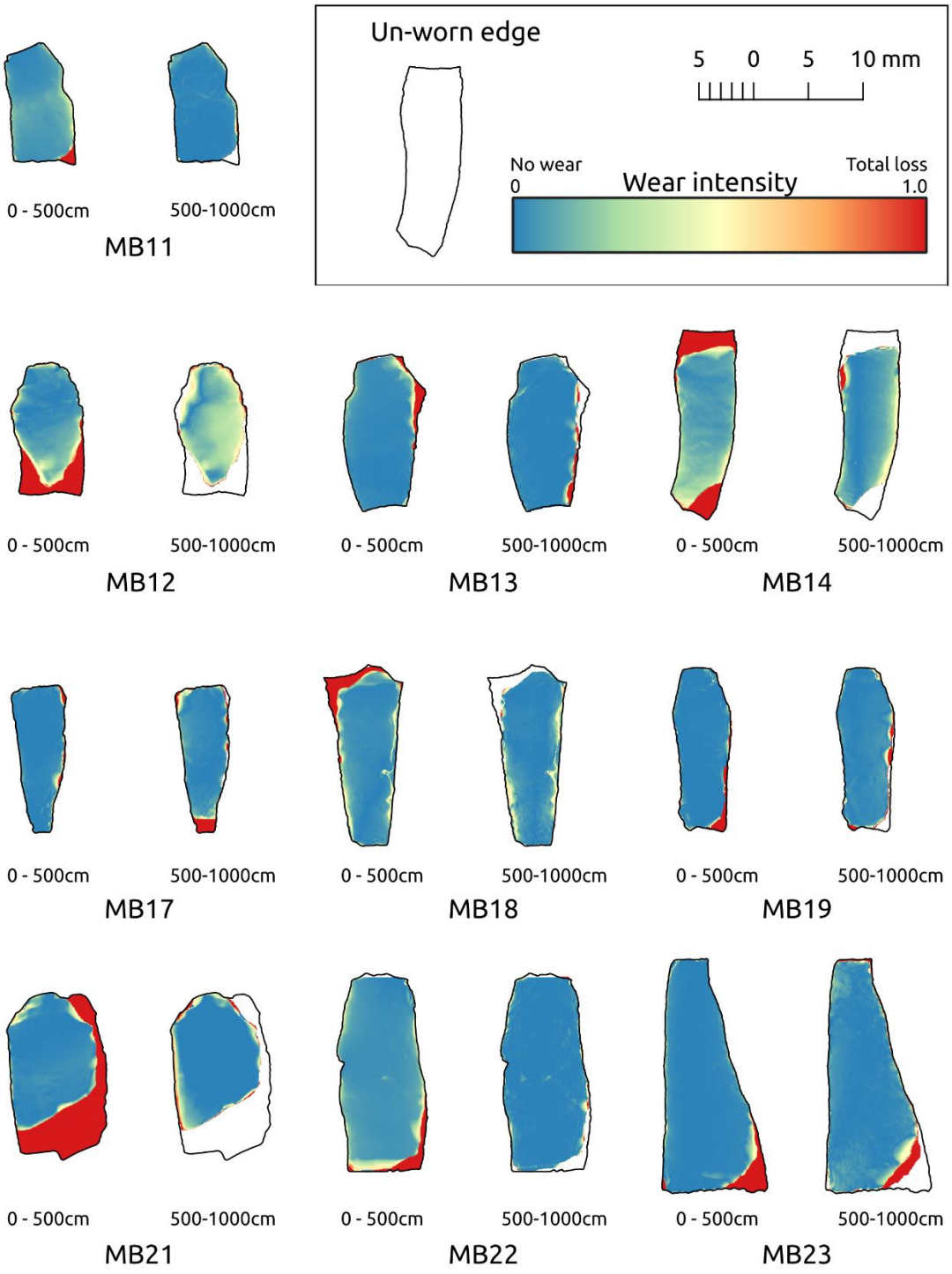

Figure 3.7: Wear intensity indices for all experimental microblades. Bright red values indicate total material loss.

3.3 Conclusion

The experiment carried out here demonstrates that the QGIS lithic edge wear quantification package may be useful for building a better understanding of edge wear formation. Photogrammetric documentation and raster difference analysis permits researchers to record and measure morphological changes to experimental stone tools over the course of an experiment, allowing archaeologists to make meaningful associations between factors such as hafting method/prehension, force, contact material, tool form, edge sharpness, and more with the accrual of edge damage. Furthermore, the rate of wear may be closely monitored to identify if and when a lithic edge is so crushed or flaked that it is effectively stablilized and not susceptible to further edge wear from that task. The corollary is that an edge will at some stage reach a point of damage where blunting compromises a tool's effectiveness. By observing this in the lab and comparing it to archaeological analogs, we can gain insight into economic factors around toolkit design and technological strategies in the past.

The workflow itself achieves one other important goal: it makes the technology necessary for meaningful and rigorous edge wear analysis accessible to the vast majority of archaeologists. No part of this method requires equipment beyond a digital camera and a personal computer. Not having access to a lab with a laser-scanning confocal microscope or similar equipment is not a barrier to entry for this research. Furthermore, all the software used in this project is cross-platform and open source, including the QGIS scripts. People are free to use, redistribute, and alter anything and everything as they see fit. Thus, independent researchers who do not have student- or site-licenses, and researchers from communities with limited funding for lab equipment or proprietary software licenses can engage with the research at the same level as those at the best-equipped university. The author hopes that by making the lithic wear quantification accessible to the greatest number of people, a community of users will advance the method in ways beyond his capabilities and imagination. QGIS scripts and example data are available at the following URL: https://doi.org/10.5281/zenodo.2781197

3.4 Future Avenues of Research

Two main lines of future research present themselves: expansion and refinement of the QGIS scripts, and application of the raster difference calculation principles to archaeological samples. QGIS scripts, while useful, are occasionally finnicky and unforgiving with the data inputs. This is especially the case with point cloud and raster alignment, where unexpected inconsistencies between worn and unworn rasters (such as a worn surface exceeding the bounds of an unworn surface due to sloppy alignment) can result in errors and frustration. Furthermore, software updates occasionally result in changes to program behaviour and require minor updates to elements such as module parameter details (i.e., for SAGA 2.3.1 vs. 2.2.0 after a QGIS sub-version update) or major updates, such as the impending complete rewrite in a different programming language to adapt the scripts to QGIS 3. Fortunately, since the scripts are entirely open-source, anyone can update and revise the scripts, and possibly (hopefully) port the functions to other software packages with spatial analytic capabilities, such as R.

The second line of research has more immediately practical benefits for archaeologists. The application of quantitative edge wear analysis to archaeological specimens has tremendous potential for expanding our understanding of past lifeways. The socially-defined spectrum of sharpness/dullness is fundamental to technological strategies, affecting all aspects of the *chaîne opératoire* from design, raw material acquisition and manufacture through use, maintenance, and curation to discard. Perceived sharpness is the ultimate arbiter of a tool's utility, weighing heavily on the economic calculation of whether to immediately carry out a task or to re-tool. The other components of that calculation recursively affect the subjective sharpness value; a tool user with ready access to prime material and the time to craft fine implements may hold a higher cognitive model of what constitutes "sharp" than another individual in a different material context, and may be more likely to discard suboptimal tools. In contrast, the person experiencing material scarcity, time stress, a lower degree of expertise (or pickiness) may use the same tools far beyond their counterpart's point of discard. On this basis, sharpness may be defined as much by immediate external factors as by embodied cultural standards of tool optimality. The first step to a good archaeological understanding of past perspectives on sharpness is the ability to measure how intensively tools were used, and how much edge damage was enough to prompt

somebody to discard a tool and refit. The QGIS wear quantification scripts should be ideal for this, except that they require an unworn surface model for the raster difference calculation, and that model cannot yet be reliably produced from archaeological lithics to the level of precision needed for wear analysis.

Extending the theme of wear-based tool morphology, archaeologists may also benefit from using wear location data as a proxy for evidence of hafting of flake tools. Edge wear location mapping has already been used to good effect to understand hafting and application patterns of Middle Stone Age triangular flake points (Bird et al. 2007; Schoville 2010; Wilkins et al. 2012; Schoville and Brown 2010), but these studies have relied on subjective wear classification. Combining the proven wear mapping methods with robust quantified wear data would make such inferences even stronger. Furthermore, other hafted flake tool types from other places and time periods would benefit greatly from such research. Microblades are one such tool, and identifying how a blade was hafted is integral to understanding how it was used. Since microblades are too small for practical handheld use, hafts are absolutely necessary. The form of those hafts and the position of the microblades in them will reflect the design applications of the microblade edges. Scattered examples of preserved hafts include the long, side-slotted bone and antler points known from sites around the North Pacific rim (Christensen and Stafford 2005; Magne and Fedje 2007; Giria and Pitul'ko 1994; Hare et al. 2004; Waber 2011); solitary end-hafted microblade knives in cedar handles from Washington's Hoko River site (Croes 1995; Flenniken 1980); and side-hafted single microblade knives from the Arctic Small Tool Tradition (Owen 1987). In each case, the specific function(s) of the microblades, and thus the action and behaviour of the tool users (what we are ultimately interested in), are closely related to the haft form. After all, following the principles of *chaîne opératoire, a* technology is as much defined by the gestures used to apply it as it is by the objects employed in the task action (Tostevin 2011; Boëda et al. 1990; Pelegrin 1990), and hafts are the interface that articulate human gesture and the basic lithic operational unit: the edge. If quantified wear mapping can be used to identify and map hafting patterns in an objective and replicable manner it has the potential to fill in large gaps in the archaeological record where poor preservation of organic material has deprived us of the greater part of many tools.

Chapter 4 GIURF and GIS: Testing GIS Automation of Kuhn's Geometric Index of Unifacial Reduction Flaking for Digital Edge Reconstruction and Wear Quantification

4.1 Introduction

Objective and replicable lithic use wear quantification has been a long-standing goal of many archaeologists, and recent innovations in technology have contributed to this endeavour. In experimental archaeology, several different methods for lithic use wear quantification have been developed, including super-high precision microscopy for surface roughness measurement (Evans and Donahue 2008; Evans and Macdonald 2011; Evans et al. 2014; Macdonald 2013; Stemp et al. 2015), image- and reflection-based flake scar definition (Lerner 2007; Stemp and Stemp 2001; Vardi et al. 2010; Vergès and Morales 2014), and GIS-based surface attrition quantification (Waber 2015; in press). However, these methods have hitherto been applied only in experimental contexts, where the tool could be observed prior to wear, and the conditions of use were known. In archaeological contexts the unworn tool of the past is inaccessible to the present-day researcher, so in order to quantify the extent of damage sustained by the artifact during use, the original form of the unworn tool must be modeled based on the existing worn form. This is especially the case for automated surface difference calculation. Whereas human lithic analysts can employ experience with lithic tools and pattern recognition from experimental and/or ethnoarchaeological analogs to cognitively model the artifact in question in its original, pristine state, computer-based methods require a more formal and definite "unworn" benchmark for comparison. To that end, treating the dorsal and ventral surfaces of a lithic flake tool (including microblades or other tools with un-modified or smooth working edges, such as expedient flake tools, microliths, or many ground stone tools) as the convergent planes of a triangular prism, and digitally modeling the prism beyond the damaged margin of the tool, creates a virtual representation of the tool in its unworn state. This forms the basis of the comparison between worn and un-worn edges. Because the largely-intact dorsal and ventral surfaces are available to be recorded by the archaeologist, using 3D digital models of lithic artifacts, it may be possible to virtually reconstruct the original, un-retouched/damaged flake, and thereby observe the difference between that and the worn artifact.

While lithic use-wear research has not focused on virtual reconstruction of unworn tools, lithic analysts interested in quantifying deliberate retouch have investigated this possibility, starting with Kuhn's (1990) Geometric Index of Unifacial Reduction Flaking (GIURF). The guiding principle of GIURF is to provide a means of objectively quantifying the amount of edge retouch on lithic artifacts. This aligns very closely with the core goal of modern experimental use-wear quantification studies, differing mostly in the scale of retouch: deliberate unifacial knapping vs. lithic edge attrition as a result of use. On this basis, it stands to reason that melding GIURF with high-precision lithic surface 3D digital modelling techniques can potentially provide researchers with a useful tool for quantifying edge wear.

One important aspect of applying GIURF to lithic wear quantification is understanding the difference between "edge wear" (as used in this paper) and "microwear" (as used in most surface wear studies). Lithic microwear refers to microscopic polish and striations that are conventionally observed on the surfaces of worn lithic artifacts (i.e., Grace et al. 1985; Evans and Donahue 2008; van Gijn 2014; Keeley 1980). This evidence of wear is the result of friction between the lithic surface and contact material that occurs during use. The natural texture of the lithic tool is worn away and hard inclusions between the surfaces gouge striae into the lithic surface. Edge wear, in contrast, should be understood as the structural damage sustained by a lithic tool's edge during use (Akoshima 1987; Semenov 1957; Keller 1966; Bird et al. 2007; Schoville et al. 2016; Schoville and Brown 2010; Wilkins et al. 2014; Tringham et al. 1974). As force is exerted on the edge, the material chips and flakes away, generally following the same principles of conchoidal fracture as the initial production of the stone tool. This may be considered "macrowear" in contrast to "microwear". The result is that, as the material at the sharpest section of the edge is lost, the edge becomes progressively duller until the tool is either resharpened or abandoned. Notably, edge attrition "macrowear" results in lithic surfaces with accrued microwear being lost, and fresh surfaces continually exposed and subsequently worn. As a result, accurate microwear quantification is to an extent dependent on identifying surfaces that underwent wear throughout a task, and avoiding surfaces that were exposed and worn after the task was underway. Quantifying edge wear ("macrowear"), however, documents the damage sustained by a tool throughout its use life. GIURF, designed to quantify edge attrition as the

result of deliberate retouch, is readily adaptable to quantifying "macrowear". The process described in the paper is not designed for studying microwear, and all subsequent uses of terms such as "edge wear", "wear", "edge damage", etc. refer to macrowear; microwear is not discussed at length in this paper and is only referred to as "microwear". It is also worth noting that the identification and quantification of post-depositional damage is not discussed or examined in this experiment. The author follows Schoville (2014, 2018) and Tringham et al. (1974) in the assertion that post-depositional "trample" damage is more likely to be randomly distributed around the periphery of an artifact's edge rather than concentrated in one area as one would expect from use wear. On this basis, wear- and trample-related edge damage should be distinguishable from one another using the digital edge wear mapping methods presented here. I acknowledge, however, that the random distribution of post-depositional damage is not a universally accepted model, and other scholars emphasize the necessity of polish as the only "definitive" evidence of deliberate use (i.e., Flenniken and Haggarty 1979). Ideally micro- and macrowear studies of lithic edge wear could be used in concert; only the latter is discussed in this paper.

Kuhn's GIURF model to estimate the degree of flake retouch is based on trigonometric calculation of the original flake edge position derived from flake thickness, as measured at two positions on a flake cross section, with the ratio between the thickness at the junction of un-retouched and retouched surfaces (t) and the maximum flake thickness (T) representing the degree of retouch; the higher the value of t/T, the greater the retouch index (Kuhn 1990). Eren et al. (2005) further adapted Kuhn's method, using surface measurements to calculate the area of lost material in an imagined cross-section of a flake based on the angle of convergence between the dorsal and ventral surfaces (Figure 4.1). This calculation defines the original cross-section of the flake at that point, and the difference in area between the triangle and the extant flake section represents the destroyed material; a greater area represents more intensive retouch. Lost material volume could be estimated by multiplying the calculated area difference by the length of retouched edge (the area difference is calculated at several places along the edge to account for variability in retouch intensity) (Hiscock and Clarkson 2005, 2008; Eren et al. 2005). This method was further advanced to incorporate 3D digital models into the measurement procedure

(Eren and Sampson 2009; Eren and Prendergast 2008). The results are generally encouraging, with increasingly accurate and nuanced models of unifacial and bifacial artifact retouch.

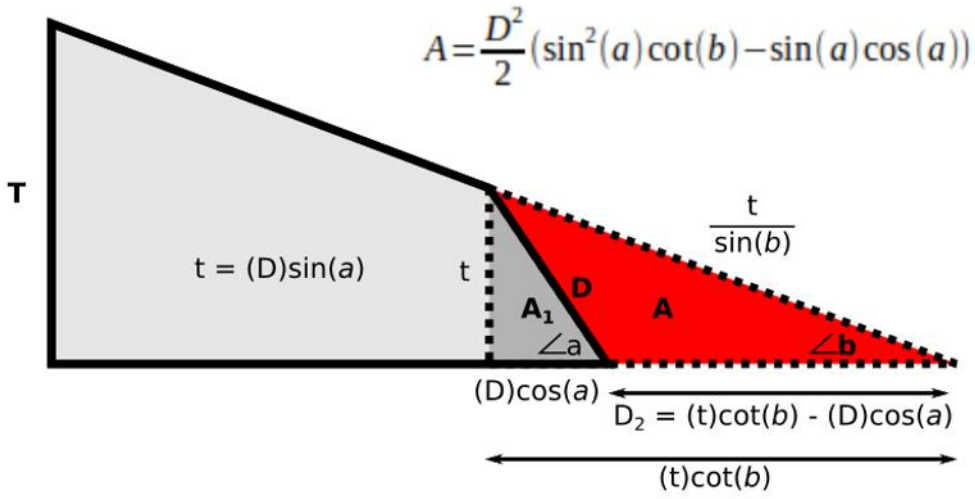

Figure 4.1: Worn edge cross-section area calculation, adapted from Eren et al. 2005.

This study examines the viability of adapting a similar geometric reduction index to the GIS-based quantification of edge damage as a result of use wear. The basic premise is that one may compare digital surface models (DSM) of pristine and worn lithics in raster format—a grid of pixels where each cell's value represents the thickness (the Z-axis) of the lithic at that position (X/Y axes)—in order to measure the amount of edge attrition that took place during use, a method that has been demonstrated to be effective for measuring wear on experimental microblades (Waber 2015; in press). Applied to archaeological artifacts, it should be possible to digitally reconstruct unworn lithic surfaces by interpolating surface points from intact areas of a worn lithic artifact with edge points based on extrapolating surface data to produce the "fresh" flake raster surface, and then use the difference between the worn and reconstructed "fresh" surfaces to precisely quantify macrowear: GIURF on a microflake-scale, or "MicroGIURF". Automated surface sampling permits the GIS to produce trigonometric edge reconstruction points at whatever interval the researcher desires, resulting in a projected edge that follows the curves and indents of the natural flake, rather than a blocky and arbitrary form imposed by unevenly and distantly spaced sampling loci. If the MicroGIURF method is successful, it will be

possible for researchers to accurately quantify the volume of lithic material worn from the edges of lithic artifacts, and thereby better understand patterns of use intensity and function. Furthermore, articulated with spatial wear distribution methods (i.e., Bird et al. 2007; Schoville and Brown 2010; Waber 2015, in press), precise use wear quantification can facilitate meaningful inferences regarding hafting/prehension, gesture, modes of use, and patterns of re-use.

The difference in analytical scale between GIURF and MicroGIURF is noteworthy, as it has relevance beyond the basic practicalities of observing modification on a flake and extends to the specific technological behaviours manifested in an artifact. GIURF examines retouch (deliberate modification) of flaked stone tools: design decisions and manufacturing practices involved with unifacial tool production. MicroGIURF is designed to measure macrowear (incidental edge flaking), especially on un-modified tools, that occurs while a tool is applied to a task. The former aims to measure the volume of (relatively) large flakes knapped away to define the tool's form (preparation for activity), while the latter quantifies the miniscule edge crushing and microflaking that occurs on a tool's edge (aftermath of activity). In this way GIURF and MicroGIURF complement each other in their analytical scopes, or rather MicroGIURF extends GIURF to later portions of the *chaîne opératoire* (Sellet 1993; Edmonds 1990; Tostevin 2011).

The major test of this method is whether or not it can position reconstructed edge points with enough accuracy and precision that the original pristine flake form may be reliably reconstructed. In the GIURF retouch tests, previous authors have had the luxury of dealing with relatively large volumes of flaked lithic material in deliberately modified unifacial tools, so small ranges of error in the subsequent flake volume quantification calculations are negligible relative to the overall flaking. However, in the context of use wear, the volume of worn material may be quite small, so hypothetical imprecision of a few millimetres that might result in a 0.1% error in volume quantification in a conventional retouch model may result in a 200% error in calculating use wear because of the difference in flake scale.

A series of experiments were performed to test the viability of GIS automated MicroGIURF. A set of experimental microblades were recorded before and after use using 3D digital photogrammetry, and digital models were used to reconstruct the original edges using a GIURF-based process adapted to QGIS. Microblades were used for three reasons: first, the relatively uniform morphology of microblades means that similar modes of hafting and use result in recognizable wear patterns along specific sections of the edge; second, microblades are produced without formal retouch, so any flaking identified on the edges through MicroGIURF analysis must therefore be the result of tool use; and finally, the author's development of the method is based on a desire to apply it specifically to existing microblade collections. As an artifact category, microblades are a global phenomenon with distribution around the circumpolar north, the north Pacific Rim, the interiors of Siberia and Western Canada, the Mississipian interaction sphere, Britain, and the Near East. Other prepared-core blade technologies are known from throughout North America, Mesoamerica, Europe, and Africa. A practical method to quantify wear on blade tools (micro- or otherwise) would be useful to archaeologists around the world.

In the present study, the accuracy of the MicroGIURF results was checked against perimeter and surface data from the microblades in their original, unworn states. The MicroGIURF process was automated using a set of QGIS Processing Toolbox models. The models incorporate a series of geoprocessing modules and functions built into QGIS to project edge points derived from blade thickness, edge angle, and edge direction on the microblade surface rasters.

4.2 GIURF in GIS: Method(s)

GIS tools are ideal for GIURF, and generally for any analysis of lithic surfaces, because the powerful geoprocessing modules designed for sophisticated landscape analyses can be easily adapted to surfaces at any size and scale, including those of lithic artifacts (Davis et al. 2015; Waber 2015; Waber in press), and data that may be tedious and difficult to collect manually with adequate precision for use-wear analysis are easily derived from a digital surface model (DSM)

of a lithic artifact. QGIS—an open source Geographic Information Systems (GIS) program—is an excellent platform for MicroGIURF as the software's Processing Toolbox provides a variety of advanced geoprocessing algorithms—including integrating modules from the powerful open-source GIS programs GRASS and SAGA—and offers a graphical modelling interface for building complex and sophisticated algorithm trees in an accessible, user-friendly environment.

Adapting GIURF to QGIS involves three primary procedures: (1) sampling the lithic surface, (2) calculating the edge projection, and (3) reconstructing the un-worn surface. Each stage involves a series of actions. First, to sample the lithic surface, one must identify sample locations. This may be done various ways, from full automation to individual manual point placement. The optimal method depends largely on the nature of the flake in question: fully automated random sampling is useful for flakes with very simple surfaces—a single dorsal arris, minimal step fractures, etc.; zone-restricted random sampling is a means of avoiding complex or suboptimal sections of surfaces; user-positioned lines are useful to target specific edge sections, such as those where edge wear has been visually identified. This study used fully automated random sampling of the entire flake dorsal surface, random sampling restricted to surfaces adjacent to flake margins, and user-guided sampling based on lines marked on the digital model.

The second stage is sampling the lithic surface for three elements: flake thickness, edge angle (sharpness), and direction in which the surface angles towards the edge. In GIS raster parlance, these are the DSM, slope, and aspect maps. Sampling is done by querying the rasters using the point layer generated earlier. The three components are then incorporated into a formula that enables the projected extension of the original edge.

Edge projection calculation essentially follows Eren et al.'s (2005) differential area method using the convergence angle of the dorsal and ventral surfaces to estimate the position of the flake's original edge relative to a sampled point. A second algorithm is applied based on surface aspect to direct the edge point placement towards the flake edge (assuming that the surface angles towards the edge of the flake). The first step is to use the DSM and slope rasters to determine how far the original flake edge is from the sampled point (Figure 4.2). This uses

the following formula: $d = \frac{z}{tan(radians(slope))}$. Next, the X and Y coordinates of the edge point must be derived from the X and Y coordinates of the sample point, the distance value from the previous formula, and the direction value from the aspect raster[3]: $X_{edge} = X_{point} + d^{*}sin(radians(aspect)); Y_{edge} = Y_{point} + d^{*}cos(radians(aspect))$ Finally, the edge points may be plotted based on the projected X/Y coordinates.

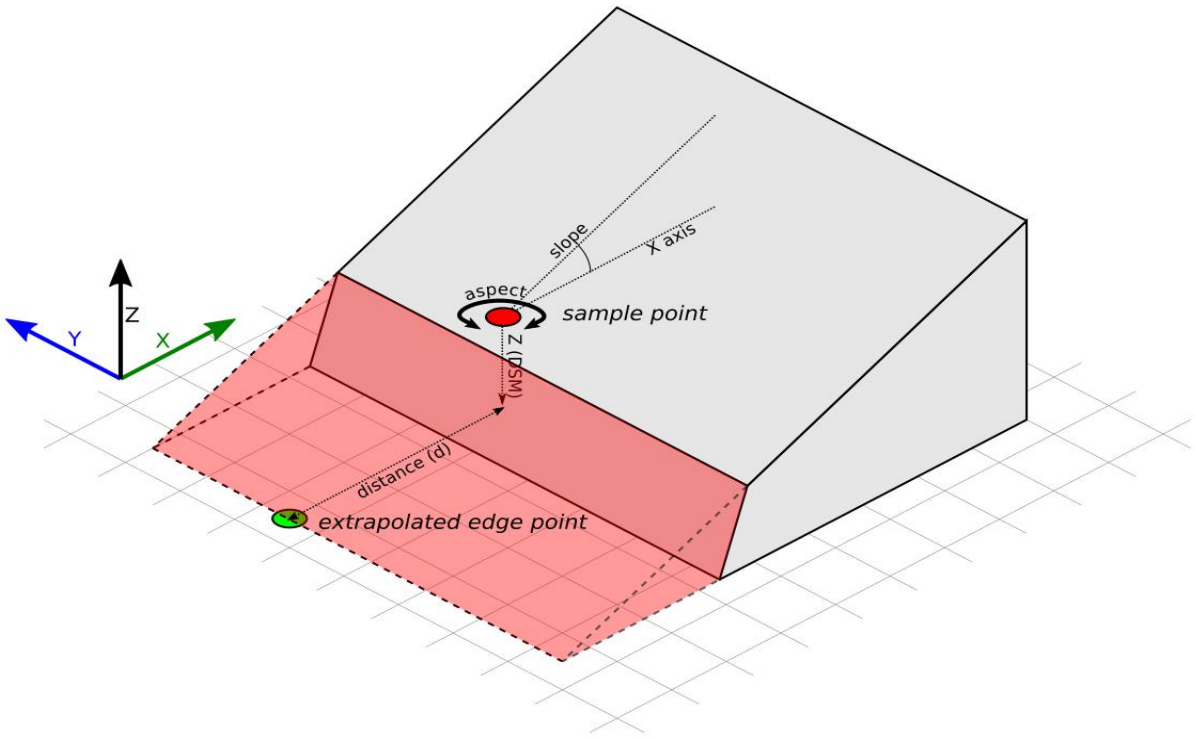

Figure 4.2: Isometric diagram of idealized edge extrapolation from lithic DSM point sampling. Translucent red represents the section of the edge that was worn away during use. In this idealized example, a sample point is given at X=4.5, Y=6.5, Z=1.5, aspect = 270°, and the slope = 19.44°. Thus, d = 4.25, and the edge point is positioned at X = 0.25, Y = 6.5, Z = 0.

In order for the GIS to interpolate a surface between the extrapolated edge points and the sampled surface points, an arbitrary Z=0 "elevation" is assigned to the edge points. To assign a

3 A direction raster produced using a proximity grid module may be substituted for an aspect raster in cases where intrusive flake scars (i.e. on a heavily retouched surface) make aspect unreliable for finding the edge.

relative Z axis value of 0 to the edge points, the flake must be digitally flattened—absolute Z axis values must be converted to relative Z values, all referring to a prime Z_0 plane fitted between edge points around the flake[4]. After this is done, a new surface raster must be interpolated that combines the newly plotted points and the unworn portion of the surface DSM.

Notably, while raster-based GIS analyses do not operate in "true" 3D (these surface models are considered "2.5D" -- while the obverse side has volume, the reverse does not exist), it is generally inconsequential because dorsal and ventral faces of most flaked stone tools may be treated as individual terrains for analysis, and the results combined afterwards. It is worth noting, however, as the edge-adjacent dorsal surfaces of most flakes intersect with the Z0 plane at a considerably steeper angle than do ventral surfaces, there is considerably less error in edge point extrapolation based on dorsal surface sampling alone.

4.3 Evaluating MicroGIURF: Tests and Results

Answers to two questions are required in order to evaluate the efficacy of MicroGIURF:

1) How accurately can MicroGIURF place edge points based on lithic surface measurements?

Accurate 3D surface reconstruction requires accurate edge point placement. The extrapolated edge points combine with points sampled on unworn surface areas to facilitate interpolation of the reconstructed flake DSM. Therefore, if points are consistently too far from the original edge line, the original flake volume will be overestimated and measured wear will be exaggerated; if they are undershot (within the original perimeter), then volume will be underestimated. The goal is for MicroGIURF to position reconstructed edge points directly on the original edge. To test this, edge points are placed using the QGIS module, and the distances between those positions and the unworn microblade perimeter are measured. Lower distances reflect more accurate point placement. Ideally, all of the points will be placed correctly right on the line.

[4] See Waber (in press) and appendix I for details on how to digitally flatten a curved flake model.

2) Is it possible to reconstruct an accurate and precise 3D digital model of the original microblade surface using only a worn microblade?

This test generates a virtual microblade surface based only on the worn microblade DSM, and quantifies edge wear using that surface. The resulting reconstructed DSM is then compared against a DSM of the original microblade to identify how closely the virtual reconstruction matches the unworn lithic. The difference between the unworn- and the reconstructed microblade DSMs is quantified in two ways: first, the horizontal area is considered: how closely does the reconstructed edge adhere to the path of the original edge? Second, each cell of the unworn microblade DSM is compared with the reconstruction in order to determine what the difference is at that specific point on the blade. If the horizontal and vertical reconstruction values are suitably accurate, the method may be considered successful. In this case, "suitable accuracy" is defined on an arbitrary basis, with 75% of the sampled cells being within 25% of the correct value.

All of the experiments are based on a set of microblades made and used by the author. The microblades were side-hafted in a slotted handle, and were used to incise green Douglas fir wood (*Pseudotsuga menziesii*). The microblades were recorded before and after use using 3D digital photogrammetry. Approximately 40 photos were taken of each side of each blade. The MicMac photogrammetry software package (Rupnik et al. 2017; Pierrot-deseilligny et al. 2015) was used to produce high-resolution 3D point clouds of the microblades, that were then processed in CloudCompare (2017). The resulting point clouds were interpolated into digital surface models (DSM) in QGIS. GIS-based lithic wear quantification follows the methods developed by Waber (2015; in press), producing very precise volumetric and spatial measurements of edge damage on microblades.

One important aspect of the GIURF process is understanding how the ventral surface of the subject flake is accounted for in the edge extrapolation calculation. In the original GIURF (Kuhn 1990) and subsequent permutations (Eren et al. 2005; Eren and Sampson 2009; Hiscock and Tabrett 2010), the flake cross-section was treated as a polygon with intersecting straight sides, one of which was the ventral surface. The MicroGIURF method adapts this to some degree by digitally isolating the dorsal surface and establishing a Z_0 plane based on a natural

neighbour interpolation between points around the edge of the flake (Waber in press). The effect is that the edges of the flake are always oriented on a horizontal plane relative to the "vertical" Z-axis (Figure 4.3). This reduces the error resulting from using the naturally curved ventral surface as the horizontal intercept for edge point triangulation. Given the conchoidally curved form of the natural ventral surface of a flake, a direct Z-axis projection from a sample point on the dorsal DSM will not intersect with the ventral surface at the 90° angle that is needed to calculate the edge distance. For this reason, the ventral surface is omitted from the MicroGIURF edge extrapolation process, instead using the Z_0 plane as the horizontal intercept.

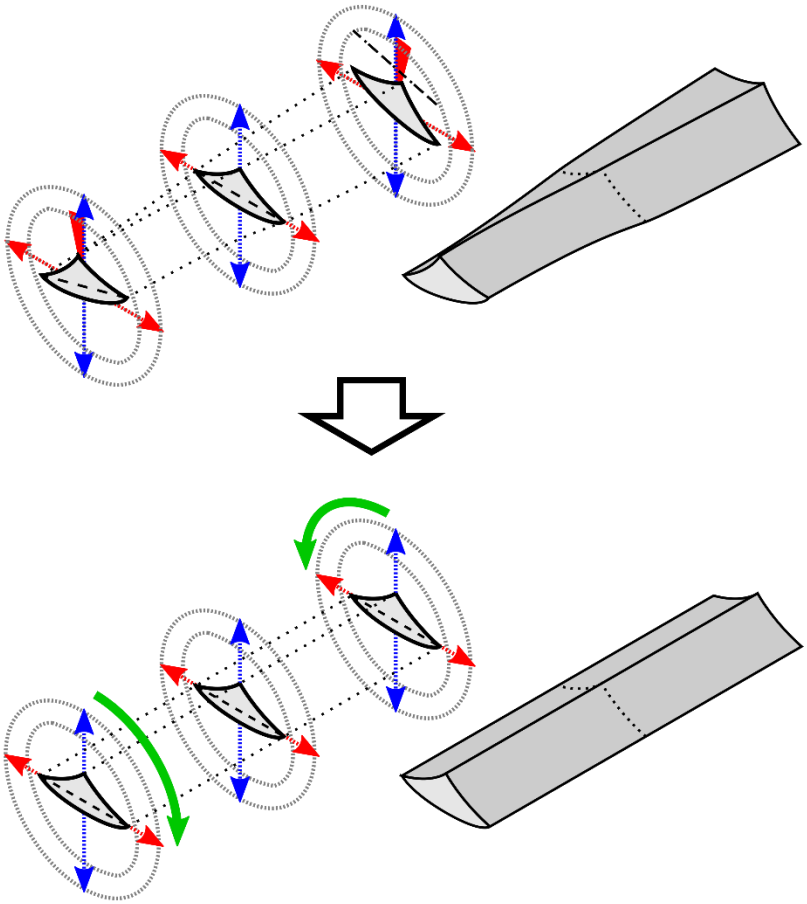

Figure 4.3: Idealized diagram of flake flattening.

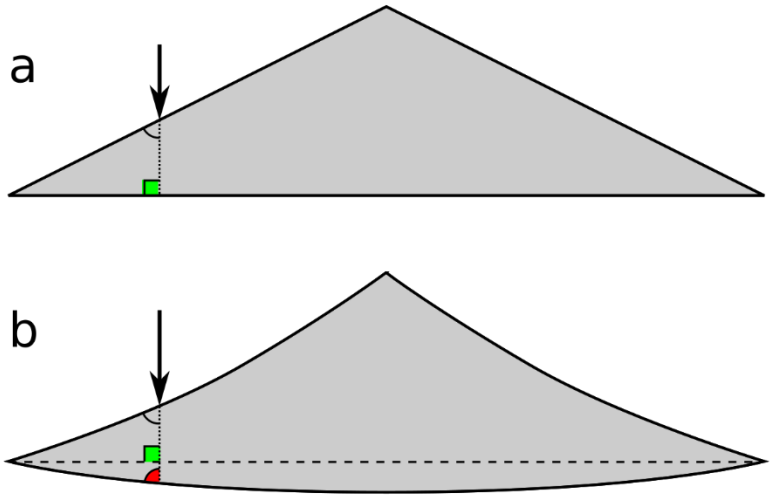

Figure 4.4: Idealized intersections between Z-axis sample point extension and ventral surface: a) simplified flake cross-section with flat ventral surface intersected at 90°; b) conchoidally curved ventral surface intersected at <90°. Note that the inter-edge Z_0 plane in image "b" is intersected at 90°

The Z_0 plane is essentially a DSM based only on elevation data from points around the edge of the flake. That DSM is then subtracted from the original lithic DSM. The result is a digital model of the flake as if it had been "flattened"; all of the horizontal curvature and twisting of the flake is removed, and only the "true" volume remains. This effectively corrects the 2.5D effect of raster-based surface computation, permitting one to measure the absolute thickness of the flake at any point. However, it is important to note that this plane also effectively bisects the dorsal and ventral volumes of the flake, so total volume (and thickness) is the sum of two rasters (or point samples). The QGIS Processing models account for flake bisection in their volume calculations.

Test 1: How accurate is MicroGIURF edge reconstruction?

This test assesses the accuracy of MicroGIURF's point positioning calculations for virtually reconstructing unworn flake edges. The base data set used was a set of unworn

microblade DSMs and perimeters. Three sets of sample points were used to assess the method. The first was a set of 500 points randomly placed within the unworn blade perimeter. The second was a subset of these 500 points, selected from edge-adjacent surfaces, sampled to test the extent to which multiple dorsal arrises affect edge reconstruction. Subsets ranged from 132 to 398 points, depending on the area of the edge-adjacent surfaces. The edge-adjacent selection omitted the randomly generated sample points on central dorsal flake scars (those between parallel dorsal arrises) or other "convoluted" structures away from the lateral margins, such as stacked step fractures near the striking platform. The third set was generated along a line defined on the DSM by the user, based on visual assessment of the DSM. The lines were generally positioned approximately 1.5-2.5mm from the edge, on smooth dorsal surfaces narrowing to straight margins that look likely to have been used as cutting edges.

Each set of points sampled the DSM, slope, and aspect rasters, and extrapolated a matching point position based on the GIURF model. The minimum distance between the extrapolated points and the blade perimeter line was measured, as well as the "true" distance between the sample points and the edge. The difference between the calculated/expected distance to the edge, and actual distance to the edge shows the algorithm's accuracy.

The randomly placed sample points resulted in poor accuracy in edge point placement (Table 4.1, Figure 4.5). Comparison with the edge-adjacent points (Figure 4.6) indicates that this is, to some extent, the result of points on non-edge-adjacent surfaces projecting virtual edge points far beyond the desired target. This is somewhat analogous to the "flat flake" problem identified by Dibble (1995) and Hiscock and Clarkson (2005) in regards to Kuhn's GIURF model, where parallel dorsal and ventral surfaces confound the edge triangulation formula. The inaccuracy observed in edge point extrapolation by fully-random point samples taken from central flake scars (perhaps better understood as inter-arris troughs) is reflected in the substantial difference between the median and mean point distances as well as the very high maximum inaccuracy value, even though the median inaccuracy is only 0.23mm (Table 4.1). This shows that major outliers at the extreme upper end of the scale are skewing the distribution—outliers that one would expect from samples on flat (minimally convergent) surfaces with relatively high

Z-values casting their extrapolated points far beyond the microblade perimeter. On this basis, the median point distance-from-edge is most useful for understanding point placement accuracy. Table 4.1 shows the consistent improvement from random sampling through semi-random edge-adjacent sampling to user-guided line sampling, with accuracy increasing by 50% between each method. This pattern demonstrates the importance of sample surface quality to extrapolated point accuracy.

Despite being more accurate than the fully random sample set, the lower accuracy of the semi-random edge-adjacent points compared with the user-positioned line-based points indicates that samples taken from central flake scars in the fully random placement model is not the only confounding factor (see Figure 4.6, Table 4.1). The line-based points consistently clustered very close to the true edge, although some scattering was still evident in places (i.e., the left edges of MB13 and MB22, Figure 4.8). Line-based points also exhibit substantial improvement in consistent accuracy over the edge-adjacent points, as seen in comparisons of the median values and interquartile ranges of the two groups (Table 4.1). All of the methods result in points clustering fairly densely around the expected perimeter, but the user-positioned line method is by far the most accurate. The improvement in accuracy over the three sampling methods is evident in Table 4.1 and Figure 4.8 where the concentration of line-sampled points immediately around the central 0mm marker stands in sharp contrast with the considerably less accurate random- and semi-random sampling methods. The superior results of the user-guided line sample method stands to reason as the lines—positioned based on visual assessment of the lithic surfaces—avoided step fractures, inter-arris troughs, ridges, and other suboptimal surfaces that would skew the slope and/or aspect values for the sample points. However, avoiding surface irregularities comes at the cost of the sample depending on user input, and is prone to omitting surface areas that are subjectively suboptimal, and thus possibly missing wear on the adjacent edges. Random- and edge-adjacent sampling at least ensures that all the edge is considered in the analysis.

Despite underperforming against the user-positioned line sampling method, the tight clustering of extrapolated points around the perimeter line by the random- and edge-adjacent

sampling methods shows that full automation may yet be feasible (Figure 4.5 and Figure 4.6). The key to this is to concentrate on the central tendencies of the point placements, and to allow median points to guide perimeter reconstruction (see experiment 2 for an example method). Ideally, future versions of the MicroGIURF method will see optimal surfaces for random point placement be identified programmatically rather than by user direction.

	Min.	Q1	Median	Mean	Q3	Max.	Std.Dev.	IQR	CV
Random Pts	0	0.07	0.23	40.69	0.62	202.51	7.06	.55	0.174
Edge-adjacent Pts	0	0.05	0.13	1.26	0.29	5.85	0.29	.24	0.230
User-positioned Sample line	0	0.04	0.08	0.26	0.15	1.02	0.14	.11	0.538

Table 4.1: Distance (mm) of extrapolated points from real microblade perimeter.

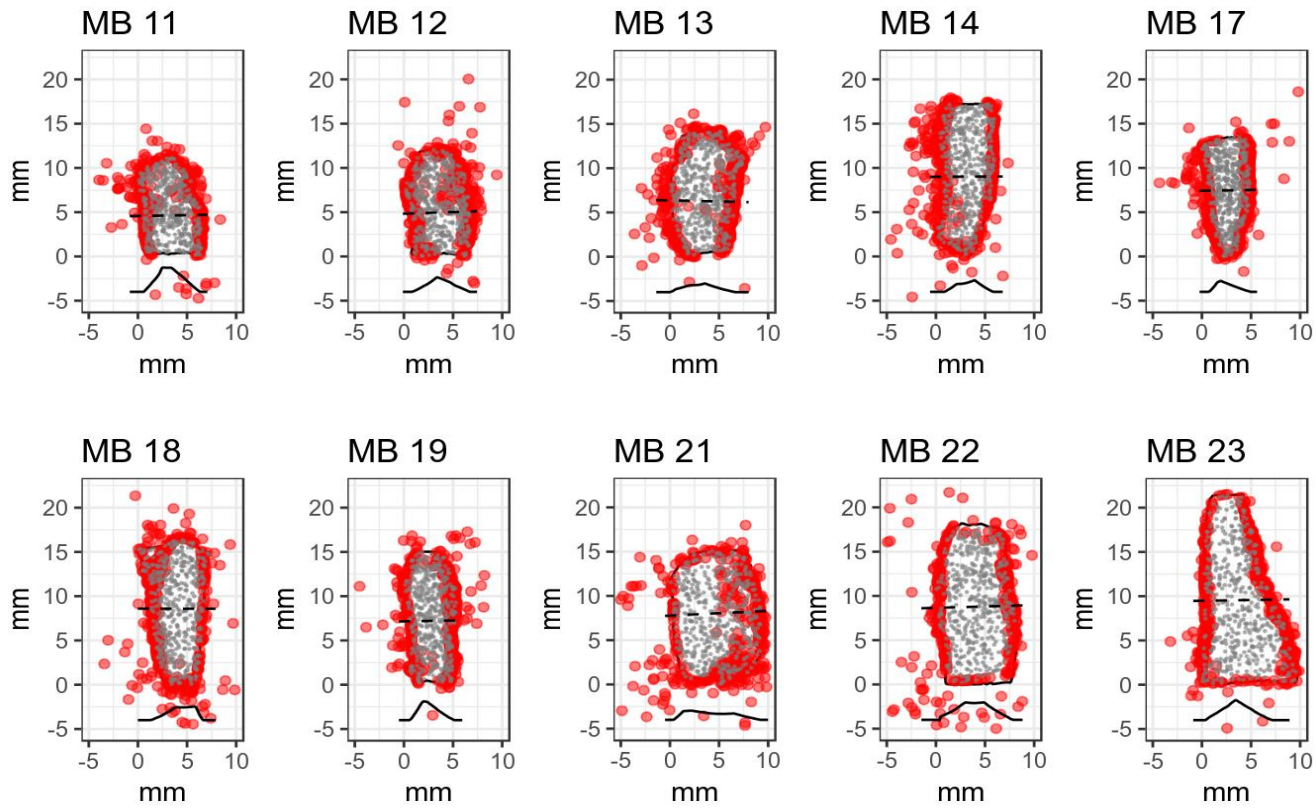

Figure 4.5: Edge point placement generated by total coverage raster sampling. Grey points are samples; red points are extrapolated edge points.

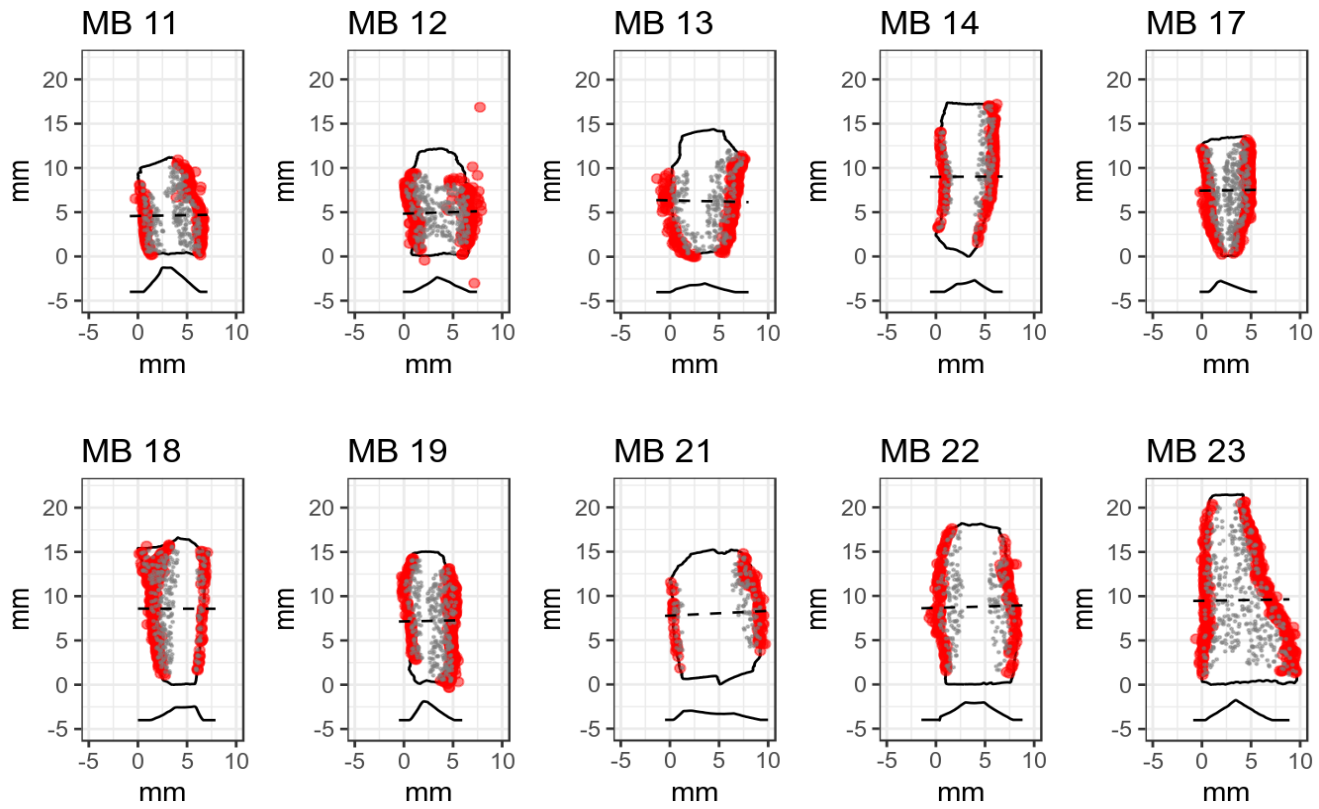

Figure 4.6: Edge point placement generated by raster sampling restricted to edge-adjacent surfaces.

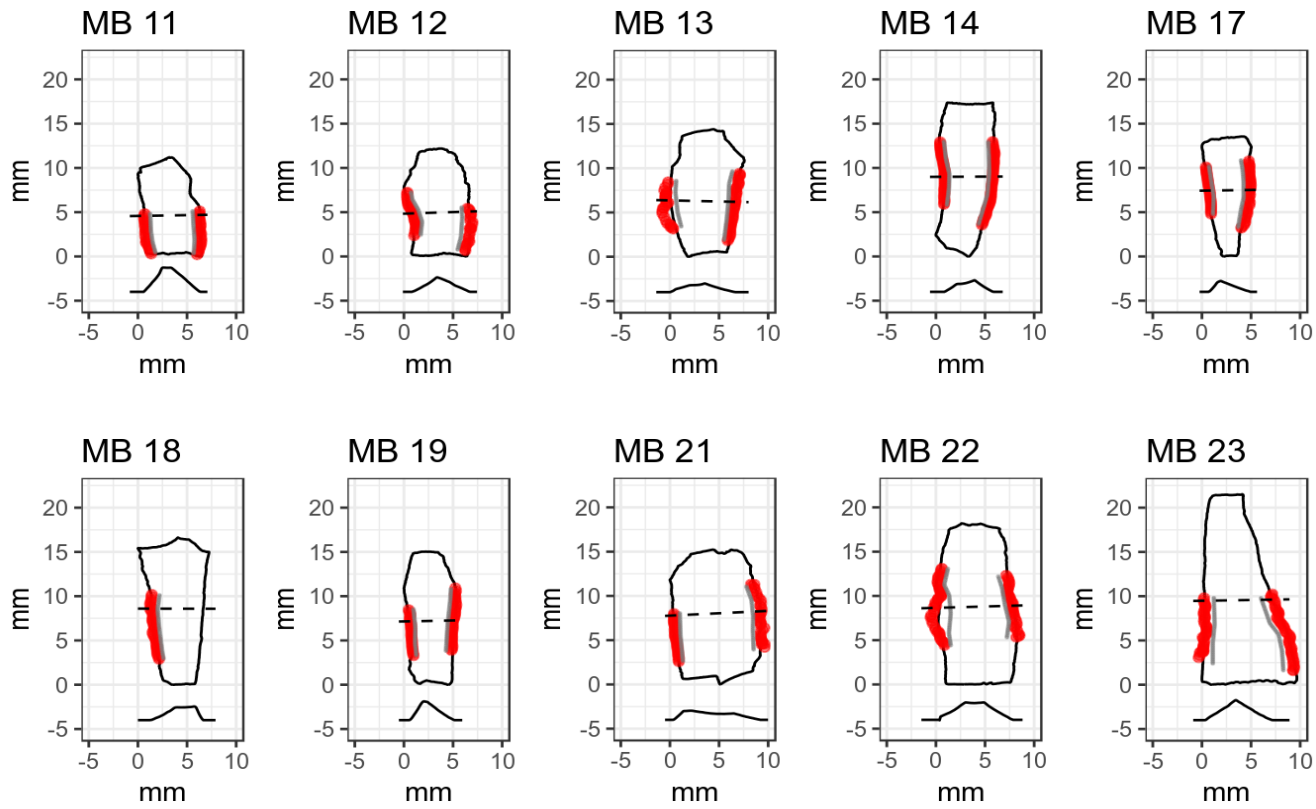

Figure 4.7: Edge point placement generated by user-positioned sampling lines. Line placement was based on subjective assessment of likely edge angles and visual identification of edge wear.

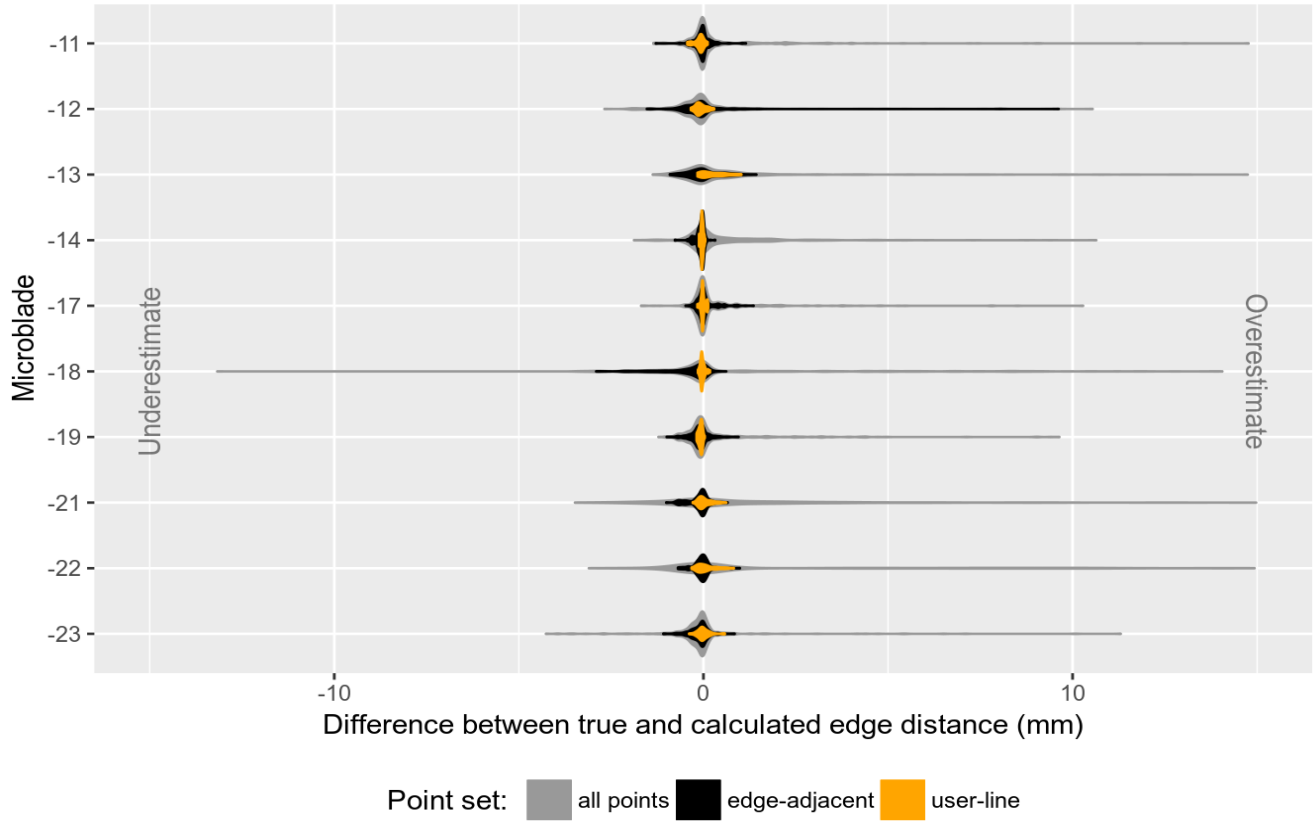

Figure 4.8: Violin plot showing edge point placement accuracy based on three sampling methods: unrestricted automated random point placement; random points restricted to edge-adjacent surfaces; user-positioned lines. Outliers beyond 15mm from the edge are omitted.

Test 2: Digital surface reconstruction

The second experiment tests the accuracy of virtual lithic surface reconstruction based on point samples taken only from a worn lithic DSM. The guiding hypothesis is that if the edge reconstruction is accurate, most of the reconstructed surface will be within an acceptable range of error[5] of the correct position based on the DSM of the original, un-worn microblade. The difference between the reconstructed model and the unworn flake model is determined using QGIS lithic use-wear quantification modules (Waber, in press), which perform raster difference

5 For the purposes of this experiment two arbitrary levels are applied in order to define "most of a surface" and "acceptable range of error": moderate stringency expects at least 75% of a reconstructed surface to be within 25% range of error of the un-worn microblade DSM reference model.

calculations on paired lithic DSM sets (dorsal and ventral surfaces) and return wear volume and wear intensity maps on 0.05 x 0.05 mm grids. In this case rather than use wear, difference values represent error in surface reconstruction. Higher volume values equal greater absolute difference between the original and reconstructed surfaces. The wear index is used to represent relative wear in any given position, using the formula: $\frac{U-W}{U}$ where U is the unworn microblade DSM and W is the worn microblade DSM; a value of 0 represents no wear, and 1.0 indicates complete material loss (Waber, in press). In the context of a reconstructed microblade DSM, however, 0 indicates a perfect flake reconstruction and 1.0 represent areas that were missed altogether by the reconstruction model.

Method:

For this test, virtual surface reconstruction used the semi-automated edge-adjacent surface sampling method and the QGIS reconstruction Processing model. This method was selected for two reasons: first, it provides substantially higher quality edge point placements than does the total surface random sampling method; second, this method overcomes the major selection bias inherent with the user-positioned-line surface sampling method. A guiding design principle for the reconstruction model is to obviate subjective wear interpretation and create an analysis tool useful to non-experts, and edge-adjacent sampling does not require a user to identify likely edge wear.

Despite the majority of high-quality edge reconstruction points that are extrapolated using the edge-adjacent method, remaining inaccurate points threaten to severely skew the DSM interpolation if they are included in the ultimate surface reconstruction. A QGIS Processing model was created to produce a microblade edge perimeter line using only optimal extrapolated edge points rather than using the entire population (Appendix J). The QGIS Processing model is predicated on creating a density raster (a heatmap) of the extrapolated points and using peaks in the density raster (positions where the density is highest within each cluster) to position edge points.

For this experiment, two parallel criteria are used to assess the quality of the flake reconstruction: horizontal reconstruction accuracy (how well the reconstructed microblade edge matches the original microblade edge) and vertical accuracy (how closely the reconstructed surface fits the original surface at each sample location). Structured as testable hypotheses, these are:

- Hypothesis 1: if the QGIS lithic surface reconstruction model accurately extrapolates the original edge profile of a worn lithic tool using only data collected from the worn lithic surface, the reconstructed area will closely match the horizontal area that was worn away during use. For the purposes of this test, "close" is set at an arbitrary threshold of 75%. Thus, if there is greater than a 25% discrepancy between the worn-away area and the reconstructed area, the hypothesis may be rejected.
- Hypothesis 2: if the surface interpolation is accurate, there will be minimal vertical (Z-axis) difference between reconstructed lithic digital surface model (DSM) and the original un-worn lithic DSM. "Minimal vertical difference" is defined here as no greater than 25% of vertical difference (75% or better accuracy) over 75% of the sampled area[6].

Observations:

Hypothesis 1: horizontal area difference

None of the digital microblade reconstructions achieved the 75% area reconstruction threshold, or even over 50% correct reconstruction (Table 4.2). The microblade edge reconstruction inaccuracy appears to be due to two complimentary factors: catastrophic snapping/spalling, and the density-based edge point calculation model. First, microblade ends frequently broke off during the use wear experiment. MB12, MB14, and MB21 are excellent examples of this spalling effect, where catastrophic material loss is evident on one or both ends when comparing the worn microblade models to the original microblade perimeters, and as a result the percentage of correctly reconstructed area is very low (Table 4.2, Figure 4.7). This breakage was of a different nature to the conventional micro-flakes and chips concentrated on the contacted edge(s) of the microblades; end-breakage was the result of twisting, prying, or other such leverage-inducing situations that caused large sections of the blade to snap or spall off.

[6] The arbitrary thresholds of 75% accuracy over 75% area were selected in order to strike a balance between accuracy high enough to be analytically useful, and achievable criteria.

The spalling is analogous to flaking that occurs during burin manufacture, where the resulting flake scar occupies the distal/proximal end of the flake, rather than being on the dorsal or ventral surface. Obviously, in cases of catastrophic snapping or spalling, where material loss meant that the microblade was actually reduced in length, all of the dorsal surface was on the broken section was also lost, so edge-reconstruction that relies on sampling the surface is impossible. As a result, the reconstructed edge points do not extend beyond last section of intact surface; the QGIS reconstruction method assumes that the sampled flake is complete. Because flake breakage (especially microblade breakage) is a reality of lithic tool use, this assumption reveals a severe limitation of this method: QGIS wear quantification based on semi-automated surface reconstruction requires optimal samples (flakes that have not been broken), and that optimality must be identified by an expert analyst, which contradicts the stated goal of objective replicability in the method. While the need to identify (and omit) broken samples from MicroGIURF analysis constrains the utility of the method, it does not entirely cripple it; microblades without post-depositional snap damage are still viable subjects for digital reconstruction, and other tool types that are less prone to catastrophic breakage may be better options as well.

Microblade	Expected area	Absent	Correct	Reconstructed %
11	2.193	1.487	0.706	32.19
12	15.125	13.448	1.677	11.09
13	6.038	3.331	2.707	44.83
14	16.193	14.927	1.266	7.82
17	2.054	1.075	0.979	47.66
18	8.152	6.685	1.467	18
19	3.335	2.62	0.715	21.44
21	39.665	38.532	1.133	2.86
22	7.133	4.653	2.48	34.77
23	9.415	6.705	2.71	28.78

Table 4.2: Reconstructed areas by reconstruction result (mm^2)

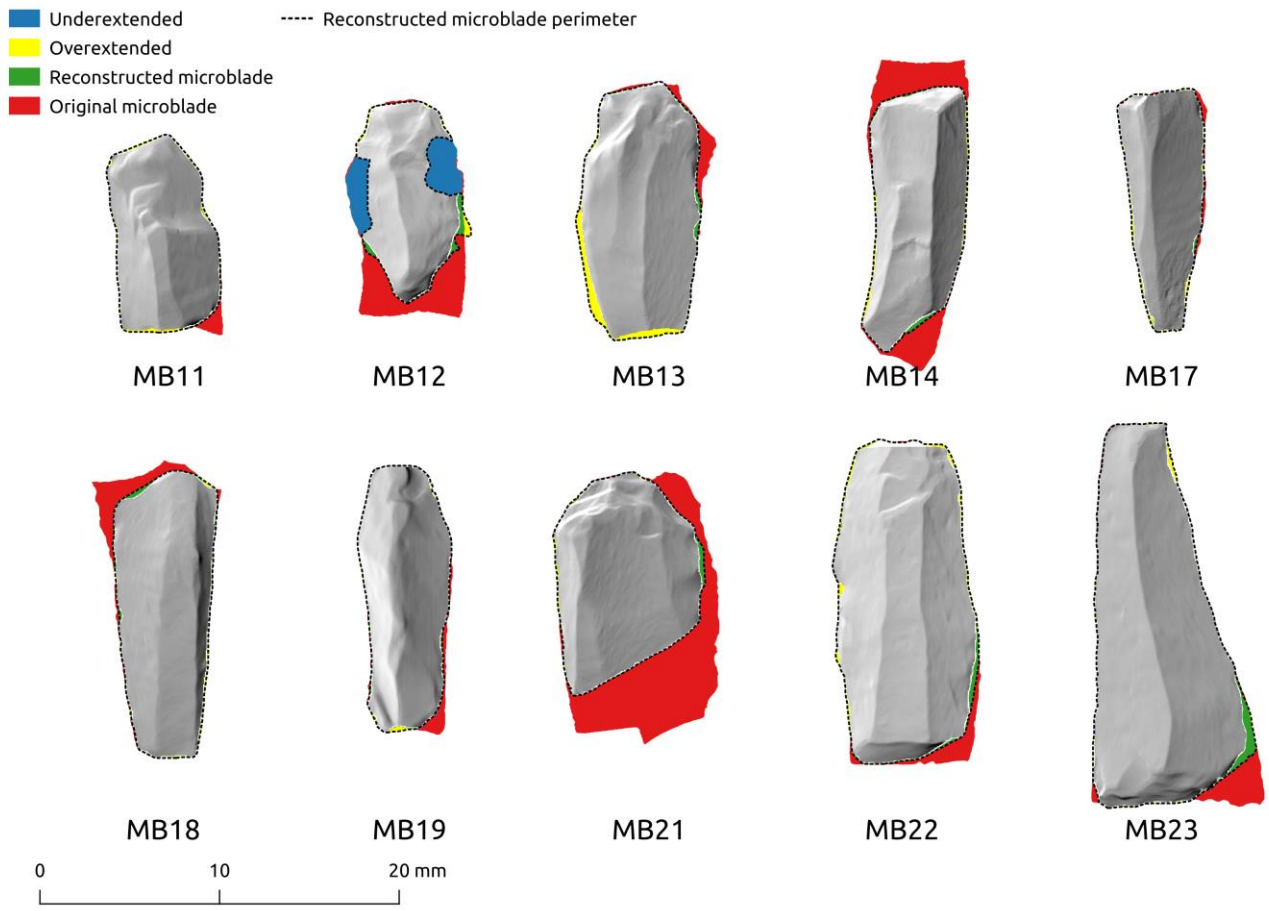

Figure 4.9: Reconstructed area categories. Red indicates sections that were missed by the reconstruction algorithm.

Second, the point-density-based edge reconstruction method also performed poorly near the ends of the microblades. In this case, the problem stemmed from the end of the sample surface resulting in lower extrapolated point densities at the microblade's extremities. In order to avoid edge reconstruction errors due to otherwise-isolated points creating false density peaks, the edge reconstruction model is set to ignore points generated by low-density clusters. Unfortunately, as surface sample point frequency tapers near the ends of microblades, so does extrapolated point density (points-per-surface area is consistent, but the surface ends). The result

is that the reconstructed edge produced from the density cluster peaks does not extend to the full extent of the original edge, but rather tapers in along the worn edge. This effect is clearly visible on a comparison of the original, worn, and reconstructed edges of MB23: the original edge extends the full length of the microblade, but the reconstructed edge tapers following the most intact section of the worn surface (and matches the original edge with robust accuracy), but tapers sharply towards the midline at the distal end (Figure 4.9). Without a more complex model capable of recognizing reasons behind extrapolated point density variation and adjusting density peak value accordingly, the operator is forced to choose between a looser or more stringent standard for density cluster peak inclusion. Informal trials found that a looser standard introduces edge extrapolation overestimation due to scattered high-error point clusters, and a tighter standard imposes very conservative edge extrapolation by restricting peaks to only the highest quality (most dense) clusters.

Given the failure in reaching the minimum threshold for accurate area reconstruction, and subjective visual comparison of worn areas and reconstructed areas, Hypothesis 1 is rejected: edge profile reconstruction using the proposed QGIS method is not sufficiently accurate to model the original (unworn) perimeter of a microblade using only samples from a worn microblade DSM.

Hypothesis 2: Vertical surface difference

To calculate the vertical surface differences, every grid cell of each unworn microblade DSM was converted to a point that was used to sample the worn DSM, reconstructed DSM, and sharpness raster. The points also registered which section of the horizontal area map they fell within. Points were then compared based on the criteria of 75% (or better) accuracy over at least 75% of the samples.

Given the horizontal area reconstruction analysis showed that none of the worn microblade sections were reconstructed to the 75% area threshold, the criterion was adjusted and applied only to samples from the reconstructed section. That is, only cells that fell within the reconstructed microblade perimeter as designated by the extrapolated points were included.

Even with that adjustment, none of the microblades achieved the 75% accuracy threshold (Table 4.3).

Microblade	Sampled cells	Reconstructed cells within 0.25 threshold	Accurate %
11	78	19	24.36
12	494	105	21.26
13	273	123	45.05
14	327	84	25.69
17	169	57	33.73
18	319	119	37.3
19	166	16	9.64
21	403	238	59.06
22	537	264	49.16
23	873	142	16.27

Table 4.3: Vertical accuracy of individual cells on reconstructed microblade DSMs: reconstruction index.

One possible explanation for these results is that by using the raster difference index, the accuracy threshold criteria are disproportionately stringent on reconstruction nearer microblade edges versus more central surfaces. A hypothetical surface error of 0.1mm on a section of edge that is 0.2mm thick results in an index inaccuracy of 50%, while a 0.1mm surface error in a 2.0mm thick midsection would only be 5% error, and thus well within the acceptable range. To counteract this effect, the DSM vertical differences were re-assessed using the absolute vertical difference between the original and reconstructed surfaces. In this case, an arbitrary quality threshold of 0.1 mm was used.

Examining the reconstruction results using the absolute surface difference returns accuracy values that are somewhat, but not substantially, better than the index threshold test. One microblade has greater than 75% accurate reconstruction, and four others are marginally "better than a coin flip" (Table 4.4). While this is an improvement, it is insufficient support for hypothesis 2, which is therefore rejected: the semi-automated QGIS microblade reconstruction method does not achieve accurate results in surface reconstruction.

Microblade	Sampled cells	Reconstructed cells within 0.1 mm	Accurate %
11	78	42	53.85
12	494	271	54.86
13	273	126	46.15
14	327	32	9.79
17	169	146	86.39
18	319	86	26.96
19	166	89	53.61
21	403	238	59.06
22	537	118	21.97
23	873	432	49.48

Table 4.4: Vertical accuracy of individual cells on reconstructed microblade DSMs: absolute vertical difference.

The results of the semi-automated reconstruction test are disappointing for the viability of the QGIS lithic surface reconstruction model as it is currently configured. Using MicroGIURF with a semi-automated edge-adjacent polygon sampling method provides neither accurate edge reconstruction nor accurate surface reconstruction. At this stage, the tool is not useful for quantifying edge wear on archaeological lithic artifacts.

4.4 Discussion: Towards functional MicroGIURF

Despite the negative results of the experiments, MicroGIURF showed some promise under optimal conditions, that is encouraging to consider future development of the method. The QGIS edge reconstruction method returns strong results when applied to specific sections of certain microblades with the worn DSM sampled using the user-placed line method. Two sections of the MB11 and MB23 DSMs were cropped and analyzed following the criteria used previously for the semi-automated sampling method. These two sections were selected on three bases: both exhibit substantial edge wear both in terms of edge attrition and material volume loss (MB17, which scored highly in the absolute vertical difference DSM test, exhibits a much lower volume and area of wear), both reconstructed edges aligned closely with the original un-worn edge, and visual comparison of overlain 3D surfaces shows high similarity between original- and

reconstructed DSMs (Figure 4.10). Edge- and surface reconstruction in these cases is quite encouraging, but there remains the question of why these particular sections, sampled with the user-positioned lines, resulted in reconstruction accuracy superior to the other microblades, or the semi-automated edge-adjacent polygon method.

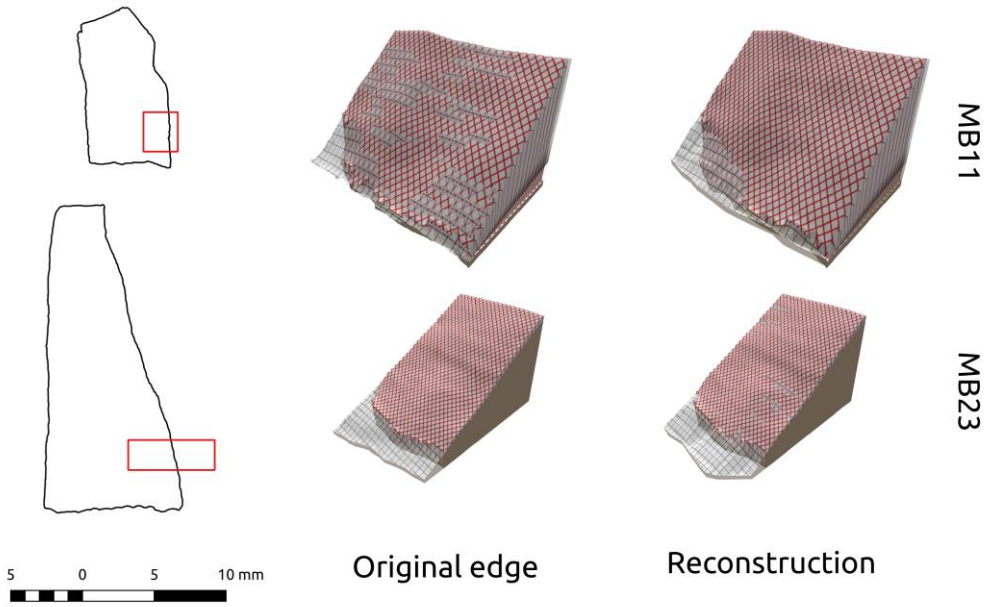

Figure 4.10: Optimal DSM sections. The original (un-worn) and reconstructed DSMs are transluscent; the worn DSM is solid grey.

Using the same criteria as in the two hypotheses above, the initial positive impression of the optimized reconstruction sections generally holds true. First, the horizontal area comparisons were extremely accurate, with both sections at over 93% correct reconstruction (Table 4.5). This is sustained in the volumetric comparisons, with over 98% of the original volume reconstructed (Table 4.6). In this case, it is worth noting that the MB11 reconstruction at 100.41% reflects a slight over-estimation of the surface during interpolation. In any case, both the horizontal area and the volume-within-area scores far exceed the 75% threshold set forth in H1, and the hypothesis is supported in this case.

Microblade	Expected area	Absent	Correct	Reconstructed %
11	0.468	0.018	0.45	96.15
23	1.249	0.079	1.17	93.67

Table 4.5: Reconstructed area by result (mm^2)

Microblade	Expected volume	Absent	Correct	Reconstructed %
11	11.9	-0.05	11.95	100.41
23	65.19	0.72	64.47	98.89

Table 4.6: Reconstructed volume by area (mm^3)

The vertical surface difference comparison analysis also returned better accuracy results for the optimized microblade sections, but not as strong as the horizontal area comparison; the wear index analysis did not meet the minimum accuracy threshold for either microblade (Table 4.7), and only MB11 achieved adequate accuracy for the 0.1mm absolute difference threshold (Table 4.8). It is worth noting that the MB23 reconstructed surface was generally within 0.05mm of the 0.1mm threshold, with a median difference of 0.14mm.

Microblade	Sampled cells	Reconstructed cells within 0.25 threshold	Accurate %
11	135	11	8.15
23	393	90	22.9

Table 4.7: Vertical accuracy of individual cells on optimal sections of reconstructed microblade DSMs: reconstruction index.

Microblade	Sampled cells	Reconstructed cells within 0.1 mm	Accurate %
11	135	126	93.33
23	393	113	28.75

Table 4.8: Vertical accuracy of individual cells on optimal sections of reconstructed microblade DSMs: absolute vertical difference (mm).

The reasons why edge reconstruction worked well for these sections are unclear, and several possible factors can be considered. First, neither of the sampled sections are at the proximal or distal extremes of the microblade edge. As a result, the Y-axis ranges of extrapolated edge points extend beyond the bounds of the respective sampled sections so the reconstructed edges do not taper towards the microblades' centrelines as extrapolated point frequency drops (and thus reduces point density clusters).

A second potential factor is the sampling method, which records data at points along a user-positioned line, avoids surface features such as step fractures, dorsal arrises, flake scars, undulations, etc., and thus only samples the "cleanest" parts of the DSM (based on user assessment of the surface). While this evidently results in somewhat better reconstruction and wear quantification results than the edge-adjacent polygon sampling (which does not detect and avoid step fractures, etc.), it has the major disadvantage that it depends on the user's visual identification of edge wear to inform line placement. This means that if the user does not spot edge wear, that wear would not be recorded at all.

Finally, the sampled surfaces are smooth and quite flat; that is, the slope towards the lateral margin is direct and consistent. Neither the MB11 nor MB23 dorsal surface adjacent to the operational edge is a deep (i.e., low-diameter) flake scar. This is relevant to the GIURF formula, as the slope angle of the dorsal surface dictates the intercept between the projected surface plane and the horizontal edge-to-edge plane (or the horizontal ventral plane in traditional GIURF models). A deep flake scar running longitudinally on a dorsal surface results in variable slope angles oriented towards the lateral margin. Closer to the dorsal arris, the slope will be steep and projected edge points will undershoot the true edge, while only slope samples taken very close to the edge will result in accurate edge point projections. Of course, as the edge is worn away the preferable slope samples are also worn away. Therefore, the minimal curvature of the flake scars on the MB11 and MB23 dorsal surfaces meant that the slope was quite

consistent from the dorsal arris to the blade edge, which contributed to highly accurate edge reconstruction.

4.5 Conclusion:

Although the MicroGIURF tests returned mixed results, they indicate considerable promise for the future. The point extrapolation experiment demonstrated that many points placed using GIS surface sampling and GIURF calculation result in outstanding accuracy (<0.25mm median distance from the true edge for all tests), but substantial error was observed, especially near the microblade ends. Point extrapolation accuracy is also dependent on the sample selection method, where fully automated selection (the ideal analytical goal) results in less accuracy, while fully manual selection based on user-interpretation of lithic surface quality provides the best results. This presents a dilemma: with some relatively minor developments, MicroGIURF may be usable in conjunction with manual sampling, and may thus facilitate at least some macrowear quantification, which is an appealing tool. However, with its dependence on user input, this non-automated MicroGIURF is essentially just another subjective wear quantification method clad in a technological veneer of scientific pseudorigour; it is little better than "eyeballing" where the edge "should be" and basing the macrowear quantification on that supposition. Nevertheless, something is better than nothing, and early and encouraging results from MicroGIURF through manual sampling will lead to greater impetus to develop the automated sampling method. After all, the MicroGIURF module for QGIS is open source, and users are encouraged to edit, alter, distribute, hack, tear down, repair, and ultimately improve the tool themselves. The key is to avoid complacency: MicroGIURF is not good enough (yet)!

The surface reconstruction experiment was less encouraging; neither horizontal area reconstruction nor vertical surface reconstruction was satisfactory using GIS-based MicroGIURF. With optimal lithic surface conditions, using manual sample selection, MicroGIURF reconstruction achieved very strong results for horizontal accuracy and considerable improvements for vertical accuracy. However, these results are dependent on very selective sampling based on subjective user identification of ideal test cases on sections of

microblades. The goal of MicroGIURF is to reduce reliance on subjective analysis by a lithic expert, to automate the process to the greatest possible extent, and to be applicable to more artifacts (not just those with characteristics that are "perfect" for the MicroGIURF tool). The issue of optimal subjects for MicroGIURF analysis may be extended from the ideal surfaces used in the "cheat" test to the entire artifact category: microblades are artifacts with fairly simple morphology—an easily identifiable long axis, two relatively straight sharp lateral margins, no transverse flake scars—and thus should be well suited to automated sampling and edge extrapolation; they are essentially triangular (or trapezoidal) prisms. However, as the experiments showed, even with this optimal form edge extrapolation, surface reconstruction is unreliable, especially near the ends of the microblade, where wear is often most heavily concentrated. On this basis, MicroGIURF is not currently useful for reliably quantifying edge wear on lithic artifacts.

A further challenge for MicroGIURF will be extending functionality to other artifact categories. Microblades are morphologically very simple: they are triangular or trapezoidal prisms with sharp lateral edges. As such, it is not overly difficult to construct a QGIS script that can distinguish the lateral edges from the non-functional proximal and distal margins. For expedient flake tools with less standardized morphology it will be considerably more difficult to programmetrically identify the "business end", and to ignore step fractures, platforms, etc. Modified lithics, whether expedient flaked stone tools or intensively knapped formal tools, offer additional difficulties due to the large flake scars (often overlapping) intersecting with the edge. Quantifying wear on these tool types requires reconstruction of the intact-but-worked edge. This means that only the surfaces of the last flake scars are valid sampling sites, and that the algorithm must be modified to account for the conchoidal fracture pattern of the flake scars—the simple slope calculation adapted from GIURF combined with the aspect-based edge direction calculation will not work because the sides of the flake scars form a trough angling towards the middle of the flake scar as well as towards the lateral margin of the artifact. The ultimate version of MicroGIURF will have to detect flake scars and account for flake scar surface morphology. Furtunately this is not an insurmountable obstacle, and Davis et al.'s GLiMR toolkit (2015) has

employed a means of identifying large flake scars in their innovative GIS-based geometric morphometric analyses of projectile points. A GLiMR-MicroGIURF hybrid is a worthy goal.

Despite the failure of the DSM reconstructions, there are positive aspects of MicroGIURF that can provide a foundation for further endeavours in the area of automated edge wear quantification: First, the generally high quality of edge point placement shows that the underlying GIURF principle is strong, and may ultimately be applicable to DSM reconstruction. The strong results from the optimal microblade sections demonstrate that when lithic surfaces are ideal, slope towards an edge is highly consistent, and dense extrapolated points extend beyond the ends of the sampled surface, then reconstruction is more likely to be a good match for the original artifact. Future iterations of MicroGIURF (or similar methods) should build on these factors, hopefully automating optimal surface sampling and extending point extrapolation in order to bring better DSM reconstruction to the entire extent of artifacts, and to more lithic artifact categories. The other conclusion one may draw from the optimal surface experiment is that direct sampling guidance from an experienced lithic analyst (one with an understanding of MicroGIURF's algorithm and the effects of lithic surface characteristics on point extrapolation results) improves the accuracy of edge reconstruction. Surface quality matters, and if a human is capable of identifying suitable surfaces on a lithic artifact, MicroGIURF may yet be useful with only minimal tweaks to improve point extrapolation at the ends of blades. Rather than a fully automated computerized process (the ideal in terms of objectivity and replicability), MicroGIURF may be better understood as a technology to augment and amplify, rather than obviate, the skills of expert lithic analysts. Lithic analysts can rest easy: we will not be replaced by robots (yet).

Chapter 5 Conclusion and Future Directions

5.1 Summary and Conclusion

The three methods developed and evaluated in this dissertation were designed to address specific methodological gaps in lithic use wear studies. OpenHaft and Cumulative Stroke Distance (CSD) were designed to quantify aspects of gesture in lithic use wear experiments. OpenHaft is an electronic handle that measures the amount of force exerted on a stone tool's edge during use, and CSD is the total distance the tool is moved while in contact with the worked material. Together, these provide accurate data regarding the intensity and volume of action involved in creating use wear—important factors in wear formation (Rabinowicz 1995; Stachowiak and Batchelor 2014). The QGIS Experimental Edge Wear script collection was developed to accurately and objectively quantify edge damage on experimental stone tools. The basic premise was to apply GIS surface analysis modules to lithic surfaces, and thereby measure how much material was lost to macrowear (edge crushing, microflaking, etc.), and from where on the edge it was lost. The resulting QGIS scripts provide lithic analysts with tools to precisely monitor lithic edge attrition during use wear experiments, providing a means to better understand rates of wear, how wear is spatially distributed depending on hafting, gesture, tool form, etc., and thus build a more sophisticated model for analogy with archaeological artifacts. Finally, MicroGIURF is a set of QGIS scripts designed to adapt the principles and methods of Kuhn's (1990) Geometric Index of Unifacial Reduction Flaking (GIURF) and its successors (Eren et al. 2005; Eren and Sampson 2009) to quantifying edge damage on archaeological lithics. By programmatically sampling a digital surface model (DSM) of a worn microblade, MicroGIURF calculates the original position of the edge in an un-worn state and interpolates a DSM of the "pristine" microblade. Then the QGIS edge wear scripts can be used to quantify the edge wear. Like the QGIS experimental wear scripts, MicroGIURF is designed to study macrowear, and give researchers the ability to measure the intensity of tool use in archaeological contexts and build inferences regarding patterns of curation and discard as they pertain to the condition of a tool's edge. The MicroGIURF experiments yielded mixed results, and the algorithm requires further development to become useful. All of the tools and methods presented in this dissertation

use open source software and hardware, and all of the code and schematics needed for users to copy, distribute, alter, and develop tools further are provided.

5.1.1 OpenHaft and CSD

The impetus behind OpenHaft and Cumulative Stroke Distance (CSD) was to address the lack of control over gesture in most use wear experiments. While increasingly impressive technologies have been used to gather extremely precise data on the effects of use on experimental lithic artifacts, most use-wear experiments have focused more on the material components of technological behaviour rather than the behaviour itself. In recent years, two teams of researchers have tried to address this issue, with a load-sensing platform on which lithics may be used to process materials (similar to a cutting board) (Key et al. 2014; Key 2013; Stemp et al. 2015), and with a hand-held load- and torque-sensing device used in conjunction with a robotic arm (Pfleging et al. 2015; Iovita et al. 2017; Pfleging et al. 2018). While the load-sensing platform is a relatively simple, low-cost device, it forces the user to perform lithic use tasks on a cutting board, severely restricting the user's range of possible activities and actions. Pfleging et al.'s handheld device suffers no such gestural restrictions (though it does only work with hafted stone tools) but is exorbitantly expensive and utterly inaccessible to most researchers. OpenHaft is designed to incorporate the best aspects of each of the preceding tools without falling prey to their shortcomings; it is an electronic handle for hafted stone tools that wirelessly records precise load data. It is constructed with low-cost, off-the-shelf electronic components, and is built on the open source Arduino platform, allowing (and encouraging!) users to freely use, distribute, and develop the tool as they desire. Users can view—in real time—how much force they are exerting while using a stone tool, and how that changes across tasks and materials, and between users and tool types. CSD is simply a method to record the distance a tool covers under load; it is effectively a lithic odometer.

Experiments with OpenHaft and CSD showed that both use intensity (load) and use volume (cumulative distance) contribute substantially to use wear. The experiment was structured in two parts: a device trial using a graphite rod as a proxy for the stone tool (the rod is

a regular shape and of uniform consistency, so it may be expected to wear at a constant rate), and a material processing experiment using replicated dacite microblades. The graphite experiment yielded three main results: 1) subjective categories of "light" and "heavy" load are distinguishable in quantitative load data; 2) greater edge loading over equivalent CSD results in more wear; and 3) wear also increases as CSD increases, with equivalent maximum loading. The dacite experiment demonstrated that, for a single user performing two sets of the same activity, maximum edge loading remains generally consistent between task sessions. Despite the overall maximum loading consistency, edge loading varies between individual microblades; and that microblades with consistent edge loading and CSD will exhibit a high rate of wear (substantial volume of material loss) early in their use lives, and the rate of wear will taper as the microblade is used more and the edges stabilizes (becomes dull).

While several components of the OpenHaft/CSD experiment were designed specifically as methodological test-cases and demonstrations, both the graphite and the microblade experiments yielded results with implications for better archaeological understanding of lithic technologies. Specifically, in both the graphite and microblade experiments, the edge load data collected from OpenHaft revealed the extents to which tools form a reflexive relationship with the user, and directly inform gestural decisions. In the graphite experiment, the two hardness grades of graphite (HB and 2B) show significant differences between each other in the subjective "heavy" and "light" load categories, respectively. In both cases, the hardness of the graphite affects tool use. In the "light" load case, the task goal was to draw lines using the minimum amount of force needed to achieve a clear, consistent result, and the harder HB graphite required greater loading than the 2B to meet the task requirement. In the "heavy" load case, the task goal was to draw lines using the maximum amount of force that could achieve the result without breaking the graphite. Again, the HB set registered higher loading, because the user had to use less force with the 2B graphite, adapting the gesture to the less robust tool material. A similar, but less formal result was observed in the microblades, where less robust tools (either thinner, more delicate microblades or those in less firm haft settings) were applied with lesser loads than their sturdier counterparts.

Thus, OpenHaft data clearly illustrated how tool users adapt their gestures to the characteristics and capacities of individual tools, even if the ultimate task at hand is identical, which adds "hard science" support to extant social theory regarding technological behaviour (Ingold 2000; Pelegrin 1990; Tostevin 2011). In addition to the specific relevance of user/tool-adaptive loading in tasks, the experiment results show great promise for future applications of OpenHaft/CSD; gesture is visible in the data. Specific human actions and decisions related to the application of tool to task resulted in distinguishable patterns in load data, which means that future lithic experiments can engage with research topics such as decision making, strength and endurance, and learning and expertise in technological behaviour. The last is particularly productive ground for future research as tool edge loading may be among the ultimate indicators of expert performance in technological contexts: an expert practitioner may be expected to be efficient of movement, effort, and material in execution of a task, while an apprentice may be wasteful of all three. Efficiency improves as an apprentice's skills develop. By using OpenHaft, the gulf between master and apprentice may be witnessed by external observers, and the apprentice's improvement quantified. Analyzed in conjunction with tool wear data, it may be possible to observe how differences in skill and efficiency affect the tools' durability, which is in turn applicable to analogous artifacts in the archaeological record.

5.1.2 QGIS Edge Wear Quantification

The QGIS lithic edge wear quantification scripts are digital tools designed to address a specific gap in the lithic use-wear analyst's toolbox: observing, recording, and describing the effects of the chipping, microflaking, and edge rounding that many artifacts undergo. For decades, lithic "use wear" has been effectively synonymous with "microwear"-- the polish and microscopic striae that develop on the surface of tools as the result of use, and which are studied by lithic analysts in an effort to identify the specific use(s) of a given tool (e.g., Keeley 1974; Evans and Macdonald 2011; Stemp et al. 2015; Grace et al. 1985). "Macrowear"—the remains of the chipping and microflaking—has received relatively less attention and has generally been of a more qualitative than quantitative nature (e.g., Akoshima 1987; Semenov 1957; Tringham et al. 1974). Subjective description has been a proxy for quantification, with categories such as

"light" and "severe" wear (Bird et al. 2007). Quantifying microwear is ultimately a question of spatial analysis: how much does a surface change between its pristine and worn states, and where does that change take place? GIS is an ideal tool for this sort of lithic analysis as it is possessed of sophisticated surface analysis tools and is ultimately indifferent to scale, so by treating lithic surfaces as "micro-landscapes" one can apply powerful spatial analysis methods to wear quantification research. QGIS, a popular free and open source GIS, includes the capacity to compose one's own scripts combining multiple modules from the program's Processing Toolbox. The QGIS lithic edge wear scripts presented in this dissertation offer a method for quantifying macrowear in lithic experiment contexts, by measuring and mapping the difference between digital surface models of a lithic tool in pristine and worn condition. The volume of material difference between the two states is reported, as well as the wear index (the proportion of material lost at each position), and the precise location on the artifact where the material was worn away.

Perhaps the most important outcome of the QGIS experimental edge wear quantification script collection is its accessibility: any researcher with a personal computer and a digital camera (even a cell phone) can use this analytical toolkit to study edge damage formation on stone tools with precise, accurate, and replicable methods. From a research perspective, the QGIS script collection is likely most relevant in building an understanding of how different actions and materials affect tool edge stability. Simply stated, this is a means of watching lithic tools deteriorate, either through breakage or dulling (or both). This ability to precisely track macrowear accrual allows users to document how tool function and operational efficiency is affected by edge structure, and thus gain insight, through experimental analogy, into archaeological decisions surrounding curation, retouch, and discard. Sharp tools presumably cut better than dull tools, but how dull is "too dull"? Similarly, associating contact materials with rates of edge attrition is useful for more traditional microscopic surface use-wear studies. For example, 1000 strokes is a common task action in many wear studies that compare the wear effects of different materials (e.g., Giusca et al. 2012; Macdonald 2013), but it is unclear whether 1000 strokes is appropriate; 1000 strokes on unyielding material (i.e., antler, bone) may be far beyond the effective capacity of a stone edge, and the tool may have ceased to be effective many

strokes earlier. In an archaeological context, the tool might have been discarded or retouched long before the thousandth stroke, but a research assistant instructed to reach 1000 strokes may be indifferent to the actual cutting efficacy of the experimental tool. In such a case, the wear observed on the antler processing tool would be an unrealistic analogy for archaeological wear. By monitoring the development of macrowear and comparing that to archaeological specimens, experimenters could determine how many strokes (or preferably what CSD under a given edge load) constitute the maximum effective use life of a stone tool used for different tasks. Thus, instead of comparing equivalent stroke counts (or CSDs) the researchers could compare tools with equivalent proportions of dullness. Macrowear studies are very fertile ground for further research and advancing the archaeological understanding of tool function, toolkit design, and technological strategies.

5.1.3 MicroGIURF

The MicroGIURF QGIS script collection is intended to extend the QGIS edge wear quantification script functionality to archaeological artifacts by applying the principles of Kuhn's Geometric Index of Unifacial Reduction Flaking (GIURF) (Kuhn 1990; Eren and Sampson 2009; Eren et al. 2005) to 3D digital models of microblades. The premise is that by sampling the thickness (the Z-value or "elevation"), slope, and aspect of a digital surface model (DSM) of a worn microblade, the MicroGIURF algorithm can triangulate the position of the original, intact edge. Performed simultaneously for hundreds (or thousands) of samples, the entire length of the pristine edge can be extrapolated and then interpolated with the samples to create a digital reconstruction of the lithic in its (calculated) unworn state. Once that unworn DSM is produced, the QGIS edge wear quantification script may be used to precisely quantify the edge wear.

The MicroGIURF experiment had two components: reconstructing the edge and reconstructing the DSM. Both tests returned mixed results. The edge reconstruction test showed that the core premise of GIS-applied GIURF is generally sound, with most extrapolated edge points being positioned very close to the edge; less the 0.25 mm from the original edge for the least accurate sampling method (fully random sampling), less than 0.15 mm from the original

edge for surface sampling restricted to edge-adjacent polygons, and within 0.1 mm of the edge for samples taken along a user-positioned line. However, deriving the edge from the points proved to be more of a challenge, and the proportion of correct edge reconstruction, as measured by the amount of worn area (digitally) restored, was just over 47% at best, and under 3% at worst. The overall high quality of the edge point placement is encouraging for future improvement of the edge tracing component of the algorithm—there is a strong foundation to build on. The DSM reconstruction results were also mixed, with surface reconstruction accuracy (in this case defined as the reconstructed surface being within 0.1mm of the true surface) ranging from greater than 87% to less than 10%, and most results being around 50%. Despite the inaccurate MicroGIURF DSM reconstruction there is reason for encouragement. Very accurate edge and DSM reconstruction of hand-picked sections of two microblades shows that user-guided sampling and optimal surface morphology can achieve the desired results. Further development of MicroGIURF—especially point clustering for edge reconstruction around the ends of flakes—should yield a complete, usable toolkit for quantifying edge wear on archaeological artifacts.

Precise macrowear quantification is important in terms of building our understanding of how technological strategies were employed within past societies, as being able to identify connections between edge damage and tool applications, tool curation, raw material husbandry, and toolkit portability is integral to understanding economic and technical decision making in the lithic cultures that account for most of humanity's existence. This may extend to such factors as socioeconomic hierarchies and social control of resources; an individual who controls access to raw material may discard dulling tools earlier than someone without the ability to readily replace discarded tools. Mapping wear is also important because it identifies the "business end" of a stone tool. Many tools, such as microblades, require hafts in order to be used, but if the organic haft material decayed, the details of how a lithic component was articulated in a composite tool are often lost. Mapping wear offers a solution to this problem: identifying macrowear allows researchers to know which part of the tool was exposed and came into contact with contact materials, and which parts were shielded from damage by the haft. On an assemblage scale, wear-inferred hafting patterns reveal formal tool morphology and can indicate the cognitive ideal

for that specific implement (Bird et al. 2007; Schoville 2010). Automating this process and mapping precisely and objectively quantified wear would be a boon to researchers and is a worthy (and achievable) goal for MicroGIURF user/developers.

5.2 Open Source for Open Science

While specific research goals structured the development of the tools and methods presented in this dissertation, the project was also guided by an underlying philosophy: to develop tools and techniques following the precepts of the open science movement. Following Marwick et al. (2017), there are three pillars that make up the foundations of open science as it pertains to archaeology: open access (publications are accessible to the public without cost), open data (the data used to conduct studies are freely and readily available), and open methods; "Open methods are methods of data collection, analysis, and visualization that are available for inspection and reuse by the public." (Marwick et al. 2017, p.10).

This philosophy—that the tools of science are available to the public—is vitally important to one of the most basic principles of science: replicability. Scientific results are expected to be objective (as much as that is possible), and replicable. Replicability is severely compromised when access to the tools of science is restricted. While free, open source analytical software, such as R, is widely used and readily available, this is only one component of open methods as described by Marwick et al. (2017). Open data collection is equally important. If the means of archaeological data collection are sequestered in the halls of academia, archaeological data is hardly public. In North America, the subjects of most archaeology are the ancestors of the Indigenous peoples of this continent; a population that has historically been grossly underrepresented in academe. Data collection that depends on specialized and expensive resources available only to academic institutions systematically divorces descent communities from archaeological data. While recent years have seen increasing efforts towards meaningful

collaboration between academic institutions and indigenous communities (UBC's Reciprocal Research Network and the Indigenous/Science Research Cluster are examples of this), and many First Nations communities employ archaeologists, a great deal of data gathering, analysis, and knowledge production still occurs outside the descent communities. Open science is an avenue to decentralize the tools of data collection and analysis and empower communities in the stewardship and control of their own archaeology.

In an effort to adhere to, and promote the principles of open methods, public accessibility was a primary guiding principle for every methodological component of this project: OpenHaft was designed to be genuinely low cost (<$60 CAD) and uses the popular, easy, and open source Arduino platform, with the sketch (the term for Arduino code) released under the MIT license for open use; QGIS is free open source software, and both the experimental lithic wear quantification scripts and the MicroGIURF scripts are freely available for download, distribution, and alteration. Furthermore, every procedure and analysis in this dissertation was undertaken using free, open source software. Photogrammetry was performed using MicMac (Rupnik et al. 2017); point clouds were cleaned and aligned using CloudCompare (CloudCompare 2017); analyses were carried out using QGIS (QGIS Development Team 2015) and R (R Core Team 2016), with plots using the package ggplot2 (Wickham 2009); image analysis in FiJi (Schindelin et al. 2012); diagrams and illustrations in Inkscape (Harrington et al. 2015); photo editing using GIMP (The GIMP team 2013); and all text written on LibreOffice (LibreOffice Contributors 2000) and citation management through Zotero (Zotero Community 2013). The equipment used for research was either conventional, commercially available, and affordable (i.e., a Nikon D7000 digital camera, a soldering iron, stereomicroscope), or self-built (i.e., OpenHaft, a small microscope stage for photogrammetry). The project was both a research undertaking and an exercise in applying open methods to scientific problems, demonstrating that use wear studies can be performed on a limited budget, and freely providing the tools for other people to do so as well.

Using only open source and free or low-cost tools proved to be a challenge in some ways and resulted in some compromises in the research program. Obviously, the digital 3D models

produced using digital photogrammetry and a conventional digital camera are less precise than what might be achieved with laser-scanning confocal microscopy (e.g., Evans and Donahue 2008). Also, some open source programs are more difficult to use than their commercial counterparts, especially where graphical user interfaces (GUI) are incomplete or non-existent, such as the MicMac photogrammetry suite compared with the more popular commercial photogrammetry software AgiSoft Metashape (formerly called PhotoScan). Nonetheless, open source options allow users to fine-tune the functions of the program, and the intimate familiarity that is built with command-line interfaces and custom scripts results in a degree of understanding (and even mastery) of data collection, processing, and analysis processes that may not be feasible in easy but opaque proprietary software packages.

5.3 Future Directions of Research

While future directions of research are a common component of most dissertations, the usual format seems to follow a theme of "if I had more time..." or "my next project will be...", building directly on the research carried out for the project. In this case, since OpenHaft/CSD, the QGIS experimental wear quantification scripts, and "MicroGIURF" are all tools or methods, and are open source, this "future directions" section takes a slightly different perspective: limitations in the tools/methods, and possible ways to address them. Like the open-source tools and methods, these "future directions" should be considered open source as well: readers are encouraged to go ahead and do them!

5.3.1 Shortcomings and Solutions

5.3.1.1 OpenHaft and Cumulative Stroke Distance: Motion detection and recording

The first priority for future development of the OpenHaft/Cumulative Stroke Distance system is to incorporate a more precise and robust means of documenting tool motion rather than simply relying on the user to consistently perform strokes of identical length, or take manual measurements of individual strokes. A limitation of the OpenHaft/Cumulative Stroke Distance

method is the precise recording of individual stroke lengths (and thus the range of error in calculating CSD). In lab-based constant-length stoke experiments, the true individual stroke lengths are the result of the user's ability to keep the tool within the desired gesture parameters. While marked distances on contact material provide useful guidelines for the tool user, the inherent vagaries of human movement necessarily result in some minor variation in stroke length.

Pfleging et al.'s (2015) use of the AprilTag system of motion tracking through visual landmarks offers a possible solution for the CSD precision problem, where the position and orientation of the tool is monitored by video on a frame-by-frame basis (Olson 2011). AprilTag relies on visual landmarks printed on a card fixed to the tool and oriented towards the camera. The specific geometry of the April Tags landmark image (essentially a QR code) changes in the camera viewfield based on perspective, permitting the software to record tool orientation as well as motion (Olson 2011). Informal trials of OpenHaft motion tracking through visual landmarks using ImageJ achieved basic two-dimensional tool position monitoring, though AprilTag is definitely a superior system with its ability to calculate three-dimensional position and tool orientation.

While augmented video recording is certainly useful, two issues are evident in the AprilTag system: 1) AprilTag motion tracking requires the fiducial mark to be positioned on the tool in such a way that it can be recorded by a fixed camera, and 2) AprilTag only marks the tool and not the contact material. First, the dependence of AprilTag on video recording means that any gesture performed with an AprilTag-enabled tool must be carried out with the tag and pattern in view of the camera throughout the motion. Although the system can record some twisting and turning of the subject tool, the range of motion is restricted by the need to keep the tag facing the lens and the view unobstructed. The tool user could not, for instance, reposition his or her body in order to better carry out a butchering task on a large mammal cadaver if that movement resulted in the researcher blocking the camera or significantly changing the tool's orientation relative to the camera. The problem of tag visibility is not insurmountable, though it does require (and early trials are underway) to incorporate a 9-degrees-of-freedom (9DoF)

Inertial Measurement Unit (IMU) into the next iteration of OpenHaft. A 9DoF IMU is a circuit with an accelerometer, gyroscope, and magnetometer, all operating on three axes. Together the three components can detect the orientation and movement of the tool through space. Because the IMU is connected directly to the OpenHaft's microcontroller, the orientation and movement data is packaged and transmitted by Bluetooth along with the load and stroke data. This means that the tool user can move around and perform a task without regard to restrictions placed by a camera's field of view.

Unfortunately, the IMU does not solve the issue of only the tool's movement being registered, and not that of the contact material. Both the IMU and AprilTag systems suffer in that the tool orientations and movements are recorded in relation to "real space", which does not matter; what matters is how a tool moves relative to the contact material. That tool-material movement relationship is what defines the task, and unless the contact material is totally stationary, recording the tool's absolute movement is meaningless. Thus, both the AprilTag and the IMU systems are hindered by the same fixed position issue that hindered the "digital cutting board" studies in lithic edge loading (Stemp et al. 2015; Key et al. 2014; Key 2013; Key and Lycett 2014). CSD is currently the most accurate and robust means of quantifying tool motion in tasks involving hand-held contact material.

One possible solution to the tool-material relative motion problem would be to record the position of the contact material as well, either using a second AprilTag or a second IMU. This way, while the tool user wields the motion-tracking OpenHaft in one hand, the other hand can support the worked object, which is also motion-tracked. While this is viable for small hand-held objects with specific orientations and limited surfaces, or objects with specific planes such as stretched hides, it does not address the issue of larger, more complex objects with multiple surfaces, such as the hypothetical large mammal cadaver. In that context, it may be possible to incorporate the OpenHaft IMU with a 3D motion tracking system such as a re-purposed Microsoft Kinect (Tian et al. 2015), but combining OpenHaft and Kinect is beyond my abilities. Thus, this future direction of research depends on others taking advantage of OpenHaft's opensource nature to further develop the system.

5.3.1.2 QGIS Experimental Use-wear Toolkit: Plugin Development

For the most part, the QGIS experimental use-wear toolkit is ready to use. There is a learning curve inherent with the photogrammetry and point cloud matching, but that is easy enough to overcome. The most pressing improvement has to do with distribution of the scripts. Currently the scripts are hosted on Github at https://github.com/nwaber/QGIS_lithics and must be installed in the QGIS program folder. Ideally the scripts would be converted to a QGIS Plugin and distributed through the plugin repository. This has not yet been done, largely due to my limited fluency in Python 3, the language of the current QGIS version.

5.3.1.3. Blind testing of QGIS wear quantification

A cogent question regarding the utility of QGIS wear quantification is whether it can achieve results that are demonstrably superior to subjective assessments of wear intensity by lithic analysis experts. To test this, it would be necessary to conduct double-blind experiments where lithic analysts independently examine a set of experimental lithics using traditional visual/macroscopic (<100x magnification) methods, as well as the QGIS wear quantification toolkit. Wear quantification quality can be gauged by comparing inter-analyst variation in human and machine tests. The optimal method may be the one where multiple researchers working independently with the same artifact achieve the highest degree of agreement with each other.

Blind-testing QGIS wear quantification against human analysts is valuable because it provides a dual benchmark: human analysts are currently (and traditionally have been) the "state of the art" for lithic wear assessment, and visual/macroscopic examination is by far the most commonly used method of wear analysis. Human subjectivity and inter-analyst variation, are major driving factors behind the development of digital wear quantification methods in order to improve objectivity and replicability. Using the QGIS wear quantification method optimally means that two independent operators can achieve identical results in independent assessments of the same artifact. However, if a blind test of expert analyst wear assessments against the QGIS method shows that humans are capable of comparable results to machines, then we may infer that analyst subjectivity is less of a factor than was initially supposed, that independent computer operators result in as much (or more) variance as traditional analysts, or a combination of both.

5.3.1.4. MicroGIURF: Broken microblade extensions

MicroGIURF offers fertile ground for improvement. First, a means of estimating the lost volume of broken microblades would be useful for estimating the original form of fragmentary tools (of which there are many), and how much edge damage occurred. In the MicroGIURF experiment, edge wear volume calculations were particularly inaccurate for microblades where the proximal or distal end was snapped off because the missing surface was not sampled, and thus the edge was not reconstructed. If one could digitally extend the surface based on a microblade length estimate, it may be possible to calculate the volume of lost material for that portion of the blade.

Digitally extending broken microblades involves two basic functions: defining the form of the extension and determining how far it should extend. The extension form is relatively simple: a line of point samples of the most intact surface spanning the majority of the width of the microblade adjacent to the break may act as a model for the extension. A more nuanced algorithm may use a series of such lines and calculate a linear regression model to account for the microblade tapering in width or thickness. The more difficult aspect of the algorithm is calculating how far to extend the microblade. This could potentially be calculated based on length tendencies observed in complete microblades of the same material in the same assemblage. It may be reasonable to infer that, since (complete) microblade length is largely a function of the length of a microblade core's fluted face (Magne 1996; Callahan 1985; Flenniken 1987), multiple microblades from the same core are likely to have been of similar lengths when intact. This assumes that none of the microblades were step terminations, that the microblades were not deliberately broken prior to use, and it precludes major core rejuvenation flaking that significantly changed the length of the fluted face.

Based on these two premises, the hypothetical QGIS microblade-extending script may function as follows: the user inputs the worn microblade DSMs and perimeter, indicates whether the script should extend the proximal or distal end (or both), enters the median length of comparable microblades in the assemblage, and indicates a line drawn at the desired sample

location(s). Another line marking the axis of the microblade would also be necessary, though it could be derived by calculating the azimuth values of the lateral edges of the microblade perimeter file. The script then generates points along the sample line and records the DSM values. The points are replicated, maintaining the same X- and Z-axis values, but iterating along the Y-axis until the median complete blade length is reached. These points may then be interpolated to produce a new microblade surface for that broken section of blade, and then MicroGIURF may be used to extrapolate the edges for that section. Finally, the lost volume may be estimated.

This method of estimating snapped flake volume has several shortcomings: the microblade length estimation is based on several tenuous assumptions regarding morphological consistency of the core and other microblades (as mentioned above); the lost material volume estimation will be imprecise, especially compared with edge wear volumes; and even after reconstructing the broken end of the microblade, MicroGIURF may not extrapolate the edges correctly (especially at the distal or proximal corners). Nonetheless, it may be argued that it is desirable to at least have a working estimate of broken microblade volume when analyzing microblade use in an assemblage.

5.3.1.3 MicroGIURF: Quality point mapping at distal/proximal ends

One of the most obvious shortcomings of the MicroGIURF script is the poor edge extrapolation near the proximal and distal corners. Given the cluster density edge extrapolation method considers the maximum density value of each cluster against those of the other clusters, areas with more sample points (the medial section of a microblade) have more extrapolated points, and thus higher density values in the clusters than areas where the sample points are restricted by the end of the microblade. The result is that density clusters near the microblade ends are erroneously omitted from the edge reconstruction. Ideally, they would be included.

Two possible solutions to the cluster density issue are to weight cluster density values based on the sample points' proximity to the proximal or distal ends, or to place multiple extrapolation points for each sample in order to "overshoot" the ends. Either method is feasible using the existing QGIS script. To weight the density values, one simply needs to query the relevant points layer for the minimum, median, and maximum Y-axis values, and then apply a multiplier in relation to the difference between each point's Y-axis value and the median value. A greater difference results in a higher weight. This method would require substantial experimentation with the weight value to ensure that the revised algorithm is picking out sound clusters at the proximal and distal ends but is still omitting outliers. That may be a challenge if complex surface topology results in outlying clusters that are assigned very high density values because they are so far from the midpoint (and the rest!) of the microblade.

The second option involves the extrapolation from the surface samples to the edge. The MicroGIURF point extrapolation algorithm already calculates the aspect angle of the surface, which in the case of microblades is (theoretically) perpendicular to the lateral margin. Once the extrapolated point is positioned, it should not be difficult to use a simple Python command to place a handful of points adjacent to the extrapolated point along an axis 90° to the aspect angle. The existing algorithm already places a point at a specified interval along a given bearing, so it is simply a matter of iterating that command but from a different reference point (the extrapolated point's coordinates), at a much shorter interval, and at 90° off the previous bearing. The result should be a fan of extrapolated points from each sample location, extending beyond the ends of the microblade, and thus resulting in point clusters at least up to the ends.

5.3.1.4 MicroGIURF: General Surface Quality: Automated vs. User-selected Sampling

The substantial differences observed in edge point accuracy between the fully-automated sampling, edge-adjacent sampling, and user-positioned line sampling in the MicroGIURF experiment reveal another avenue for development with the MicroGIURF script: improving automated surface sampling so the sampled surface quality matches that of the user-positioned

line sampling. This will be a challenge as it requires the algorithm to recognize surface quality—to seek smooth surfaces with minimal curvature and no step fractures—in order to select optimal sample locations. One way to achieve this may be to capitalize on the strong tendency observed in all three methods for most extrapolated points to be positioned quite close to the edge line, from which one may infer that most surface samples are from clear sections of the DSM. Taking this one step further, surface slope is the factor most responsible for the distance an extrapolated point is placed from the sample point. Therefore, if most extrapolated points are placed close to the edge line, most samples must have accurate slope values, and thus the interquartile range (IQR) of slope values most likely reflects the highest quality surface samples. By only using sample points with slope values within that IQR for point extrapolation, it should be possible for randomly sampled points to at least mimic the subjective surface quality identification that is seen in the user-guided method.

One important consideration with the IQR quality metric is that it must be calculated individually for each edge-adjacent surface. In the case of microblades with a single dorsal arris, this should be a relatively simple matter, where slope aspect values for each of the two dorsal faces (the surface on either side of the arris) will likely fall into a consistent range and thus be easy to classify as “left-oriented” or “right-oriented”. The problem becomes more complex when a microblade has two dorsal arrises—one on either side of a central longitudinal flake scar. That central flake scar forms a third major surface that, depending on its size, could override the slope IQR calculation of the edge-adjacent surfaces. The semi-automated edge-adjacent sampling method was developed specifically for this reason. Theoretically, it may be possible to programmatically define edge-adjacent surfaces based on the ridge or break in slope created by the dorsal arrises combined with their proximity to the blade margin. A similar method was applied to the task of finding shore-adjacent low-angle landforms in Prince Rupert Harbour using LiDAR data (Letham et al. 2018).

Unfortunately, the QGIS model used in Letham et al.’s 2018 study cannot be transposed seamlessly to microblades—whereas the sought-after coastal landforms could be picked out due to their predictable slope between 0 - 7°, the slopes of the edge-adjacent microblade surfaces are

unknown so it is necessary to define the surfaces another way. One possibility is to use a topographic position index (TPI) based landform classification system, which identifies areas of a DEM relative to each other and labels them using 10 terms from canyon through valleys, plains, and slopes to mountain tops and high ridges (Weiss 2001). Although informal trials have hitherto been unsuccessful, it should be yet possible to use TPI to define the boundaries of each dorsal flake scar surface to create a contiguous area bounded by ridges (arrises), and then use a modified version of the portion of Letham et al.'s workflow (2018) for picking landforms near shorelines to omit the central trough. If this method works, it should result in a fully-automated sampling strategy that selects highest-quality dorsal flake surfaces with as much success as the user-guided method, except that this would not only result in hundreds (or thousands!) more samples, but would also be unaffected by the natural variances and biases of manual surface assessment. It would not matter how many good or poor surfaces preceded each iteration (a factor that may affect subjective assessments of surface quality), or whether wear is visible (or even likely), but would apply the same sample quality assessment rubric every time. Ultimately this should result in higher quality reconstructions.

5.3.1.5 QGIS Edge Wear and MicroGIURF: Extend Reconstruction Algorithm to Other Tool Types

While microblades are a fascinating subject and a lithic artifact type with global relevance, the basic premise behind MicroGIURF (and GIURF)—that the volume of material removed from a flake edge may be quantified using a projected edge model—should be generally applicable to other lithic edged tool types. The development of MicroGIURF and the QGIS edge wear quantification toolkit was driven by a specific interest in microblades, which resulted in several assumptions and structures being microblade-centric. One example of this is that the QGIS use wear algorithm default setting is for "linear artifacts"; long, straight flakes with parallel lateral margins (microblades or other blade-like tools). This results in wear positions being expressed in millimetres distant from the proximal end, and being in the proximal, medial, or distal portion of the left or right lateral margin. In the QGIS 2.X versions of the script the user may select an option for non-linear (radial) wear mapping which describes wear locations based on the bearing from the geographic centre of the artifact. This is similar to

the radial coordinate method used by previous authors to map wear on Middle Stone Age points (Bird et al. 2007; Schoville 2014, 2010, 2016). However, the radial mapping function has not yet been ported to a QGIS 3.X Processing Toolbox model.

MicroGIURF is perhaps less microblade-centric and more unmodified flake-centric. The surface sampling and edge extrapolation algorithm is based on samples being taken from a surface with an aspect (slope direction) oriented directly towards the flake's nearest edge. Once again, because MicroGIURF was designed with microblades in mind, this is not a problem: because of the nature of microblade manufacture, the prepared-core fluted surface is produced by long, thin, parallel flake scars where (ideally) the nadirs of paired conchoidal troughs become the lateral margins of a microblade and the inter-scar ridge is the dorsal arris. Therefore, the MicroGIURF sampling algorithm simply reads the surface aspect at a given point and projects the corresponding edge point on that bearing.

The MicroGIURF algorithm may not seamlessly work for flakes with more complex dorsal surface topography, however, because multi-directional flake scars mean that the surface aspects do not necessarily face the nearest edge. This is compounded in the case of heavily retouched surfaces, such as those of projectile points or retouched flake tools with intrusive flaking, where multiple flake scars originate from the edges. If the tool is relatively thin, the surface aspect of a point within a scar may be more likely to register a bearing towards the centre of the scar rather than the edge of the artifact at all (Figure 5.1). In this case the aspect is the result of two slope axes compounding: the general flake form sloping towards the lateral margin, and the flake scar surface sloping towards the conchoidal trough of the scar.

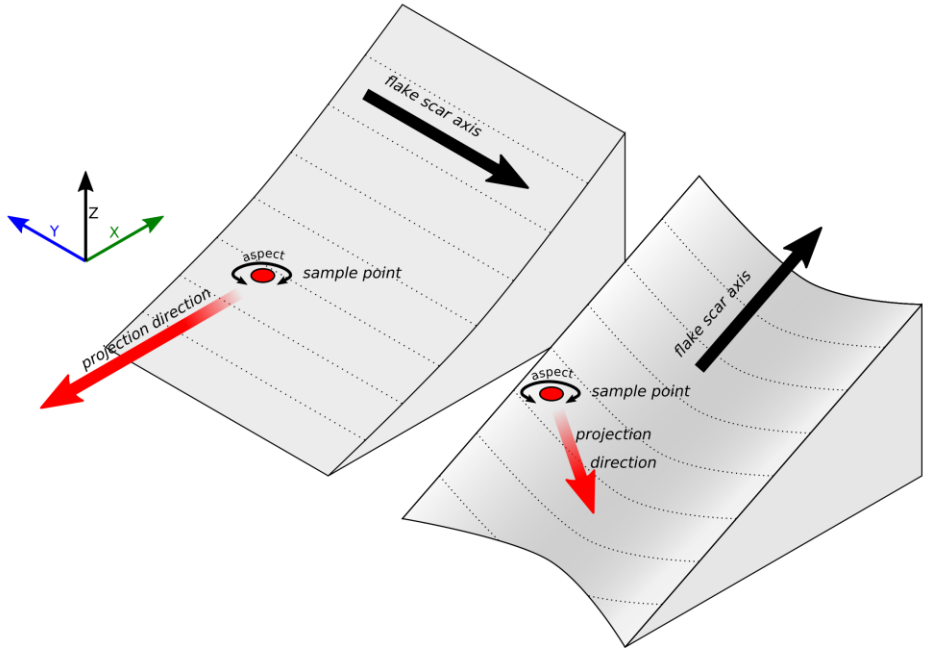

Figure 5.1: Idealized diagrams of edge point projection based on a longitudinal flake scar axis as seen on a microblade (left) and a lateral axis as seen on heavily retouced artifacts (right).

The solution to this issue is unclear at present. Davis et al.'s GLiMR (2015) applied an aggressive GIS surface filter that effectively erased flake scars on projectile points, leaving a smooth flattened scar-less surface. While the precision of this method was inadequate for accurate edge wear quantification (I tried this method before switching to GIURF), it may be suitable for determining general surface aspect. Alternatively, it is also possible to replace the aspect value with a direction value generated from the QGIS distance matrix module. This simply directs the point projection towards the nearest place on the edge. In microblade contexts the distance matrix is useful until samples are taken nearer to the proximal or distal ends than to the lateral margins; in those cases, the point is projected towards the end instead of the edge of the microblade. In flake tools without the clear linear morphology of microblades, the proximity grid may be the optimal method of determining the direction in which to extrapolate the edge points. However, it should be noted that when the aspect value does not correspond to the direction to the edge, slope must be sampled in a different way. As discussed above, non-edge-oriented aspect means that that slope is the product of combining a global slope and a local slope.

The key is to identify and use the global slope. Again, a filtering method similar to Davis et al.'s GLiMR may be useful. Alternatively, it may be possible to use the proximity grid direction value to generate multiple aligned sample points, sample the DSM, and derive the slope from those points.

Adapted MicroGIURF to other lithic tool categories would be very useful. Although microblades have near-global distribution and tens of millennia of history, they are far less common in most sites than conventional expedient flake tools. Being able to accurately quantify wear on all tool categories would not only be valuable for understanding patterns of tool use at non-microblade sites, but it would also permit researchers to compare wear intensity between tool types. Quantified wear on utilized flakes could reveal aspects of decision making around how immediate task needs guide expedient tool design. This would open a window into tool use and curation practices by letting us understand how (or if) past peoples associated tool types or flake characteristics with specific tasks or duty levels.

5.4 The Big Picture: Relevance Beyond the Lab

The tools and methods presented in this dissertation have relevance to archaeology and anthropology at three scales: immediate technical utility in lithic analysis; future applicability to archaeological studies of tool use and technological strategies in past societies; and ultimately better understanding of human engagement with technology, and the relationships between tool users and their equipment. Technology is the means by which humans interface with their environments (Ingold 2004), and the nature of that interface affects and informs the interaction. Therefore, in order to best understand human behaviour, it is necessary to understand how that behaviour is (and was) structured, filtered, and articulated through technology. This includes examining perceptions and emic categorization of tools—avenues of inquiry that are generally closed to archaeologists. The open source tools presented here initiate this effort by providing researchers with means of better recording gesture in experimental contexts, and accurately tracking how gesture, tool morphology, contact material, and other task parameters affect edge longevity.

At their core, OpenHaft, cumulative stroke distance (CSD), the QGIS edge wear quantification modules, and MicroGIURF are all designed to examine technological behaviour, but they approach that behaviour from opposite ends: OpenHaft/CSD focus on what "goes in" to a task: the human-tool engagement that is technological action. This follows the model of technology as skilled practice, where human and tool are not considered to be mutual externalities, but rather an integrated whole (Franklin 2004; Harvey 1997; Ingold 2000). In this case, instead of understanding the tool as an implement that facilitates execution of a task by an individual (and therefore the tool-as-object is an inert entity with identical functionality regardless of user—and objectified object), the tool and the gesture are indivisible (Tostevin 2011), and therefore personal and unique to individual people, albeit within a general oeuvre. In archaeology, obviously, the gestural component of technology does not preserve, so it is only the tool—and even then, often only a fragment of the tool—that is available to archaeologists. The premise is that by close observation of that fragmentary physical component we can reconstruct the gesture, task, and thereby indirectly observe the past behaviour. Experimental archaeology extends this premise to incorporate modern analogy: if the results (i.e., edge wear) of directly observable actions taken by an experimenter with replica tools are analogous to results observed on archaeological specimens, we may infer that similar actions were likely performed in the past (Amick and Mauldin 1989; Bamforth 2010). OpenHaft and CSD contribute to the analogical by recording the intensity and duration of a tool's active use life, thereby revealing aspects of technique—the skilled application of gesture—that are otherwise invisible. "How hard" and "how much" a tool was used, variables that have hitherto been largely subjective in experimental use wear studies, can be accurately and consistently quantified, and their effects studied in relation to quantifiable wear.

In addition to providing data regarding input variables, quantified gesture in lithic experiments is important to the study of technological behaviour because it permits individuals external to the technological system to observe how the recursive relationship between user, tool, and material affects action. It reveals decisions and adaptations made by the user, and therefore provides information regarding skill, practice, learning, and even task goal hierarchies. Technological actions exist in an economy where aspects such as gestural precision and output quality are

weighed against energy and time budgets, and material capacities. Hasty work may lead to sloppy results but overly fastidious work takes too much time to complete a task; too much force results in broken tools and/or contact material but too little force is ineffective; balance is necessary for efficient and successful task execution. Directly monitoring how an operator adapts his or her action to a task can bring these balances to light, and using a tool such as OpenHaft means that monitoring is not limited to visual observation and subjective description of perceived effort and action. Instead researchers can effectively "see inside" gesture. Efficiency and success are also socially-constructed concepts, and by identifying how the varying factors of a technological strategy (*sensu* Nelson 1991) are balanced, and thus scrying the hierarchy of priorities within the technological system, we are able to better understand what constitutes technological "success" in different contexts, gaining insight into a set of enculturated values that are not otherwise available to people outside of that society (especially not to those displaced by millennia).

Measuring how the object and material components of technological behaviour inform gesture is an intriguing and important line of inquiry beyond archaeological contexts because it is relevant to understanding skill, learning, and the novice-expert continuum of practice in all fields. Optimal balance in the task action economy is a marker of expert performance, and measuring how variation in competence and experience are manifested in differences in gesture (differences which may otherwise be invisible to external observers) gives us new means of studying how expertise is developed, what the effects of mentorship/apprenticeship are, and even how skill-plateaus, "knowledge-dumps", breakthroughs, and deliberate practice (Ericsson et al. 1993) affect trajectories of learning. Expert performance is a growing field of study across several disciplines, and has relevance throughout society, and examining the phenomena in archaeological contexts offers a unique perspective regarding how it manifested over time.

Following gesture and action as input, in the context of lithic activity, edge wear represents the output of technological behaviour. The QGIS experimental edge wear quantification suite and MicroGIURF are both designed to let lithic researchers precisely and accurately quantify edge wear, and thereby describe the degradation of lithic tools over their use

lives. This description is important in archaeological contexts, as it may provide insight into the emic categories of "sharp" and "dull", as perceived by people of the past. The fundamental functional criteria of an edged tool is its ability to cut—a function that depends on a sharp edge. Sharpness, however, is a socially constructed parameter, based on effectiveness within a specific task but ultimately arbitrated with a learned threshold on a sharp-dull spectrum. The threshold is not a fixed benchmark dictated by objective criteria, but rather a negotiated position, informed by functionality (of course), but also material availability, desired outcome, and practitioner skill, knowledge, and even personal taste. A response to a lithic tool reaching the threshold of "too dull", and thereby passing out of the subjective category of viability, is to either re-sharpen or discard the exhausted implement, the latter of which transfers the tool into the archaeological record.

Based on the premise of discard marking either tool exhaustion (or breakage) or task completion, artifacts recovered from archaeological contexts provide researchers with the fundamental operating parameters as understood by the original tool designers, makers, and users. Used edges represent tools that were at one point within those parameters. If one can reconstruct the pristine edge of a worn tool, and determine how much the tool was used before it was discarded, one may gain insight into how original design parameters articulate with the exigencies of manufacture and use. To achieve that, two data sets are needed: an accurate edge reconstruction with precise wear mapping and quantification, and a comparative collection of worn experimental tools with known use-life histories. The QGIS experimental edge wear quantification suite and MicroGIURF were designed specifically to provide this data.

Understanding the dullness threshold, and how it was determined in different social and ecological contexts in archaeological settings provides insight into how humans structure economic priorities such as weighing optimal task execution or tool efficiency against the material and opportunity costs of tool modification or repair. Viewed across societies and throughout time, patterns in these economic decisions express enculturated values regarding what constitutes "success", and how to go about achieving it. Technological strategies are design responses to problems (Bleed 1986; Bousman 1993; Nelson 1991), and the specific cost-

benefit balances that are struck within a strategy are the tactics by which it is executed. The trade-offs around defining, attaining, and maintaining optimal edge sharpness are one such tactic, and understanding how costs are weighed in tactical contexts is an imported component of understanding the strategy as a whole. Ultimately, seeing how past cultures employed technological strategies is valuable in regards to how modern societies do the same: we can observe how circumstances came to be, what strategies were employed in that context, and what the consequences were of the design response. The external perspective gained by looking into the deep past permits us to view the trajectories of history far beyond the timescales of human lifespans, but at risk of entrenching the viewer's etic position. Using tools such as those presented in this dissertation, the archaeological observer may gain better access to the emic perspective—the "inside view"—of the technological strategies of the past.

5.5 Concluding Statement

The primary goal of this dissertation was to produce a useful, accessible, and novel toolkit for lithic researchers interested in quantifying use wear. The research split into three primary streams: managing gesture in lithic use experiments, quantifying macrowear on experimental lithics, and quantifying macrowear on archaeological lithics. The experimental components produced methods and tools—both hardware and software—that address core problems in wear quantification experiments. OpenHaft and CSD combine to record and control the volume and intensity of tool use. Now lithic experimenters, whether studying the effects of lithic raw material, tool morphology, or contact materials, can isolate and control for how much and how hard a tool is used in a test. The QGIS experimental wear quantification script collection is complimentary to OpenHaft and CSD; the workflow permits researchers to identify precisely where damage occurs on a flake edge, and to quantify the volume of lithic material lost to that flaking or crushing at any given point. In addition to individual artifact analyses, this can be applied to assemblage-wide studies, tracking wear tendencies across multiple artifacts.

MicroGIURF was designed to apply established principles in lithic tool analysis to the study of edge wear. The core premise is that, using a QGIS script based on Kuhn's GIURF

(Kuhn 1990) and Eren's later developments of that model (Eren and Sampson 2009; Eren et al. 2005), one may derive the original flake edge from the flake surface, and quantify wear by comparing the digitally reconstructed edge to the worn edge. This essentially would apply the QGIS experimental edge wear tools to archaeological lithics. Unfortunately, the edge reconstruction component of MicroGIURF was unsuccessful, which prevents accurate wear calculation. However, the experiment worked well enough to show promise for future development and refinement based on the MicroGIURF foundation.

Ultimately, the three components of this dissertation—OpenHaft/CSD, QGIS experimental edge wear scripts, and MicroGIURF—are designed to function in conjunction with each other. OpenHaft/CSD and the experimental edge wear scripts already work well together, as demonstrated in the microblade wear experiment. Controlled gestures with recorded edge load account for two important (and oft-neglected) variables in any use wear experiment. Precise macrowear quantification provides data regarding the intensity of a tool's use life. Together, these are methodological components that account for the input controls and output data collection of a robust use wear experiment; it is up to the user to select variables: tool form, lithic material, contact material, task (nature or duration), etc. MicroGIURF will hopefully develop into the vital third piece: the means of applying the experimental findings directly to archaeological examples. All of the tools are released under open licenses; therefore, all of the internal workings are visible and accessible to users, and updates, improvements, new features, and adaptations to other methods, techniques, or technologies are all immediately available to anyone with the time and expertise (and/or enthusiasm!) to tinker.

References

Akoshima, K. (1987). Microflaking Quantification. In G. D. G. Sieveking & M. H. Newcomer (Eds.), *The Human Uses of Flint and Chert: Proceedings of the Fourth International Flint Symposium Held at Brighton Polytechnic 10-15 April 1983* (pp. 71–80). Cambridge University Press.

Akoshima, K., & Kanomata, Y. (2015). Technological organization and lithic microwear analysis: An alternative methodology. *Journal of Anthropological Archaeology*, *38*, 17–24. https://doi.org/10.1016/j.jaa.2014.09.003

Amick, D. S., & Mauldin, R. P. (1989). *Experiments in Lithic Technology*. Oxford: B.A.R. International Series.

Andrefsky, W. (2005). *Lithics: Macroscopic Approaches to Analysis*. New York: Cambridge University Press.

Bamforth, D. B. (2010). Conducting Experimental Research as a Basis for Microwear Analysis. In J. R. Ferguson (Ed.), *Experimental Research in Archaeology: Examining Technology through Production and Use* (pp. 93–111). Boulder, CO: University of Colorado Press.

Beutel, A., Mølhave, T., & Agarwal, P. K. (2010). Natural Neighbor Interpolation Based Grid DEM Construction Using a GPU ∗. *GIS 2010*.

Bird, C., Minichillo, T., & Marean, C. W. (2007). Edge damage distribution at the assemblage level on Middle Stone Age lithics: an image-based GIS approach. *Journal of Archaeological Science*, *34*(5), 771–780. https://doi.org/10.1016/j.jas.2006.08.005

Bleed, P. (1986). The Optimal Design of Hunting Weapons: Maintainability or Reliability. *American Antiquity*, *51*(4), 737–747.

Boëda, E., Geneste, J.-M., & Meignen, L. (1990). Identification de chaînes opératoires lithiques du Paléolithique ancien et moyen. *Paléo*, *2*, 43–80.

Borchers, H. W. (2016). *pracma: Practical Numerical Math Functions. R package version 1.9.3.* https://cran.r-project.org/package=pracma

Bousman, C. B. (1993). Hunter-Gatherer Adaptations, Economic Risk and Tool Design. *Lithic Technology*, *18*(1 & 2), 59–86.

Callahan, E. (1985). Experiments with Danish Mesolithic Microblade Technology. *Journal of Danish Archaeology*, *4*, 23–39.

Chiabrando, F., Donadio, E., & Rinaudo, F. (2015). SfM for Orthophoto to Generation: A Winning Approach for Cultural Heritage Knowledge. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-5/W7*, 91–98. https://doi.org/10.5194/isprsarchives-XL-5-W7-91-2015

Christensen, T., & Stafford, J. (2005). Raised Beach Archaeology in Northern Haida Gwaii: Preliminary Results from the Cohoe Creek Site. In D. W. Fedje & R. W. Mathewes (Eds.), *Haida Gwaii: Human History and Environment from the Time of the Loon to the Time of the Iron People* (pp. 245–273). Vancouver: University of British Columbia Press.

Clark, J. E., & Woods, J. C. (2014). Squeezing Life from Stones: The Human Side of Replication Experiments. *Works in Stone*, *086*, 197–212.

CloudCompare. (2017). *CloudCompare v2.10*. http://www.cloudcompare.org

Croes, D. R. (1995). *The Hoko River archaeological site complex: the wet/dry site (45CA213), 3,000-1,700 BP*. Pullman, WA: Washington State University Press.

Davis, L. G., Bean, D. W., Nyers, A. J., & Brauner, D. R. (2015). GLiMR: A GIS-based method for the geometric morphometric analysis of artifacts. *Lithic Technology*, *40*(3), 199–217. https://doi.org/10.1179/2051618515Y.0000000007

Dickie, R. (2015). *Identifying Microblade Function at EeRb-140 and EeRb-144, Kamloops, British Columbia* (Unpublished M.A. thesis). Simon Fraser University, Burnaby, BC.

Duarte, L., Teodoro, A. C., Moutinho, O., & Gonçalves, J. A. (2016). Open-source GIS application for UAV photogrammetry based on MicMac. *International Journal of Remote Sensing*, *38*(8–10), 3181–3202. https://doi.org/10.1080/01431161.2016.1259685

Edmonds, M. (1990). Description, Understanding and the Chaine Operatoire. *Archaeological Review from Cambridge*, *9*(1), 55–70.

Eren, M. I., Dominguez-Rodrigo, M., Kuhn, S. L., Adler, D. S., Le, I., & Bar-Yosef, O. (2005). Defining and Measuring Reduction in Unifacial Stone Tools. *Journal of Archaeological Science*, *32*(8), 1190–1201. https://doi.org/10.1016/j.jas.2005.03.003

Eren, M. I., & Prendergast, M. E. (2008). Comparing and Synthesizing Unifacial Stone Tool Reduction Indices. In WilliamEditor Andrefsky Jr (Ed.), *Lithic Technology: Measures of Production, Use and Curation* (pp. 49–85). Cambridge University Press. https://doi.org/10.1017/CBO9780511499661.004

Eren, M. I., & Sampson, C. G. (2009). Kuhn's Geometric Index of Unifacial Stone Tool Reduction (GIUR): does it measure missing flake mass? *Journal of Archaeological Science*, *36*(6), 1243–1247. https://doi.org/10.1016/j.jas.2009.01.011

Ericsson, K. A., Krampe, R. T., & Tesch-Romer, C. (1993). The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*, *100*(3), 363–40644.

Evans, A. a., & Donahue, R. E. (2008). Laser scanning confocal microscopy: a potential technique for the study of lithic microwear. *Journal of Archaeological Science*, *35*(8), 2223–2230. https://doi.org/10.1016/j.jas.2008.02.006

Evans, A. A., & Macdonald, D. (2011). Using metrology in early prehistoric stone tool research: further work and a brief instrument comparison. *Scanning*, *33*(5), 294–303. https://doi.org/10.1002/sca.20272

Evans, A. A., Macdonald, D. A., Giusca, C. L., & Leach, R. K. (2014). New Method Development in Prehistoric Stone Tool Research: Evaluating Use Duration and Data Analysis Protocols. *Micron*, *65*(October), 69–75.

Evans, A. A., Maxwell, M. L., & Cruickshanks, G. L. (2013). From lidar to LSCM : micro-topographies of archaeological finds. In R. S. Opitz & D. C. Cowley (Eds.), *Interpreting Archaeological Topography: 3D Data, Visualization and Observation* (pp. 124–135). Oxford: Oxbow.

Magne, M. P. R., & Fedje, D. W. (2007). The Spread of Microblade Technology in Northwestern North America. In Y. V. Kuzmin, S. G. Keates, & C. Shen (Eds.), *Origin and Spread of Microblade Technology in Northern Asia and North America* (pp. 171–188). Burnaby, BC: SFU Archaeology Press.

Flenniken, J. J. (1981). *Replicative Systems Analysis: A Model Applied to the Vein Quartz Artifacts from the Hoko River Site*. Pullman, WA: Washington State University Laboratory of Archaeology.

Flenniken, J. J. (1987). The Paleolithic Dyuktai Pressure Blade Technique of Siberia. *Arctic Anthropology*, *24*(2), 117–132.

Franklin, U. M. (2004). *The real world of technology* (Rev. ed.). Toronto, Ont. : Berkeley, CA: House of Anansi Press ; Distributed in the United States by Publishers Group West.

Gajski, D., Solter, A., & Gašparovic, M. (2016). Applications of Macro Photogrammetry in Archaeology. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLI-B5*, 263–266. https://doi.org/10.5194/isprsarchives-XLI-B5-263-2016

Giria, E. Y., & Pitul'ko, V. V. (1994). A High Arctic Mesolithic Industry on Zhokov Island: Inset Tools and Knapping Technology. *Arctic Anthropology*, *31*(2), 31–44.

Giusca, C. L., Evans, A. a., Macdonald, D., & Leach, R. K. (2012). *The Effect of Use Duration on Surface Roughness Measurements of Stone Tools NPL Report ENG 35*. Teddington, Middlesex, England.

Grace, R., Graham, I. D. G., & Newcomer, M. H. (1985). The quantification of microwear polishes. *World Archaeology*, *17*(1), 112–120.

Hare, P. G., Greer, S., Gotthardt, R., Farnell, R., Bowyer, V., Schweger, C., & Strand, D. (2004). Ethnographic and Archaeological Investigations of Alpine Ice Patches in Southwest Yukon, Canada. *Arctic*, *57*(3), 260–273.

Harrington, B., & et al. (2015). *Inkscape v.0.98*. inkscape.org

Harvey, P. (1997). Technology as Skilled Practice: approaches from Anthropology, History and Psychology. *Social Analysis: The International Journal of Anthropology*, *41*(1), 3–14.

Helwig, K., Monahan, V., & Poulin, J. (2008). The Identification of Hafting Adhesive on a Slotted Antler Point from a Southwest Yukon Ice Patch. *American Antiquity*, *73*(2), 279–288.

Hiscock, P., & Clarkson, C. (2005a). Measuring Artefact Reduction: An evaluation of Kuhn's Geometric Index of Unifacial Reduction. In C. J. Clarkson & L. Lamb (Eds.), *Lithics "down*

under": Australian perspectives on lithic reduction, use and classification. (p. 15). Oxford: Archaeopress.

Hiscock, P., & Clarkson, C. (2005b). Experimental evaluation of Kuhn's geometric index of reduction and the flat-flake problem. *Journal of Archaeological Science*, *32*(7), 1015–1022. https://doi.org/10.1016/j.jas.2005.02.002

Hiscock, P., & Clarkson, C. (2008). The Construction of Morphological Diversity: A Study of Mousterian Implement Retouching at Combe Grenal. In WilliamEditor Andrefsky Jr (Ed.), *Lithic Technology: Measures of Production, Use and Curation* (pp. 106–135). Cambridge University Press. https://doi.org/10.1017/CBO9780511499661.006

Hiscock, P., & Tabrett, A. (2010). Generalization, inference and the quantification of lithic reduction. *World Archaeology*, *42*(4), 545–561. https://doi.org/10.1080/00438243.2010.517669

Ingold, T. (2000). Making Culture and Weaving the World. In P. M. Graves-Brown (Ed.), *Matter, Materiality and Modern Culture* (pp. 50–71). London: Routledge.

Ingold, T. (2004). Culture on the Ground: The World Perceived Through the Feet. *Journal of Material Culture*, *9*(3), 315–340. https://doi.org/10.1177/1359183504046896

Iovita, R., Pfleging, J., & Buchli, J. (2017). Evaluating the Effect of Force and Duration on Lithic Use-Wear Using a Force—and Impedance— Controlled Robot. In *82nd Annual Meeting of the Society for American Archaeology. March 29th-April 2nd, 2017.*

Jurasinski, G., Koebsch, F., Guenther, A., & Beetz, S. (2014). *flux: Flux rate calculation from dynamic closed chamber measurements. R package version 0.3-0.* https://cran.r-project.org/package=flux

Kay, M., & Mainfort, R. C. (2014). Functional analysis of prismatic blades and bladelets from Pinson Mounds, Tennessee. *Journal of Archaeological Science, 50,* 63–83. https://doi.org/10.1016/j.jas.2014.06.019

Keeley, L. H. (1974). Technique and Methodology in Microwear Studies: A Critical Review. *World Archaeology*, *5*(3), 323–336.

Keeley, L. H. (1980). *Experimental determination of stone tool uses: a microwear analysis*. Chicago: University of Chicago Press.

Keller, C. M. (1966). The Development of Edge Damage Patterns on Stone Tools. *Man*, *1*(4), 501. https://doi.org/10.2307/2798343

Key, A. J. M. (2013). Applied Force as a Determining Factor in Lithic Use-Wear Accrual: An Experimental Investigation of its Validity as a Method with which to Infer Hominin Upper Limb Biomechanics. *Lithic Technology*, *38*(1), 32–45. https://doi.org/10.1179/0197726113Z.0000000001

Key, A. J. M., & Lycett, S. J. (2011). Technology based evolution? A biometric test of the effects of handsize versus tool form on efficiency in an experimental cutting task. *Journal of Archaeological Science, 38*(7), 1663–1670. https://doi.org/10.1016/j.jas.2011.02.032

Key, A. J. M., & Lycett, S. J. (2014). Are bigger flakes always better? An experimental assessment of flake size variation on cutting efficiency and loading. *Journal of Archaeological Science, 41*, 140–146. https://doi.org/10.1016/j.jas.2013.07.033

Key, A. J. M., Stemp, W. J., Morozov, M., Proffitt, T., & de la Torre, I. (2014). Is Loading a Significantly Influential Factor in the Development of Lithic Microwear? An Experimental Test Using LSCM on Basalt from Olduvai Gorge. *Journal of Archaeological Method and Theory*, *22*(4), 1193–1214. https://doi.org/10.1007/s10816-014-9224-9

Kuhn, S. L. (1990). A geometric index of reduction for unifacial stone tools. *Journal of Archaeological Science*, *17*(5), 583–593. https://doi.org/10.1016/0305-4403(90)90038-7

Kuzmin, Yaroslav V. (2007). Geoarchaeological Aspects of the Origin and Spread of Microblade Technology in Northern and Central Asia. In Kuzmin, Yaroslav V., S. G. Keates, & S. Chen (Eds.), *Origin and Spread of Microblade Technology in Northern Asia and North America*. Burnaby, BC: SFU Archaeology Press.

Lee, C. M. (2007). *Origin and Function of Early Holocene Microblade Technology in Southeast Alaska, USA* (Ph.D.). University of Colorado.

Lerner, H. J. (2007). Digital Image Analysis and Use-wear Accrual as a Function of Raw Material: An Example from Northwestern New. *Lithic Technology*, *32*(1), 51–67.

Letham, B., Martindale, A., Waber, N., & Ames, K. M. (2018). Archaeological Survey of Dynamic Coastal Landscapes and Paleoshorelines: Locating Early Holocene Sites in the Prince Rupert Harbour Area, British Columbia, Canada. *Journal of Field Archaeology*, *43*(3), 181–199. https://doi.org/10.1080/00934690.2018.1441575

LibreOffice Contributors. (2000). *LibreOffice*. The Document Foundation. libreoffice.org

Lin, S. C., Rezek, Z., & Dibble, H. L. (2017). Experimental Design and Experimental Inference in Stone Artifact Archaeology. *Journal of Archaeological Method and Theory*. https://doi.org/10.1007/s10816-017-9351-1

Macdonald, D. a. (2013). The application of focus variation microscopy for lithic use-wear quantification. *Journal of Archaeological Science*, 1–8. https://doi.org/10.1016/j.jas.2013.10.003

Magne, M. P. R. (1996). Comparative Analysis of Microblade Cores from Haida Gwaii. In R. L. Carlson & L. Dalla Bona (Eds.), *Early Human Occupation in British Columbia* (pp. 151–158). Vancouver: University of British Columbia Press.

Magne, M. P. R., & Fedje, D. W. (2007). The Spread of Microblade Technology in Northwestern North America. In Y. V. Kuzmin, S. G. Keates, & C. Shen (Eds.), *Origin and Spread of Microblade Technology in Northern Asia and North America* (pp. 171–188). Burnaby, BC: SFU Archaeology Press.

Marwick, B. (2015). Reproducible Research in Archaeology: Basic Principles and Common Tools. Presented at the The 80th Annual Meeting of the Society for American Archaeology, April 15-19, 2015, San Francisco.

Marwick, B., d'Alpoim Guedes, J., Barton, C. M., Bates, L. A., Baxter, M., Bevan, A., et al. (2017). Open science in archaeology. *SAA Archaeological Record*, *17*(4), 8–14.

Morales, J. I., Lorenzo, C., & Vergès, J. M. (2013). Measuring Retouch Intensity in Lithic Tools: A New Proposal Using 3D Scan Data. *Journal of Archaeological Method and Theory*, 1–16. https://doi.org/10.1007/s10816-013-9189-0

Nelson, M. C. (1991). The Study of Technological Organization. *Archaeological Method and Theory Vol.3*, *3*(1991), 57–100.

Newcomer, M., Grace, R., & Unger-Hamilton, R. (1986). Investigating microwear polishes with blind tests. *Journal of Archaeological Science*, *13*(3), 203–217. https://doi.org/10.1016/0305-4403(86)90059-2

Odell, G. H., & Odell-Vereecken, F. (1980). Verifying the reliability of lithic use-wear assessments by "blind tests": the low-power approach. *Journal of field Archaeology*, *7*, 87–120.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3400–3407). Presented at the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China: IEEE. https://doi.org/10.1109/ICRA.2011.5979561

Owen, L. R. (1987). Hafting Microblades : Examples from the Dorset Culture of the North American Arctic. *La Main et l'Outil. Manches et emmanchements préhistoriques. Table Ronde C.N.R.S. tenue à lyon du 26 au 29 novembre 1984, sous la direction de D. Stordeur. Lyon : Maison de l'Orient et de la Méditerranée Jean Pouilloux, 1987. (Travaux de la Maison de l'Or*, 147–150.

Owen, L. R. (1988). *Blade and Microblade Technology: Selected Assemblages from the North American Arctic and the Upper Paleolithic of Southwest Germany*. Oxford: B.A.R. International Series.

Pelegrin, J. (1990). Prehistoric Lithic Technology: Some Aspects of Research. *Archaeological Review from Cambridge*, *9*(1), 116–125.

Pétillon, J.-M., Bignon, O., Bodu, P., Cattelain, P., Debout, G., Langlais, M., et al. (2011). Hard Core and Cutting Edge: Experimental Manufacture and Use of Magdalenian Composite Projectile Tips. *Journal of Archaeological Science*, *38*(6), 1266–1283. https://doi.org/10.1016/j.jas.2011.01.002

Pfleging, J., Iovita, R., & Buchli, J. (2018). Influence of force and duration on stone tool wear: results from experiments with a force-controlled robot. *Archaeological and Anthropological Sciences*. https://doi.org/10.1007/s12520-018-0729-0

Pfleging, J., Stücheli, M., Iovita, R., & Buchli, J. (2015). Dynamic monitoring Reveals Motor Task Characteristics in Prehistoric Technical Gestures. *PLoS ONE*, *10*(8), 1–20. https://doi.org/10.1371/journal.pone.0134570

Pierrot-deseilligny, M., Deveau, M., Belvaux, J., Choqueux, G., G.Maillet, & Girod, L. (2015). MicMac, Apero, Pastis and Other Beverages in a Nutshell!

Pokotylo, D. L., & Mitchell, D. H. (1998). Prehistory of the Northern Plateau. In D. E. Walker (Ed.), *Handbook of North American Indians Vol.12: The Plateau* (Vol. 11, pp. 81–102). Washington, DC: Smithsonian Institution.

Prentiss, W. C., & Kuijt, I. (2004). The Evolution of Collector Systems on the Canadian Plateau. In W. C. Prentiss & I. Kuijt (Eds.), *Complex Hunter-Gatherers: Evolution and Organization of Prehistoric Communities on the Plateau of Northwest North America* (pp. 49–65). Salt Lake City: University of Utah Press.

QGIS Development Team. (2015). *QGIS Geographic Information System.* Open Source Geospatial Foundation Project. qgis.org

R Core Team. (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.r-project.org/

Rabinowicz, E. (1995). *Friction and Wear of Materials* (Second Edi.). New York: John Wiley & Sons, Inc.

Rots, V, Pirnay, L., Pirson, P., & Baudoux, O. (2006). Blind tests shed light on possibilities and limitations for identifying stone tool prehension and hafting. *Journal of Archaeological Science, 33*, 935–952. https://doi.org/10.1016/j.jas.2005.10.018

Rots, Veerle. (2003). Towards an understanding of hafting: the macro-microscopic evidence. *Antiquity*, *298*(77), 806–815.

Rots, Veerle. (2004). Prehensile Wear on Flint Tools. *Lithic Technology*, *29*(1), 7–32.

Rots, Veerle. (2010). *Prehension and Hafting Traces on Flint Tools: a Methodology*. Leuven: Leuven University Press.

Rousseau, M. K. (2000). Results of the Keatley Creek Archaeological Project Lithic Source Study. In B. Hayden (Ed.), *The Ancient Past of Keatley Creek Vol.1: Taphonomy* (pp. 165–184). Burnaby, BC: SFU Archaeology Press.

Rousseau, M. K. (2015). Primary Toolstone Sources and Pre-Contact Period Quarrying Behaviour in the Thompson River Drainage of South Central British Columbia. In T. L. Ozbun & R. L. Adams (Eds.), *Toolstone Geography of the Pacific Northwest* (pp. 29–48). Burnaby, BC: SFU Archaeology Press.

Rupnik, E., Daakir, M., & Pierrot Deseilligny, M. (2017). MicMac – a free , open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*, *2*(14), 1–9. https://doi.org/10.1186/s40965-017-0027-2

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, *9*(7), 676.

Schlanger, N. (1990). Techniques as Human Action - Two Perspectives. *1Archaeological Review from Cambridge*, *9*(1), 18–26.

Schoville, B. J. (2010). Frequency and distribution of edge damage on Middle Stone Age lithic points, Pinnacle Point 13B, South Africa. *Journal of Human Evolution*, *59*(3–4), 378–391. https://doi.org/10.1016/j.jhevol.2010.07.015

Schoville, B. J. (2014). Testing a taphonomic predictive model of edge damage formation with Middle Stone Age points from Pinnacle Point Cave 13B and Die Kelders Cave 1, South Africa. *Journal of Archaeological Science*, *48*, 84–95. https://doi.org/10.1016/j.jas.2013.10.002

Schoville, B. J. (2016, April 2). *Landscape variability in tool-use and edge damage formation in South African Middle Stone Age lithic assemblages* (Ph.D.). Arizona State University.

Schoville, B. J., & Brown, K. S. (2010). Comparing Lithic Assemblage Edge Damage Distributions: Examples from the Late Pleistocene and Preliminary Experimental Results. *Vis-a-vis: Explorations in Anthropology*, *10*(2), 34–49.

Schoville, B. J., Brown, K. S., Harris, J. A., & Wilkins, J. (2016). New Experiments and a Model-Driven Approach for Interpreting Middle Stone Age Lithic Point Function Using the Edge Damage Distribution Method. *PLoS ONE*, 1–32. https://doi.org/10.1371/journal.pone.0164088

Sellet, F. (1993). Chaine Operatoire; the Concept and its Applications. *Lithic Technology*, *18*(1), 106–112.

Semenov, S. A. (1957). *Prehistoric Technology: an Experimental Study of the oldest Tools and Artefacts from traces of Manufacture and Wear*. Bath: Adams and Dart.

Stachowiak, G. W., & Batchelor, A. W. (2014). *Engineering Tribology (4th Edition)*. Boston, MA: Butterworth-Heinemann.

Stemp, W. J. (2013). A review of quantification of lithic use-wear using laser profilometry: a method based on metrology and fractal analysis. *Journal of Archaeological Science*, *48*, 15–25. https://doi.org/10.1016/j.jas.2013.04.027

Stemp, W. J., Morozov, M., & Key, A. J. M. (2015). Quantifying lithic microwear with load variation on experimental basalt flakes using LSCM and area-scale fractal complexity (Asfc). *Surface Topography: Metrology and Properties*, *3*, 1–20.

Stemp, W. J., & Stemp, M. (2001). UBM Laser Profilometry and Lithic Use-Wear Analysis: A Variable Length Scale Investigation of Surface Topography. *Journal of Archaeological Science*, *28*(1), 81–88. https://doi.org/10.1006/jasc.2000.0547

Stevens, N. E., Harro, D. R., & Hicklin, A. (2010). Practical Quantitative Lithic Use-wear Analysis Using Multiple Classifiers. *Journal of Archaeological Science, 37*(10), 2671–2678. https://doi.org/10.1016/j.jas.2010.06.004

The GIMP team. (2013). *GIMP*. www.gimp.org

Tian, Y., Meng, X., Tao, D., Liu, D., & Feng, C. (2015). Upper limb motion tracking with the integration of IMU and Kinect. *Neurocomputing, 159*, 207–218. https://doi.org/10.1016/j.neucom.2015.01.071

Tostevin, G. B. (2011). Levels of Theory and Social Practice in Reduction Sequence and Chaîne Opératoire Methods of Lithic Analysis. *PaleoAnthropology*, 351–375. https://doi.org/10.4207/PA.2011.ART64

Tringham, R., Cooper, G., Odell, G., Voytek, B., & Whitman, A. (1974). Experimentation in the Formation of Edge Damage: A New Approach to Lithic Analysis. *Journal of Field Archaeology*, *1*(1–2), 171–196. https://doi.org/10.1179/jfa.1974.1.1-2.171

van Gijn, A. L. (2014). Science and interpretation in microwear studies. *Journal of Archaeological Science*, *48*, 166–169. https://doi.org/10.1016/j.jas.2013.10.024

Vardi, J., Golan, A., Levy, D., & Gilead, I. (2010). Tracing sickle blade levels of wear and discard patterns: a new sickle gloss quantification method. *Journal of Archaeological Science, 37*(7), 1716–1724. https://doi.org/10.1016/j.jas.2010.01.031

Vergès, J. M., & Ollé, A. (2011). Technical microwear and residues in identifying bipolar knapping on an anvil: experimental data. *Journal of Archaeological Science*, *38*(5), 1016–1025. https://doi.org/10.1016/j.jas.2010.11.016

Waber, N. H. (2011). *The Development of the Microblade Industry at the Richardson Island Site, Haida Gwaii, British Columbia* (Unpublished M.A. thesis). Dept. and Anthropology, University of Victoria.

Waber, N. H. (2015). 3D Photogrammetry and GIS for Tracking Edge Wear Accumulation in Lithic Experiments. Presented at the 80th Annual Meeting of the Society for American Archaeology, San Francisco.

Waber, N. H. (in press). An Open-source GIS Toolkit for Lithic Wear Quantification. *Archaeological and Anthropological Science.*

Weiss, A. D. (2001). *Topographic Position and Landforms Analysis*. Poster presented at the ESRI user conference, San Diego.

Whittaker, J. C. (1994). *Flintknapping: Making and Understanding Stone Tools*. Austin: University of Texas Press.

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer-Verlag.

Wilkins, J., & Chazan, M. (2012). Blade production ~500 thousand years ago at Kathu Pan 1, South Africa: support for a multiple origins hypothesis for early Middle Pleistocene blade technologies. *Journal of Archaeological Science*, *39*(6), 1883–1900. https://doi.org/10.1016/j.jas.2012.01.031

Wilkins, J., Schoville, B. J., & Brown, K. S. (2014). An Experimental Investigation of the Functional Hypothesis and Evolutionary Advantage of Stone-Tipped Spears. *PLoS ONE*, *9*(8), e104514–e104514. https://doi.org/10.1371/journal.pone.0104514

Zotero Community. (2013). *Zotero*. Corporation for Digital Scholarship and the Roy Rosenzweig Center for History and New Media at George Mason University. zotero.org

Appendix A: OpenHaft 1.0 Hardware Description and Wiring Schematic

OpenHaft is based on inexpensive, easily obtained hardware and free, open-source software. At the heart of the unit is a Teensy 3.2 single-board microcontroller with 32bit ARM processor and microUSB interface. The Teensy runs code written for Arduino (a popular open-source microcontroller) through Teensyduino, a freely available add-on for the Arduino IDE. The Teensy is connected to a load cell through an HX711 load cell amplifier. The load cell is an aluminium bar (55.3mm x 12.6mm x 12.6mm) with resistors bonded across the mid-point on two long sides of it in a formation known as a Wheatstone bridge circuit. These resistors act as a strain gauge; as the bar experiences shearing forces and flexes very slightly, the resistors on either side deform and the voltage travelling through the resistors changes. The change in voltage is amplified by the HX711, and read in analog format on the Teensy. The Teensy registers the incoming voltage value and converts it to kg through a calibration algorithm for the load cell amp. The unit must be calibrated using a known-mass reference weight prior to use in order to find the correct setting (the calibration sketch is included in the HX711.h library). The load cell precision is 10g (±0.05% of its maximum load of 20kg), which is roughly equivalent to a 10 bit depth (10420 steps), which is well within the Teensy's 16bit analog potential. The sample rate is set using a dial (a linear potentiometer) which can select an interval from 1Hz to 10Hz. Slightly faster sample rates may be achievable (~12Hz), but it is easier to detect errors, sort data, and calculate stroke duration with an even 10 samples per second. The communication rate with the HX711 amplifier means that substantially faster sample intervals are not currently feasible, and higher settings (i.e., 100Hz) result in uneven rates of data recording. 10Hz seems to provide an adequate volume of data for gesture-load analysis, reliably identifying variation in load throughout each stroke. As the Teensy reads and parses the load data, it transmits the output through a Bluetooth module that is readable (and recordable) from any device capable of reading a Bluetooth serial input (most laptop computers, smart phones, tablets, etc.). The data was recorded on a laptop running Ubuntu 17.10, using PuTTY serial port interface software. An additional program (KST 2.0) was also used to visualize the incoming data in real time (see Appendix C for the detailed data acquisition workflow). The haft unit is powered by a rechargeable 3.7V 1s lithium polymer battery, but may also be run using power from a USB connection.

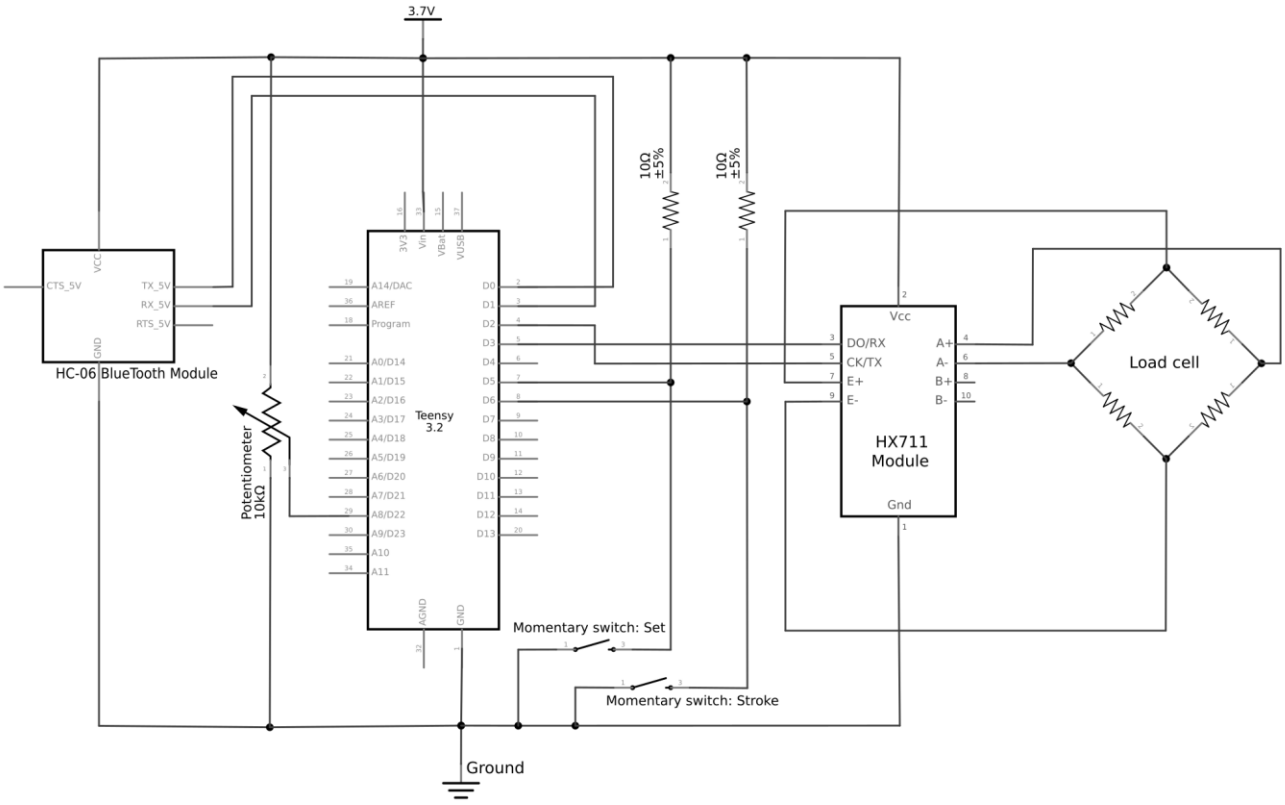

Figure A.1: Wiring schematic for OpenHaft

Appendix B: OpenHaft 1.0 Sketch

The Teensy/Arduino code (termed a "sketch") is written to record five values: sample interval (in milliseconds), experiment timestamp (in milliseconds), experiment set number, stroke number, and load (in kg). The OpenHaft sketch is also available for download from: https://github.com/nwaber/OpenHaft/blob/master/OpenHaft-1.0_2018-08-23/OpenHaft-1.0_2018-08-23.ino

The code requires the HX711.h Arduino library, available from https://github.com/bogde/HX711 or https://github.com/nwaber/HX711.h-old-version . Previous HX711 library updates occasionally did not work with alpha versions of OpenHaft, so the older version may be needed.

```
/*
************ OpenHaft 1.0 ***************
Pressure recording knife haft for microblade experimentation
2018/08/23
Nick Waber nick.waber@telus.net
https://github.com/nwaber/OpenHaft

This software is released under the MIT License
Copyright (c) 2018 Nicholas Waber

Permission is hereby granted, free of charge, to any person
obtaining a copy
of this software and associated documentation files (the
"Software"), to deal
in the Software without restriction, including without limitation
the rights
to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell
copies of the Software, and to permit persons to whom the
Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
```

```
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
SOFTWARE.

Load recording based on the Sparkfun HX711 Example sketch by
Nathan Seidle

Goal: To build a haft that outputs recordable data
regarding the force exerted on a hafted lithic edge during use.

Hardware:
Teensy 3.2 Microcontroller
HC-06 Bluetooth Module
10K Potentiometer
HX711 load cell amplifier
20kg load cell (wheatstone bridge)
3.7V 250mAh 1S LiPo battery, harness

Hardware setup:
1) HX711 connects to Teensy D2 (CLK), D3 (DAT), 3.3V (VCC), GND
2) HX711 connects to load cell:
  a) HX711 E+ (red): red wire
  b) HX711 E- (black): black wire
  c) HX711 A+ (white): white wire
  d) HX711 A- (green): green wire
3) Potentiometer connects to A8
4) HC-06:
  a) TX to d0
  b) RX to d1
  c) VCC to 3.3V
  d) GND to GND
5) Pushbutton #1 connects to Teensy 7
6) Pushbutton #2 connects to Teensy 8


Software:
Coolterm is used to collect/export serial data on Windows.
Putty is used to collect/export serial data on Ubuntu 16.04.
*/

  #include "HX711.h"
//If errors are encountered, try using the older HX711.h library,
available at https://github.com/nwaber/HX711.h-old-version

  #define calibration_factor -121230.00 //This value is obtained
using the SparkFun_HX711_Calibration sketch

  #define DOUT  3
```

```
  #define CLK  2
  const int potPin = 21;  //potentiometer input for interval
adjustment on Analog port 7 (digital 21)
  const int buttonPin = 7;  //momentary pushbutton for stroke
count advancement on digital port 7.
  const int buttonPin2 = 8;  //momentary pushbutton for test
count advancement on digital port 8.


  HX711 scale(DOUT, CLK);

  int test_num; //experiment set number
  int stroke_num;  //individual stroke number

void setup() {

  analogReadResolution(11); // set analog bit depth resolution
for potentiometer.  Teensy max resolution is 16 bit, 13 "usable"


/*
Resolution reference
8 bit = 256
10 bit = 1024
11 bit = 2048
12 bit = 4096
13 bit = 8192
16 bit = 65536
*/
  pinMode(buttonPin, INPUT);  //reads input from button
  pinMode(buttonPin2, INPUT); //reads input from button 2


  Serial1.begin(9600);      // open the hardware serial port #1
(pins d0,d1) at default 9600 bps
  //replace all references to Serial1 with Serial to use USB
serial data communication

    delay(20000);       // Lets the serial monitor connect to the
bluetooth before printing data



  scale.set_scale(calibration_factor); //This value is obtained
by using the SparkFun_HX711_Calibration sketch
    //
https://codebender.cc/sketch:123175#SparkFun_HX711_Calibration.in
o
  scale.tare();  //Assuming there is no weight on the scale at
start up, reset the scale to 0

//set up header line for spreadsheet:
```

```
  Serial1.print("interval");  //how quickly the samples are
taken, i.e., every 5 milliseconds
  Serial1.print(",");
  Serial1.print("time");    //how much time has passed in the
experiment
  Serial1.print(",");
  Serial1.print("set");
  Serial1.print(",");
  Serial1.print("stroke");
  Serial1.print(",");
  Serial1.print("kg");  //force exerted on the tool edge in kg.
Multiply by 981 for N.
  Serial1.println("");

delay(5000);
}

void loop() {
  int potInterval;    //sample interval value mapped from
intervalStep
  int intervalStep;   //interval read from potentiometer
  intervalStep = analogRead(potPin); //read potentiometer to set
speed
  potInterval = map(intervalStep, 1, 2047, 1000, 100); //low = 1
second interval, high = 100ms
   //set to correspond with analogRead bit depth!!!
  int buttonState;
  buttonState = digitalRead(buttonPin);
  int button2State;
  button2State = digitalRead(buttonPin2);

 if ((buttonState == HIGH) && (button2State == HIGH))
  {
delay(1000); //if both buttons are pressed, then delay loop.
  }
   else
   {


  Serial1.print(potInterval);  //default: potInterval
  Serial1.print(",");
  Serial1.print(millis());  //Experiment time stamp
  Serial1.print(",");
  Serial1.print(test_num);
  Serial1.print(",");
  Serial1.print(stroke_num);
  Serial1.print(",");
  Serial1.print(scale.get_units(), 2); //scale.get_units()
returns a float with two decimal place precision
  //this measures to the nearest 10 grams
  //sensor accuracy is +/-0.05% of max load (20kg)
```

```
  Serial1.println("");

  if ((buttonState == HIGH) && (button2State == LOW))
  {
    stroke_num += 1;  // push the button once to advance the
stroke number by one
    delay(100);   //wait 100 ms before continuing the loop
                  //prevents double-advancement if the button is
not released quickly enough
  }
    else{
    stroke_num +=0;  //if the button is not pushed, the stroke
number stays the same
  }

  if ((buttonState == LOW) && (button2State == HIGH))
  {
    test_num += 1;  // push the button once to advance the test
number by one
    delay(100);   //wait 100 ms before continuing the loop
                  //prevents double-advancement if the button is
not released quickly enough
  }
    else{
    test_num +=0;  //if the button is not pushed, the test number
stays the same
  }

  if(Serial1.available())
  {
    char temp = Serial1.read();
    if(temp == '+' || temp == 'a')   //send "+" or "a" to advance
the experimental set number one
      test_num += 1;
    else if(temp == '-' || temp == 'z')  //send "-" or "z" to
decrease the set number one
      test_num -= 1;
  }
  delay(potInterval);  //use potentiometer to adjust delay
interval
   }
}
```

Appendix C: Data Acquisition Workflow.

1. Set up Bluetooth connectivity between OpenHaft and PC.
    1. Windows/OSX:
        1. Make sure your PC has Bluetooth turned on.
        2. Turn on OpenHaft.
        3. In your PC's Bluetooth settings, search for new connections. The OpenHaft will likely be detectable as "HC-06". Connect to that.
        4. Note: A PIN code may be required. By default the HC-06 PIN is 1234.
        5. The HC-06 device should now be identified as "Connected" on your PC, and the blinking LED on the HC-06 should be steadily illuminated.
        6. Test the connection in the Arduino IDE.
            1. Open the Arduino IDE.
            2. Set the board to correspond to the model Teensy you are using.
            3. Set the port to match your Bluetooth serial port (this may take some trial and error).
            4. Open the serial monitor. The OpenHaft should automatically have started transmitting data, which will appear in the serial monitor.
    2. Ubuntu 18.04 (if the above does not work)
        1. Open Terminal
        2. Use hcitool scan to find the HC-06's MAC address.
            1. In Terminal, enter: `hcitool scan`

               The resulting printout will identify the HC-06 device.
        3. In etc/bluetooth/rfcomm.conf create an entry like this:

```
rfcomm0 {
        bind no;
        device 20:15:02:04:12:81;
        channel 1;
        comment "Serial Port";
```

```
}
```

Note: next to “device” enter your MAC address.

4. In Terminal, enter the following: `sudo rfcomm connect 0`
   The light on the HC-06 should glow solid.

2. Activate OpenHaft.
   1. Connect battery leads to activate OpenHaft.
   2. Connect to PC via Bluetooth.

   After the initial pairing, subsequent connection should be automatic. In Ubuntu, enter `sudo rfcomm connect /dev/rfcomm0 20:15:02:04:12:81` (with your MAC address) in Terminal to connect again.

3. Run serial interface software.
   A variety of SSH serial interface applications are available for most operating systems. These applications provide a link between a device such as OpenHaft and the PC. They read, display, and record the data transmitted from OpenHaft. The instructions here provide a short guide for using serial interface applications to communicate with OpenHaft on Windows and Ubuntu. The OSX process should be similar.

   1. Windows:
      1. Install Coolterm. The software is available from: http://freeware.the-meiers.org/
      2. Run Coolterm.
         1. Set up the connection
            1. Click the “Options” icon.
               1. Set the port to whatever port your Teensy communicates to the computer through. You might need to have the teensy connected and turned on.
               2. Buadrate: 9600
               3. Data bits: 8

4. Parity: none
5. Stop bits: 1

2. Click OK. The settings will stay that way on reboot until you change them.

2. Set the data log file
    1. Click on the “Connection” menu (not the “Connect” button)
    2. Select “Capture to text file”.
    3. Select “Start”
    4. Enter a file name and save location.
    5. Click “Save”

    NOTE: CoolTerm is actually recording data as of now. You may wish to click “pause recording” in the Capture to text file menu if you are not going to immediately commence with the experiment..

3. Open the connection
    1. Turn on the OpenHaft.
    2. In CoolTerm, click “Connect”.
    3. Data transmission should commence within approximately 20 seconds.

2. Ubuntu:
    1. Install PuTTY: sudo apt-get install putty
    2. Turn on OpenHaft.
    3. Run PuTTY from Terminal: sudo putty

    Note: sudo is required for Putty to have access to ports.

    Note: Under Ubuntu 17.x and later, you must run Ubuntu on Xorg.

    To do this, at the Ubuntu sign-on screen, click the gear next to the password box, select “Ubuntu on Xorg”, and log in.

1. In PuTTY SSH client:
    1. Under the “Session” tab:
        1. Select Connection type: Serial
        2. Enter Serial Line to match the rfcomm number from rfcomm.conf (i.e., `/dev/rfcomm3`)
        3. Set Speed to match the baudrate set in the OpenHaft sketch (default: 9600).
    2. In the “Logging” tab:
        1. Set “Session logging” to “All session output”.
        2. Set “Log file name” to the desired save location and file name (.txt format).
    3. (Optional): Under the “Session” tab, save the session. Henceforth you may load the session settings very quickly (including log file settings), which is very convenient if you have multiple OpenHafts, and thus multiple HC-06 modules with individual MAC addresses.
    4. Click “Open”. A Terminal window should appear to display the serial data stream.

4. Run data visualization software (optional).
   KST 2.0 was used to display a live plot of streaming OpenHaft data. While this does not affect data collection, it makes it easier to track strokes, and is quite interesting.
    1. Download and install KST 2.0. KST2 is available for download from: https://sourceforge.net/projects/kst/files/Kst%202.0.8/
    2. Activate OpenHaft and connect using PuTTY or CoolTerm.
    3. Open KST 2.0.
    4. Using the “Data Wizard”:
        1. Navigate to and select the PuTTY/CoolTerm log file.
        2. Next to “ASCII file”, click “Configure...”
            1. Adjust the settings for data lines and delimeter to match your log file output format, and click “OK”.

3. Set “Update type” to “Time Interval”. Click “Next”.
4. Select the “stroke” and “kg” fields, use the arrow icons to move them to the “Selected data” pane, and click “Next”.
5. Under “Create XY plots” set “Create from field” to the “time” field, or leave at “Index”. Click “Next”.
6. Leave the curve placement and plot placement settings at their defaults, and click “Finish”. A pair of time series plots will appear.

5. Right-click the “stroke” plot and select “delete”.
6. Right-click the empty space, select “Clean up layout”, and select “Automatic”. The load plot will re-size to fill the empty space.
7. Create a live label for the stroke number.
    1. Select “Create” , “Annotation”, “Label”. A “Create Label Dialog” window will appear.
    2. (Recommended) set “Font size” to 40.
    3. In the “Label:” field, enter: [stroke:Last (X15)] and click “OK”.
       Note: The “X15” digit may vary depending on your data setup. As soon as you type the square-open-bracket, a pulldown menu will appear; use it to select “stroke:Last”.
    4. Click in the top left of the plot (or wherever you prefer) to place the stroke count display.
       Each time the “stroke” advancement button on the OpenHaft is clicked the displayed number will advance one.

5. Perform the experiment.
    1. Use the OpenHaft-mounted tool to perform the task being tested.
       Tip: Make sure you do not press on the back of the load cell or lithic implement, as this will affect the integrity of the load data. A haft designed to shield the load cell from inadvertent contact with the user’s hand is optimal.
    2. If the task involves discrete strokes, click the stroke button between strokes.

NOTE: To prevent accidentally advancing the stroke value with a long click, the OpenHaft 1.0 sketch is set to pause for 0.1 seconds after the button click. If your strokes are in quick succession, the delay may affect the data. In this case, either do not use the stroke button, or adjust the post-click delay (line 161) to be shorter.

3. After the task is complete, the stroke count is reached, or the tool is exhausted, power down the OpenHaft. Quit Putty (log will be saved automatically) or CoolTerm (make sure you have saved the log).

   Tip: Between sets or when adjusting the haft, use the potentiometer to lower the sample rate to 1Hz. The "interval" field in the data may then be used to find an delete cases that were recorded at an interval setting slower than the maximum sample rate.

   Note: If the automatic logging function is set in CoolTerm or PuTTY, be sure to re-name your CoolTerm/PuTTY output log in order to avoid overwriting the file the next time you run an experiment.

Appendix D: Notes on the Study of Wear

The engineering field of tribology (the study of wear in mechanical systems) uses variations of a formula for a coefficient of wear (*K*) in order to quantify the wear relationship between two materials: $K = \frac{V}{m*N}$ (Rabinowicz 1995; Stachowiak and Batchelor 2014). In this formula, "*V*" represents volume of material worn off of the object under observation, "*N*" is load on the system, and "*m*" is distance travelled during the test. Each value is ostensibly dimensionless, though it is standard practice to express volume in cubic millimetres, load in Newtons, and distance in metres (Stachowiak and Batchelor 2014). The CSD and OpenHaft provide the distance and load values, and the graphite measurements provide volume loss, facilitating calculation of the coefficient of wear in archaeological lithic experiments: a single number that describes the relationship between lithic material and contact material. Material hardness is sometimes also included in the calculation of the coefficient of wear, changing the formula slightly to: $K = \frac{V*p}{m*N}$ (Rabinowicz 1995). Material hardness has generally been measured in Vickers pyramid hardness (HV) based on indentation values relative to force over area for a specified amount of time. The HB and 2B leads used in the first case study experiment are approximately 12HV and 14HV, respectively.

Multiple forms of wear are identified in the field of tribology. The two with the greatest relevance to lithic wear analysis are adhesive wear and abrasive wear. Adhesive wear occurs when asperities on the surface of a hard material come into contact with a softer material, and the strength of the bond between the harder and softer material overcomes the tensile strength of the asperity, at which point the asperity breaks off of the parent material and is transported away by the contact medium (Rabinowicz 1995; Stachowiak and Batchelor 2014). This is one process by which polish occurs on lithic tool edges: the microtopographical peaks and ridges on the lithic surface break before they can gouge through the softer contact material. Of course, the soft material is also being scraped off by the harder material, filling cavities and creating a smooth aggregate surface. Abrasive wear occurs when a harder material removes mass from a softer material. This generally results in gouges or striae on the material surface as asperities on the harder material bite and cut the softer substance (Rabinowicz 1995). In a lithic wear context,

since most stone tools were applied to working softer material, striae observed on the tool's surface are often due to three-body wear systems, where hard particles suspended between the two primary surface scrape material from both during contact. The microscopic asperities broken off a lithic surface by adhesive wear may become the abrasive third body in such a system, gouging striae into the surface of the tool from which they just detached. Rabinowitz (1995) considers the three-body wear system in such a context to be a component of adhesive wear, rather than a class of abrasive wear. In any case, the formula for the coefficient of abrasive wear (K_{abr}) and the coefficient of adhesive wear (K_{adh}) are identical, so the coefficient of wear may theoretically be applied fairly broadly to lithic wear systems.

In the case of the graphite wear experiment described in Chapter 2, the mechanism of wear is best considered adhesive wear. Strictly speaking, the graphite is a harder substance than the paper. However, the tensile strength of the graphite (or more specifically, the clay matrix suspending the graphite particles) is quite low so when the graphite is rubbed across the paper's surface the friction between paper and graphite crumbles the latter, and the particles are deposited in the microtopographic "valleys". In contrast to most three-body wear mechanisms, the graphite is so fine that, by filling the recesses in the paper texture, it acts as a lubricant, flattening the surface and reducing friction (and thus wear) rather than introducing additional hard and loose particles that gouge the material and increase wear.

One additional aspect of wear that is somewhat confounding is fatigue wear, which occurs when material fractures due to "cyclic mechanical stresses" (Stachowiak and Batchelor 2014). This is the type of wear that occurs when the edge of a tool flakes or crushes away due to load rather than friction. In the case of lithic artifacts, specifically, this flaking often manifests in conchoidal (micro)fractures, similar to those that were used to form the tool in the original flintknapping process. This fracture pattern may result in more material loss than from abrasive or adhesive wear. The edge damage observed in the microblade experiment is most closely aligned with fatigue wear. In lithic use-wear analysis studies, it is important to recognize that fatigue wear may result in the destruction or obfuscation of the evidence of abrasive or adhesive wear on lithic artifact surfaces, which may "refresh" as flakes are removed at the edge.

Appendix E: Preliminary OpenHaft Data Processing Workflow in R

There are two primary data processing tasks that must be applied to OpenHaft load data: finding peak values and calculating the load integral.. This R Markdown script provides a simple introduction for new users to be able to derive those products from an example data set. Simply copy-paste it into a an empty markdown file, or download the markdown file from: https://github.com/nwaber/OpenHaft/blob/master/OpenHaft.Rmd . The example data (microblade #12, set #2) is available from: https://github.com/nwaber/OpenHaft/blob/master/oh.data.csv . Very basic familiarity with R, and the use of RStudio is assumed.

````
---
title: "OpenHaft demo"
author: "Nick Waber"
date: "November 30, 2018"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r, echo=FALSE}
#Load the required libraries
library(ggplot2) #only needed for plotting data
library(pracma)  #for finding peaks
library(flux)    #for calculating area under curve
library(RCurl)  #to load demo data from Github
library(readr)

#load the example data
oh_url <-
getURL("https://raw.githubusercontent.com/nwaber/OpenHaft/master/oh.da
ta.csv")
oh.data <- read.csv(text=oh_url)

#Create an index column for the data
oh.data$idx <- as.numeric(row.names(oh.data))

#Plot the data (optional, bust useful to see if there is noise from
handling the OpenHaft or adjusting the tool setting that needs to be
trimmed)
````
````

```
ggplot(data=oh.data, aes(x=time,y=kg)) +
  geom_line() +
  geom_point(color="red",size=0.75) +
  labs(x = "Time (ms)", y = "Load (kg)", title = "Load values",
caption = "Sampling events indicated by red points.")

#Identify the peaks in oh.data and rename the columns
oh.pks <- findpeaks(oh.data$kg, nups=3, minpeakheight = 2,
minpeakdistance = 4, npeaks = 50) #Experiment with the settings to
find what works for the data.
oh.pks <- as.data.frame(oh.pks)  #reformat the findpeaks() output to a
data frame
colnames(oh.pks) <- c("pk.y","pk.x","pk.start","pk.end")  #rename the
columns

#Plot the peak data on the load data to confirm that the correct peaks
are identified.
oh.pks <- merge(oh.pks, oh.data, by.x = 2, by.y = 6)  #join the load
data columns to the peak data, matched by the index number.  This
assigns the time stamp to the peak data.

ggplot(data=oh.data, aes(x=time,y=kg)) +
  geom_line() +
  geom_point(data=oh.pks, aes(x=time, y=pk.y), color="red") +
  geom_text(data = oh.pks, aes(x = time, y = kg+0.2 , label = stroke),
size=2.5) +
  labs(x = "Time (ms)", y = "Load (kg)", title = "Load values with
peaks marked")

#(optional) calculate stroke duration for each peak
oh.pks$dur <- (oh.pks$pk.end - oh.pks$pk.start) * oh.pks$interval

#Calculate the integral of the load for the OpenHaft data
oh.load.int <- auc(oh.data$time, oh.data$kg, thresh = 0)
```

Further analysis may now take place in R, or the oh.pks data set may be exported as a CSV for analysis or manipulation other software.

Appendix F: MicMac Procedure

See https://micmac.ensg.eu/index.php/Accueil for further details on how to use MicMac.

1. Photograph both sides of the lithic artifact from multiple camera positions.
    - Tips:
        - The content of each image must overlap the neighbouring images by 65-85%.
        - There should not be an angular difference greater than 10° between neighbouring images.
        - It is absolutely vital to only include high quality, in-focus images.
        - Dark, shiny, transluscent, and/or monotone surfaces should be treated prior to photography. Dilute white ink and fine chalk (i.e., "liquid chalk") are ideal solutions.

2. Install MicMac: https://micmac.ensg.eu/index.php/Install

3. The script lines below are used to match the photos and calculate camera and surface positions.
    1. The "cd..." line directs the script to the relevant folder. Change the line to match the folder in which you have the images for that particular lithic surface.
    2. The number at the end of the Tapioca line refers to the desired image downsample resolution. Higher resolution results in higher quality point clouds (and higher point density), but takes a long time and requires a great deal more harddrive space. Lower resolution is faster but lower quality.
    3. MicMac requires the command line case to match the image file suffix case (jpg vs JPG). Change as needed.
    4. The last line (rm...) deletes the temporary files in order to free up harddrive space.

5. The script may be run as a batch process by creating the appropriate batch file for your operating system (i.e., an .sh bash shell for Linux) and copy-pasting multiple iterations of the script into the file. Change the "cd..." line for each iteration so it matches the desired folders, and let the script run overnight.

```
cd /Path/to/your/image/folder/
mm3d Tapioca All ".*.jpg" 2500
mm3d Tapas RadialBasic ".*.jpg" Out=Arbitrary
mm3d AperiCloud ".*.jpg" Arbitrary
rm -r Tmp-MM-Dir/
```

4. After the initial matching for an arbitrary orientation, identify and enter ground control points.
    1. Run the script below to activate the SaisieAppuisInitQT module.
    2. Click your known coordinate point in the first image and right-click to validate the point. Then add and validate in the other images. REMEMBER THE ORDER OF THE COORDINATES YOU CLICKED
    3. Exit Saisie... to generate the output files.
    4. The output will include two .xml files: GCP-S2D.xml and GCP-S3D.xml (unless you renamed them something else.)
    5. Open "GCP-S3D.xml" in a text editing program wuch as Wordpad.
    6. Manually edit each of the (arbitrary) 3D coordinates so it matches your known coords. The coords are in XYZ order, with a space between each value. So if the first known coordinate is X=45 Y=13 Z=10, it would be entered as: 45 13 10
       The order of the coordinates here must match the order in which the GCPs were clicked.
    7. Once all of the GCP coordinates have been entered, save the file.

```
mm3d SaisieAppuisInitQT ".*.jpg" Arbitrary GCP GCP.xml
```

5. [optional] Mask the master image in order to restrict point cloud production to exclude background points.
    1. Run the script below.
    2. Drag the master image onto the SaisieMasqQT window.
    3. Click around the lithic tool to define the mask boundary.
    4. When finished, save the mask and exit.

```
mm3d SaisieMasqQT
```

6. Run the script below to create georectified dense point clouds. Edit the relevant lines ("cd..." ad "Nuage2Ply...") to match your input folder and output file.
    1. In "Malt..." make sure the master image matches the master image used in SaisieMasqQT.
    2. This script may be run as a batch file as well.

```
cd /Path/to/your/image/folder/
mm3d GCPBascule ".*.jpg" Arbitrary Ground_Init GCP-S3D.xml GCP-S2D.xml
mm3d Campari ".*.jpg" Ground_Init Ground
mm3d AperiCloud ".*.jpg" Ground
mm3d Malt GeomImage ".*.jpg" Ground Master="img_000.jpg" ZoomF=1
mm3d Nuage2Ply "MM-Malt-Img-img_000/MMLastNuage.xml"
Attr="img_000.jpg" Out=../mb01_d00_dense.ply RatioAttrCarte=1
rm -r Tmp-MM-Dir/
```

Appendix G: CloudCompare Workflow

1. Download and install CloudCompare: http://www.danielgm.net/cc/
2. Point cloud cleaning
    1. Open the dense point cloud produced using MicMac (i.e., "mb14_d00_dense.ply")
    2. The point cloud should already be free of background points. If not, use the "Segment" tool (scissor icon) to trim the unwanted background.
    3. Examine the point cloud to determine how aggressively to filter inaccurate points.
    4. Basic cleaning:

        This is suitable for most high-quality point clouds that do not have a great deal of noise points.

        1. Use the "Statistical Outlier Removal" filter. (Found in the [Tools]->[Clean] menu.)
        2. Increasing the number of points used for calculation causes the filter to identify closer outliers.
        3. Decreasing the standard deviation value causes the filter to trim out more surface points.
        4. Use the Noise filter. (Found in the [Tools]->[Clean] menu.)
            1. Check the "Remove isolated points" option.
            2. Set the relative error to 0.5.
        - The resulting point cloud should have a relatively smooth surface. Occasionally a second round of filtering is needed.
    5. Aggressive cleaning

        This method is useful for point clouds with fairly substantial surface roughness problems, often the result of performing 3D photogrammetry on dark monochrome objects (such as black dacite microblades without surface treatments).

        1. Use the Cloth Simulation Filter (CSF).
            1. Set cloth resolution to 0.1. Leave the other settings at their defaults (or experiment as you see fit.)

- For small objects (smaller than 30 units) it is necessary to scale up the model in order to achieve adequate filter resolution. For the microblades used in this study, scaling up 10x was successful.
- The "ground points" output is the desired product.

2. Use the "Convert clouds to raster" tool. (in the [Tools] -> [Projection] menu.)
    1. Set the grid step to 0.05
    2. Set the active layer to "Height grid values"
    3. Set the projection cell height to "minimum height"
    4. Set the empty cells "Fill with" to "Leave empty"
    5. Click "Update grid".
    6. Under "Export", click "Cloud".

       A new point cloud will be produced in the main CloudCompare window.
3. Save the cleaned cloud.

3. <u>Dorsal/ventral surface alignment</u>
    1. Load both point clouds into CloudCompare.
    2. Use the "Align" tool. (In the [Tools] -> [Registration] menu.)
        1. Select the two point clouds to align.
        2. Activate the "Align" tool.
        3. Select the dorsal surface as "Reference" and ventral as "Align"

           This ensures that the dorsal surface will be "up" in subsequent procedures.
        4. Click common edge points on either surface. Small points, notches, corners, and other such landmarks where the dorsal and ventral surfaces converge are ideal.
        5. If the ventral surface model was precisely scaled using MicMac, be sure to un-check the "adjust scale" box.
        6. Click "Align" and confirm the alignment. Repeat as necessary to fine-tune the alignment.

4. <u>Sample iteration alignment</u>
    1. Load both time-series surfaces into CloudCompare.

2. Use the “Align” tool. (In the [Tools] -> [Registration] menu.)
    1. Follow the same process as for dorsal and ventral surfaces, but instead of edge points (because the edges may have been altered through wear damage) use common surface points whenever possible. It is sometimes useful to mark the surface with a pen or graver prior to starting the experiment in order to have alignment marks available for this step.
    - Make sure that the earliest sample (i.e., the unworn surface) is always the reference model.
    - Often a fairly rough alignment is adequate.
3. Once the surfaces have been generally aligned, use the “Fine Registration” tool.
    1. Select the two surfaces.
    2. Activate the “Fine Registration” tool. (In the [Tools] -> [Registration] menu.)
    3. Make sure the earlier model is the reference.
    4. Leave “adjust scale” un-checked (unless the point clouds are not to scale with each other).
    5. Click “OK”.
    6. Check the alignment. This is very important; because of changes to edge morphology, the fine registration tool occasionally grossly misaligns lithic surfaces.
4. Export the aligned surfaces as CSV files for use in QGIS.

Appendix H: R Script for Edge Wear Analysis

This script assumes very basic knowledge of R and R Studio. Copy-paste the following into an R markdown file, or download the markdown file from: https://github.com/nwaber/QGIS_lithics/blob/master/GIS_wear_MarkDownGH.Rmd

````
---
title: "GIS Wear"
author: "Nick Waber"
date: "October 31, 2018"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## GIS Use Wear Quantification
```{r, include=FALSE}
library(ggplot2)
library(plotly)
library(psych)
library(Gmisc)
library(janitor)
library(data.table)
library(ggpubr)
library(plyr)
library(dplyr)
library(reshape)
library(reshape2)
```
````

```{r, include=FALSE}

###################

# Load wear data from Github

###################

library(RCurl)

data_url <-
getURL("https://raw.githubusercontent.com/nwaber/QGIS_lithics/master/MB_all_wear.csv")

wear.all.df <- read.csv(text=data_url)


#wear.all.df file is composed of all wear data from 10 microblades- originally 10 seperate files

#sharpness data has been stripped


```



```{r, include=FALSE}

#aggregate values and merge aggregates into single DF

wear.agg.vol <- aggregate(volume ~ src + position, data = wear.all.df, FUN = "sum")

wear.agg.vol.V <- aggregate(volume.V ~ src + position, data = wear.all.df, FUN = "sum")

wear.agg.vol.D <- aggregate(volume.D ~ src + position, data = wear.all.df, FUN = "sum")

wear.agg.idx.mn <- aggregate(index ~ src + position, data = wear.all.df, FUN = "mean")

wear.agg.idx.mdn <- aggregate(index ~ src + position, data = wear.all.df, FUN = "median")

wear.agg.idx.mx <- aggregate(index ~ src + position, data = wear.all.df, FUN = "max")

wear.agg.idx.mn.V <- aggregate(index.V ~ src + position, data = wear.all.df, FUN = "mean")
```

```
wear.agg.idx.mdn.V <- aggregate(index.V ~ src + position, data =
wear.all.df, FUN = "median")

wear.agg.idx.mx.V <- aggregate(index.V ~ src + position, data =
wear.all.df, FUN = "max")

wear.agg.idx.mn.D <- aggregate(index.D ~ src + position, data =
wear.all.df, FUN = "mean")

wear.agg.idx.mdn.D <- aggregate(index.D ~ src + position, data =
wear.all.df, FUN = "median")

wear.agg.idx.mx.D <- aggregate(index.D ~ src + position, data =
wear.all.df, FUN = "max")

#wear.agg.sharp <- aggregate(mean.sharp ~ src + position, data =
wear.all.df, FUN = "mean")


wear.agg <- merge(wear.agg.vol, wear.agg.vol.D,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.vol.V,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mn,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mn.D,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mn.V,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mdn,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mdn.D,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mdn.V,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mx,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mx.D,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.idx.mx.V,
by=c("src","position"))

wear.agg <- merge(wear.agg, wear.agg.sharp,
by=c("src","position"))
```

```
colnames(wear.agg) <- c("src","position","volume","volume D",
"volume V", "index mn", "index D mn", "index V mn", "index
mdn","index D mdn", "index V mdn","index max", "index D max",
"index V max", "sharp")


wear.agg$set <- ifelse(wear.agg$src %like% "-1",1,2)

#distinguish between experimental sets 1 and 2

wear.agg$MB <- substring(wear.agg$src,0,4)

#add microblade labels


```


```{r, include=FALSE}

#vol.sum is the total wear on each microblade in each set.

#vol.sum is calculated by summing all of the wear volume,
grouped by MB source

#vol.pct is the wear volume of each individual MB section
devided by the total wear of that microblade.

Vsum <- rowsum(wear.agg$volume, group = wear.agg$src)

Vsum <- setDT(as.data.frame(Vsum), keep.rownames = TRUE)[]

colnames(Vsum) <- c("src","vol.sum")

wear.agg <- merge(wear.agg, Vsum,by.x = "src", by.y = "src")

wear.agg$vol.pct <-
floor(((wear.agg$volume/wear.agg$vol.sum)*100 )*100)/100

wear.agg$CSD <- wear.agg$set*500
```


Wear plots

Using ggplot and ggarrange to display wear volume and wear
percentage (index) barplots


```{r, echo=FALSE}
```

```
#Wear volume


gg.wear.right.1 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "right" & wear.agg$set==1),
aes(x=position, y=volume))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Volume of wear (mm^3)",
title="Set 1", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,3))  +coord_flip()


gg.wear.left.1 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "left" & wear.agg$set==1),
aes(x=position, y=volume))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Volume of wear
(mm^3)",title="Set 1", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,3))  +coord_flip()


gg.wear.right.2 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "right" & wear.agg$set==2),
aes(x=position, y=volume))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Volume of wear
(mm^3)",title="Set 2", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,3))  +coord_flip()


gg.wear.left.2 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "left" & wear.agg$set==2),
aes(x=position, y=volume))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +
```

```
scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Volume of wear
(mm^3)",title="Set 2", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,3))  +coord_flip()


p1 <- ggarrange(gg.wear.left.1, gg.wear.right.1, gg.wear.left.2,
gg.wear.right.2, nrow = 2, ncol = 2, common.legend = TRUE,
legend = "bottom")

annotate_figure(p1, top=text_grob("Wear volume by microblade
section: sets 1 and 2", color = "black", size=14))
```

```{r, echo=FALSE}
#Wear percent


gg.wearPCT.right.1 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "right" & wear.agg$set==1),
aes(x=position, y=vol.pct))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Percentage of total wear",
title="Set 1", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,100))  +coord_flip()


gg.wearPCT.left.1 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "left" & wear.agg$set==1),
aes(x=position, y=vol.pct))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +
```

```
labs(x="Microblade section", y="Percentage of total
wear",title="Set 1", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,100))  +coord_flip()


gg.wearPCT.right.2 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "right" & wear.agg$set==2),
aes(x=position, y=vol.pct))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Percentage of total
wear",title="Set 2", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,100))  +coord_flip()


gg.wearPCT.left.2 <- ggplot(data= subset( wear.agg ,
wear.agg$position %like% "left" & wear.agg$set==2),
aes(x=position, y=vol.pct))  +

stat_summary(fun.y = "sum", geom = "bar", aes(fill=MB),
position="dodge") +

scale_fill_brewer(palette = "Paired") +

labs(x="Microblade section", y="Percentage of total
wear",title="Set 2", fill="Microblade ID")+

theme(legend.position = "bottom")+ylim(c(0,100))  +coord_flip()


p2 <- ggarrange(gg.wearPCT.left.1, gg.wearPCT.right.1,
gg.wearPCT.left.2, gg.wearPCT.right.2, nrow = 2, ncol = 2,
common.legend = TRUE, legend = "bottom")

annotate_figure(p2, top=text_grob("Sectional wear as a
percentage of total blade wear volume: sets 1 and 2", color =
"black", size=14))



```


```{r, echo=FALSE}
```

```
p3 <- ggarrange(gg.wear.left.1, gg.wear.right.1, gg.wear.left.2,
gg.wear.right.2, gg.wearPCT.left.1, gg.wearPCT.right.1,
gg.wearPCT.left.2, gg.wearPCT.right.2, nrow = 4, ncol = 2,
common.legend = TRUE, legend = "bottom")

fig5 <- annotate_figure(p3, top=text_grob("Sectional wear as a
percentage of total blade wear volume: sets 1 and 2", color =
"black", size=14))


fig5
```
```

####Wear progression

Tracking wear progression over two experimental sets.

```{r, echo=FALSE}

#create a blade-wide data frame for sequential wear tracking

#create a dummy DF for set 0 (0cm CSD, 0mm^3 wear)

wear.agg.dummy <- subset(wear.agg, wear.agg$set==1)

wear.agg.dummy$set <- 0

wear.agg.dummy$CSD <- 0

wear.agg.dummy[,3:14] <- 0

wear.agg.dummy$vol.sum <- 0

wear.agg.dummy$vol.pct <- 0

wear.agg.dummy$src <- paste(wear.agg.dummy$MB,
wear.agg.dummy$set, sep="-")

#merge the dummy data with the main data and aggregate it.

wear.agg <- rbind(wear.agg, wear.agg.dummy)

wear.agg.blade <- aggregate(volume ~ MB + set, data=wear.agg,
FUN="sum")

wear.agg.blade$CSD <- wear.agg.blade$set*500


#plot the sequential data

fig4_line <-ggplot(data=wear.agg.blade, aes(x=CSD, y=volume))
+geom_line(aes(color=MB),size=1) +labs(x="Cumulative stroke
distance (CSD) cm", y="Volume (mm^3)", color="Microblade")
```
```

```
#barplot alternative

fig4_bar <- ggplot(data= subset(wear.agg, wear.agg$set > 0) ,
aes(x=MB, y=volume))  +

stat_summary(fun.y = "sum", geom = "bar",
aes(fill=as.factor(set)), position="dodge", size = 1.5) +

scale_fill_manual(values = c("grey","black")) +

labs(x="Microblade", y="Volume of wear (mm^3)", fill =
"Experiment set") +

theme(legend.position = "bottom")  +ylim(c(0,6))


fig4_bar #used in paper

fig4_line #not used
```

Appendix I: Flattening a Flake

One challenging aspect of using GIS to quantify edge wear is accounting for the natural curvature and twists of many flakes. Although microblades are generally thought of as a particularly straight variety of flake, the curvature still exists; a microblade placed on a flat surface, ventral side down, will almost always be curved enough that the proximal end (often the bulb of percussion) and the distal end are in contact with the surface and the medial section is elevated (see Andrefsky 1986 for a detailed discussion of flake curvature). This is relevant to GIS-based wear quantification because raster data is not truly three dimensional, but rather is "2.5D". That is, the X and Y axes are bidirectional (an object can both start and end in horizontal space), but the Z-axis only has a single side: the top surface. Objects have no underside. As a result, raster surfaces have only a single Z-axis value, which in the case of lithics is *either* the ventral or the dorsal side, but not both. The single-sided nature of rasterized lithics means that when quantifying wear with a raster difference algorithm, surficial flake scars are measured as the Z-axis difference between the pristine surface and the worn surface, but places where the edge is destroyed (i.e., the trajectory of the margin has changed due to total material loss at that horizontal coordinate), the wear is measured as the difference between the unworn surface and the minimum Z-axis value of the entire raster. The goal of flake flattening is to change the minimum Z-axis value reference for each cell of the DSM from the overall minimum Z-axis value to the Z-axis value for the adjacent edge. By fitting a spline surface between all points around the flake's edges one can create a vertical reference plane: a Z_0 plane (the term used in this paper).

Flattening a flake using QGIS is, at its core, simply subtracting the Z0 plane from the DSM. The basic premise is based on methods developed for studying terrain topography relative to rivers: the river is used to generate a trend surface which is subtracted from the surrounding landscape, leaving elevation data the reflects elevation above the closest point on the river rather than elevation above sea level. Applied to lithic flakes, the perimeter edge of the flake is treated as the river and is sampled at a regular interval (0.2mm in this case). These points are then interpolated to create a trend surface (the Z_0 plane), which is subtracted from the un-adjusted lithic "elevation" (or added to negative elevations) (Figure I.1). The outcome is a flattened flake

(Figure I.2). The required base data is a point vector file (CSV or SHP with X/Y/Z coordinates) for the dorsal or ventral surface and a line vector file for the flake perimeter. All steps are performed using modules in the QGIS Processing Toolbox. A complete Processing Toolbox python script is provided below and can be downloaded from: https://github.com/nwaber/QGIS_lithics/releases/tag/v0.9 . Commented lines for each command explain the relevance of the command to the overall procedure.

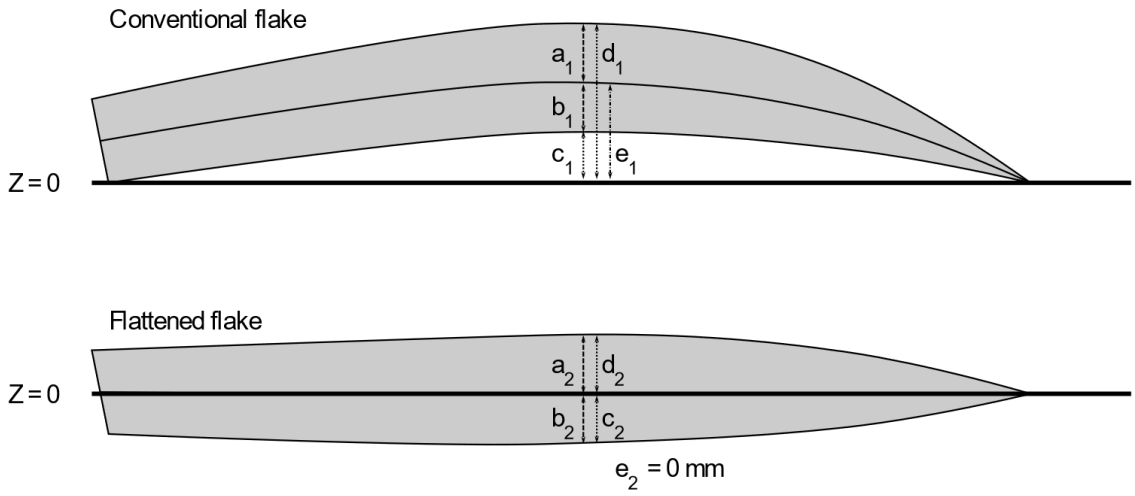

Figure I.1: Idealized flake conversion from curved to flattened. a1 = a2; b1 = b2; c1 = distance from ventral surface to Z0 plane; c2 = b2; d1 = a1 + e1.

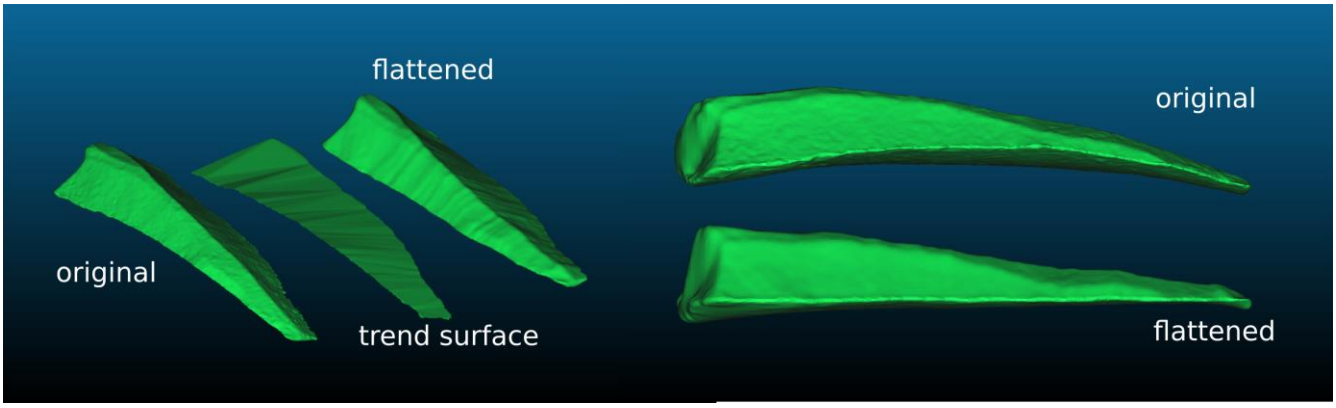

Figure I.2: Original (curved) flake, trend surface, and flattened flake comparison using Poisson Surface generated from point clouds.

QGIS Flake Flattening Script:

```
from qgis.core import QgsProcessing
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingParameterVectorLayer
from qgis.core import QgsProcessingParameterField
from qgis.core import QgsProcessingParameterRasterDestination
import processing


class TrendSurface(QgsProcessingAlgorithm):

    def initAlgorithm(self, config=None):

self.addParameter(QgsProcessingParameterVectorLayer('perimeter',
'Perimeter', types=[QgsProcessing.TypeVectorLine], defaultValue=None))
        self.addParameter(QgsProcessingParameterVectorLayer('points',
'Points', types=[QgsProcessing.TypeVectorPoint], defaultValue=None))
        self.addParameter(QgsProcessingParameterField('zfield', 'Z
field', type=QgsProcessingParameterField.Numeric,
parentLayerParameterName='points', allowMultiple=False,
defaultValue=None))

self.addParameter(QgsProcessingParameterRasterDestination('Idw', 'IDW',
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterRasterDestination('Trend',
'TREND', createByDefault=True, defaultValue=None))

    def processAlgorithm(self, parameters, context, model_feedback):
        # Use a multi-step feedback, so that individual child algorithm
progress reports are adjusted for the
        # overall progress through the model
        feedback = QgsProcessingMultiStepFeedback(9, model_feedback)
        results = {}
        outputs = {}

        # Convert lines to polygons
        # The polygon is needed for buffer layer to define  output
extent of Inverse-Distance-Weighted Interpolation;
        # IDW interpolation must extent beyond the perimeter points in
order to be sampled for Z0 trend plane interpolation
        alg_params = {
            'LINES': parameters['perimeter'],
            'POLYGONS': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ConvertLinesToPolygons'] =
processing.run('saga:convertlinestopolygons', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(1)
        if feedback.isCanceled():
            return {}
```

```
        # Convert lines to points
        # Points will be used to sample the IDW surface at the
perimeter of the flake.  Point spacing interval is 0.2mm.
        alg_params = {
            'ADD         ': True,
            'DIST': 0.2,
            'LINES': parameters['perimeter'],
            'POINTS': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ConvertLinesToPoints'] =
processing.run('saga:convertlinestopoints', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(2)
        if feedback.isCanceled():
            return {}

        # Buffer
        # Needed to define  output extent of Inverse-Distance-Weighted
Interpolation;
        # Interpolation must extent beyond the perimeter points in
order to be sampled for Z0 trend plane interpolation
        alg_params = {
            'DISSOLVE': False,
            'DISTANCE': 0.1,
            'END_CAP_STYLE': 0,
            'INPUT': outputs['ConvertLinesToPolygons']['POLYGONS'],
            'JOIN_STYLE': 0,
            'MITER_LIMIT': 2,
            'SEGMENTS': 5,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['Buffer'] = processing.run('native:buffer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(3)
        if feedback.isCanceled():
            return {}

        # Inverse distance weighted interpolation
        # IDW: interpolate flake surface.  IDW used with power = 1.5,
resolution = 0.05mm.  Precision is good, and outlying points are
negated.
        alg_params = {
            'DW_BANDWIDTH': 1,
            'DW_IDW_OFFSET': False,
            'DW_IDW_POWER': 1.5,
            'DW_WEIGHTING': 1,
            'FIELD': parameters['zfield'],
            'SEARCH_DIRECTION': 0,
            'SEARCH_POINTS_ALL': 0,
            'SEARCH_POINTS_MAX': 20,
            'SEARCH_POINTS_MIN': -1,
            'SEARCH_RADIUS': 1000,
            'SEARCH_RANGE': 0,
            'SHAPES': parameters['points'],
            'TARGET_DEFINITION': 0,
```

```
            'TARGET_TEMPLATE': None,
            'TARGET_USER_FITS': 0,
            'TARGET_USER_SIZE': 0.05,
            'TARGET_USER_XMIN TARGET_USER_XMAX TARGET_USER_YMIN
TARGET_USER_YMAX': outputs['Buffer']['OUTPUT'],
            'TARGET_OUT_GRID': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['InverseDistanceWeightedInterpolation'] =
processing.run('saga:inversedistanceweightedinterpolation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(4)
        if feedback.isCanceled():
            return {}

        # Translate (convert format)
        # SAGA raster output in QGIS 3.X defaults to SGRID and must be
changed to TIF in order to be used and exported successfully. It's a
pain in the ass and I wish I could fix it.
        alg_params = {
            'COPY_SUBDATASETS': False,
            'DATA_TYPE': 0,
            'INPUT':
outputs['InverseDistanceWeightedInterpolation']['TARGET_OUT_GRID'],
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_CRS': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['TranslateConvertFormat'] =
processing.run('gdal:translate', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(5)
        if feedback.isCanceled():
            return {}

        # Add raster values to points
        # Points created along perimeter line sample Z values from IDW
surface.
        alg_params = {
            'GRIDS': outputs['TranslateConvertFormat']['OUTPUT'],
            'RESAMPLING': 0,
            'SHAPES': outputs['ConvertLinesToPoints']['POINTS'],
            'RESULT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['AddRasterValuesToPoints'] =
processing.run('saga:addrastervaluestopoints', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(6)
        if feedback.isCanceled():
            return {}

        # Clip raster with polygon
```

```
        # Clip the IDW surface and export it.  Exporting makes it easy
to check that the IDW surface was currectly interpolated. (Compare with
the real artifact)
        alg_params = {
            'INPUT':
outputs['InverseDistanceWeightedInterpolation']['TARGET_OUT_GRID'],
            'POLYGONS': outputs['ConvertLinesToPolygons']['POLYGONS'],
            'OUTPUT': parameters['Idw']
        }
        outputs['ClipRasterWithPolygon'] =
processing.run('saga:cliprasterwithpolygon', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
        results['Idw'] = outputs['ClipRasterWithPolygon']['OUTPUT']

        feedback.setCurrentStep(7)
        if feedback.isCanceled():
            return {}

        # Natural neighbour
        # Z0 "trend" surface is interpolated between perimeter points.
        alg_params = {
            'FIELD': parameters['zfield'],
            'METHOD': 0,
            'SHAPES': outputs['AddRasterValuesToPoints']['RESULT'],
            'TARGET_TEMPLATE': None,
            'TARGET_USER_FITS': 0,
            'TARGET_USER_SIZE': 0.05,
            'TARGET_USER_XMIN TARGET_USER_XMAX TARGET_USER_YMIN
TARGET_USER_YMAX': None,
            'WEIGHT': 0,
            'TARGET_OUT_GRID': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['NaturalNeighbour'] =
processing.run('saga:naturalneighbour', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(8)
        if feedback.isCanceled():
            return {}

        # Clip raster with polygon
        # Natural Neighbour surface (Z0 "trend" surface) is clipped and
exported to "Trend".
        alg_params = {
            'INPUT': outputs['NaturalNeighbour']['TARGET_OUT_GRID'],
            'POLYGONS': outputs['ConvertLinesToPolygons']['POLYGONS'],
            'OUTPUT': parameters['Trend']
        }
        outputs['ClipRasterWithPolygon'] =
processing.run('saga:cliprasterwithpolygon', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
        results['Trend'] = outputs['ClipRasterWithPolygon']['OUTPUT']
        return results

    def name(self):
        return 'trend surface'
```

```
def displayName(self):
    return 'trend surface'

def group(self):
    return 'lithic analysis'

def groupId(self):
    return 'lithic analysis'

def createInstance(self):
    return TrendSurface()
```

Appendix J: Point Clusters for Edge and Surface Reconstruction

The MicroGIURF edge point extrapolation experiments showed that most points clustered within fractions of a millimetre of the original flake edge, and those further out were scattered relatively widely. This is borne out in visual analysis of the point extrapolation plots (Figures 5-7). On this basis we may infer that points cluster close to the correct position, and that the centres of the clusters may be used to guide reconstructed edge trajectories. This inference is the basis of the QGIS edge reconstruction process.

The QGIS edge reconstruction process uses modules from the Processing Toolbox to identify point clusters, select the densest clusters (or rather exclude minor clusters), mark their centres, and create a new flake edge on the basis of those centres. The QGIS script and Processing model is downloadable from https://github.com/nwaber/QGIS_lithics/releases/tag/v0.9. At its core, the process involves two primary workflows: tracing the original edge from the extrapolated points, and interpolating a surface between those points and points sampled on the pristine surface areas of the worn artifact. Input layers are the worn dorsal and ventral surfaces (raster files), the worn flake perimeter (vector line), the extrapolated points (vector points), a polygon marking the striking platform and/or distal step termination, and a number indicating the expected maximum flake scar length (default: 0.8mm). The platform/step termination polygon functions to mask sections of the flake perimeter that are not expected to intersect the Z0 plane. The expected flake scar length instructs the model to sample the surface further from the edges than that value. This avoids sampling in flake scars and thus simply replicating the worn edge. The basic (and heavily abridged) process is described below; refer to the QGIS Python script or Processing Model for the complete process of 42 linked QGIS Processing modules.

1. Generate the edge based on the extrapolated points (Figure A2.1).
    1. Generate a kernel density map (a “heatmap”) from the extrapolated points. This identifies cluster areas.
    2. Use SAGA: Local Minima and Maxima to mark the most dense place in each cluster. The resulting points are henceforth referred to as “density points”.

3. Extract the density points with density values above the mean.
4. Order the density points by azimuth relative to the centre of the flake perimeter. This allows QGIS to draw a line linking the points in clockwise order, rather than the order of densities or arbitrary ID field which both result in a zig-zagging path.
5. Convert the newly-drawn density point line to a polygon.
6. Merge that polygon with a polygon generated from the worn flake perimeter. This avoids instances where the density point line falls within the worn perimeter, which would result in the wear quantification algorithm identifying that area as lithic material "growing" on the flake during use.
7. Convert the merged polygons to points and assign a 0 Z value.

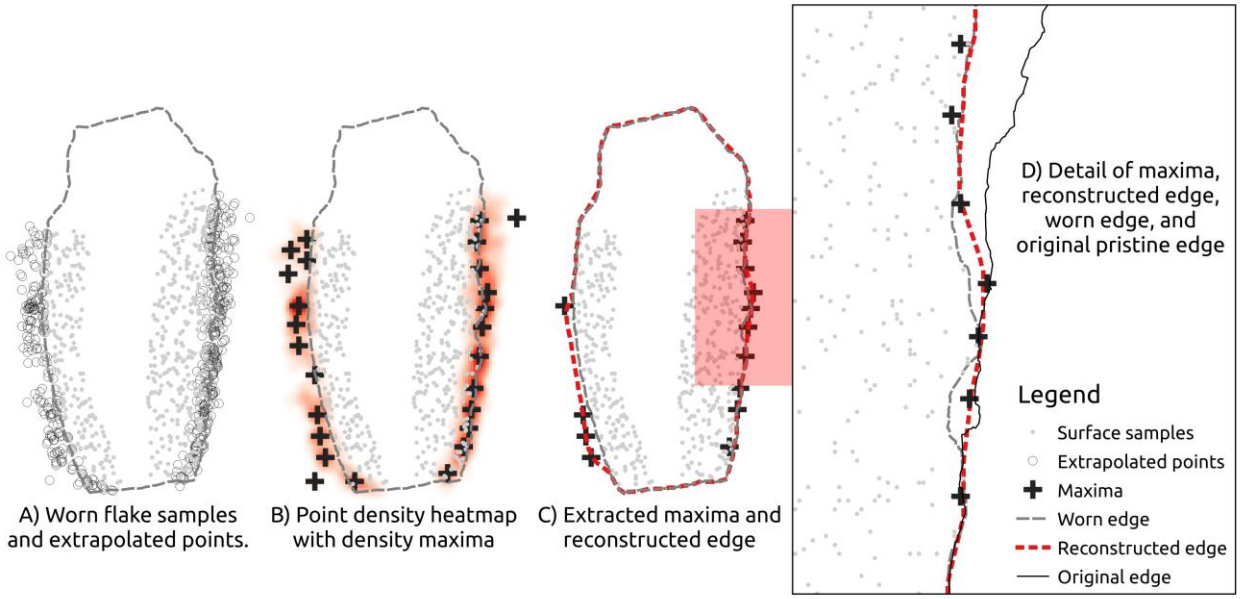

Figure 1: Example of point density-based edge reconstruction. Image D shows both the sought-after result in the more medial section, and under-estimation resulting in edge innacuracy in the more proximal section.. Note the inaccuracy of the maxima and reconstructed edge near the left distal section of the microblade in image C whether the extrapolated points were placed with overestimated distance values.

2. Sample the dorsal and ventral DSMs.
    1. Draw buffer for difference platforms and perimeter.
    2. Clip rasters to exclude buffered areas.

    3. Convert raster values to points.
3. Interpolate the surface between the sampled points and the extrapolated edge points.
    1. Refactor density points and raster points to match fields.
    2. Merge density points and raster points.
    3. Interpolate dorsal and ventral reconstructed surfaces to 0.05mm grid resolution.
    4. Clip surface to exclude data outside the extrapolated perimeter polygon.
    5. Export reconstructed surfaces.

QGIS Python Script for edge reconstruction:

```
from qgis.core import QgsProcessing
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingParameterNumber
from qgis.core import QgsProcessingParameterRasterLayer
from qgis.core import QgsProcessingParameterVectorLayer
from qgis.core import QgsProcessingParameterRasterDestination
import processing


class EdgeFromProjection04(QgsProcessingAlgorithm):

    def initAlgorithm(self, config=None):

self.addParameter(QgsProcessingParameterNumber('expectedflakescarlength
mm', 'Expected flake scar length (mm)',
type=QgsProcessingParameterNumber.Double, minValue=0, maxValue=25,
defaultValue=0.8))

self.addParameter(QgsProcessingParameterRasterLayer('lithicsurface',
'Worn dorsal surface', defaultValue=None))

self.addParameter(QgsProcessingParameterVectorLayer('perimeter',
'Perimeter', types=[QgsProcessing.TypeVectorLine], defaultValue=None))

self.addParameter(QgsProcessingParameterVectorLayer('platformspolygon',
'Platform(s) [polygon]', types=[QgsProcessing.TypeVectorPolygon],
defaultValue=None))

self.addParameter(QgsProcessingParameterVectorLayer('projectedpoints',
'Projected points', types=[QgsProcessing.TypeVectorPoint],
defaultValue=None))

self.addParameter(QgsProcessingParameterRasterLayer('wornventralsurface
', 'Worn ventral surface', defaultValue=None))

self.addParameter(QgsProcessingParameterRasterDestination('DorsalRecons
```

```
truction', 'DORSAL RECONSTRUCTION', createByDefault=True,
defaultValue=None))

self.addParameter(QgsProcessingParameterRasterDestination('VentralRecon
struction', 'VENTRAL RECONSTRUCTION', createByDefault=True,
defaultValue=None))

    def processAlgorithm(self, parameters, context, model_feedback):
        # Use a multi-step feedback, so that individual child algorithm
progress reports are adjusted for the
        # overall progress through the model
        feedback = QgsProcessingMultiStepFeedback(42, model_feedback)
        results = {}
        outputs = {}

        # Translate (convert format) DUMMY VENTRAL
        # This is to change the name of the input layer to the default
layer name 'OUTPUT'.  QGIS scripts involving Refactor Fields and Field
Calculator need predictable layer names.
        alg_params = {
            'COPY_SUBDATASETS': False,
            'DATA_TYPE': 0,
            'INPUT': parameters['wornventralsurface'],
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_CRS': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['TranslateConvertFormatDummyVentral'] =
processing.run('gdal:translate', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(1)
        if feedback.isCanceled():
            return {}

        # Centroids
        # Creates the PERIM centroid for hub lines.
        alg_params = {
            'ALL_PARTS': False,
            'INPUT': parameters['perimeter'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['Centroids'] = processing.run('native:centroids',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

        feedback.setCurrentStep(2)
        if feedback.isCanceled():
            return {}

        # Convert lines to polygons PERIM
        # Converts the perimeter line to a polygon so it can merge with
the cluster-based perimeter polygon
        alg_params = {
            'LINES': parameters['perimeter'],
            'POLYGONS': QgsProcessing.TEMPORARY_OUTPUT
```

```
        }
        outputs['ConvertLinesToPolygonsPerim'] =
processing.run('saga:convertlinestopolygons', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(3)
        if feedback.isCanceled():
            return {}

        # Heatmap (Kernel Density Estimation)
        # Quantify density of extrapolated points
        alg_params = {
            'DECAY': 0,
            'INPUT': parameters['projectedpoints'],
            'KERNEL': 0,
            'OUTPUT_VALUE': 0,
            'PIXEL_SIZE': 0.1,
            'RADIUS': 0.5,
            'RADIUS_FIELD': None,
            'WEIGHT_FIELD': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['HeatmapKernelDensityEstimation'] =
processing.run('qgis:heatmapkerneldensityestimation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(4)
        if feedback.isCanceled():
            return {}

        # Points along geometry PERIM
        # Creates points along the perimeter for sections not recorded
by the cluster-generated points (i.e., proximal/distal ends)
        alg_params = {
            'DISTANCE': 0.2,
            'END_OFFSET': 0,
            'INPUT': parameters['perimeter'],
            'START_OFFSET': 0,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['PointsAlongGeometryPerim'] =
processing.run('qgis:pointsalonglines', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(5)
        if feedback.isCanceled():
            return {}

        # Translate DUMMY DEM
        # Changes DEM name, as above
        alg_params = {
            'COPY_SUBDATASETS': False,
            'DATA_TYPE': 0,
            'INPUT': parameters['lithicsurface'],
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_CRS': 'ProjectCrs',
```

```
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['TranslateDummyDem'] = processing.run('gdal:translate',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

        feedback.setCurrentStep(6)
        if feedback.isCanceled():
            return {}

        # Local minima and maxima
        # Finds the peak densities in the heatmap
        alg_params = {
            'GRID':
outputs['HeatmapKernelDensityEstimation']['OUTPUT'],
            'MAXIMA': QgsProcessing.TEMPORARY_OUTPUT,
            'MINIMA': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['LocalMinimaAndMaxima'] =
processing.run('saga:localminimaandmaxima', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(7)
        if feedback.isCanceled():
            return {}

        # Extract by expression
        # Extracts density peaks that are above the mean density value.
        alg_params = {
            'EXPRESSION': '\"Z\"  > mean(\"Z\")',
            'INPUT': outputs['LocalMinimaAndMaxima']['MAXIMA'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ExtractByExpression'] =
processing.run('native:extractbyexpression', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(8)
        if feedback.isCanceled():
            return {}

        # Field calculator DUMMY ID
        # Adds a dummy ID column to the density points layer for hub
line generation
        alg_params = {
            'FIELD_LENGTH': 10,
            'FIELD_NAME': 'ID',
            'FIELD_PRECISION': 3,
            'FIELD_TYPE': 0,
            'FORMULA': '1',
            'INPUT': outputs['ExtractByExpression']['OUTPUT'],
            'NEW_FIELD': True,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['FieldCalculatorDummyId'] =
processing.run('qgis:fieldcalculator', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)
```

```
        feedback.setCurrentStep(9)
        if feedback.isCanceled():
            return {}

        # Join by lines (hub lines)
        # Connects centroid to density points by hub lines.
        alg_params = {
            'ANTIMERIDIAN_SPLIT': False,
            'GEODESIC': False,
            'GEODESIC_DISTANCE': 1000,
            'HUBS': outputs['Centroids']['OUTPUT'],
            'HUB_FIELD': 'local_idx',
            'HUB_FIELDS': None,
            'SPOKES': outputs['FieldCalculatorDummyId']['OUTPUT'],
            'SPOKE_FIELD': 'ID',
            'SPOKE_FIELDS': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['JoinByLinesHubLines'] =
processing.run('native:hublines', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(10)
        if feedback.isCanceled():
            return {}

        # Field calculator AZIMUTH
        # Gives points azimuth value so the edge line is drawn in the
correct order.
        alg_params = {
            'FIELD_LENGTH': 10,
            'FIELD_NAME': 'AZIMUTH',
            'FIELD_PRECISION': 3,
            'FIELD_TYPE': 0,
            'FORMULA':
'azimuth(start_point($geometry),end_point($geometry))',
            'INPUT': outputs['JoinByLinesHubLines']['OUTPUT'],
            'NEW_FIELD': True,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['FieldCalculatorAzimuth'] =
processing.run('qgis:fieldcalculator', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(11)
        if feedback.isCanceled():
            return {}

        # Extract vertices
        alg_params = {
            'INPUT': outputs['FieldCalculatorAzimuth']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ExtractVertices'] =
processing.run('native:extractvertices', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)
```

```
        feedback.setCurrentStep(12)
        if feedback.isCanceled():
            return {}

        # Extract by location
        # Extracts aximuth line vertices where they patch the density
points.
        alg_params = {
            'INPUT': outputs['ExtractVertices']['OUTPUT'],
            'INTERSECT': outputs['ExtractByExpression']['OUTPUT'],
            'PREDICATE': 0,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ExtractByLocation'] =
processing.run('native:extractbylocation', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(13)
        if feedback.isCanceled():
            return {}

        # Field calculator DUMMY Z
        # Adds 0 Z value to ordered density points.
        alg_params = {
            'FIELD_LENGTH': 10,
            'FIELD_NAME': 'Z',
            'FIELD_PRECISION': 3,
            'FIELD_TYPE': 0,
            'FORMULA': '0',
            'INPUT': outputs['ExtractByLocation']['OUTPUT'],
            'NEW_FIELD': False,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['FieldCalculatorDummyZ'] =
processing.run('qgis:fieldcalculator', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(14)
        if feedback.isCanceled():
            return {}

        # Points to path
        # Draws edge line along the density points, in azimuth order.
        alg_params = {
            'DATE_FORMAT': '',
            'GROUP_FIELD': None,
            'INPUT': outputs['FieldCalculatorDummyZ']['OUTPUT'],
            'ORDER_FIELD': 'AZIMUTH',
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['PointsToPath'] = processing.run('qgis:pointstopath',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

        feedback.setCurrentStep(15)
        if feedback.isCanceled():
```

```
            return {}

        # Explode lines
        # I forget why this happens, but it's needed.
        alg_params = {
            'INPUT': outputs['PointsToPath']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ExplodeLines'] = processing.run('native:explodelines',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

        feedback.setCurrentStep(16)
        if feedback.isCanceled():
            return {}

        # Convert lines to polygons NEWEDGE
        # Creates a polygon from the newly drawn edge line.
        alg_params = {
            'LINES': outputs['PointsToPath']['OUTPUT'],
            'POLYGONS': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ConvertLinesToPolygonsNewedge'] =
processing.run('saga:convertlinestopolygons', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(17)
        if feedback.isCanceled():
            return {}

        # Extract by expression DROP CROSSLINES
        # This relates to the Explode Lines, but it was a while ago.
        alg_params = {
            'EXPRESSION': '$length < maximum($length)',
            'INPUT': outputs['ExplodeLines']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ExtractByExpressionDropCrosslines'] =
processing.run('native:extractbyexpression', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(18)
        if feedback.isCanceled():
            return {}

        # Difference PERIM PTS
        # Isolates points generated along the worn perimeter that fall
outside the new perimeter polygon.
        alg_params = {
            'INPUT': outputs['PointsAlongGeometryPerim']['OUTPUT'],
            'OVERLAY':
outputs['ConvertLinesToPolygonsNewedge']['POLYGONS'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['DifferencePerimPts'] =
processing.run('native:difference', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)
```

```
        feedback.setCurrentStep(19)
        if feedback.isCanceled():
            return {}

        # Points along geometry
        # Create points along the new edge lines
        alg_params = {
            'DISTANCE': 0.2,
            'END_OFFSET': 0,
            'INPUT':
outputs['ExtractByExpressionDropCrosslines']['OUTPUT'],
            'START_OFFSET': 0,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['PointsAlongGeometry'] =
processing.run('qgis:pointsalonglines', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(20)
        if feedback.isCanceled():
            return {}

        # Field calculator DUMMY Z
        # Adds dummy Z value for interpolation.  I feel like it might
be redundant, but IF IT AIN'T BROKE...
        alg_params = {
            'FIELD_LENGTH': 10,
            'FIELD_NAME': 'Z',
            'FIELD_PRECISION': 3,
            'FIELD_TYPE': 0,
            'FORMULA': '0',
            'INPUT': outputs['PointsAlongGeometry']['OUTPUT'],
            'NEW_FIELD': False,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['FieldCalculatorDummyZ'] =
processing.run('qgis:fieldcalculator', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(21)
        if feedback.isCanceled():
            return {}

        # Field calculator PERIM PTS
        # Adds another dummy Z value.
        alg_params = {
            'FIELD_LENGTH': 10,
            'FIELD_NAME': 'Z',
            'FIELD_PRECISION': 3,
            'FIELD_TYPE': 0,
            'FORMULA': '0',
            'INPUT': outputs['DifferencePerimPts']['OUTPUT'],
            'NEW_FIELD': True,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
```

```
        outputs['FieldCalculatorPerimPts'] =
processing.run('qgis:fieldcalculator', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(22)
        if feedback.isCanceled():
            return {}

        # Difference PLATFORMS
        # Polygons that mark striking platform and/or distal edge cut
out Z=0 points
        alg_params = {
            'INPUT': outputs['FieldCalculatorDummyZ']['OUTPUT'],
            'OVERLAY': parameters['platformspolygon'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['DifferencePlatforms'] =
processing.run('native:difference', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(23)
        if feedback.isCanceled():
            return {}

        # Buffer
        alg_params = {
            'DISSOLVE': True,
            'DISTANCE': parameters['expectedflakescarlengthmm'],
            'END_CAP_STYLE': 0,
            'INPUT': outputs['DifferencePlatforms']['OUTPUT'],
            'JOIN_STYLE': 0,
            'MITER_LIMIT': 2,
            'SEGMENTS': 5,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['Buffer'] = processing.run('native:buffer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(24)
        if feedback.isCanceled():
            return {}

        # Difference PTS OUTSIDE PERIM
        alg_params = {
            'INPUT': outputs['DifferencePlatforms']['OUTPUT'],
            'OVERLAY':
outputs['ConvertLinesToPolygonsPerim']['POLYGONS'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['DifferencePtsOutsidePerim'] =
processing.run('native:difference', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(25)
        if feedback.isCanceled():
            return {}
```

```
        # Difference SAMPLE v PROJECTED
        alg_params = {
            'INPUT':
outputs['ConvertLinesToPolygonsPerim']['POLYGONS'],
            'OVERLAY': outputs['Buffer']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['DifferenceSampleVProjected'] =
processing.run('native:difference', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(26)
        if feedback.isCanceled():
            return {}

        # Refactor fields PERIM PTS
        alg_params = {
            'FIELDS_MAPPING': [{'expression': '"Z"', 'length': 10,
'name': 'Z', 'precision': 3, 'type': 6}],
            'INPUT': outputs['FieldCalculatorPerimPts']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RefactorFieldsPerimPts'] =
processing.run('qgis:refactorfields', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(27)
        if feedback.isCanceled():
            return {}

        # Refactor fields EDGE
        alg_params = {
            'FIELDS_MAPPING': [{'expression': '"Z"', 'length': 10,
'name': 'Z', 'precision': 3, 'type': 6}],
            'INPUT': outputs['DifferencePtsOutsidePerim']['OUTPUT'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RefactorFieldsEdge'] =
processing.run('qgis:refactorfields', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(28)
        if feedback.isCanceled():
            return {}

        # Clip raster by mask layer DORSAL
        alg_params = {
            'ALPHA_BAND': False,
            'CROP_TO_CUTLINE': True,
            'DATA_TYPE': 0,
            'INPUT': outputs['TranslateDummyDem']['OUTPUT'],
            'KEEP_RESOLUTION': False,
            'MASK': outputs['DifferenceSampleVProjected']['OUTPUT'],
            'MULTITHREADING': False,
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_EXTENT': None,
```

```
            'TARGET_EXTENT_CRS': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ClipRasterByMaskLayerDorsal'] =
processing.run('gdal:cliprasterbymasklayer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(29)
        if feedback.isCanceled():
            return {}

        # Clip raster by mask layer VENTRAL
        alg_params = {
            'ALPHA_BAND': False,
            'CROP_TO_CUTLINE': True,
            'DATA_TYPE': 0,
            'INPUT':
outputs['TranslateConvertFormatDummyVentral']['OUTPUT'],
            'KEEP_RESOLUTION': False,
            'MASK': outputs['DifferenceSampleVProjected']['OUTPUT'],
            'MULTITHREADING': False,
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_EXTENT': None,
            'TARGET_EXTENT_CRS': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ClipRasterByMaskLayerVentral'] =
processing.run('gdal:cliprasterbymasklayer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(30)
        if feedback.isCanceled():
            return {}

        # Raster values to points VENTRAL
        alg_params = {
            'GRIDS': outputs['ClipRasterByMaskLayerVentral']['OUTPUT'],
            'NODATA': True,
            'POLYGONS': None,
            'TYPE': 0,
            'SHAPES': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RasterValuesToPointsVentral'] =
processing.run('saga:rastervaluestopoints', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(31)
        if feedback.isCanceled():
            return {}

        # Raster values to points DORSAL
        alg_params = {
            'GRIDS': outputs['ClipRasterByMaskLayerDorsal']['OUTPUT'],
            'NODATA': True,
            'POLYGONS': None,
            'TYPE': 0,
```

```
            'SHAPES': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RasterValuesToPointsDorsal'] =
processing.run('saga:rastervaluestopoints', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(32)
        if feedback.isCanceled():
            return {}

        # Refactor fields DORSAL
        alg_params = {
            'FIELDS_MAPPING': [{'expression': '"OUTPUT"', 'length': 18,
'name': 'Z', 'precision': 10, 'type': 6}],
            'INPUT': outputs['RasterValuesToPointsDorsal']['SHAPES'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RefactorFieldsDorsal'] =
processing.run('qgis:refactorfields', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(33)
        if feedback.isCanceled():
            return {}

        # Refactor fields VENTRAL
        alg_params = {
            'FIELDS_MAPPING': [{'expression': '"OUTPUT"', 'length': 18,
'name': 'Z', 'precision': 10, 'type': 6}],
            'INPUT': outputs['RasterValuesToPointsVentral']['SHAPES'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['RefactorFieldsVentral'] =
processing.run('qgis:refactorfields', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(34)
        if feedback.isCanceled():
            return {}

        # Merge vector layers VENTRAL
        alg_params = {
            'CRS': None,
            'LAYERS':
[outputs['RefactorFieldsEdge']['OUTPUT'],outputs['RefactorFieldsPerimPt
s']['OUTPUT'],outputs['RefactorFieldsVentral']['OUTPUT']],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['MergeVectorLayersVentral'] =
processing.run('native:mergevectorlayers', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(35)
        if feedback.isCanceled():
            return {}

        # Concave hull (alpha shapes) VENTRAL
```

```
        alg_params = {
            'ALPHA': 0.275,
            'HOLES': False,
            'INPUT': outputs['MergeVectorLayersVentral']['OUTPUT'],
            'NO_MULTIGEOMETRY': False,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['ConcaveHullAlphaShapesVentral'] =
processing.run('qgis:concavehull', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(36)
        if feedback.isCanceled():
            return {}

        # Merge vector layers DORSAL
        alg_params = {
            'CRS': None,
            'LAYERS':
[outputs['RefactorFieldsDorsal']['OUTPUT'],outputs['RefactorFieldsEdge'
]['OUTPUT'],outputs['RefactorFieldsPerimPts']['OUTPUT']],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['MergeVectorLayersDorsal'] =
processing.run('native:mergevectorlayers', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(37)
        if feedback.isCanceled():
            return {}

        # Grid (Linear) DORSAL
        alg_params = {
            'DATA_TYPE': 5,
            'INPUT': outputs['MergeVectorLayersDorsal']['OUTPUT'],
            'NODATA': 0,
            'OPTIONS': '',
            'RADIUS': -1,
            'Z_FIELD': 'Z',
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['GridLinearDorsal'] = processing.run('gdal:gridlinear',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

        feedback.setCurrentStep(38)
        if feedback.isCanceled():
            return {}

        # Concave hull (alpha shapes) DORSAL
        alg_params = {
            'ALPHA': 0.275,
            'HOLES': False,
            'INPUT': outputs['MergeVectorLayersDorsal']['OUTPUT'],
            'NO_MULTIGEOMETRY': False,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
```

```
        outputs['ConcaveHullAlphaShapesDorsal'] =
processing.run('qgis:concavehull', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(39)
        if feedback.isCanceled():
            return {}

        # Grid (Linear) VENTRAL
        alg_params = {
            'DATA_TYPE': 5,
            'INPUT': outputs['MergeVectorLayersVentral']['OUTPUT'],
            'NODATA': 0,
            'OPTIONS': '',
            'RADIUS': -1,
            'Z_FIELD': 'Z',
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['GridLinearVentral'] =
processing.run('gdal:gridlinear', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(40)
        if feedback.isCanceled():
            return {}

        # Clip raster by mask layer VENTRAL OUTPUT
        alg_params = {
            'ALPHA_BAND': False,
            'CROP_TO_CUTLINE': True,
            'DATA_TYPE': 0,
            'INPUT': outputs['GridLinearVentral']['OUTPUT'],
            'KEEP_RESOLUTION': False,
            'MASK': outputs['ConcaveHullAlphaShapesVentral']['OUTPUT'],
            'MULTITHREADING': False,
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_EXTENT': None,
            'TARGET_EXTENT_CRS': None,
            'OUTPUT': parameters['VentralReconstruction']
        }
        outputs['ClipRasterByMaskLayerVentralOutput'] =
processing.run('gdal:cliprasterbymasklayer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
        results['VentralReconstruction'] =
outputs['ClipRasterByMaskLayerVentralOutput']['OUTPUT']

        feedback.setCurrentStep(41)
        if feedback.isCanceled():
            return {}

        # Clip raster by mask layer DORSAL OUTPUT
        alg_params = {
            'ALPHA_BAND': False,
            'CROP_TO_CUTLINE': True,
            'DATA_TYPE': 0,
            'INPUT': outputs['GridLinearDorsal']['OUTPUT'],
```

```
            'KEEP_RESOLUTION': False,
            'MASK': outputs['ConcaveHullAlphaShapesDorsal']['OUTPUT'],
            'MULTITHREADING': False,
            'NODATA': None,
            'OPTIONS': '',
            'TARGET_EXTENT': None,
            'TARGET_EXTENT_CRS': None,
            'OUTPUT': parameters['DorsalReconstruction']
        }
        outputs['ClipRasterByMaskLayerDorsalOutput'] =
processing.run('gdal:cliprasterbymasklayer', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
        results['DorsalReconstruction'] =
outputs['ClipRasterByMaskLayerDorsalOutput']['OUTPUT']
        return results

    def name(self):
        return 'edge from projection 0.4'

    def displayName(self):
        return 'edge from projection 0.4'

    def group(self):
        return 'lithics IN PROGRESS'

    def groupId(self):
        return 'lithics IN PROGRESS'

    def createInstance(self):
        return EdgeFromProjection04()
```