

Near-Real-Time Implementation of Multiple Light Source Optical Flow

by

Cristina Elena Siegerist

M.Sc., Physics, University of Oregon, 1993.

Lic. Physics, Universidad de Buenos Aires, Argentina, 1989.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia

April 1996

© Cristina Elena Siegerist, 1996

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science
The University of British Columbia
Vancouver, Canada

Date April 22, 1996

Abstract

Multiple light source optical flow is a method to compute a dense, local representation of optical flow. The basic idea is to overcome the aperture problem using multiple images of a moving object acquired simultaneously under different conditions of illumination. Each image provides one linear constraint equation. When the optical flow and the 2D motion field coincide these equations are in the same unknowns. Two equations are enough to calculate both velocity components. Three or more images over-determine the system, allowing the calculation of confidence measures and making the computation more robust.

This thesis presents a parallel, near-real-time implementation of multiple light source optical flow. Three light sources, one red, one green, and one blue, with negligible overlap in the visible spectrum illuminate the work place from different directions. Three separate black and white images corresponding to the three light sources are acquired simultaneously through a 3 CCD 24 bit RGB camera. The processing is done using a network of six independent processors. The encoded optical flow is displayed on a video monitor. For an image of size 240x240 pixels the processing rate is 4 frames/sec. The range of linear behaviour of the system was studied. A comparison with other methods on a synthetic and a real image sequence is presented.

Contents

Abstract	ii
Contents	iv
List of Tables	vi
List of Figures	vii
Acknowledgements	xi
1 Introduction	1
2 Optical Flow	3
2.1 Introduction	3
2.2 The Optical Flow Constraint Equation	4
2.3 Multiple Light Source Optical Flow	5
2.4 Error measures	7
3 Summary of Optical Flow Methods	9
3.1 Differential Methods	9
3.2 Region-based matching methods	14

3.3	Frequency-based methods	15
4	Implementation	17
4.1	Experimental Setup	17
4.2	The C40 processors	18
4.2.1	Communication Ports	19
4.2.2	DMA Coprocessor Benefits	19
4.2.3	Memory organization	20
4.2.4	Internal and external buses	20
4.3	Processing Scheme	20
4.4	Linearity of the motion stages	23
5	Results	25
5.1	Stay-Puft Marshmallow Man	25
5.2	Expanding and contracting balloon	36
5.3	Comparison with other methods	43
5.3.1	Lambertian sphere	44
5.3.2	Stay-puft Marshmallow Man	45
6	Conclusions	80
	Appendix A Least squares solution of linear equations	82
	Bibliography	84

List of Tables

5.1	Average error for the translating sphere.	45
-----	---	----

List of Figures

2.1	The optical flow constraint equation	5
2.2	The aperture problem	6
4.1	Topology of the C40 network.	21
4.2	Processing scheme.	22
4.3	Estimation of partial derivatives using central differences.	22
4.4	Behavior of the motion stages	24
5.1	Object used in the experiments.	26
5.2	Optical flow encoding.	27
5.3	Example of real-time motion detection.	28
5.4	Example of real-time motion detection.	29
5.5	Example of real-time motion detection.	30
5.6	Example of real-time motion detection.	31
5.7	Magnitude of the average measured speed vs real speed for a 240x240 image and its average, 120x120.	32
5.8	Relative error vs average measured speed.	33
5.9	Magnitude of the average measured speed vs real speed for different differentiation schemes.	34

5.10 Relative error vs average measured speed for different derivative estimation schemes.	35
5.11 Balloon illuminated with three light sources.	36
5.12 Real-time motion detection for an expanding balloon.	37
5.13 Real-time motion detection for a contracting balloon.	38
5.14 Multiple light source optical flow for a deforming balloon.	40
5.15 Relative error for the deforming balloon.	41
5.16 Condition number for the deforming balloon.	42
5.17 Lambertian sphere illuminated from three different directions.	46
5.18 Translating Lambertian sphere, Horn's technique	47
5.19 Translating Lambertian sphere, Nagel's technique	48
5.20 Translating Lambertian sphere, Lucas and Kanade technique	49
5.21 Translating Lambertian sphere, Uras et al. technique	50
5.22 Translating Lambertian sphere, multiple light source optical flow	51
5.23 Translating Lambertian sphere. Relative error for multiple light source optical flow.	52
5.24 Translating Lambertian sphere. Condition number for multiple light source optical flow.	53
5.25 Translating Lambertian sphere, thresholded case	54
5.26 Translating Lambertian sphere. Average measured speed vs real speed for multiple light source optical flow.	55
5.27 Translating Lambertian sphere. Average relative error vs real speed for multiple light source optical flow.	56
5.28 Stay-Puft Marshmallow Man illuminated from three different directions.	58

5.29 Translating Stay-Puft Marshmallow Man, Horn's technique. Red light source.	59
5.30 Translating Stay-Puft Marshmallow Man, Horn's technique. Green light source.	60
5.31 Translating Stay-Puft Marshmallow Man, Horn's technique. Blue light source.	61
5.32 Translating Stay-Puft Marshmallow Man, Lucas and Kanade tech- nique. Red light source.	62
5.33 Translating Stay-Puft Marshmallow Man, Lucas and Kanade tech- nique. Green light source.	63
5.34 Translating Stay-Puft Marshmallow Man, Lucas and Kanade tech- nique. Blue light source.	64
5.35 Translating Stay-Puft Marshmallow Man, Uras et al. technique. Red light source.	65
5.36 Translating Stay-Puft Marshmallow Man, Uras et al. technique. Green light source.	66
5.37 Translating Stay-Puft Marshmallow Man, Uras et al. technique. Blue light source.	67
5.38 Translating Stay-Puft Marshmallow Man, Nagel's technique. Red light source.	68
5.39 Translating Stay-Puft Marshmallow Man, Nagel's technique. Green light source.	69
5.40 Translating Stay-Puft Marshmallow Man, Nagel's technique. Blue light source.	70

5.41	Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. Red light source.	71
5.42	Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. Green light source.	72
5.43	Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. Blue light source.	73
5.44	Translating Stay-Puft Marshmallow Man, Anandan's technique. Red light source.	74
5.45	Translating Stay-Puft Marshmallow Man, Anandan's technique. Green light source.	75
5.46	Translating Stay-Puft Marshmallow Man, Anandan's technique. Blue light source.	76
5.47	Translating Stay-Puft Marshmallow Man, multiple light source opti- cal flow	77
5.48	Translating Stay-Puft Marshmallow Man. Relative error for multiple light source optical flow.	78
5.49	Translating Stay-Puft Marshmallow Man. Condition number for mul- tiple light source optical flow.	79

Acknowledgements

I would like to thank my supervisor Dr. Robert J. Woodham for his guidance and support. Dr. James Little kindly accepted to be my second reader, improving this thesis with his comments.

Michael Sahota, Rod Barman and Stewart Kingdon helped me with the implementation. I am very grateful for the time they spent answering my questions and for sharing what they know.

UBC, the Computer Science department and in particular the Laboratory for Computational Intelligence have been a fantastic place to live and work.

I have to give special thanks to all my friends, they helped me in some very difficult times.

CRISTINA ELENA SIEGERIST

The University of British Columbia

April 1996

Chapter 1

Introduction

Vision systems are designed to provide information about the three-dimensional structure of the world given the sequence of two-dimensional images continuously received by a retina. One important source of information is the change in time of the intensity pattern due to the three-dimensional motion of surfaces. The first stage in the processing of this information is to analyze the change in the intensity pattern to obtain the optical flow field, defined as the 2D velocity vector of the brightness values. The second stage is to use the optical flow field, for example, to segment objects in the scene and to reconstruct the three-dimensional structure of the scene.

This thesis will focus on the implementation of a parallel near-real-time multiple light source optical flow method [3]. The basic idea is to use multiple images taken under different conditions of illumination to overcome the aperture problem locally. Each image is assumed to satisfy the standard optical flow constraint equation. When the optical flow and the 2D velocity field coincide, these multiple equations are in the same unknowns. Two equations are enough to determine the

solution. Three equations, however, over-determine the solution, avoid local degeneracies and help to make the computation more robust[3].

A near-real-time solution to this problem requires a parallel implementation. The processors used in this implementation are the Texas Instruments TMS320C40 (C40). Six processors are connected in a network with a Sun workstation as a host. Three light sources, one red, one green, and one blue, illuminate the work place from different directions. The light sources have negligible overlap in the visible spectrum. Three separate black and white images, corresponding to the different conditions of illumination are simultaneously acquired through a 3 CCD 24 bit RGB camera and sent to the frame grabber connected to one of the C40 processors. A least squares solution for the two velocity components is calculated in parallel using four processors. The display of the motion field is performed through a frame buffer connected to another node which sends the output to a monitor.

The organization of this thesis is as follows. Chapter 2 will describe in detail the optical flow problem and the approach followed in the implementation. Chapter 3 will summarize other optical flow techniques. Chapter 4 will focus on the implementation of the system using the network of C40 processors. Chapter 5 will present the results and a comparison with other methods, and Chapter 6 will present the conclusions.

Chapter 2

Optical Flow

2.1 Introduction

The motion of an object in front of a camera produces intensity changes in the recorded 2D image. These intensity changes can be used to recover the relative motions and shapes of the objects in the scene and to segment the image into parts corresponding to different objects. A first step in solving these problems is to recover the motion field, defined in this thesis as the perspective projection onto the image plane of the 3D velocity field of moving objects in space. What is available to us is the motion of the brightness pattern in the image, known as optical flow.

There are some well known examples in which the optical flow does not give any information about the motion field. This is the case for a rotating sphere with no texture, under fixed illumination. The sphere rotates but the brightness pattern in the image does not change. The motion field coincides with optical flow when a moving curved surface undergoes pure translation, the illumination is uniform and the projection is orthographic. In real images these conditions are never strictly

satisfied but it is assumed that they hold locally.

2.2 The Optical Flow Constraint Equation

One of the most common assumptions in optical flow techniques is that the intensity of an image point does not change over time. This assumption can be stated formally as

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \quad (2.1)$$

Expanding the right hand side of Equation(2.1) in a Taylor series [4] and keeping terms up to the first order we obtain the gradient constraint equation

$$\nabla E(\mathbf{x}, t) \cdot \mathbf{v} + \partial E / \partial t = 0 \quad (2.2)$$

where $\nabla E = (\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y})$, $\mathbf{x} = (x, y)^T$ and $\mathbf{v} = (u, v)^T$, $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$.

Equation(2.2) gives only the component of the image velocity in the direction of the intensity gradient. The component perpendicular to this direction can not be recovered unless more information is available. The normal velocity can be written as

$$\mathbf{v}_n = s(\mathbf{x}, t) \mathbf{n}(\mathbf{x}, t) \quad (2.3)$$

where

$$s(\mathbf{x}, t) = \frac{-E_t(\mathbf{x}, t)}{\|\nabla E(\mathbf{x}, t)\|} \quad \text{and} \quad \mathbf{n}(\mathbf{x}, t) = \frac{(\nabla E(\mathbf{x}, t))^T}{\|\nabla E(\mathbf{x}, t)\|} \quad (2.4)$$

and where E_t is the partial derivative of the image intensity with respect to time. Equation(2.2) can be interpreted as a line in velocity space as depicted in Figure(2.1). Any velocity in this line will satisfy the equation. More constraint lines are needed to determine the full velocity vector. This problem, known as the problem, is illustrated in Figure(2.2). In the presence of sufficient intensity structure, like a corner,

the full velocity vector can be estimated. If the surface is homogeneous (having no texture), or contains texture with a single orientation only the normal velocity can be recovered.

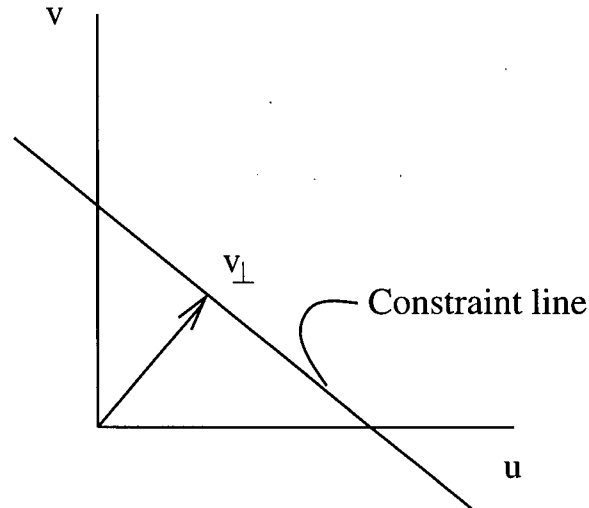


Figure 2.1: The optical flow constraint equation determines a line in velocity space. The normal velocity is the projection of the velocity vector onto the direction of the intensity gradient.

2.3 Multiple Light Source Optical Flow

There are many ways to overcome the aperture problem. Horn and Schunck [4] use the standard optical flow equation, Equation(2.2), together with a global smoothness constraint (regularization). Lucas and Kanade[5] combine optical flow constraint equations over a small window. Other methods combine second-order differential methods with global or local constraints. Barron, Fleet and Beauchemin[6], Beauchemin and Barron[7], and Chapter 3, provide a survey of the different techniques used to estimate optical flow.

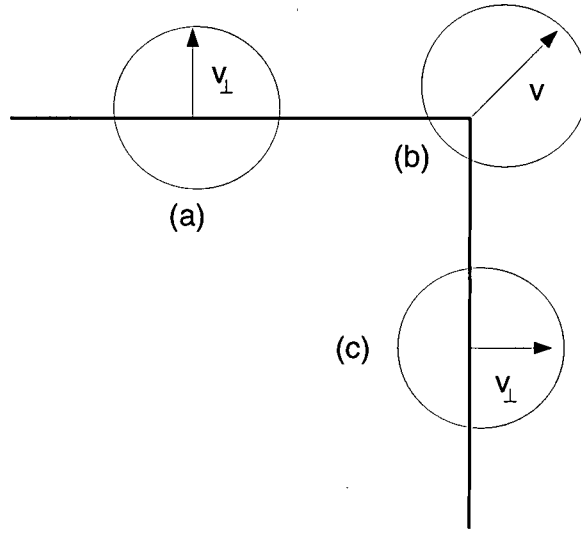


Figure 2.2: For apertures (a) and (c) only the normal component of the velocity can be recovered. The intensity structure at (b) allows the recovery of the full velocity vector.

The approach followed in this work is to use multiple light sources [3] in order to get additional constraint equations. For three light sources, we obtain,

$$\begin{aligned}
 E_{1x} u + E_{1y} v + E_{1t} &= 0 \\
 E_{2x} u + E_{2y} v + E_{2t} &= 0 \\
 E_{3x} u + E_{3y} v + E_{3t} &= 0
 \end{aligned} \tag{2.5}$$

this equation can be written as

$$\mathbf{A} \mathbf{x} = \mathbf{b} \tag{2.6}$$

where $\mathbf{x} = (u, v)^T$, $\mathbf{b} = -(E_{1t}, E_{2t}, E_{3t})^T$ and

$$\mathbf{A} = \begin{bmatrix} E_{1x} & E_{1y} \\ E_{2x} & E_{2y} \\ E_{3x} & E_{3y} \end{bmatrix} \tag{2.7}$$

The standard least squares solution, \mathbf{x} , is given by,

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (2.8)$$

The solution is unique provided that the rank of \mathbf{A} is 2. Only when the optical flow and the motion field coincide is the system of Equations (2.5) in the same unknowns[3].

2.4 Error measures

The fact that three light sources over-determine the system of Equations(2.5) can be used to test the validity of the solution. A relative error term can be defined that takes into account the fit to the model (how good is the estimate of \mathbf{x} in the equation $\mathbf{Ax} = \mathbf{b}$) and the redundancy of the measurements. Reference[3] defines this error as

$$\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} = \frac{\|\mathbf{b} - \mathbf{Ax}\|}{\|\mathbf{b}\|} \quad (2.9)$$

where $\|\cdot\|$ denotes the l_2 norm. The error is zero if $\mathbf{Ax} = \mathbf{b}$ in the case of three non-redundant measurements or when one of the intensity triplets ($E_{ix}, E_{iy}, E_{it}, i = 1, 2, 3$) is zero and equation $\mathbf{Ax} = \mathbf{b}$ is reduced to a two light source case.

Another desirable feature of the system of Equations (2.5) is to be well-conditioned so that the solution is stable. A measure of the stability of \mathbf{x} is the condition number of \mathbf{A} , $\kappa(\mathbf{A})$, which can be calculated as[8]

$$\kappa(\mathbf{A}) = \sqrt{\kappa(\mathbf{A}^T \mathbf{A})} = \frac{\sqrt{\lambda_{max}}}{\sqrt{\lambda_{min}}} \quad (2.10)$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of $\mathbf{A}^T \mathbf{A}$, respectively. We need $\kappa(\mathbf{A})$ to be small for the solution to be stable.

Chapter 3

Summary of Optical Flow

Methods

Barron, Fleet and Beauchemin[6] and Beauchemin and Barron[7] provide a survey of the different classes of optical flow techniques. Most methods lie in one of the following classes: differential methods, matching methods and frequency-based methods. Even though these methods are based on very different ways of estimating the optical flow, there are some common processing stages [6]: a prefiltering or smoothing stage using low-pass/band-pass filters to extract features of interest and to enhance the signal-to-noise ratio, the estimation of spatiotemporal derivatives or the correlation of image regions and the integration of these measurements to estimate the optical flow field. The following sections will summarize these different approaches.

3.1 Differential Methods

Differential methods compute optical flow from the spatial and time derivatives of the image brightness values. First-order differential methods assume that the

brightness of any object point in the image is constant over time. This assumption translates into the gradient constraint equation, equivalent to Equation(2.2)

$$E_x u + E_y v + E_t = 0 \quad (3.1)$$

Second-order differential methods use second order derivatives to constrain the 2D velocity field. Differentiating the gradient constraint equation we obtain

$$\begin{aligned} E_{xx}(\mathbf{x}, t)u + E_{yx}(\mathbf{x}, t)v + E_{tx}(\mathbf{x}, t) &= 0 \\ E_{xy}(\mathbf{x}, t)u + E_{yy}(\mathbf{x}, t)v + E_{ty}(\mathbf{x}, t) &= 0 \end{aligned} \quad (3.2)$$

This assumption translates into the conservation of the gradient of the brightness values over time, $d\nabla E/dt = 0$. The constraints given by Equation(3.2) can be used alone or combined with Equation(3.1) to give an over-determined system. Second-order derivatives cannot be estimated as accurately as first-order derivatives; therefore. Thus second-order methods often give sparser and less accurate optical flow fields[6].

Gradient-based algorithms have the advantage of requiring simple, local calculations and providing a dense optical flow field. The major disadvantage is that these methods require numerical differentiation which is an ill-posed problem. The basic assumption made in these methods is that the spatial and temporal sampling steps are small compared to the image features and motion amplitudes. If the brightness changes too rapidly on the scale of given by the discretization step, the accuracy of the derivative estimation decreases[10]. When aliasing effects cannot be avoided, differential techniques can be applied together with a multi-scale decomposition scheme. Reliable estimates at a coarser resolution, where aliasing is less severe, can be used as initial guesses for finer scales. Kearney, Thompson and Boley[11] analyzed the sources of error in gradient-based methods.

Horn and Schunck[4] combined the gradient Equation(3.1) with a global smoothness constraint. They minimize

$$\int_{\mathcal{D}} (E_x v + E_y u + E_t)^2 + \lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy \quad (3.3)$$

where the integration is over the domain of interest \mathcal{D} . The parameter λ gives the relative importance of the smoothness constraint and the fit to the data term. A Gauss-Seidel iterative method is used to estimate the velocity components.

Another way to solve the aperture problem is to combine constraints over a small neighborhood. Lucas and Kanade[5] and others[11],[12] implemented a least-squares fit of local first-order constraints to a constant \mathbf{v} over a small window Ω . They minimize

$$\sum_{\mathbf{x} \in \Omega} W^2(\mathbf{x}) [\nabla E(\mathbf{x}, t) \cdot \mathbf{v} + E_t(\mathbf{x}, t)]^2 \quad (3.4)$$

The function $W(\mathbf{x})$ gives more influence to constraints at the center of the neighborhood than those at the edges. The solution to this equation is given by

$$\mathbf{v} = [\mathbf{A}^T \mathbf{W}^2 \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{W}^2 \mathbf{b} \quad (3.5)$$

where

$$\mathbf{A} = [\nabla E(\mathbf{x}_1), \dots, \nabla E(\mathbf{x}_n)]^T \quad (3.6)$$

$$\mathbf{W} = \text{diag}[W(\mathbf{x}_1), \dots, W(\mathbf{x}_n)] \quad (3.7)$$

$$\mathbf{b} = -(E_t(\mathbf{x}_1), \dots, E_t(\mathbf{x}_n))^T \quad (3.8)$$

Simoncelli, Adelson and Heeger[12] proposed a probabilistic formulation of the gradient approach combined with a multi-scale pyramid decomposition. They calculate the conditional probability of the velocity \mathbf{v} given the gradient of the intensity ∇E , $P(\mathbf{v}|\nabla E)$. The errors in the estimation of the derivatives and the violation

to the brightness constancy assumption are modeled as Gaussian terms that are added to the gradient constraint equation. The aperture problem is solved by combining constraints over a small neighborhood.

Nagel[13] was one of the first to use second-order differential methods to estimate optical flow. Like Horn and Schunck, he used a global smoothness constraint, but took into account that a gray value transition might be due to an occluding edge. In this case the optical flow field should not be required to have a smooth variation. The functional that he minimized is designed in such a way that the smoothness requirement is retained only for the optical flow component perpendicular to strong gray value transitions.

Another second-order differential technique was implemented by Uras, Giroi, Verri and Torre[14]. They solve Equation(3.2) whenever the Hessian H of the intensity values is nonsingular. They divide the image into 8×8 pixel regions and they select the 8 estimates that best satisfy the constraint $\| \mathbf{M} \nabla E \| \ll \| \nabla E_t \|$ where $\mathbf{M} = (\nabla \mathbf{v})^T$. Of these estimates they choose the one with the smallest condition number of the Hessian matrix.

Battiti, Amaldi and Koch[10] proposed a gradient-based method combined with an adaptive coarse-to-fine strategy based on an error estimate. Reliable estimates at one scale are used as first estimates at a finer scale. The Horn and Schunck algorithm is used to estimate the optical flow at each scale. As a first step they build a Laplacian pyramid using three or four different spatial resolution levels. The adaptive discretization is implemented using an inhibition flag associated with each point in the pyramid. Starting at the lowest resolution, an error estimate is calculated for each pixel. If the error is below a certain threshold the grid point and its four neighboring points at the next (finer) resolution will retain their value

of the optical flow. The value of the other pixels will be obtained using bilinear interpolation. The coarsest spatial grid used should match the maximum expected motion amplitude.

Verri, Giroi and Torre[15] studied the changes in the brightness pattern in terms of elementary deformations. They combined the gradient constraint equation with stationary functions of the image brightness deduced for the cases of uniform expansion, rotation and shear. They compare the optical flow obtained in these cases and using second order constraints (Equation(3.2)).

Multiple light source optical flow makes no assumptions about the reflectance properties of the surface of the moving object. When reflectance properties are known the brightness constancy assumption can be relaxed. Hsu[16] starts by differentiating the image irradiance equation, to obtain a constraint equation given by

$$E_x \frac{dx}{dt} + E_y \frac{dy}{dt} + E_t = R_u \frac{du}{dt} + R_v \frac{dv}{dt} + R_w \frac{dw}{dt} \quad (3.9)$$

where $R(u, v, w)$ is the scene radiance at point (x, y) and $\mathbf{n} = (u, v, w)^T$ is the surface normal. Combining the constraint equations of three light sources and the additional constraint over the surface normal, $\frac{d\|\mathbf{n}\|}{dt} = 0$, with a regularization technique, the author estimates the optical flow field for a moving sphere. For an arbitrary surface the surface normal could be first estimated using photometric stereo[17].

Markandey and Flinchbaugh[18] and Ohta[19] applied multi-spectral first-order constraints to the optical flow problem. They use multiple sensors to obtain multi-spectral images with the expectation that each spectral band provides independent constraints. The independence of the constraints is a crucial factor to assure that the system is well-conditioned.

3.2 Region-based matching methods

When numerical differentiation is not accurate because of noise or aliasing, matching techniques might be more suitable to estimate optical flow. The basic idea is to define velocity as the shift, \mathbf{d} , that gives the best fit between subsequent image regions. Matching image regions is often done by maximizing a similarity measure. Little, Bülthoff and Poggio[20] look for a discrete displacement \mathbf{v} that minimizes:

$$\int [\phi(E(\mathbf{x}, t), E(\mathbf{x} + \mathbf{v}\delta t)) + \lambda(\nabla \mathbf{v} \nabla \mathbf{v}^T)] d\mathbf{x} \quad (3.10)$$

where ϕ is a comparison function. This function may include an edge detection step or some image transformation, for example, convolution by $\nabla^2 G$. Equation(3.10) is approximated by summing over overlapping patches of a certain diameter. A commonly used similarity measure is the sum-of-squared differences (SSD)

$$SSD_{1,2}(\mathbf{x}, \mathbf{d}) = \sum_{j=-n}^n \sum_{i=-n}^n W(i, j) [E_1(\mathbf{x}_{ij}) - E_2(\mathbf{x}_{ij} - \mathbf{d})]^2 \quad (3.11)$$

where $W(i, j)$ is a window function. In reference[21], Anandan presented a framework for the estimation of optical flow based on three components: spatial frequency decomposition using a Laplacian pyramid, a local, parallel match criterion within each scale using a SSD matching strategy, and a control strategy to integrate the results. The scale-based separation allows velocity estimation for large displacements. A global smoothness constraint is used to propagate reliable displacements to less reliable neighbors. The reliability measure is based on the principal curvatures of the SSD surface.

Singh[22] implemented a two-stage matching method. The first stage is the computation of SSD values for a sequence of three band-pass filtered images E_{-1} , E_0 and E_{+1}

$$SSD_0(\mathbf{x}, \mathbf{d}) = SSD_{0,1}(\mathbf{x}, \mathbf{d}) + SSD_{0,-1}(\mathbf{x}, -\mathbf{d}) \quad (3.12)$$

where $SSD_{i,j}$ is given by Equation(3.11). Adding two-frame SSD surfaces tends to average out noise. SSD_0 is then converted into a probability distribution. The velocities are estimated as the mean over the distribution. The eigenvalues of the covariance matrix associated with the velocity estimates are used as a confidence measure. The second stage is to propagate the velocity estimates using weighted-least-squares over a small neighborhood.

The major disadvantage of matching techniques is that they are sensitive to ambiguities among the features to be matched.

3.3 Frequency-based methods

Differential methods and matching methods are not appropriate in highly textured images because the direction of the brightness gradient does not vary smoothly and because there are too many features to track. In this case frequency-based methods are more appropriate. These methods are based on the use of velocity-tuned filters. The Fourier transform of a translating 2D intensity pattern $E(\mathbf{x}, t) = E(\mathbf{x} - \mathbf{v}t, 0)$ is given by

$$E(\mathbf{k}, \omega) = E_0(\mathbf{k})\delta(\omega + \mathbf{k} \cdot \mathbf{v}) \quad (3.13)$$

where $E_0(\mathbf{k})$ is the Fourier transform of $E(\mathbf{x}, 0)$, ω is the temporal frequency and \mathbf{k} is the spatial frequency. The motion constraint equation becomes $\mathbf{k} \cdot \mathbf{v} + \omega = 0$ in frequency space. All nonzero power associated with a translating 2D pattern lies in a plane through the origin in frequency space[6]. If the motion in a small region of the image can be approximated by a translation, the velocity can be estimated by finding this plane.

Image motion can be characterized by orientation in space and time and can be detected using spatiotemporally oriented filters[23]. One technique for extracting spatiotemporal energy is Gabor filtering. An example of an odd 3D Gabor filter is

$$g(x, y, t) = \frac{1}{\sqrt{2\pi^3/2}\sigma_x\sigma_y\sigma_t} \times e^{\left(-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2}\right)\right)} \times \sin(2\pi\omega_{x0}x + 2\pi\omega_{y0}y + 2\pi\omega_{t0}t) \quad (3.14)$$

where $(\omega_{x0}, \omega_{y0}, \omega_{t0})$ is the frequency for which the filter gives its greatest output and $(\sigma_x, \sigma_y, \sigma_t)$ is the width of the Gaussian window. Heeger[24] used 12 Gabor filters at each spatial scale tuned to different orientations and different temporal frequencies. The image velocity is estimated as a least-squares-fit of the filter energies to a plane in frequency space.

Phase-based methods are another type of techniques that use velocity-tuned filters. Fleet and Jepson[25] define velocity in terms of the temporal evolution of contours of constant phase. Band-pass filters are used to decompose the image according to scale, speed and orientation.

Chapter 4

Implementation

4.1 Experimental Setup

The results presented in this thesis were obtained using the Calibrated Imaging Facility (CIF) of the Laboratory for Computational Intelligence at UBC. The CIF consists of an optical bench (4 ft x 8 ft) with mounting hardware for positioning and motion of cameras, light sources and test objects. The equipment that was used includes a Sony DXC-755 3 CCD RGB camera with Fujicon 10-120mm (manual) zoom lens, three Newport MP-1000 Moire (white light) projectors with Nikon lenses and spectral filters, two linear Daedal rail tables plus one rotational, and associated controllers, motors, mounting hardware and power supplies. The CIF is integrated with a C40 network and DataCube image processing DigiColor and MaxVideo-200 subsystems.

This work requires multiple images of a scene acquired simultaneously under different conditions of illumination. This is achieved by multiplexing the spectral dimension. Three light sources with color filters, one red, one green and one

blue, illuminate the work place from different directions. The filters are the Newport FS-225 set manufactured by Corion (Holliston, Mass.) and are Corion parts CA500(blue), CA550(green), and CA600(red). There is negligible overlap in the visible spectrum between the red-light source and either the green-light source or the blue-light source. There is a small overlap between the green and the blue sources for wavelengths between 500-520 nm (see [17] for more details). The Sony 3 CCD camera is used to acquire three separate black and white images simultaneously. These images correspond to the different conditions of illumination.

4.2 The C40 processors

The Texas Instruments TMS320C40 (C40) floating point processors are specifically designed to meet the requirements of parallel processing and real-time applications. They provide on-chip hardware to facilitate interprocessor communication and concurrent I/O without degrading CPU performance. The main features of the C40 processors are[26]:

- Six communication ports for high-speed interprocessor communication.
- Six-channel DMA coprocessor for concurrent I/O and CPU operation, thereby maximizing sustained CPU performance by alleviating the CPU of burdensome I/O.
- High performance DSP CPU capable of 275 MOPS, a peak performance of 50 MFLOPS, and a total communication bandwidth of 320 Mbytes/s.

- Two identical external data and address buses supporting shared memory systems and high data rate, single-cycle transfers.
- On-chip analysis module supporting efficient, state-of-the-art parallel processing debug.
- Separate internal program, data and DMA coprocessor buses for support of massive concurrent I/O of program and data throughput, thereby maximizing sustained CPU performance.

4.2.1 Communication Ports

The six communication ports offload interprocessor communication from the memory buses increasing throughput. The main features of the communication ports are:

- Transfer rate of 20 Mbytes/s for each port.
- Simple processor to processor bi-directional communications.

4.2.2 DMA Coprocessor Benefits

The six channels of the on-chip direct memory access (DMA) coprocessor can read from or write to any location in the processor's memory map without interfering with the operation of the CPU. This allows interfacing to slow external memories and peripherals without reducing the throughput. The DMA coprocessor contains its own address generators, source and destination registers, and transfer counter. A key feature of the DMA coprocessor is to automatically reinitialize each channel following a data transfer.

4.2.3 Memory organization

The total addressable memory of the C40 is 16 Gbytes. Program memory (on-chip RAM or ROM and external memory) as well as registers affecting timers, communication ports, and DMA channels are contained in this space. Each RAM and ROM block is capable of supporting two accesses in a single cycle. A 512-byte instruction cache is provided to store often repeated sections of code reducing the number of off-chip accesses.

4.2.4 Internal and external buses

Separate buses allow for parallel program fetches, data accesses and DMA accesses: These buses connect all of the physical spaces supported by the C40. The C40 provides two identical external interfaces: the global memory interface and the local memory interface. Both buses can be used to address external program/data memory or I/O space.

4.3 Processing Scheme

Six C40 processors are connected in a network with a Sun workstation as a host. Figure(4.1) shows the topology of the network. Figure(4.2) shows the processing scheme. Three black and white images corresponding to the different conditions of illumination are simultaneously acquired by the camera and sent to the frame buffer in node 5. All input images were quantized to 8 bits-per-pixel. The images are segmented horizontally into four smaller images and sent to nodes 1-4. The least squares solution for the velocity components is calculated in parallel in these four

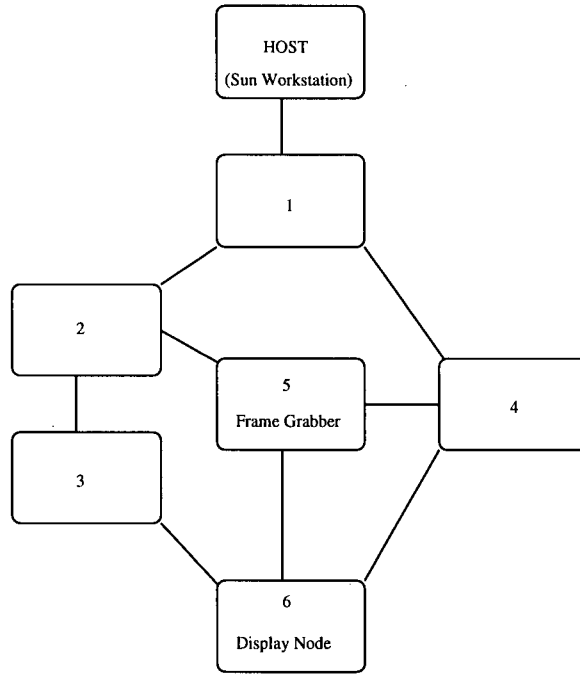


Figure 4.1: Topology of the C40 network.

nodes and the optical flow components are sent to node 6. In this node the flow components are encoded using a color encoding and displayed in a color monitor. Hue encodes direction and brightness encodes magnitude.

The derivatives were estimated using three different schemes. (1) Central differences were calculated using a $3 \times 3 \times 3$ cube of brightness values. The derivatives at any point were estimated using the mask $1/2(-1, 0, 1)$. Figure (4.3) shows three frames corresponding to times $t = k - 1$, $t = k$ and $t = k + 1$. The derivatives E_x , E_y and E_t at pixel (i, j) at time $t = k$ are given by

$$E_x(i, j, k) = \frac{1}{2}(E_{i+1jk} - E_{i-1jk}) \quad (4.1)$$

$$E_y(i, j, k) = \frac{1}{2}(E_{ij+1k} - E_{ij-1k}) \quad (4.2)$$

$$E_t(i, j, k) = \frac{1}{2}(E_{ijk+1} - E_{ijk-1}) \quad (4.3)$$

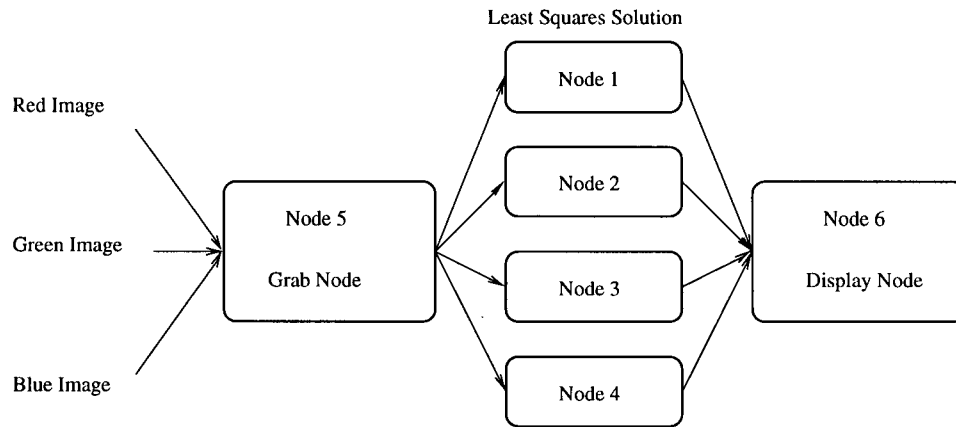


Figure 4.2: Processing scheme.

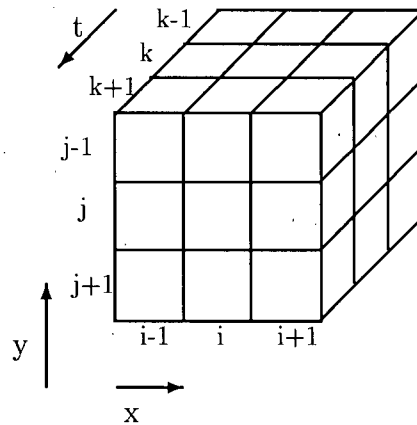


Figure 4.3: Estimation of partial derivatives using central differences.

(2) First differences were calculated using 2x2x2 cube with a mask $(-1, 1)$. In both cases spatial and temporal dimensions were treated symmetrically. (3) Four point (time) differences were calculated using a 3x3x5 cube, calculating central differences for the spatial dimensions and using a four point mask $1/12(-1, 8, 0, -8, 1)$ for the time derivatives. No presmoothing was performed on any of the images. Once the nine partial derivatives $(E_{ix}, E_{iy}, E_{it}, i = 1, 2, 3)$ are estimated, the value of the flow field is calculated following the least squares solution of Equation(2.8). Two checks have to be done to guarantee that the computation is not degenerate. First, at least two of the spatial gradients must be non-zero. This is implemented by requiring the magnitude of at least two of the spatial gradients to be greater than a threshold. Second, the rank of \mathbf{A} , given by Equation(2.7), must be two. This check is done by requiring at least two of the rows of \mathbf{A} to be linearly independent. Points that fail either of these two checks are assigned values $u = 0$ and $v = 0$.

4.4 Linearity of the motion stages

The linearity of the motion stages was tested for the velocity range used in the experiments. Figure(4.4) shows the measured velocity (pixels/sec) versus the rail speed (inches/sec). The error bars indicate an interval of two standard deviations centered at the average value. The distance was measured detecting the center of a color circle. The color circle was moved at different speeds. Two frames, separated by 6 frames, were grabbed with the frame grabber. The time between frames was measured using the C40 clock.

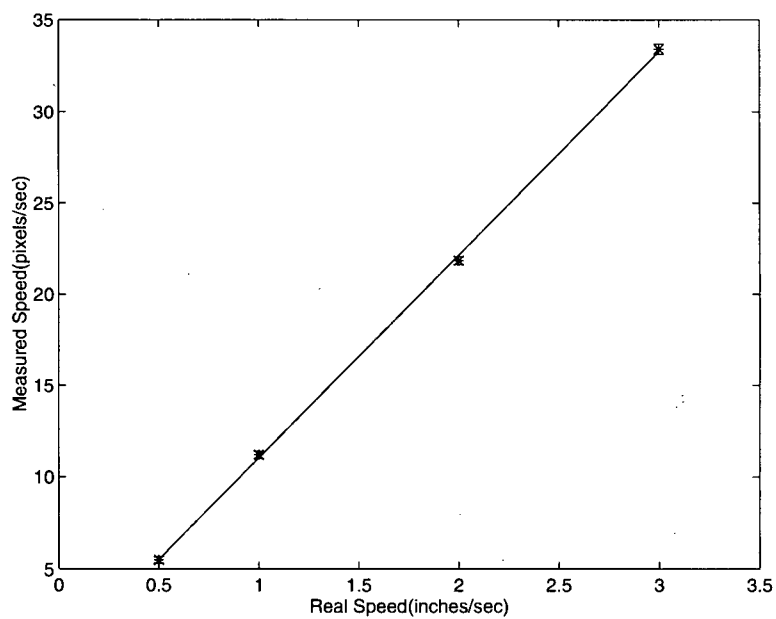


Figure 4.4: Behavior of the motion stages. The error bars indicate an interval of two standard deviations centered at the average value.

Chapter 5

Results

5.1 Stay-Puft Marshmallow Man

The system was tested with the Stay-Puft Marshmallow Man¹, shown in Figure(5.1). Most of its surface has a white semi-gloss finish. The eyes and the mouth are black, the collar and the hat are blue and the bow is red. The object was positioned on a platform that can be moved at different speeds in the x-y plane. The background was a black non-reflecting fabric. The threshold for the brightness gradient was set to 2.5 ($\|\nabla E\| \geq 2.5$). The processing rate depends on the gradient threshold and on the number of moving points. In our experiments, for an image size of 240x240 pixels, the rate was between 3.5 and 4.0 frames per second. The estimates are good at the points where the surface is smoothly shaded. The estimates are inaccurate at points where the image brightness varies very rapidly and where the brightness changes are too small. Figures(5.3)-(5.6) show an example of real-time motion detection for the object translating parallel to the image plane using the color

¹Stay-Puft Marshmallow Man is a registered trademark of Kenner Parker Toys International Inc.

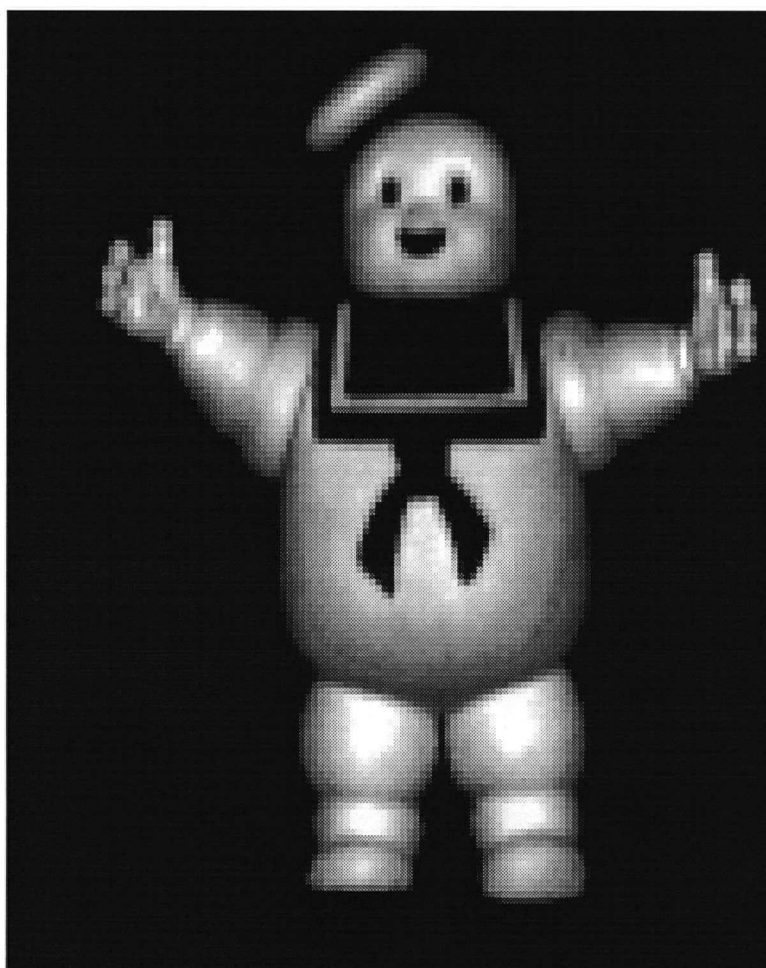


Figure 5.1: Object used in the experiments.

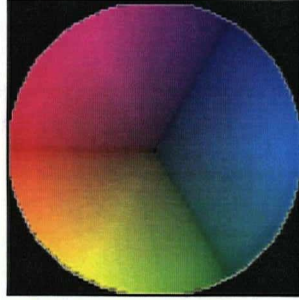


Figure 5.2: Optical flow encoding.

encoding showed in Figure(5.2). The white arrows show the direction of motion. The black pixels represent image points that were not moving or that were discarded in the calculation.

The behavior of the system was studied for motion restricted to the horizontal direction. Since differentiation operators are not rotationally invariant we expect these results to vary slightly for motion in different directions. The object was moved at different speeds and the average optical flow (over the entire image) was measured. Figure(5.7) shows the magnitude of the measured average speed versus the real speed in pixels per frame for an image of 240x240 pixels and for this image averaged to 120x120 pixels. The statistics were done for 50 – 100 frames at each speed. The error bars indicate an interval of two standard deviations centered at the average value. The behavior is linear for velocities up to 1.5 pixels/frame for the 240x240 case and up to 2.5 pixels/frame (measured in the original image size) for the averaged image. The average relative error is displayed in Figure(5.8). As expected the error is minimum at approximately 1 pixel/frame for the 240x240 case and at 2 pixels/frame for the 120x120 case. Figures(5.9)-(5.10) show the behavior of the system for the three different differentiation schemes described in section 4.3. The range of linear response is very similar in all cases. The relative error term

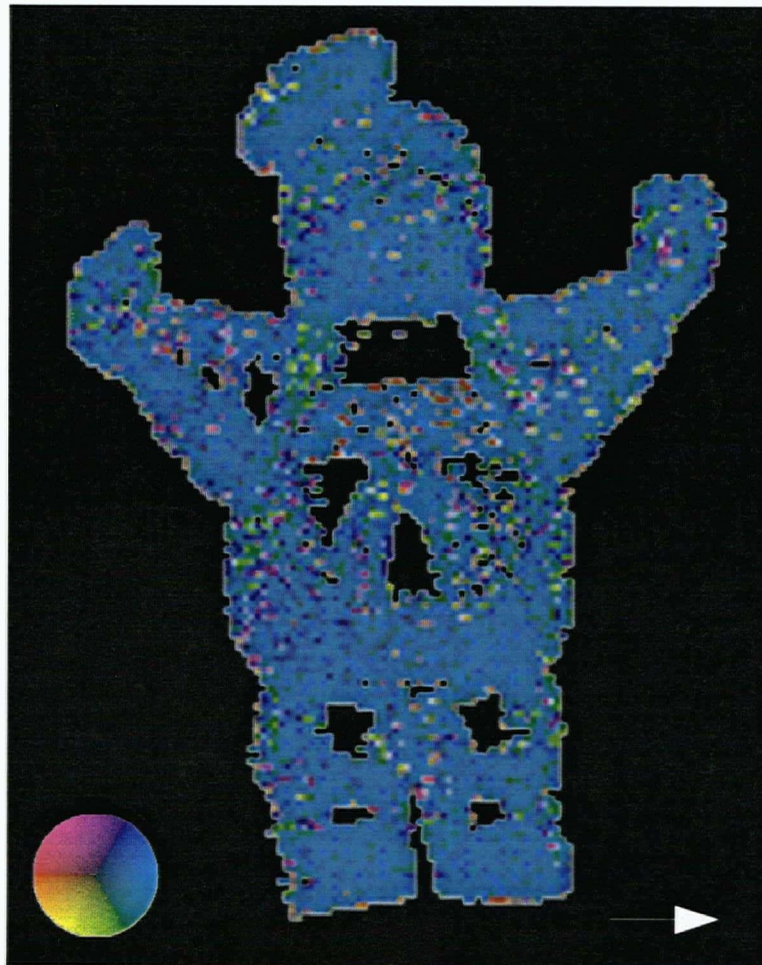


Figure 5.3: Example of real-time motion detection. Points that were not moving or that were discarded in the calculation are shown in black. The arrow shows the direction of motion. The color rosette shows the color encoding.

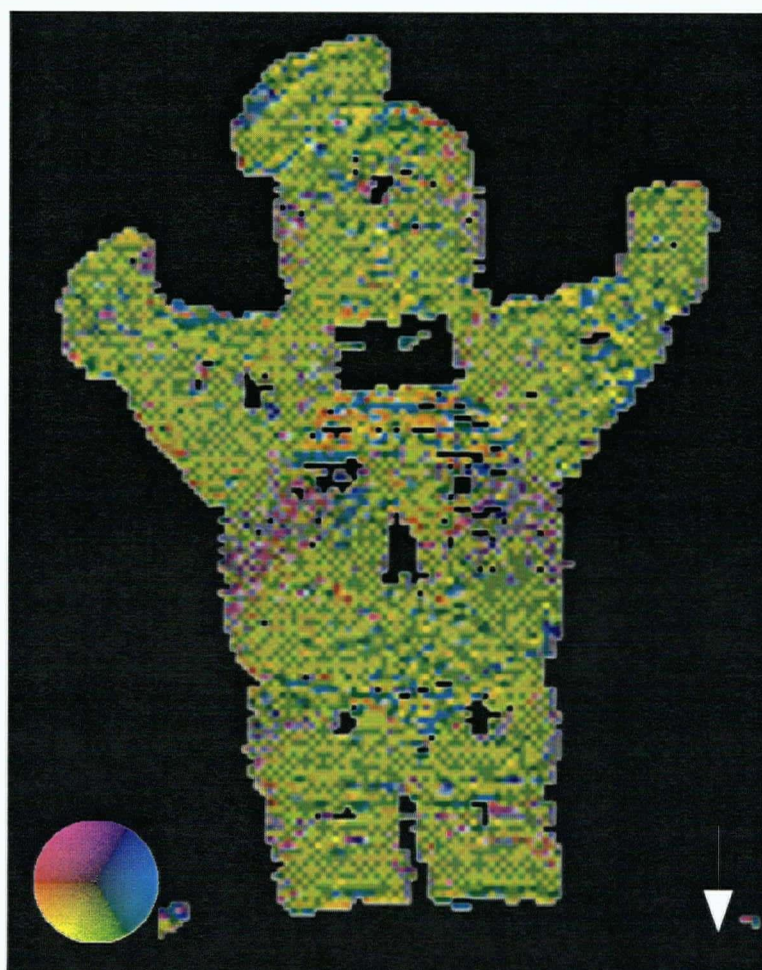


Figure 5.4: Example of real-time motion detection. Points that were not moving or that were discarded in the calculation are shown in black. The arrow shows the direction of motion. The color rosette shows the color encoding.

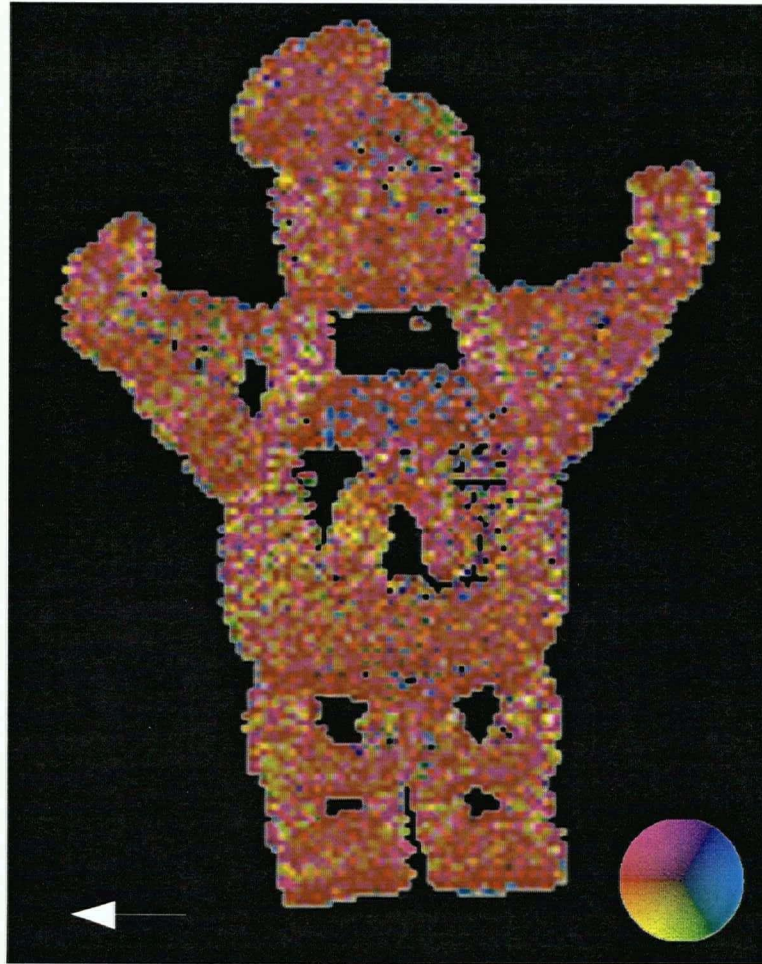


Figure 5.5: Example of real-time motion detection. Points that were not moving or that were discarded in the calculation are shown in black. The arrow shows the direction of motion. The color rosette shows the color encoding.

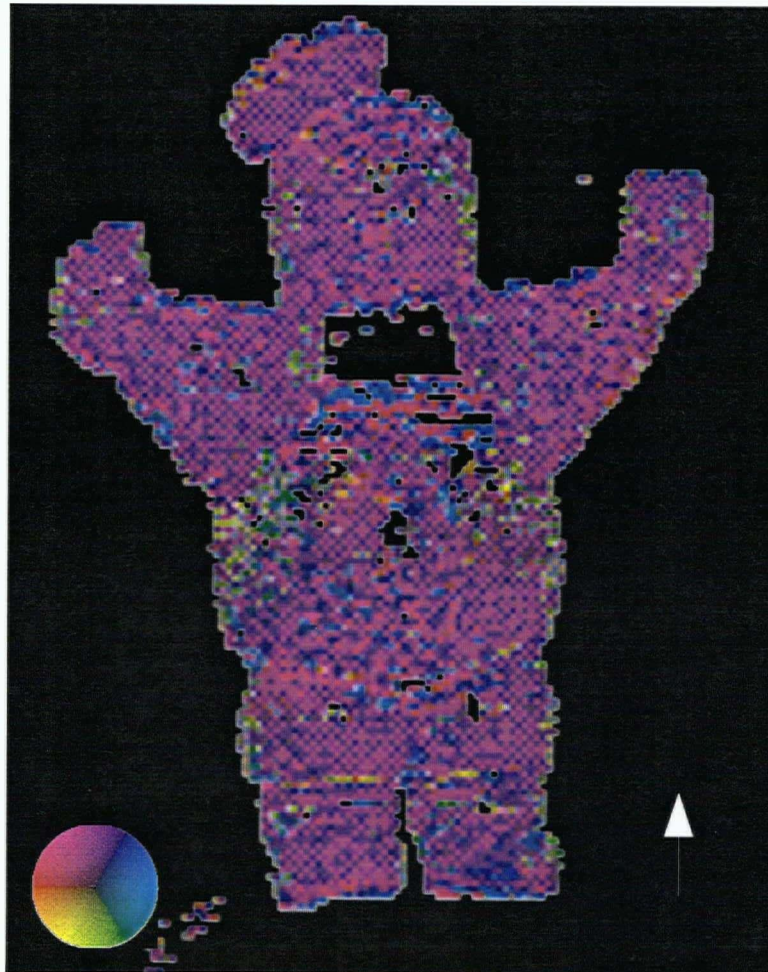


Figure 5.6: Example of real-time motion detection. Points that were not moving or that were discarded in the calculation are shown in black. The arrow shows the direction of motion. The color rosette shows the color encoding.

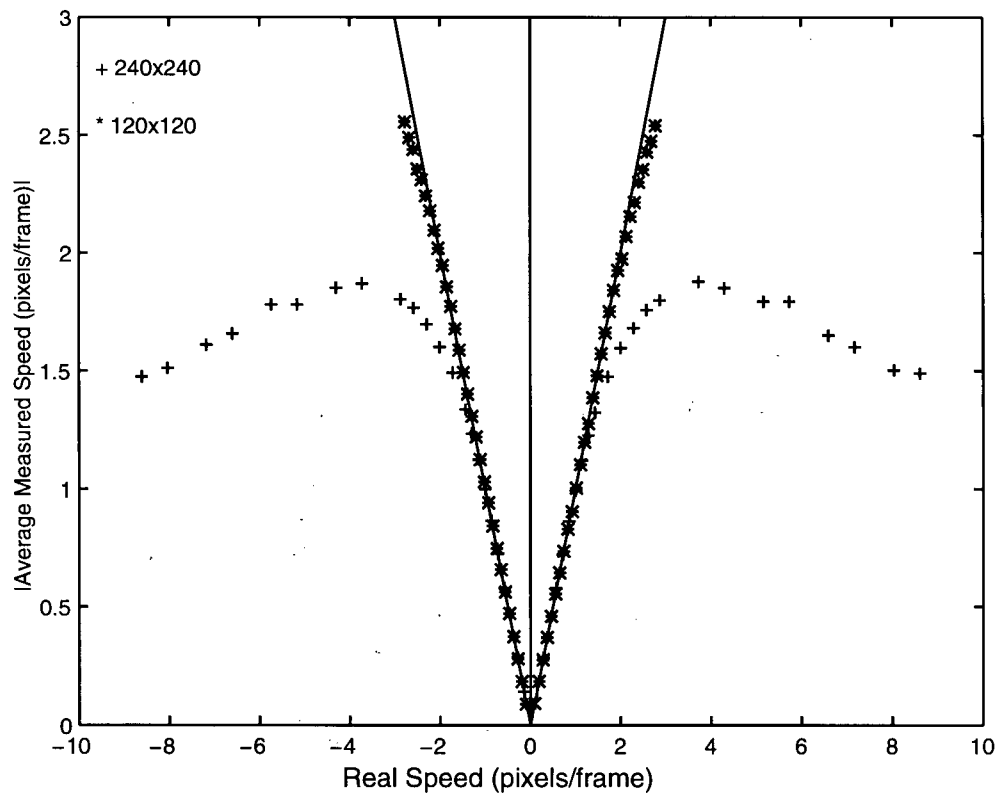


Figure 5.7: Magnitude of the average measured speed vs real speed in pixels/frame for a 240x240 image and for its average, 120x120; the derivatives were estimated using central differences.

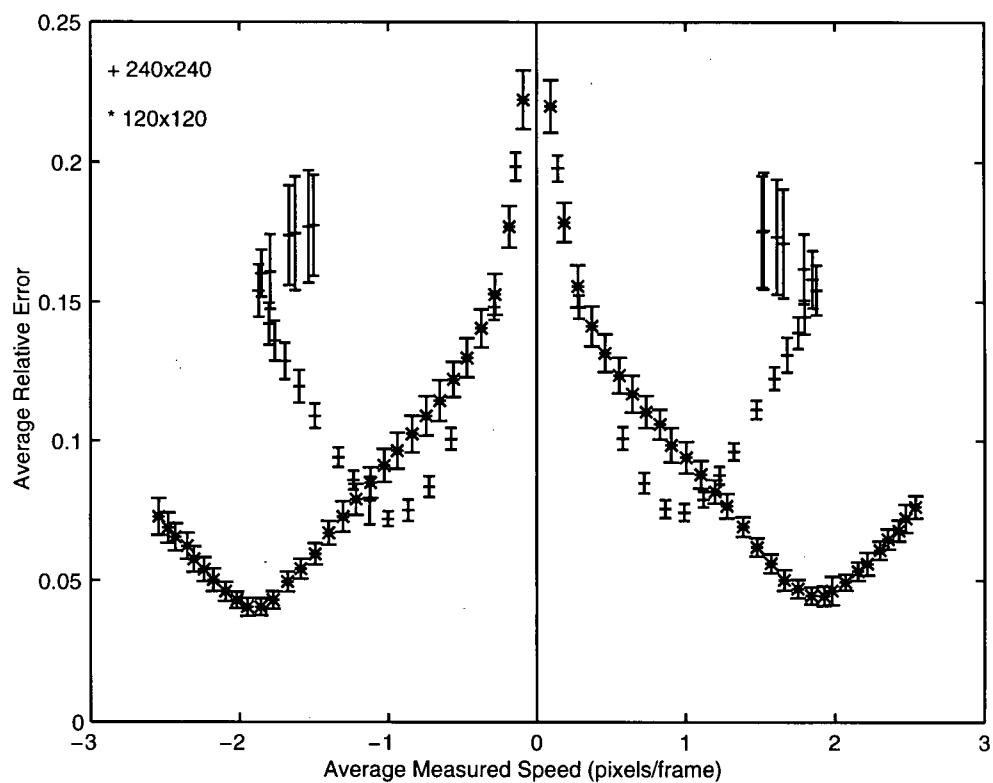


Figure 5.8: Relative error vs average measured speed for a 240x240 image and for its average, 120x120; the derivatives were estimated using central differences. The error bars indicate an interval of two standard deviations centered at the average value.

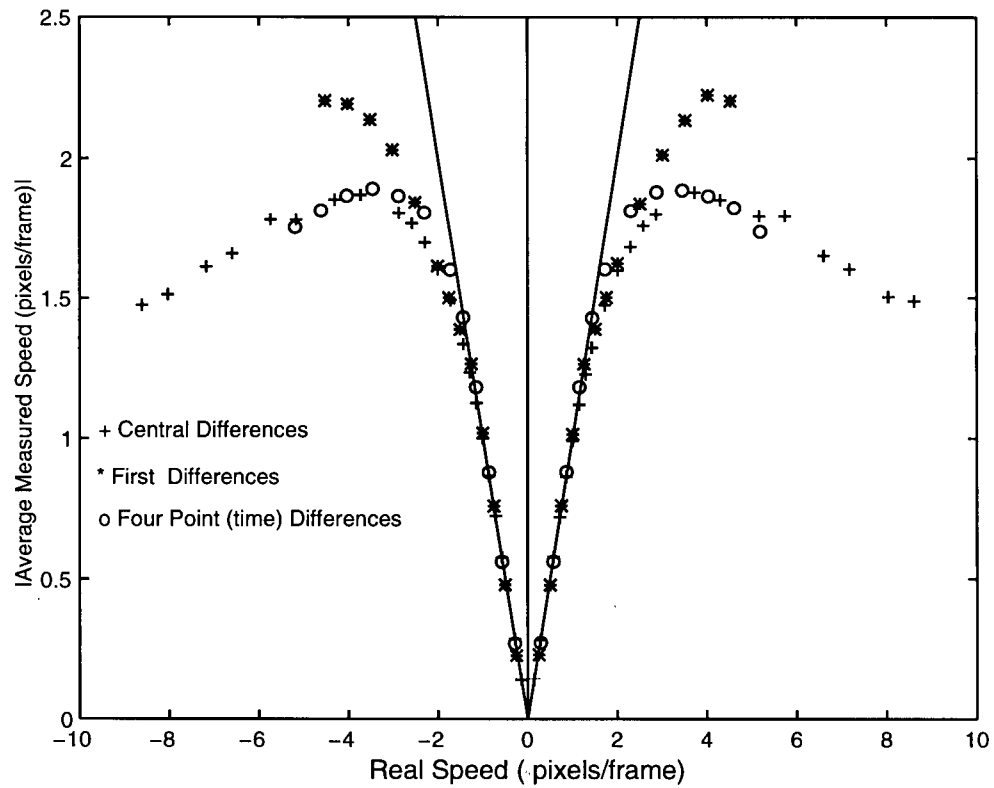


Figure 5.9: Magnitude of the average measured speed vs real speed in pixels/frame for different derivative estimation schemes.

is lower on average in the four point (time) differences, however, a difference in processing time of 200ms per frame suggests the use of central differences. All the experiments described in the following sections were done using central differences as the differentiation scheme.

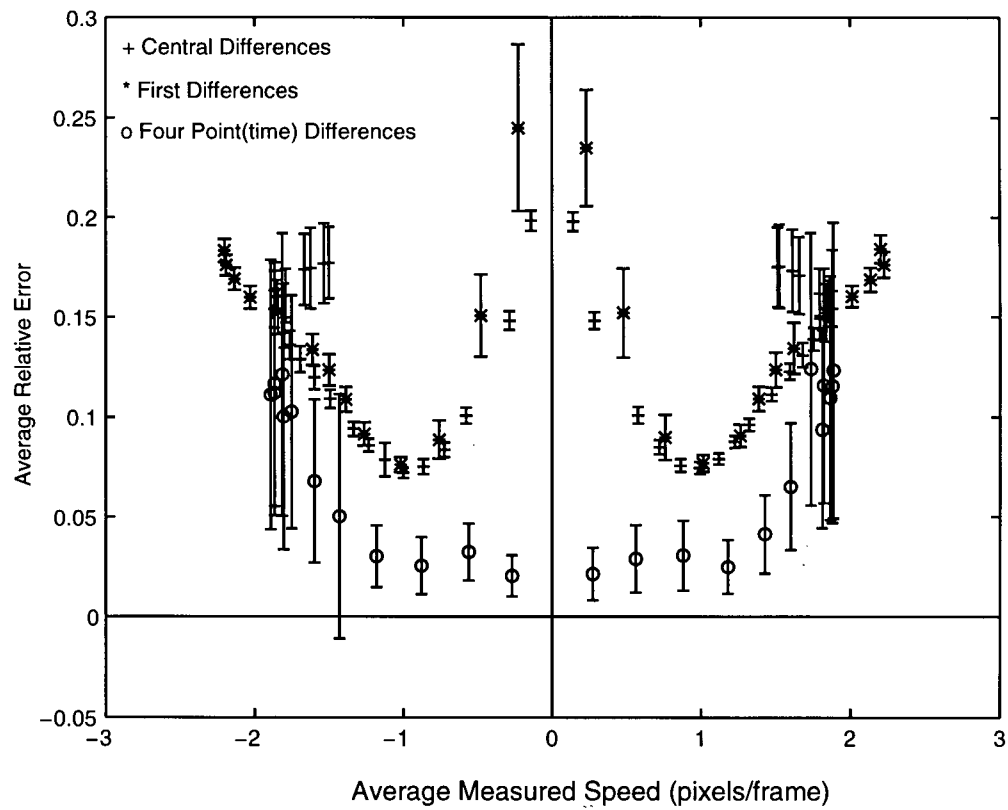


Figure 5.10: Relative error vs average measured speed for different derivative estimation schemes. The error bars indicate an interval of two standard deviations centered at the average value.

5.2 Expanding and contracting balloon

Just as a test of the algorithm for a deforming surface, a balloon was inflated and deflated in front of the camera. Figure(5.11) shows the balloon illuminated with the three light sources. Figure(5.12) and Figure(5.13) show two snap-shots of the video output for the expanding balloon and the contracting balloon respectively. The image size is 240x240 pixels and the processing rate was 4 frames/second. The velocity estimates are calculated on the well-shaded area between the edges and the center. The center is left mostly black because the intensity variation is very small and these points are discarded in the calculation.

A short sequence of the expanding balloon was grabbed and the images

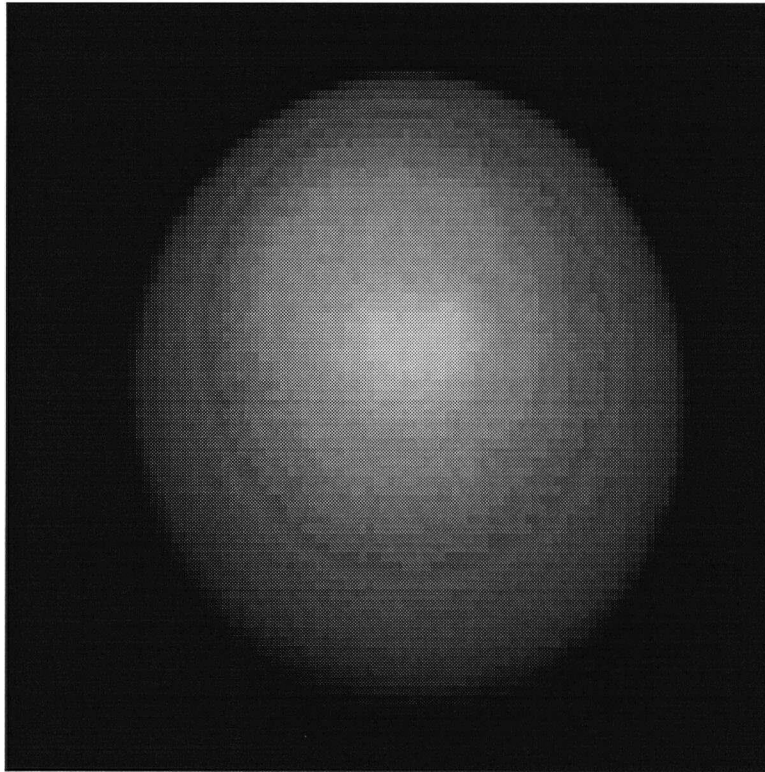


Figure 5.11: Balloon illuminated with three light sources.

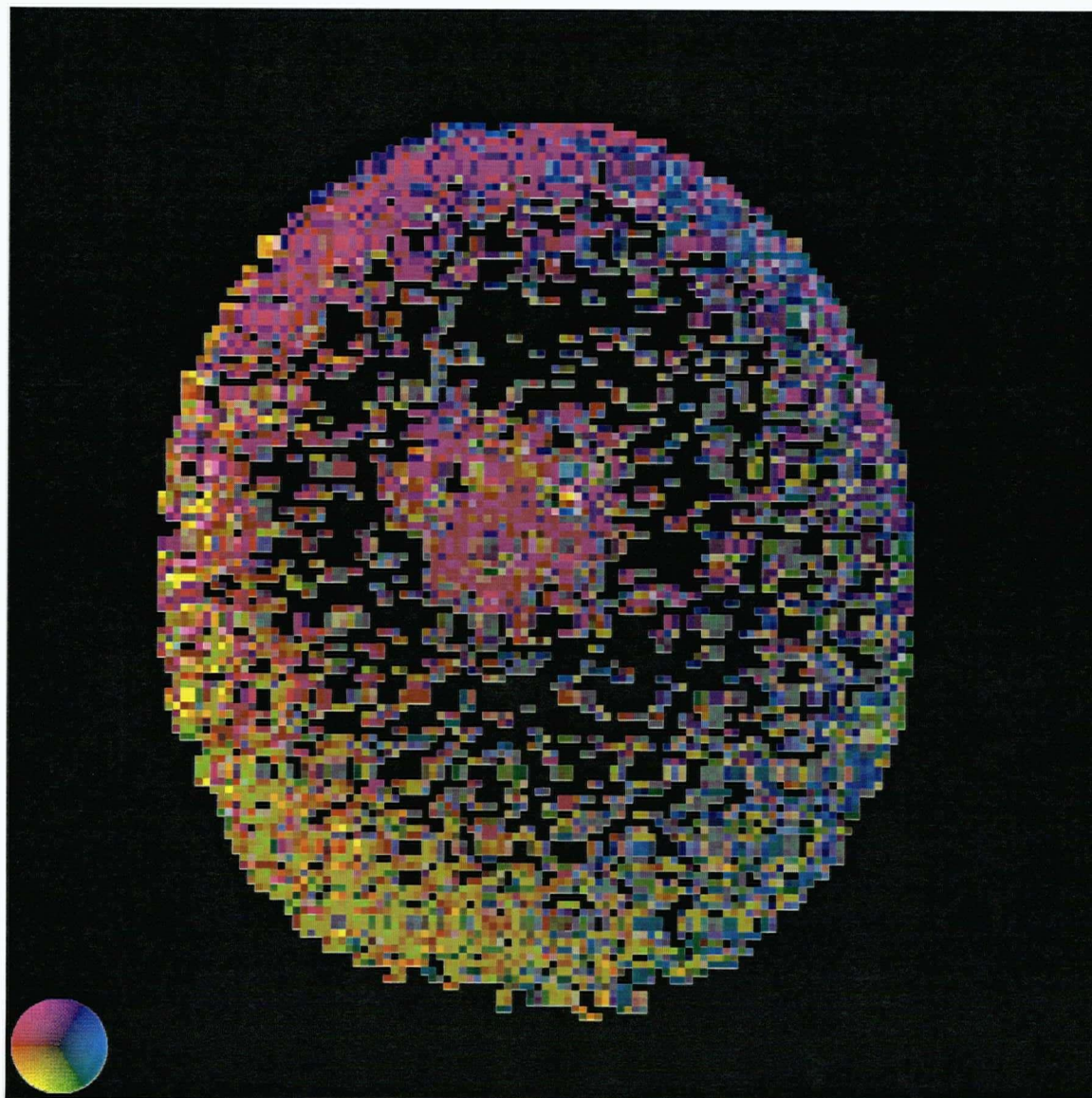


Figure 5.12: Real-time motion detection for an expanding balloon. Points that were not moving or that were discarded in the calculation are shown in black. The color rosette shows the color encoding.

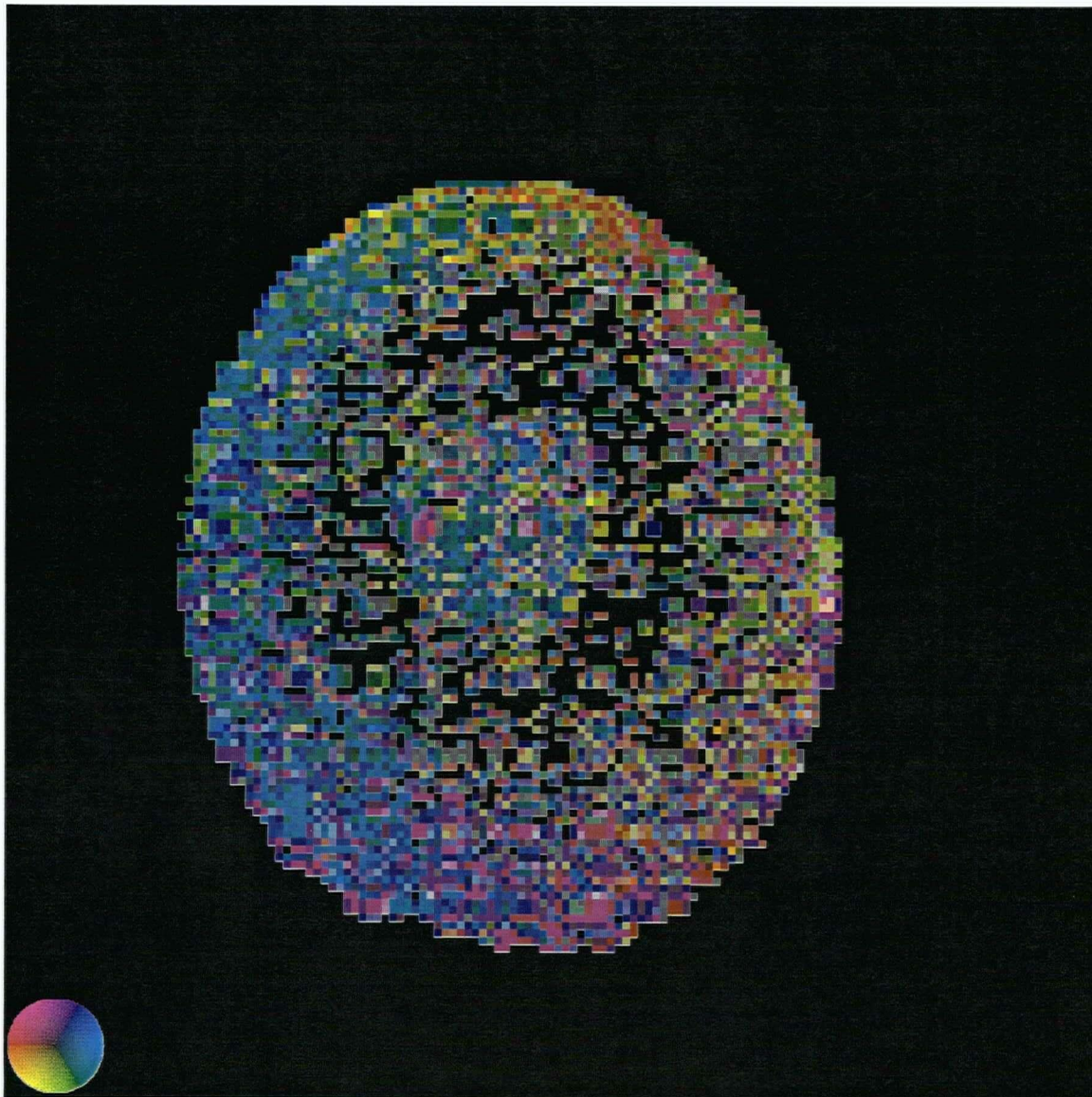


Figure 5.13: Real-time motion detection for a contracting balloon. Points that were not moving or that were discarded in the calculation are shown in black. The color rosette shows the color encoding.

were processed with a version of multiple light source optical flow running in Sun workstation. Figure(5.14) shows the optical flow; the flow field was sub-sampled every two pixels and it was scaled by two. The threshold for the magnitude of the brightness gradient was set to 2.5. Figures(5.15)-(5.16) show the relative error and the condition number. Large relative errors and large condition numbers are encoded as white and low values as black; the relative error is shown in a scale from 0 to 1, and the condition number is shown in a scale from 1 to 25. The center of the balloon has large relative errors and the edges have very large condition numbers. The specular reflections of the three light sources can be distinguished in Figure(5.15) as three white spots close to the center of the balloon.

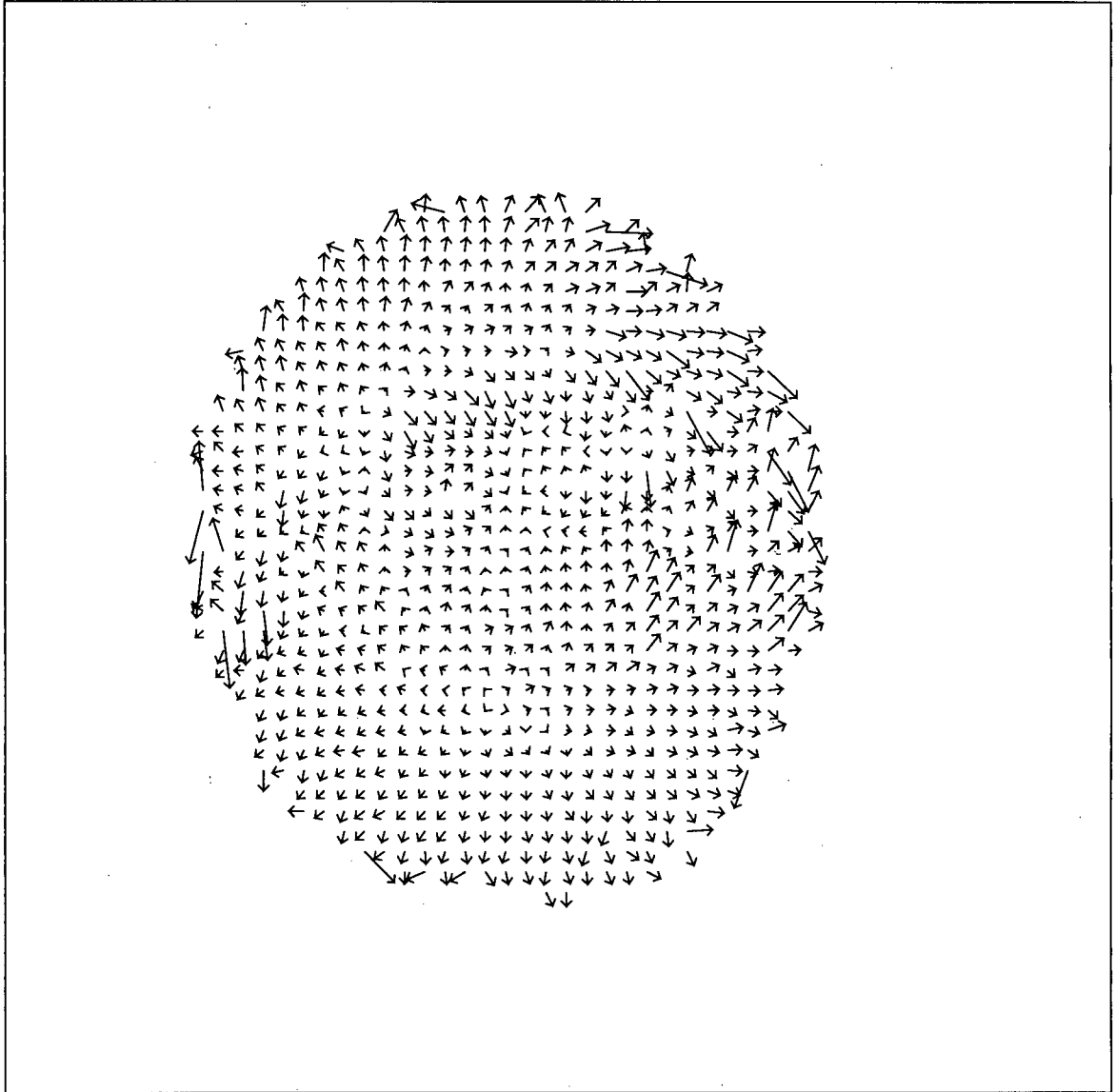


Figure 5.14: Multiple light source optical flow for a deforming balloon.

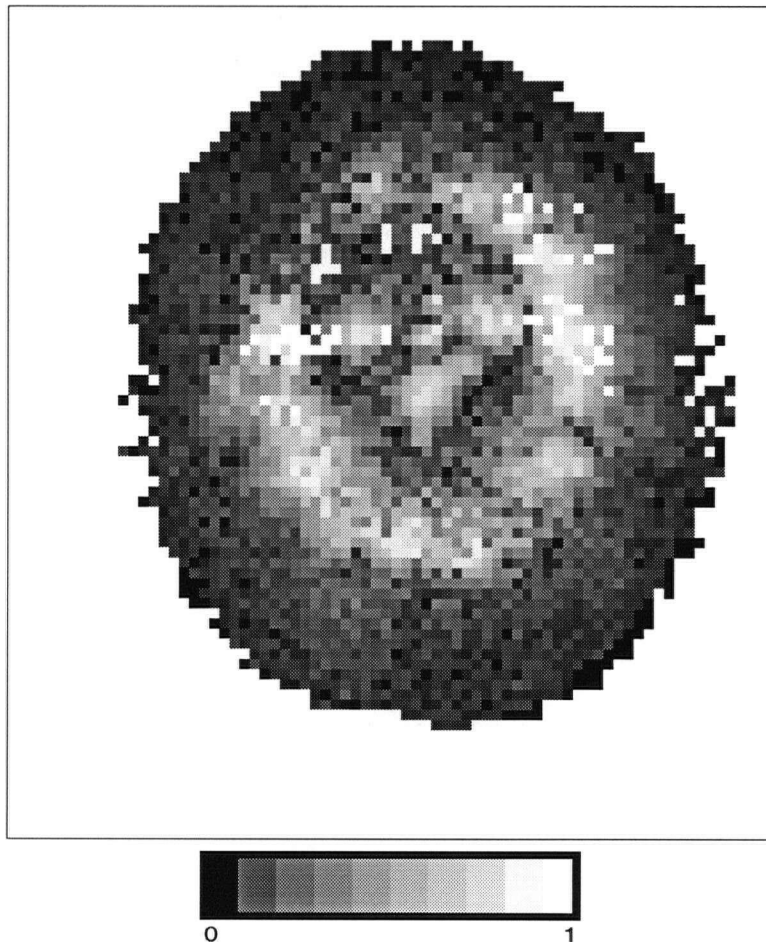


Figure 5.15: Relative error for the deforming balloon.

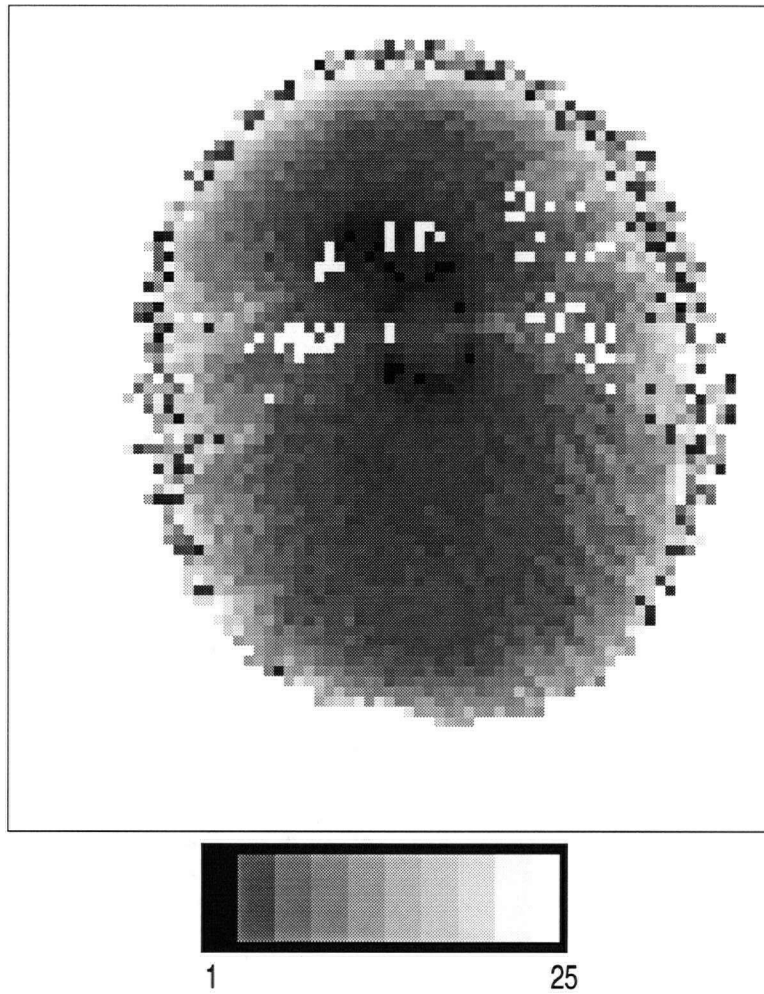


Figure 5.16: Condition number for the deforming balloon.

5.3 Comparison with other methods

To be able to make quantitative and qualitative comparisons of multiple light source optical flow with other methods a synthetic and a real image sequence were studied using some the implementations described in [6] and multiple light source optical flow running in a Sun workstation. In this implementation the images are spatially presmoothed-smoothed with a Gaussian filter with standard deviation $\sigma = 1.5$.

The methods selected for comparison are four differential techniques and two region-based matching techniques, namely, Horn and Schunck[4], Lucas and Kanade[5], Uras et al.[14], Nagel[13], Little, Bülthoff and Poggio[20], and Anandan[21]. The differential techniques include two first-order techniques and two second-order ones using global and local constraints to determine the full velocity vector. For details about these implementations see reference[6], and reference[20]. The error measure used to compare the results is the angular measure defined in references [6] and [25]. Let the estimated velocity $\mathbf{v}_e = [u, v]^T$ be written as $\mathbf{v}_e = (u^2 + v^2 + 1)^{-1/2}[u, v, 1]^T$. The error between the correct velocity \vec{v}_c and the estimated velocity is defined as

$$error = arccos((\mathbf{v}_c)^T \cdot \mathbf{v}_e) \quad (5.1)$$

The authors of reference[6] point out that this error measure is convenient because it handles large and very small speeds without the amplification of relative errors using vector differences. The bias is that directional errors at small speeds do not give angular errors as large as similar directional errors at higher speeds.

The source codes of the comparison methods and the error measure were retrieved from the ftp site referenced in [6]. Dr. James Little provided the implementation described in reference[20].

5.3.1 Lambertian sphere

As a comparison it is fair to use differential techniques since that is the classification of the method described in this thesis. This is particularly true for the case of a translating Lambertian sphere that has no particular features to match. An image sequence was created of a sphere with image brightness proportional to $\cos(i)$, illuminated from three different directions with a distant light source. The image size is 150x150 pixels and the sphere was moved by 1.3 pixels/frame along the horizontal axis. Figure(5.17) shows the different conditions of illumination. Figures(5.18), (5.19), (5.20), (5.21), (5.22) show the optical flow for four differential techniques, Horn and Schunck, Lucas and Kanade, Uras et al., Nagel, and for multiple light source optical flow. The illumination that was used for these methods is the one corresponding to Figure(5.17,(b)). No thresholds were used. The optical flow field is shown for every second point. The position of the sphere is indicated with a circle. Table(5.1) displays the average angular error for these cases.

All the methods perform well on average. Horn and Schunck, and Nagel methods give pleasing, smooth fields but they give normal velocities at the edges. The local methods of Lucas and Kanade and Uras et al. give good results for the body of the sphere but very inaccurate results on the edges. As expected, multiple light source optical flow performs very well. This test case is exactly the type where this technique is known to give good results, i.e., smooth, well-shaded surfaces with no texture. Most of the errors are in the edges of the sphere, where the condition number of the system is larger. Areas of small variation of image brightness, like the center of the sphere, give larger relative errors. Figures(5.23)-(5.24) show the relative error and the condition number for the translating sphere. The scales are the same as for the balloon sequence. Image points that are illuminated by only

Technique	Average Error	Std. Dev.
Horn and Schunck (modified) no threshold	6.41	12.60
Nagel, no threshold	5.37	11.55
Lucas and Kanade, no threshold	8.14	20.44
Uras et al., no threshold, regularization 2	9.77	21.49
Multiple Light Source, no threshold	1.17	7.49

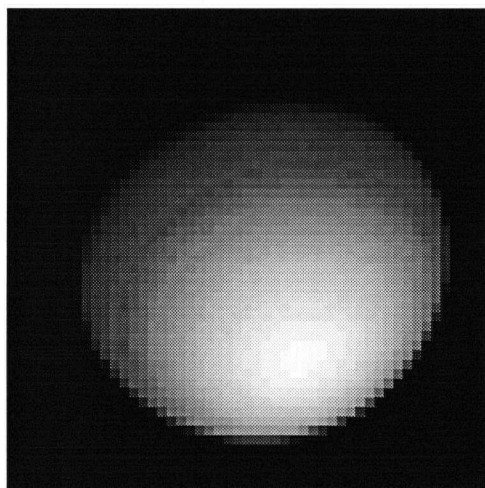
Table 5.1: Average error for the translating sphere. In all cases the density of the flow field is 100%.

two light sources have a relative error of zero and are printed as black. This is the case for the two black areas at the edge of the sphere. Points in the background are assigned white. A comparison of Horn and Schunck, Nagel, Lucas and Kanade, and multiple light source optical flow using different thresholds is shown in Figure(5.25).

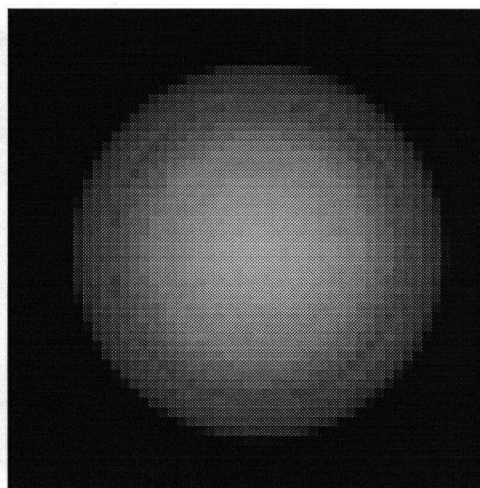
The behavior of multiple light source optical flow for different speeds is shown in Figure(5.26). The average measured speed in the horizontal, x , direction was plotted against the real speed. The linear range in this synthetic case is up to almost 4 pixels per frame. The relative error is shown in Figure(5.27), the minimum is at 1 pixel/frame. The error bars indicate an interval of two standard deviations centered at the average value.

5.3.2 Stay-puft Marshmallow Man

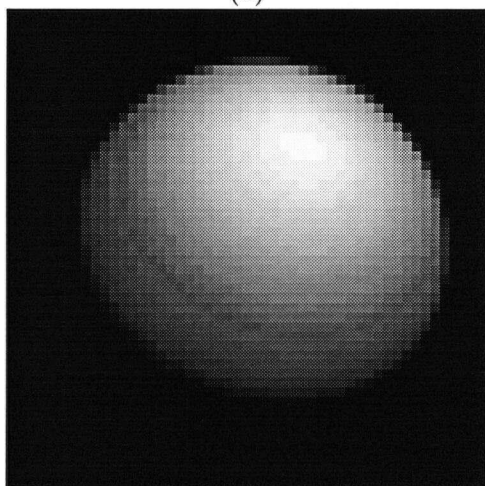
A real sequence of images was created by moving the object in the horizontal axis. Figure(5.28) shows the three different conditions of illumination. Figures(5.29)-(5.40) show the results obtained with four differential techniques, for the three different conditions of illumination; Horn and Schunck, Lucas and Kanade, Uras et al., and Nagel. Figures(5.41)-(5.43) show the result for a non-hierarchical matching technique[20]. Figures(5.44)-(5.46) show the results of a hierarchical matching



(a)



(b)



(c)

Figure 5.17: Lambertian sphere illuminated from three different directions (a), (b), (c).

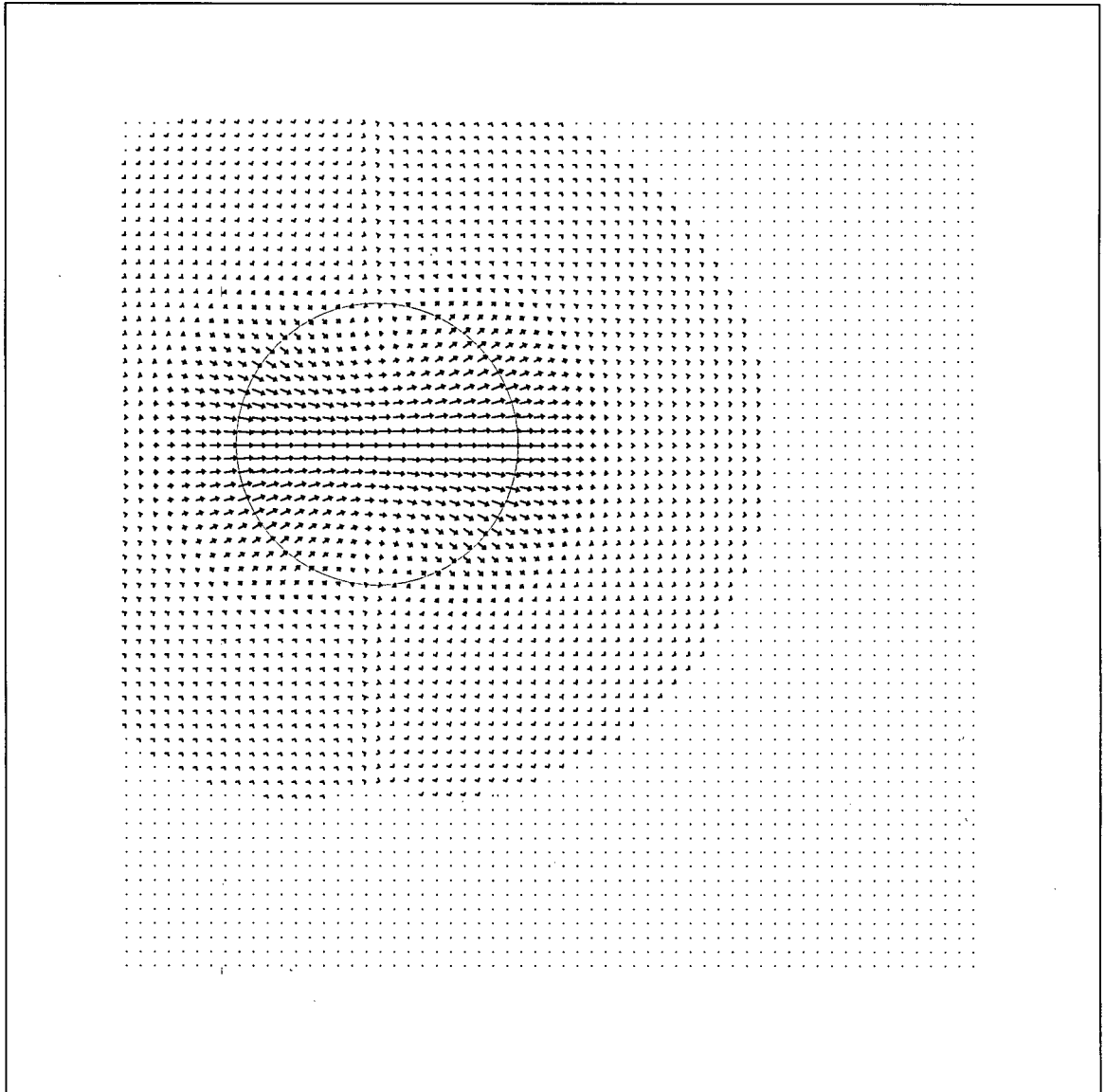


Figure 5.18: Translating Lambertian sphere, Horn's technique, no threshold, $\lambda = 0.5$, $\sigma = 1.5$, 100 iterations. The circle indicates the position of the sphere.

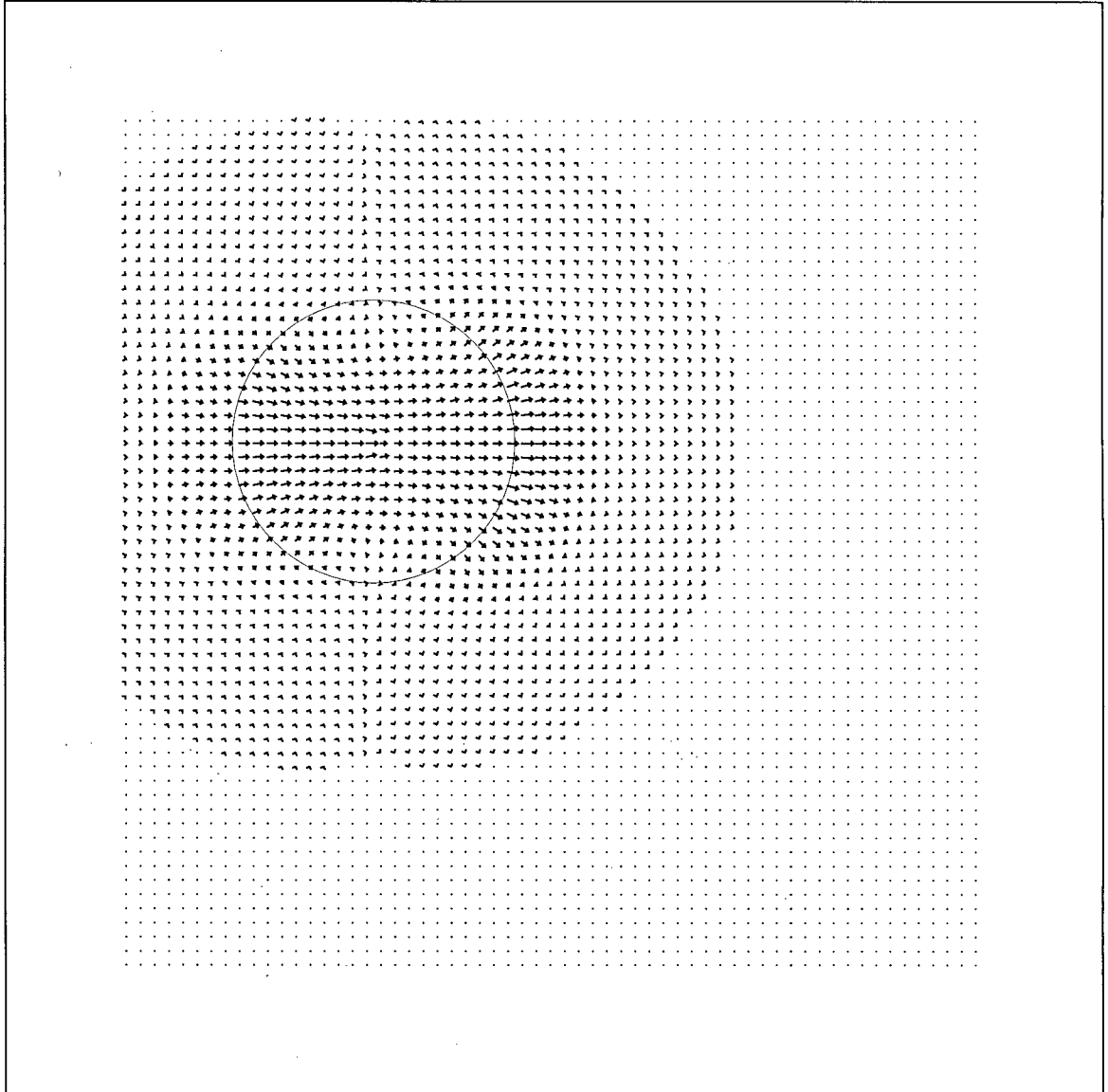


Figure 5.19: Translating Lambertian sphere, Nagel's technique, no threshold, $\alpha = 0.5$, $\delta = 1.0$, $\sigma = 1.5$, 100 iterations. The circle indicates the position of the sphere.

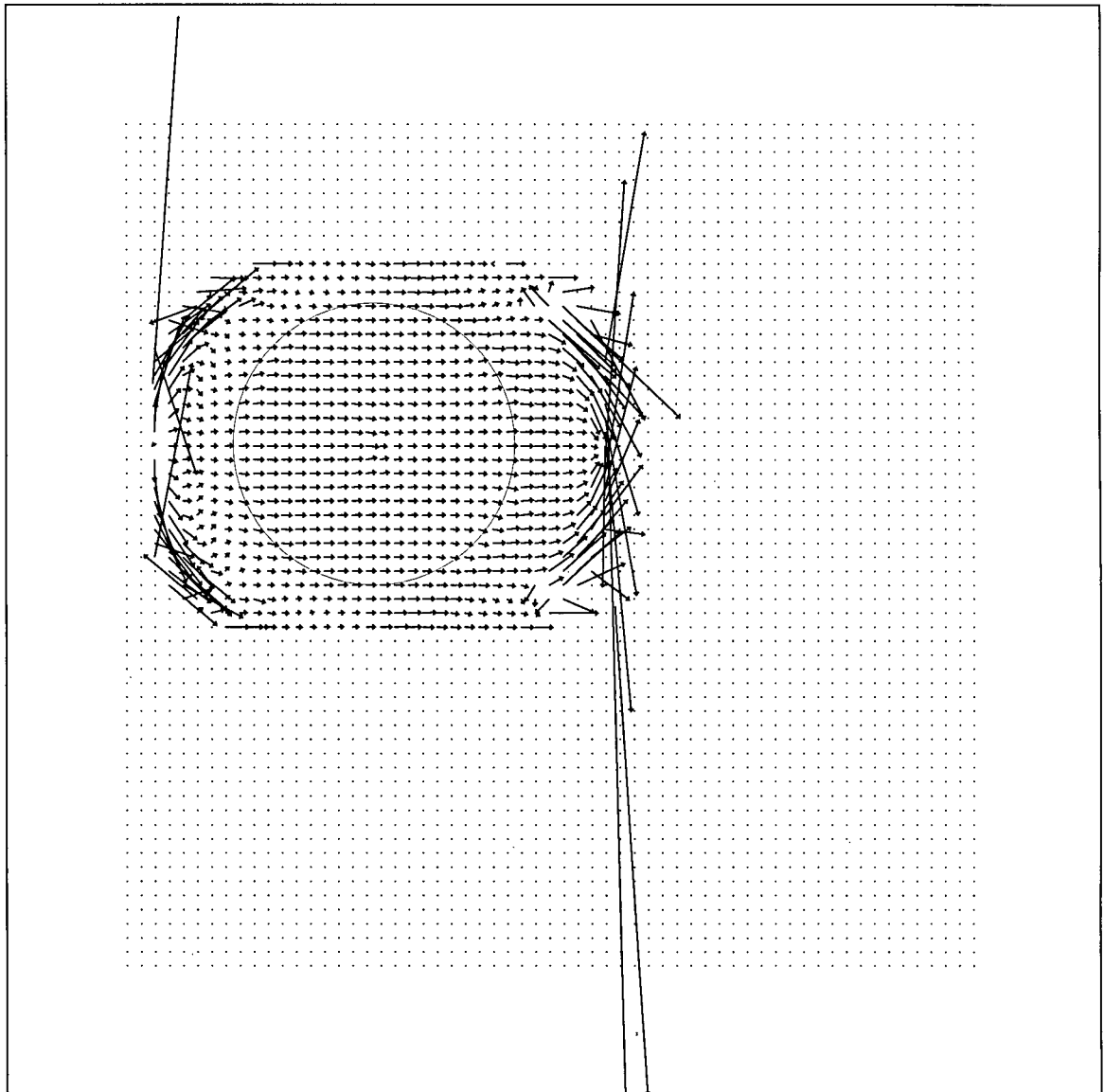


Figure 5.20: Translating Lambertian sphere, Lucas and Kanade technique, no threshold, $\sigma = 1.5$. The circle indicates the position of the sphere.

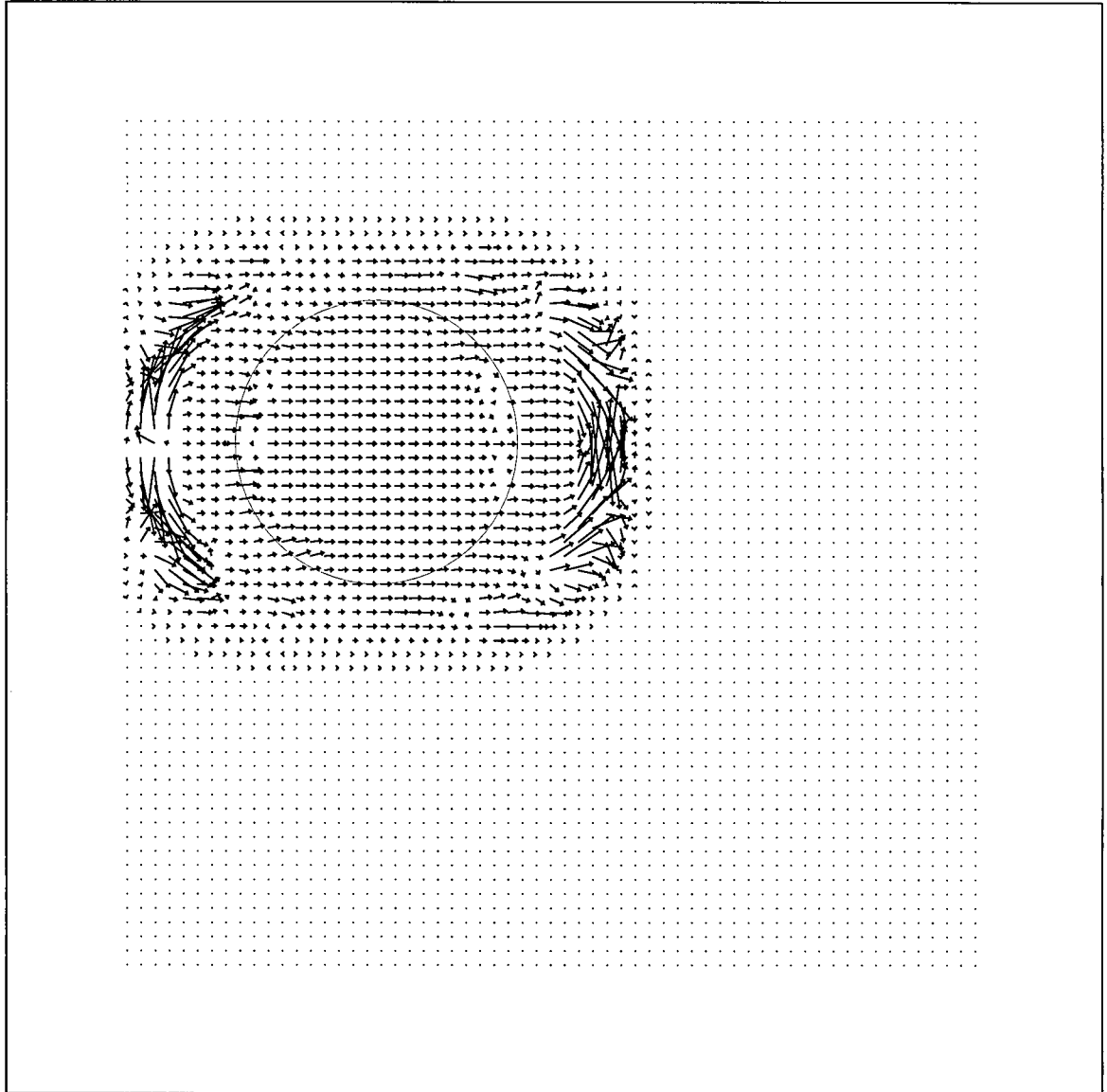


Figure 5.21: Translating Lambertian sphere, Uras et al. technique, regularization 2, no threshold, $\sigma = 1.5$. The circle indicates the position of the sphere.

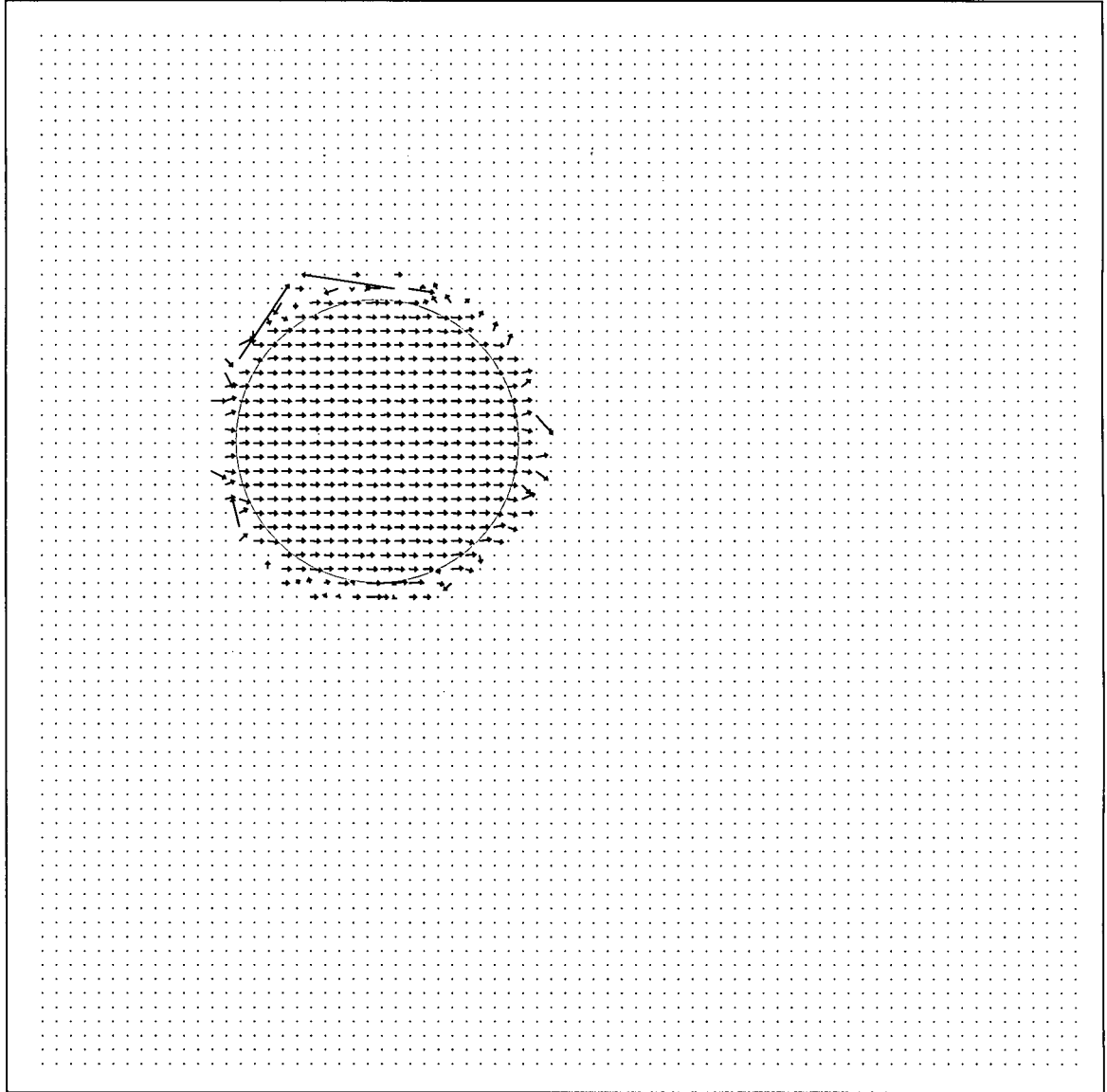


Figure 5.22: Translating Lambertian sphere, multiple light source optical flow, no threshold, $\sigma = 1.5$. The circle indicates the position of the sphere.

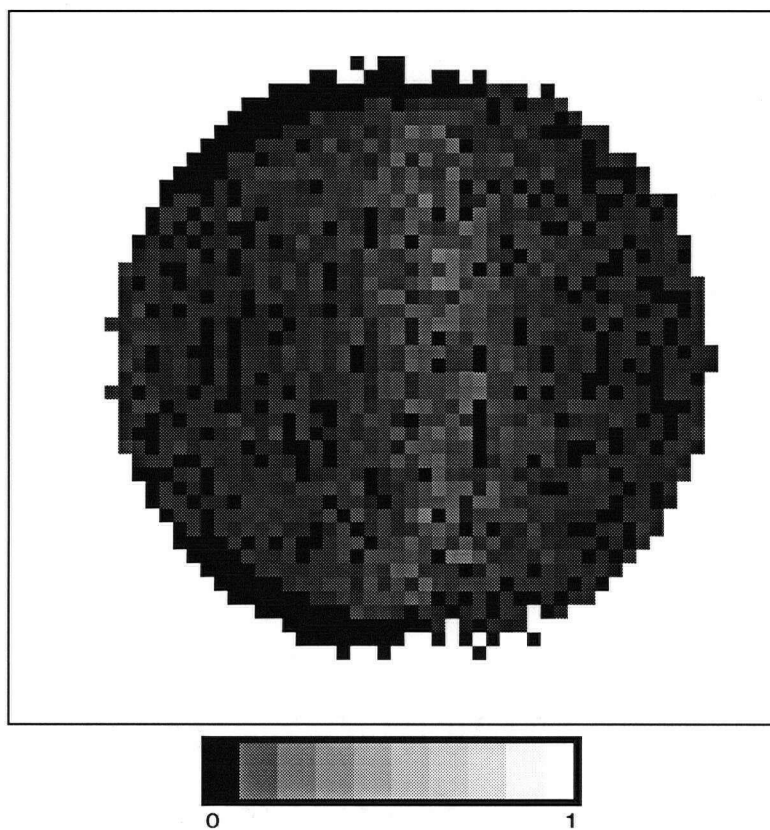


Figure 5.23: Translating Lambertian sphere. Relative error for multiple light source optical flow.

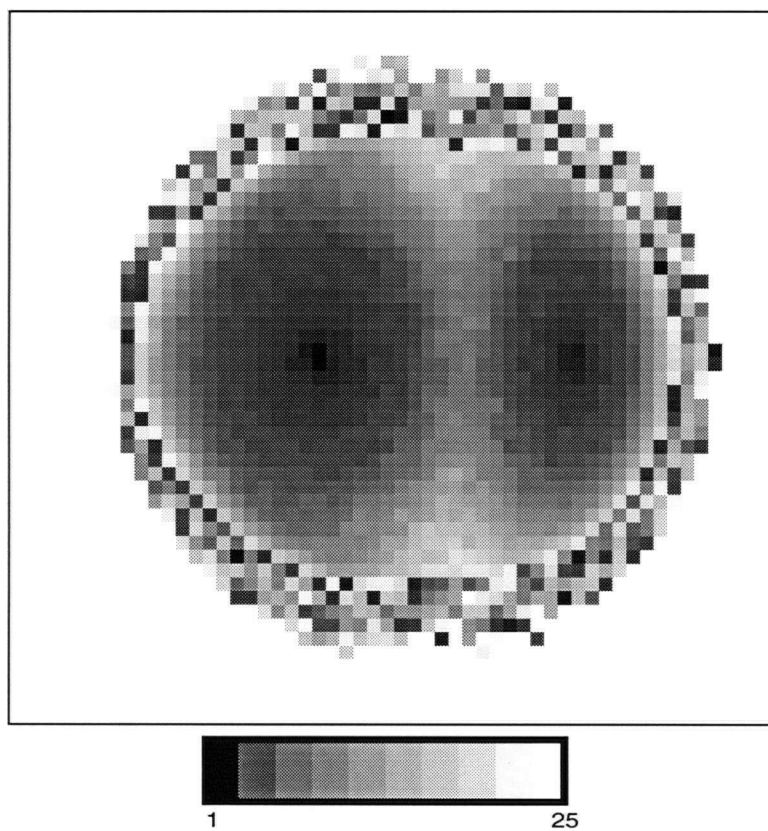


Figure 5.24: Translating Lambertian sphere. Condition number for multiple light source optical flow.

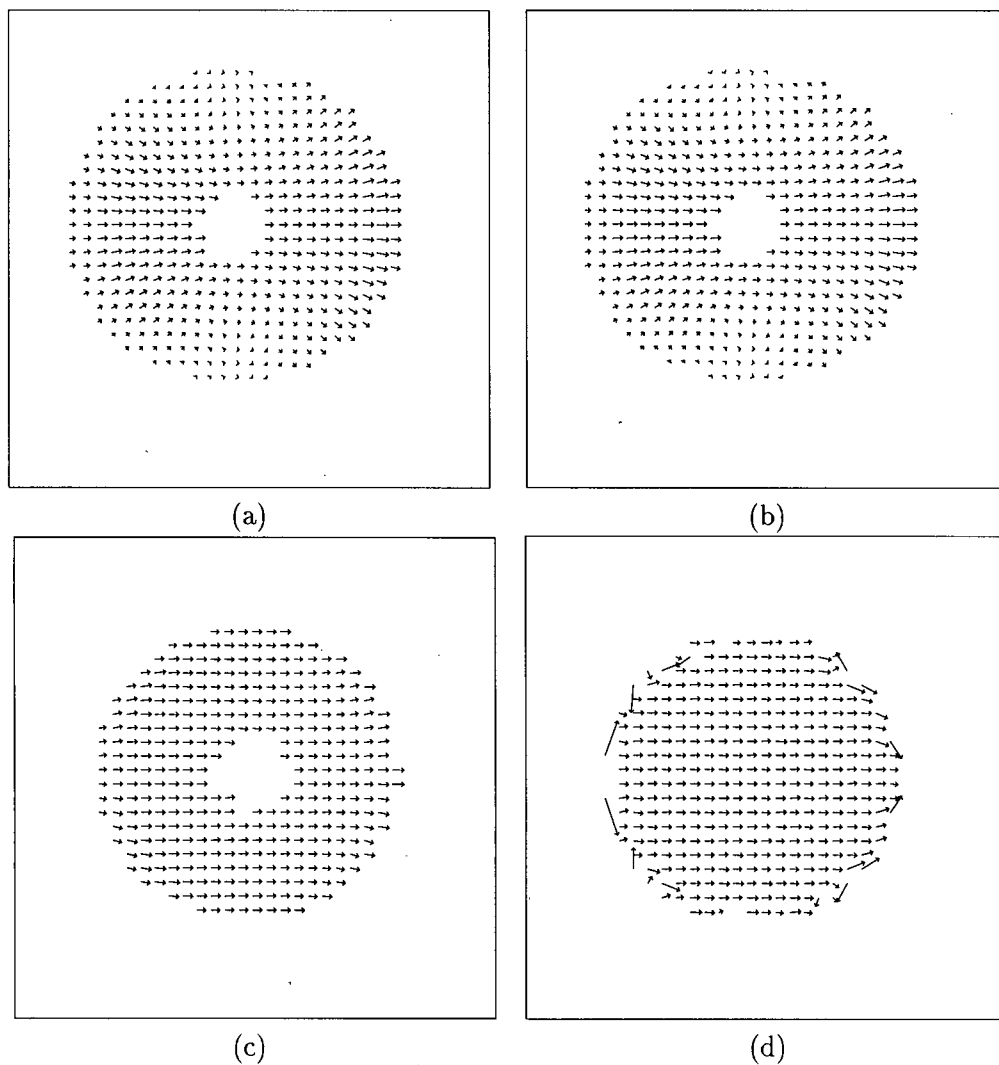


Figure 5.25: Translating Lambertian sphere. (a) Horn's method, $\| \nabla E \| \geq 2.5$. (b) Nagel's method, $\| \nabla E \| \geq 2.5$. (c) Lucas and Kanade method, $\lambda_2 \geq 0.2$. (d) Multiple light source optical flow, $\| \nabla E \| \geq 2.5$

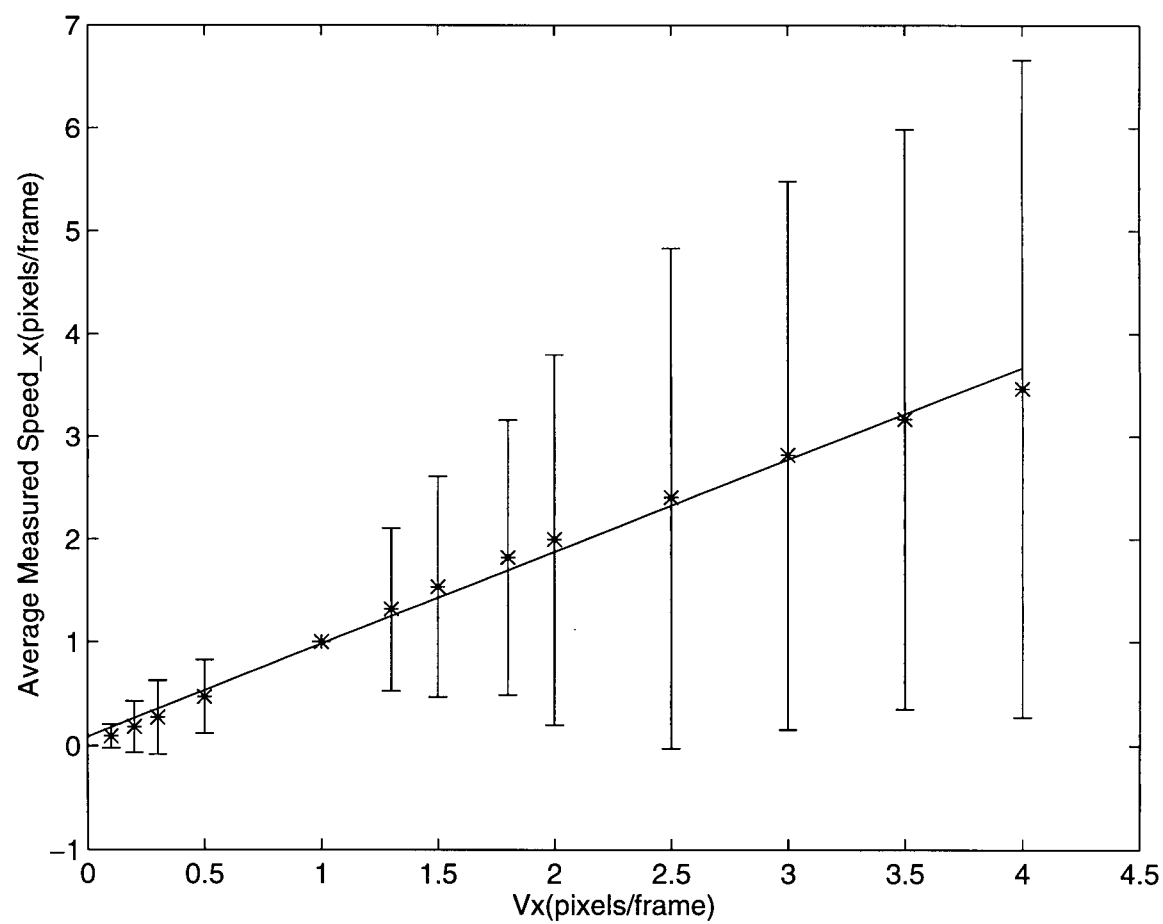


Figure 5.26: Translating Lambertian sphere. Average measured speed vs real speed for multiple light source optical flow. The error bars indicate an interval of two standard deviations centered at the average value.

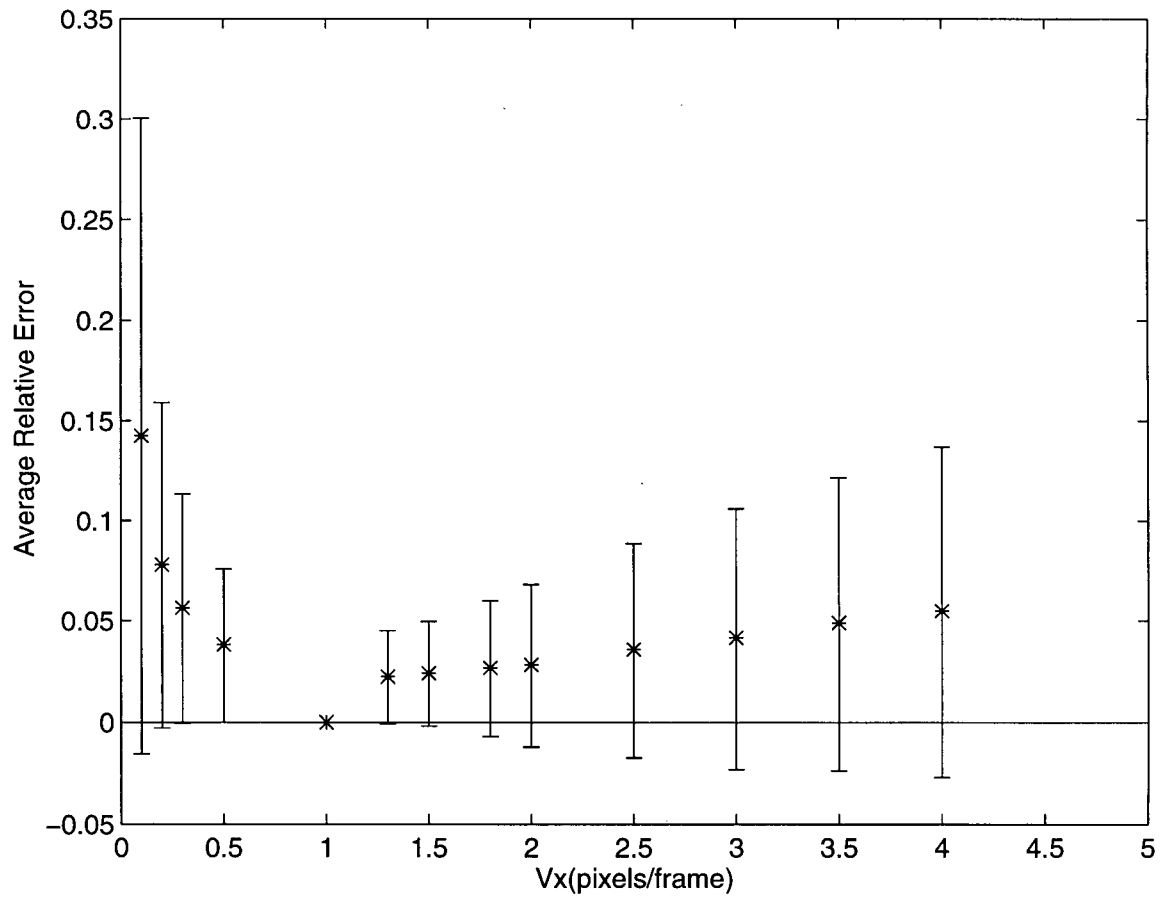
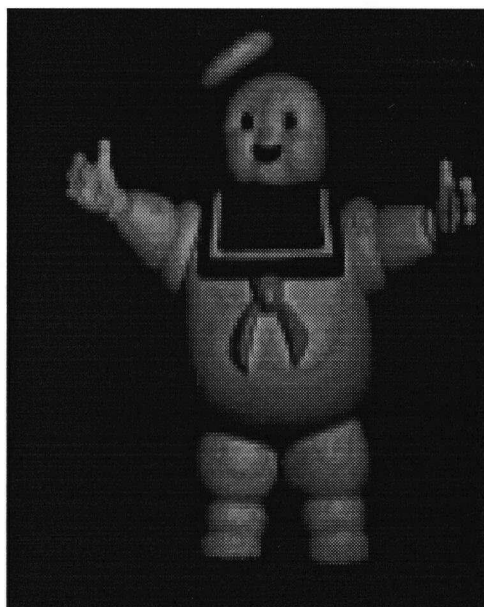
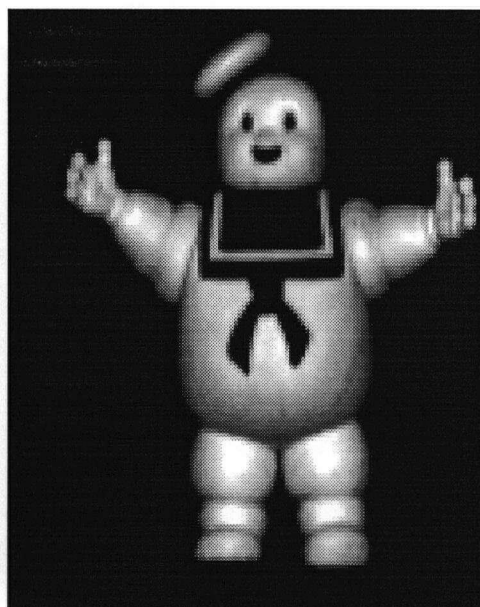


Figure 5.27: Translating Lambertian sphere. Average relative error vs real speed for multiple light source optical flow. The error bars indicate an interval of two standard deviations centered at the average value.

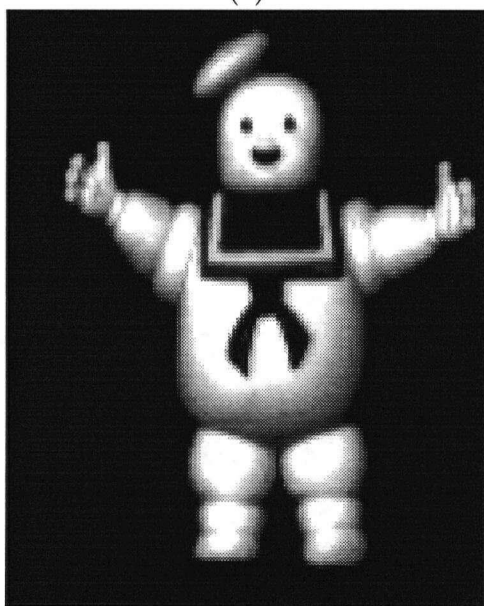
technique, Anandan, and Figure(5.47) displays the results for multiple light source optical flow. The thresholds that were used are indicated in the captions of the figures. The choice of these values was such that a similar density was obtained for the flow field. The optical flow fields were subsampled to prevent cluttering and were scaled by two. In this case there is no correct optical flow to make quantitative comparisons but the results obtained with multiple light source optical flow are qualitatively good and comparable to the other techniques. The relative error term and the condition number for these images are shown in Figures(5.48)-(5.49). The scales are as for the Lambertian sphere. Multiple light source optical flow performs well in regions that exhibit significant shading, and that are not close to boundaries and surface markings. The comparison techniques are sensitive to the conditions of illumination as can be seen in the results obtained with the different light sources.



(a)



(b)



(c)

Figure 5.28: Stay-Puft Marshmallow Man illuminated from three different directions, (a) red light source, (b) green light source, (c) blue light source.

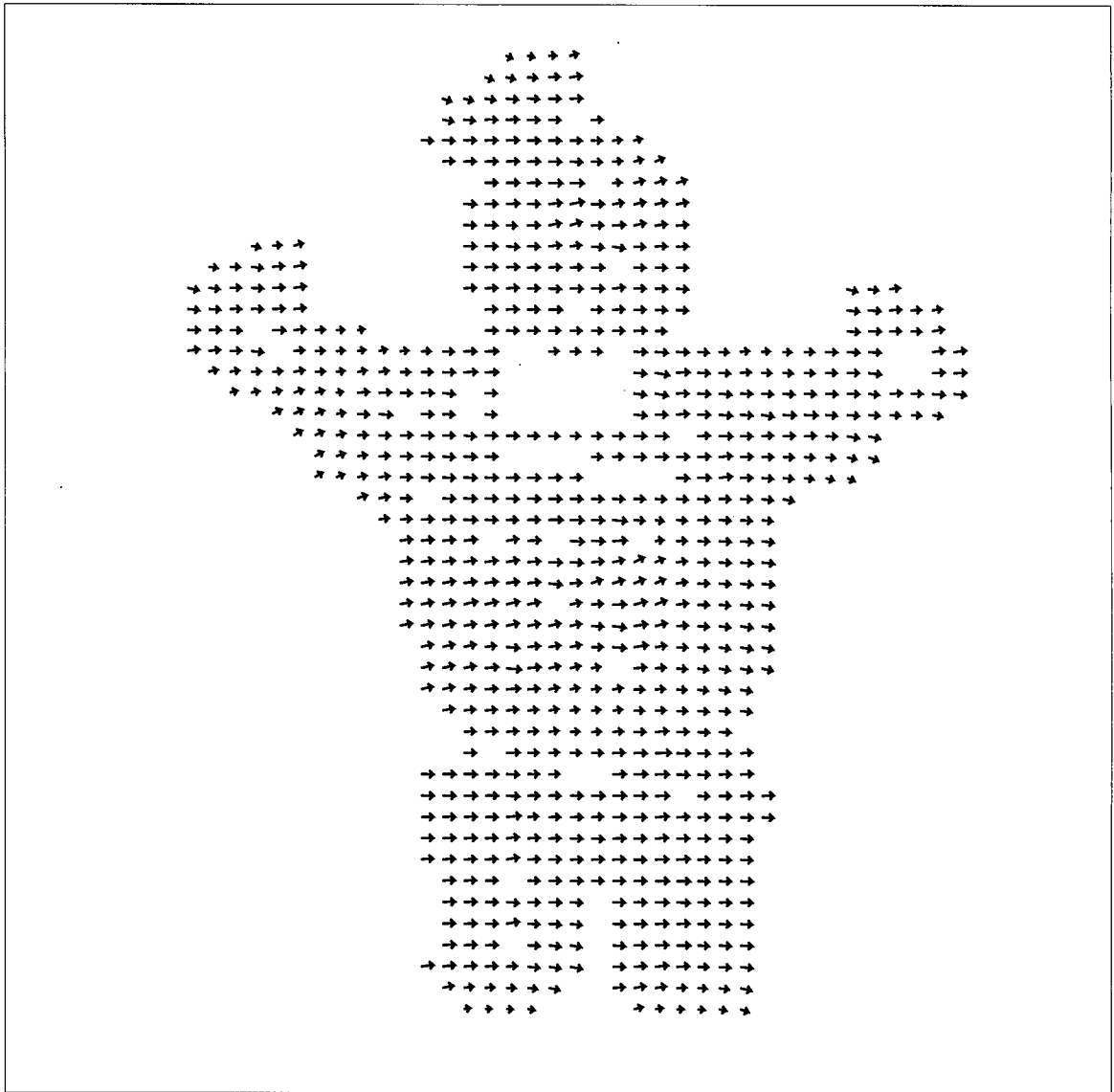


Figure 5.29: Translating Stay-Puft Marshmallow Man, Horn's technique, $\|\nabla E\| \geq 2.5$. Red light source.

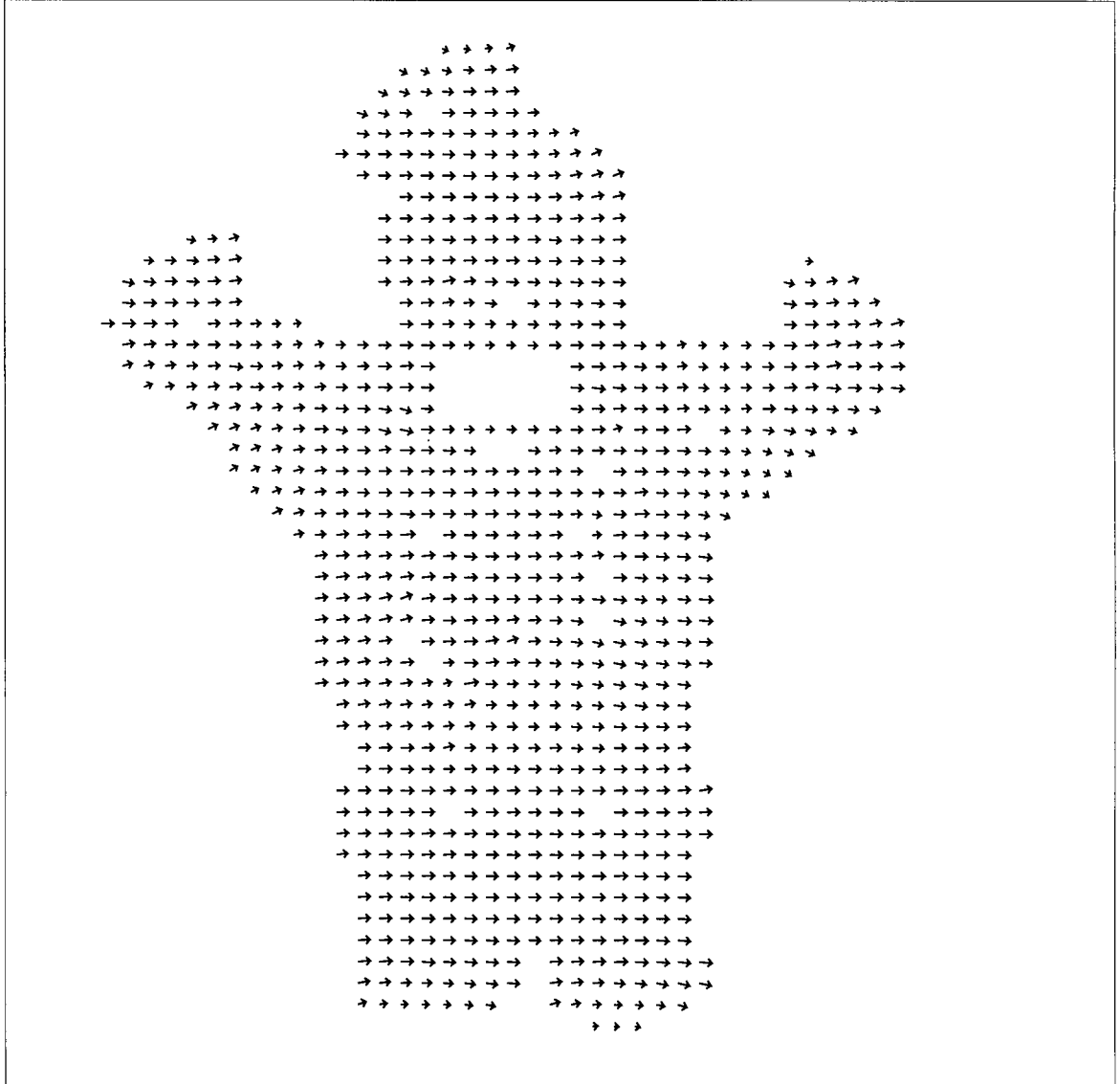


Figure 5.30: Translating Stay-Puft Marshmallow Man, Horn's technique, $\|\nabla E\| \geq 2.5$. Green light source.

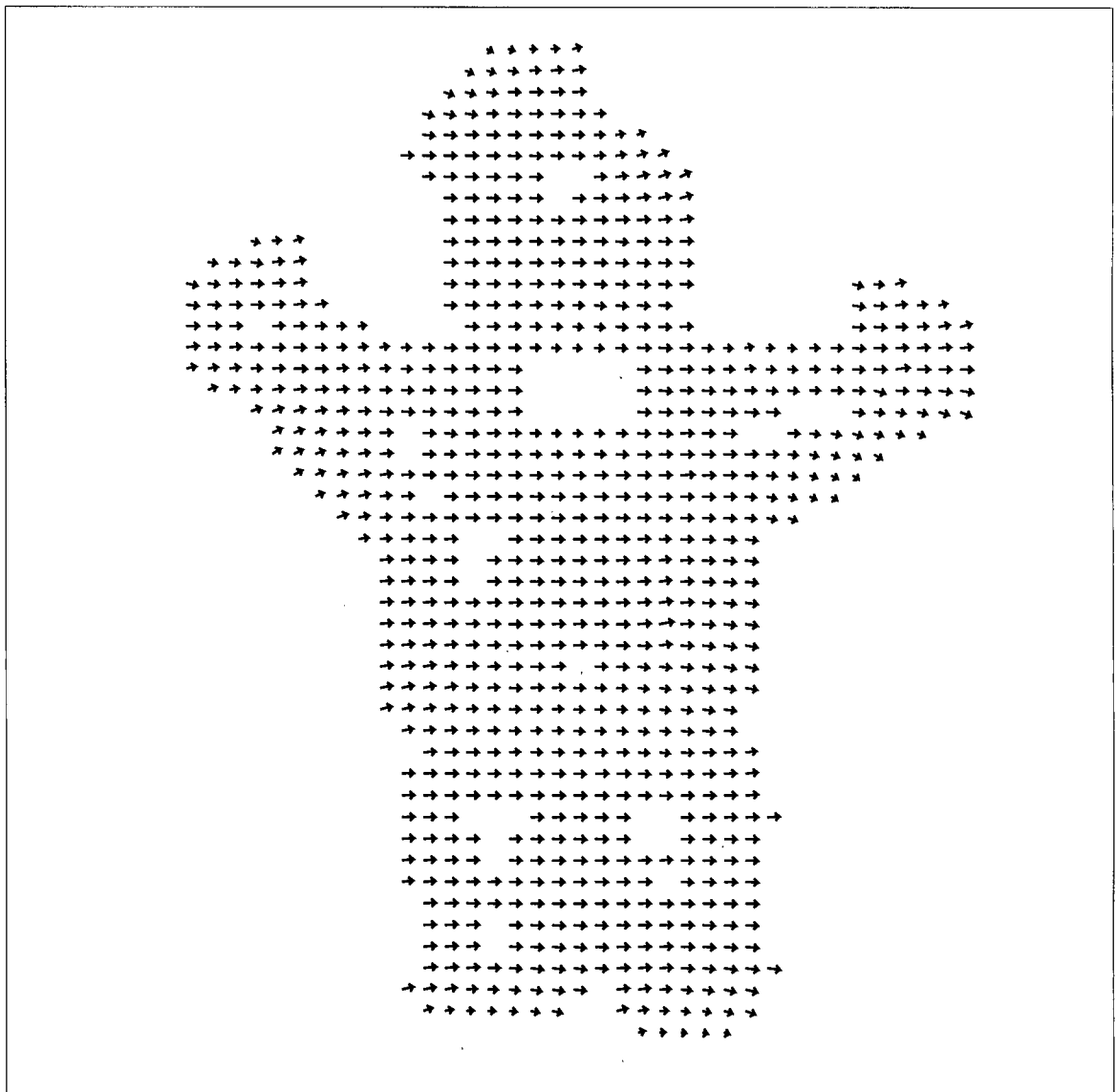


Figure 5.31: Translating Stay-Puft Marshmallow Man, Horn's technique, $\|\nabla E\| \geq 2.5$. Blue light source.

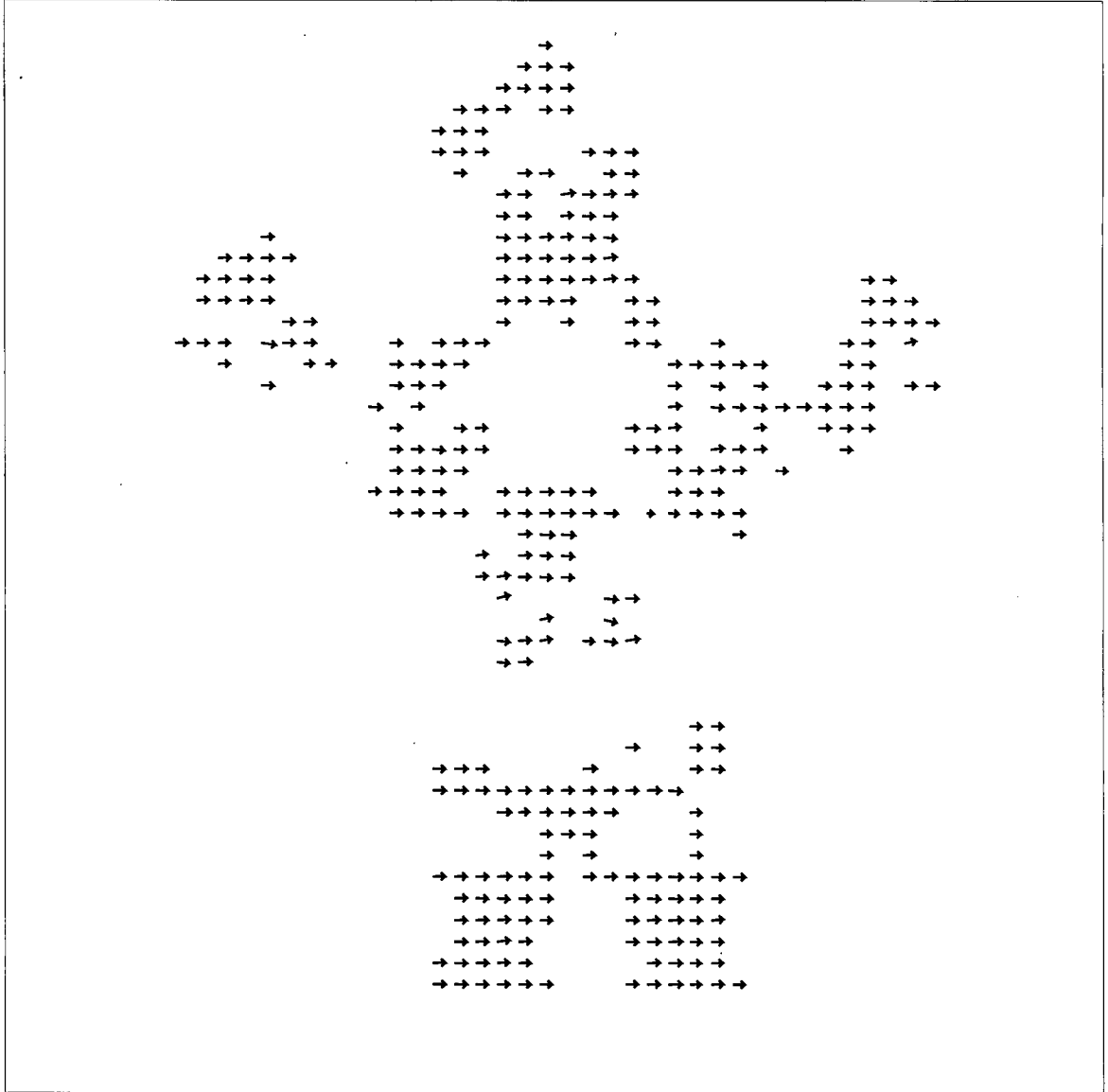


Figure 5.32: Translating Stay-Puft Marshmallow Man, Lucas and Kanade technique, $\lambda_2 \geq 2.5$, $\sigma = 1.5$. Red light source.

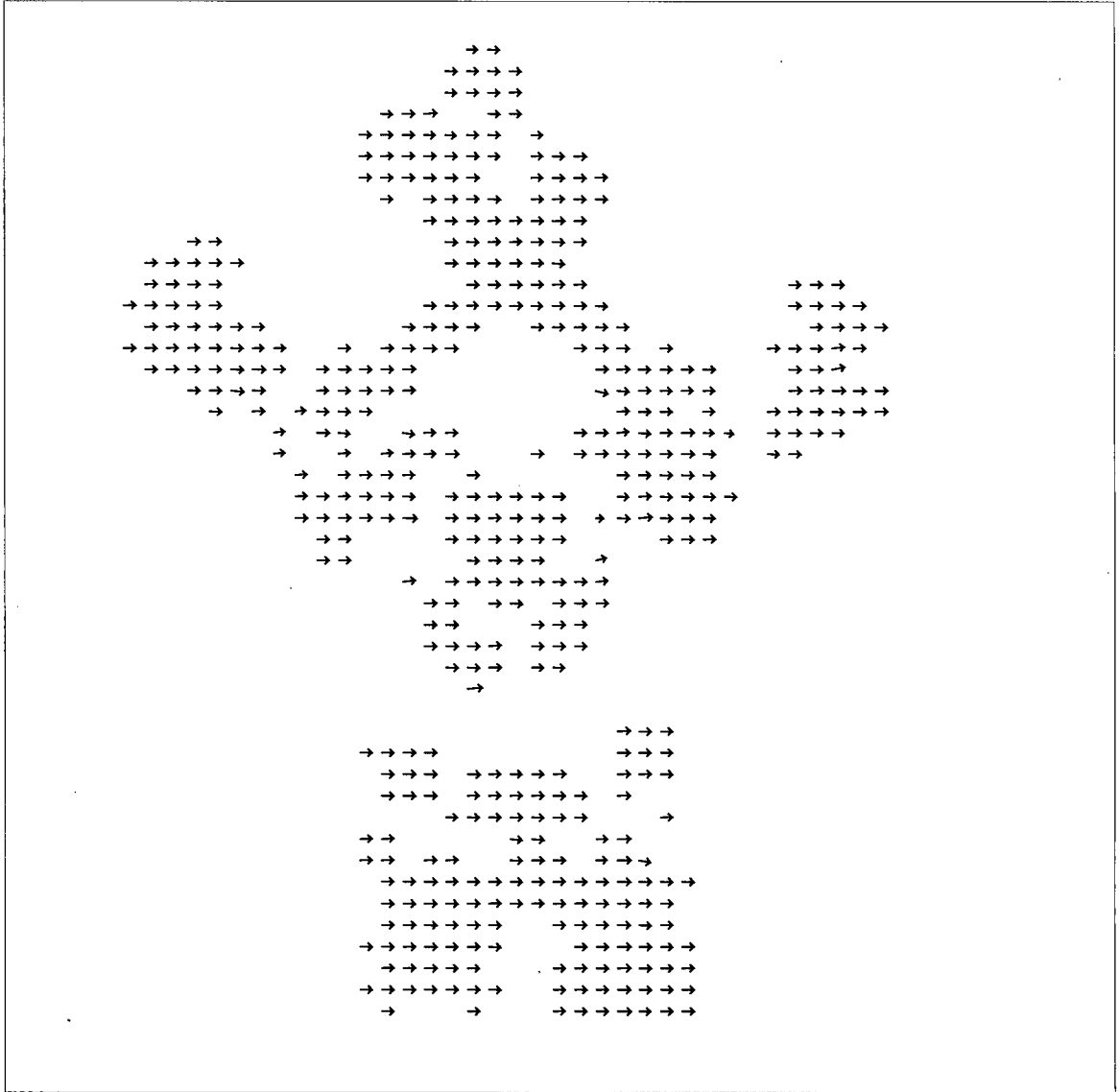


Figure 5.33: Translating Stay-Puft Marshmallow Man, Lucas and Kanade technique, $\lambda_2 \geq 2.5$, $\sigma = 1.5$. Green light source.

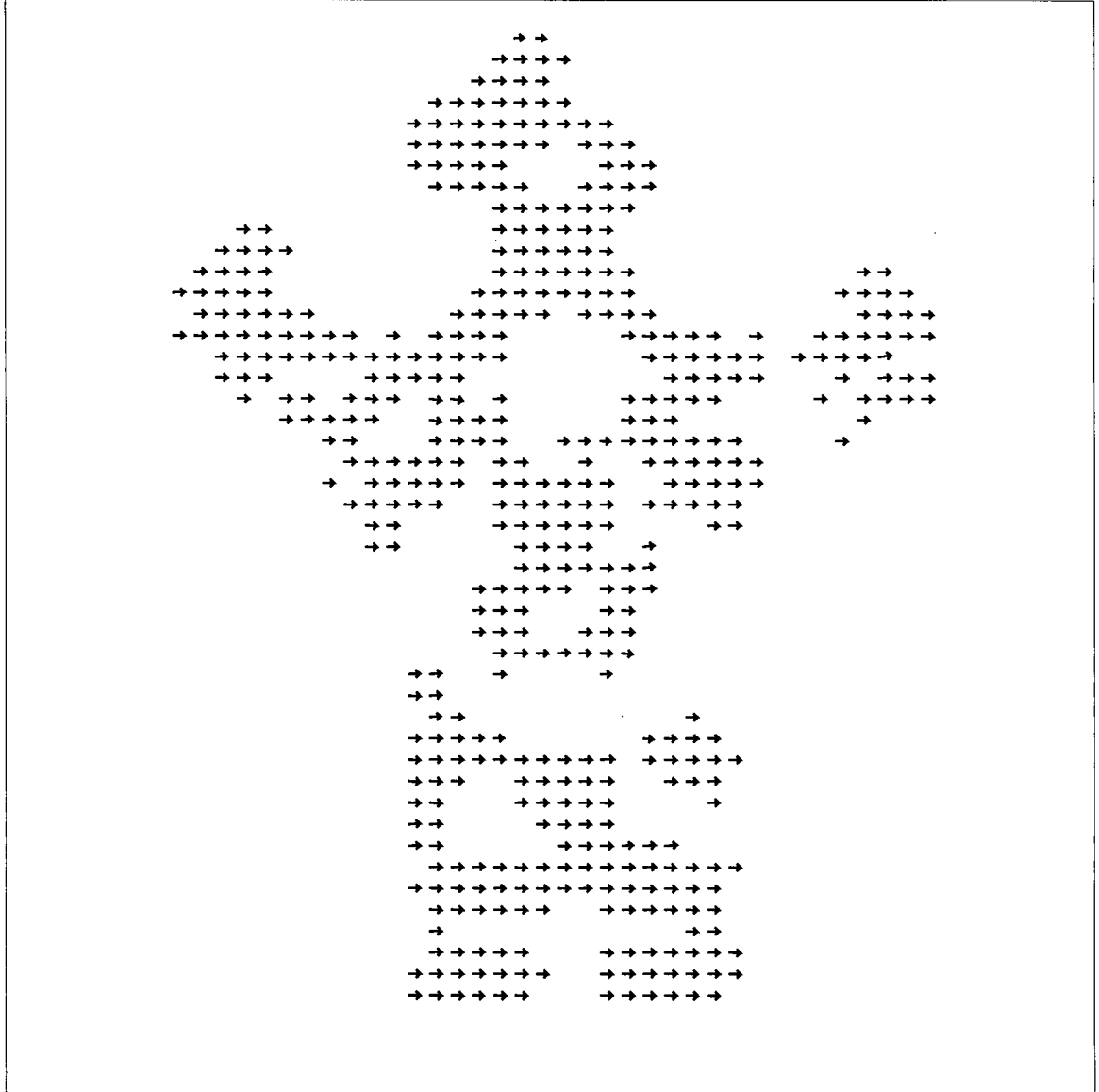


Figure 5.34: Translating Stay-Puft Marshmallow Man, Lucas and Kanade technique, $\lambda_2 \geq 2.5$, $\sigma = 1.5$. Blue light source.

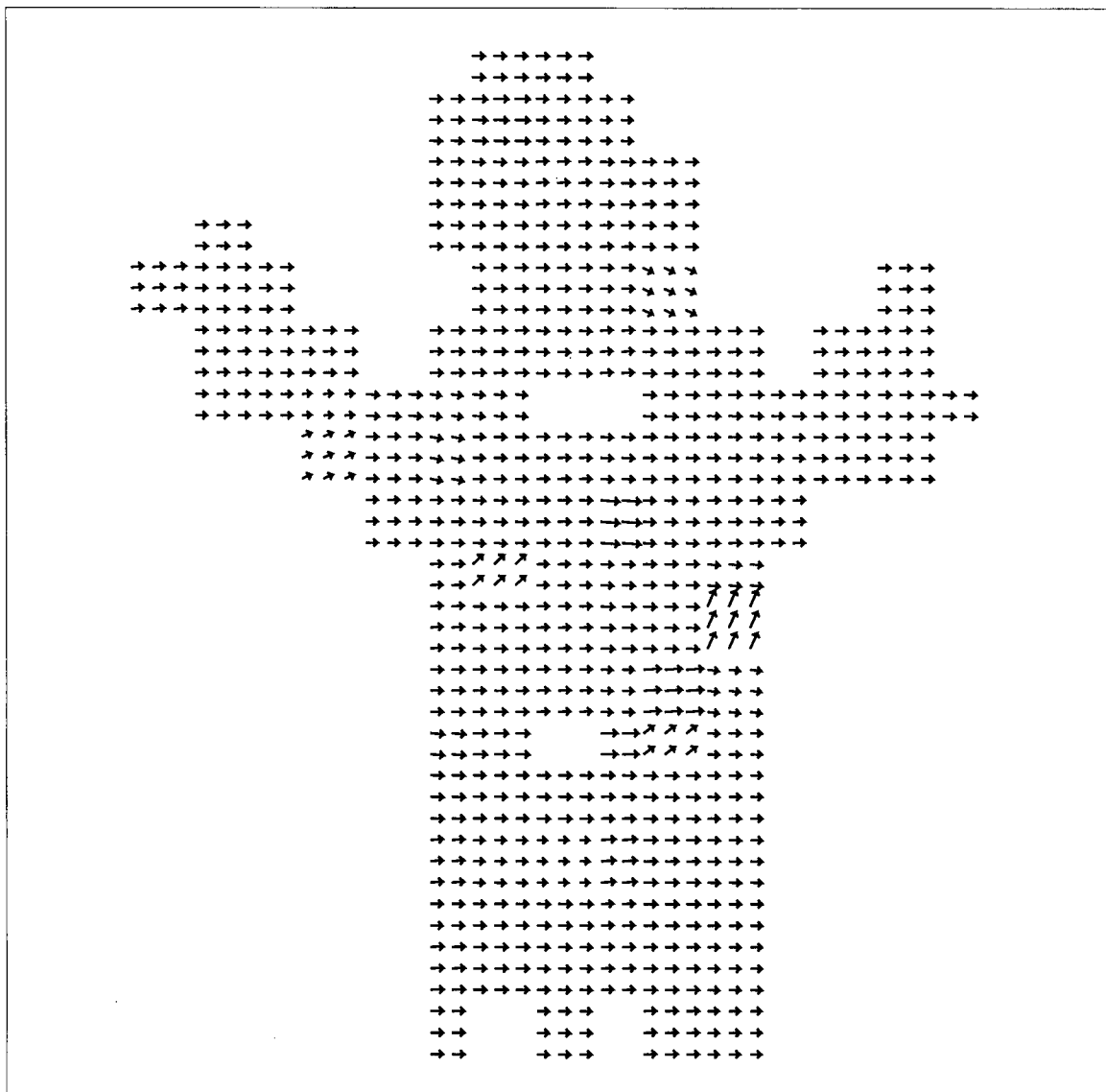


Figure 5.35: Translating Stay-Puft Marshmallow Man, Uras et al. technique, $\det(H) \geq 0.1$, $\sigma = 1.5$. Red light source.

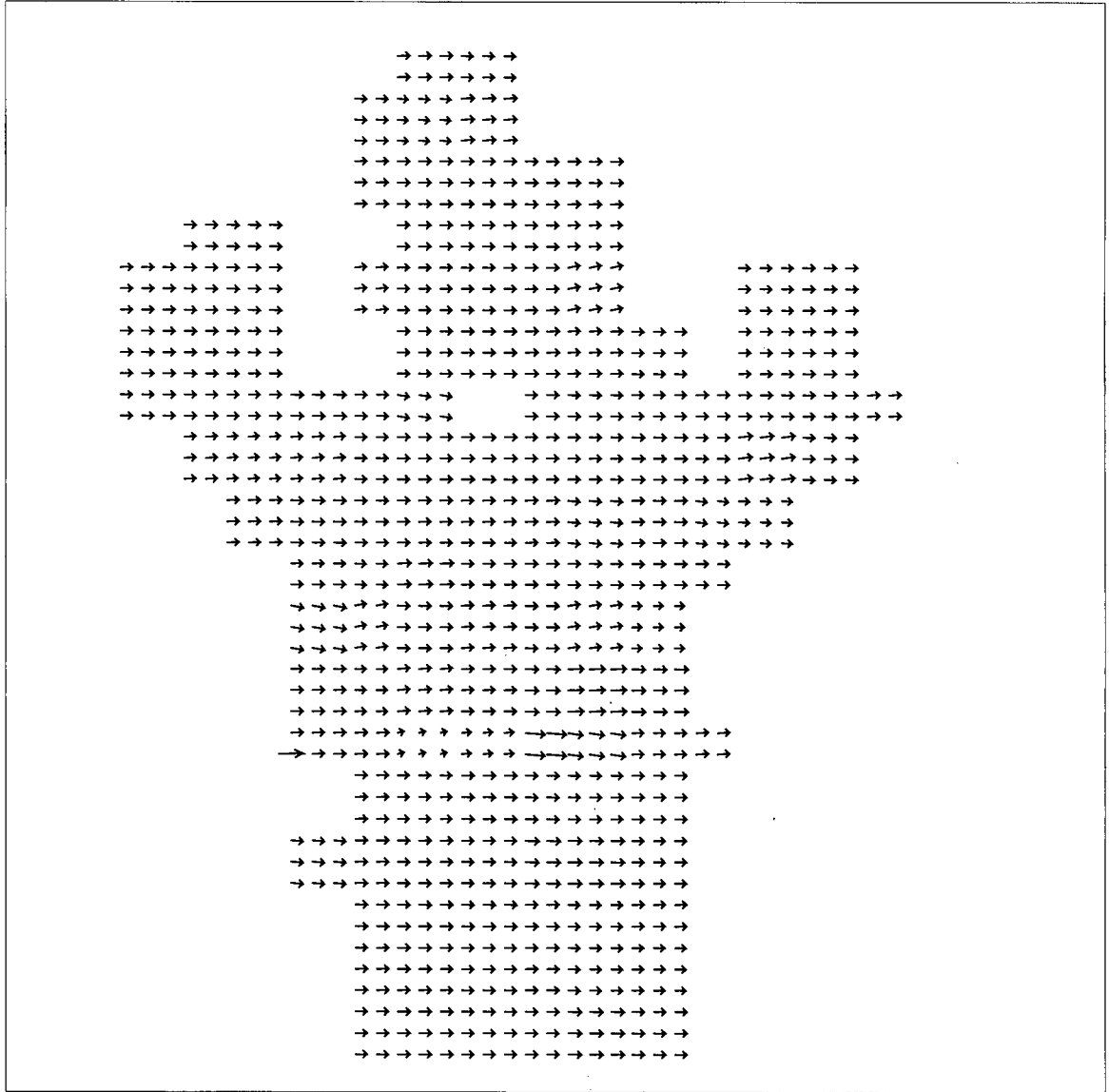


Figure 5.36: Translating Stay-Puft Marshmallow Man, Uras et al. technique, $\det(H) \geq 0.1$, $\sigma = 1.5$. Green light source.

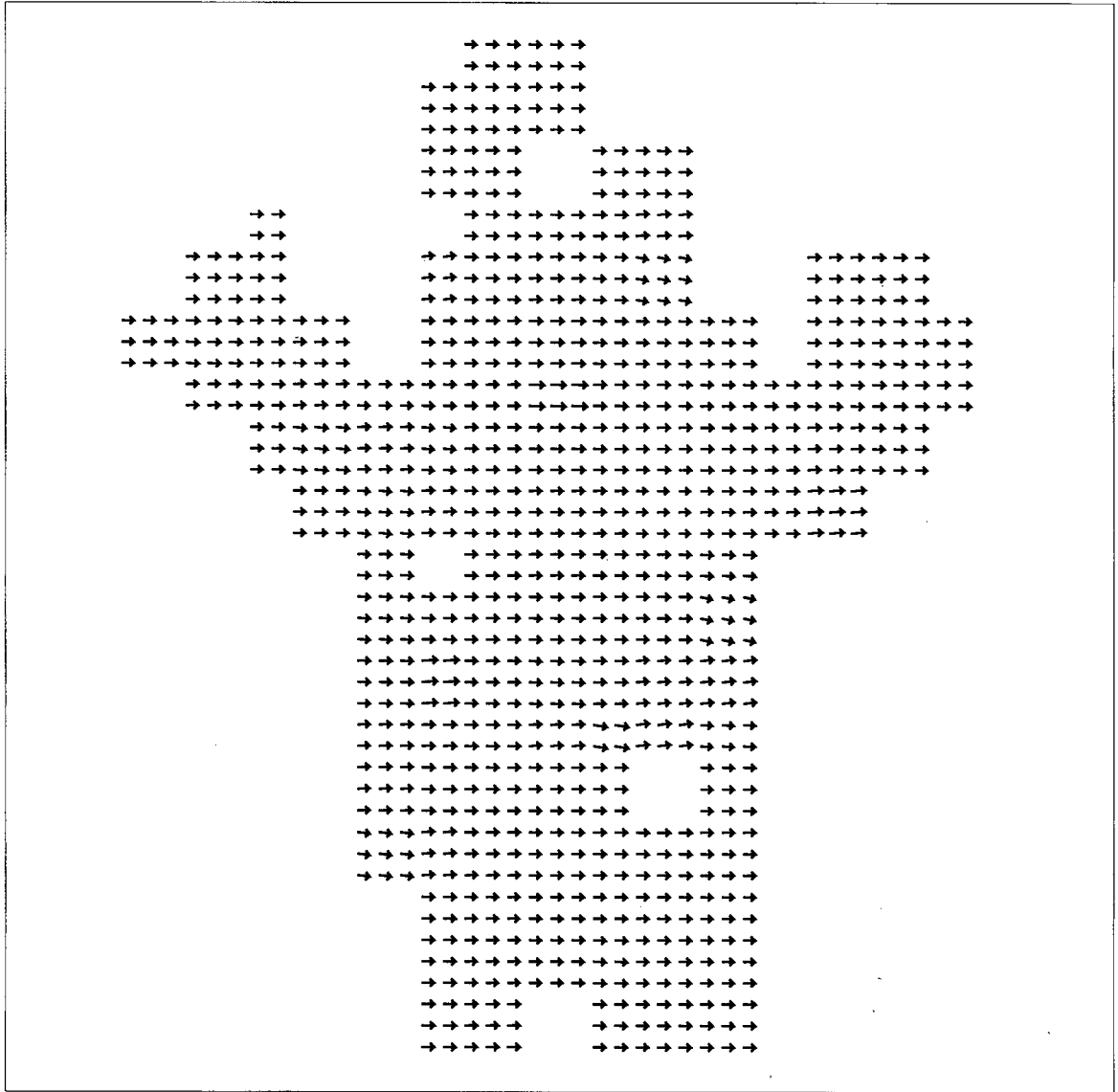


Figure 5.37: Translating Stay-Puft Marshmallow Man, Uras et al. technique, $\det(H) \geq 0.1$, $\sigma = 1.5$. Blue light source.

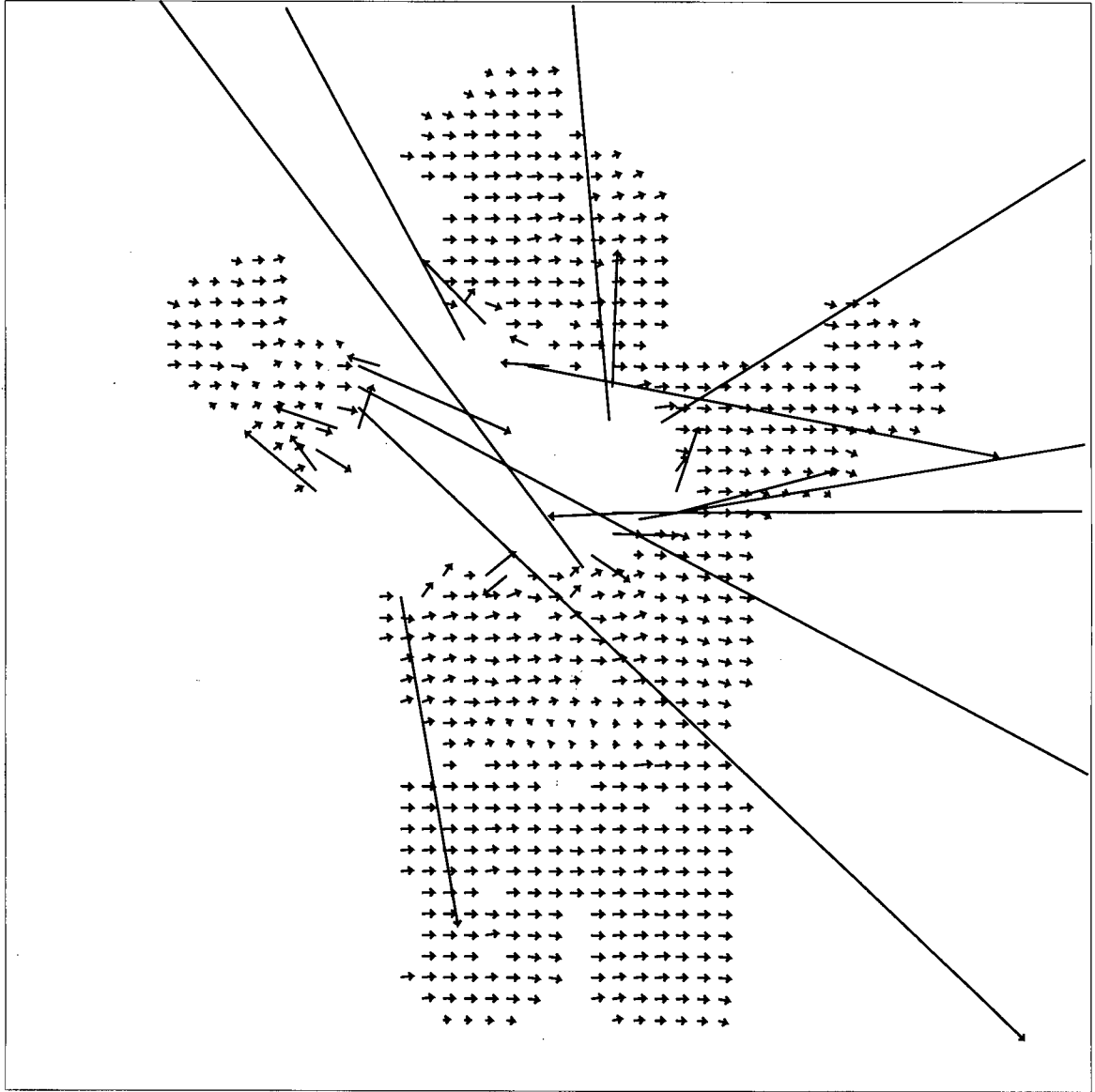


Figure 5.38: Translating Stay-Puft Marshmallow Man, Nagel's technique, $\|\nabla E\| \geq 2.5$, $\alpha = 0.5$, $\delta = 1.0$, $\sigma = 1.5$, 100 iterations. Red light source. To prevent cluttering of the image, velocities larger than 100 pixels/frame were not displayed.

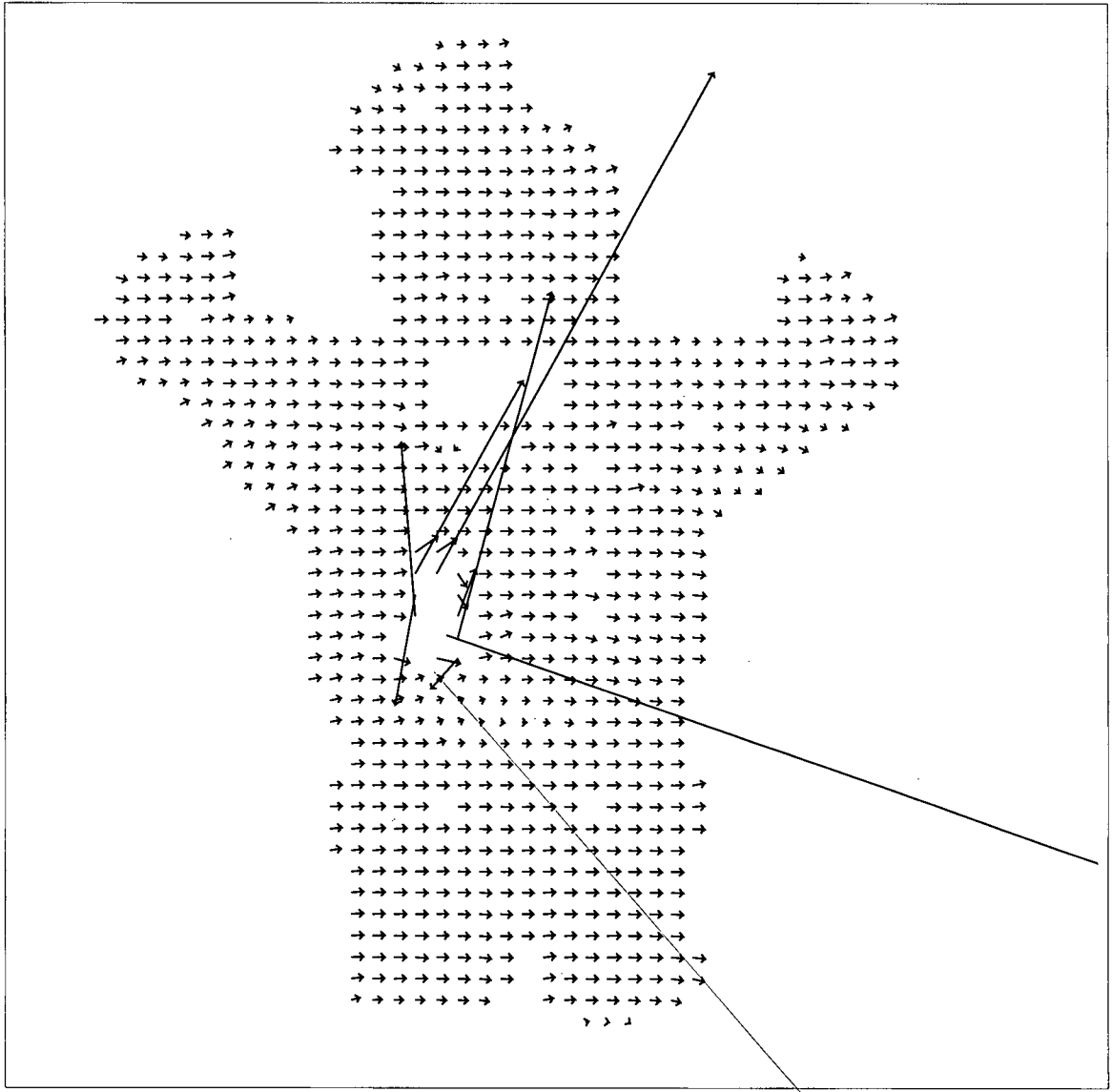


Figure 5.39: Translating Stay-Puft Marshmallow Man, Nagel's technique, $\|\nabla E\| \geq 2.5$, $\alpha = 0.5$, $\delta = 1.0$, $\sigma = 1.5$, 100 iterations. Green light source. To prevent cluttering of the image, velocities larger than 100 pixels/frame were not displayed.

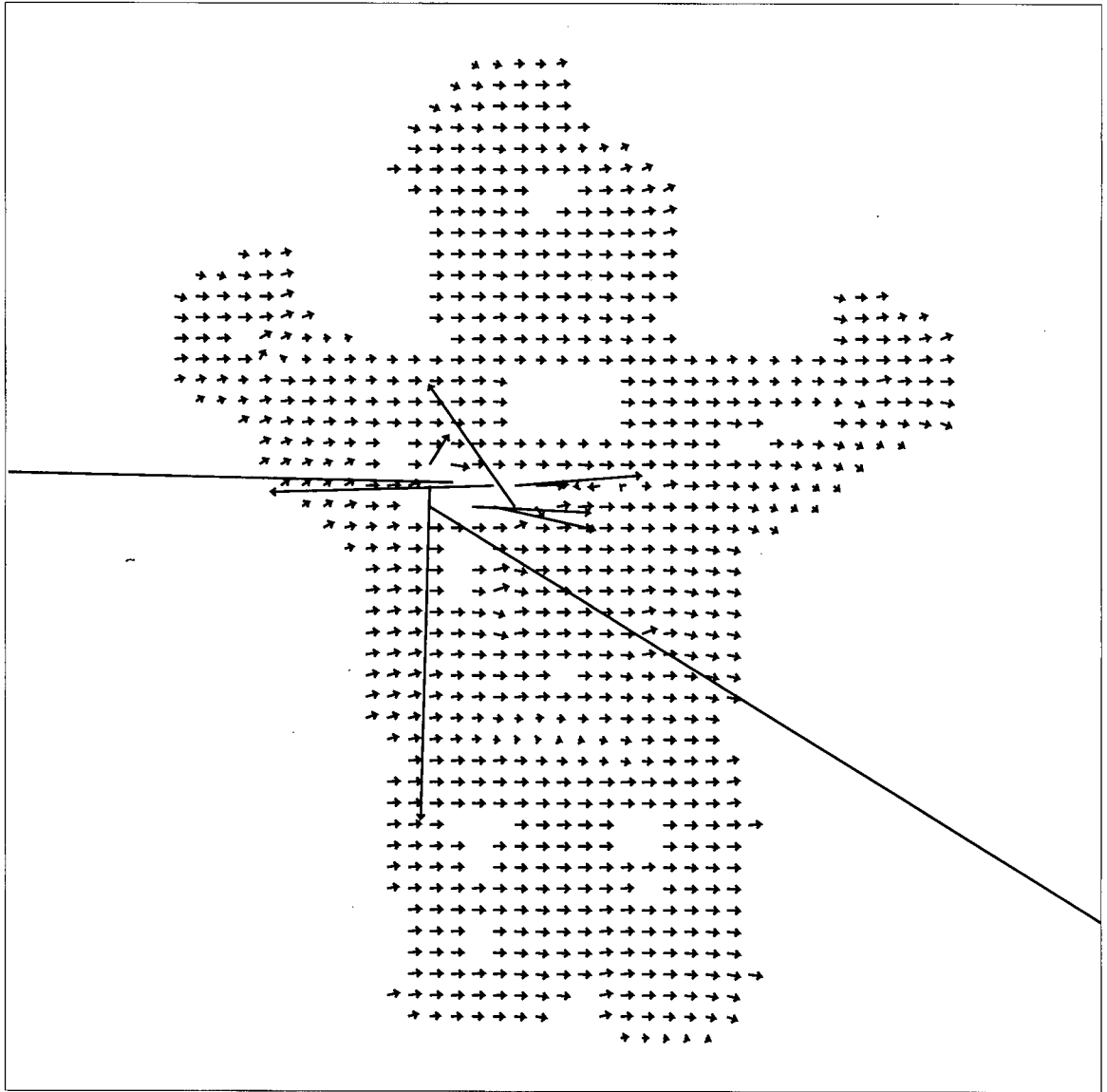


Figure 5.40: Translating Stay-Puft Marshmallow Man, Nagel's technique, $\|\nabla E\| \geq 2.5$, $\alpha = 0.5$, $\delta = 1.0$, $\sigma = 1.5$, 100 iterations. Blue light source. To prevent cluttering of the image, velocities larger than 100 pixels/frame were not displayed.

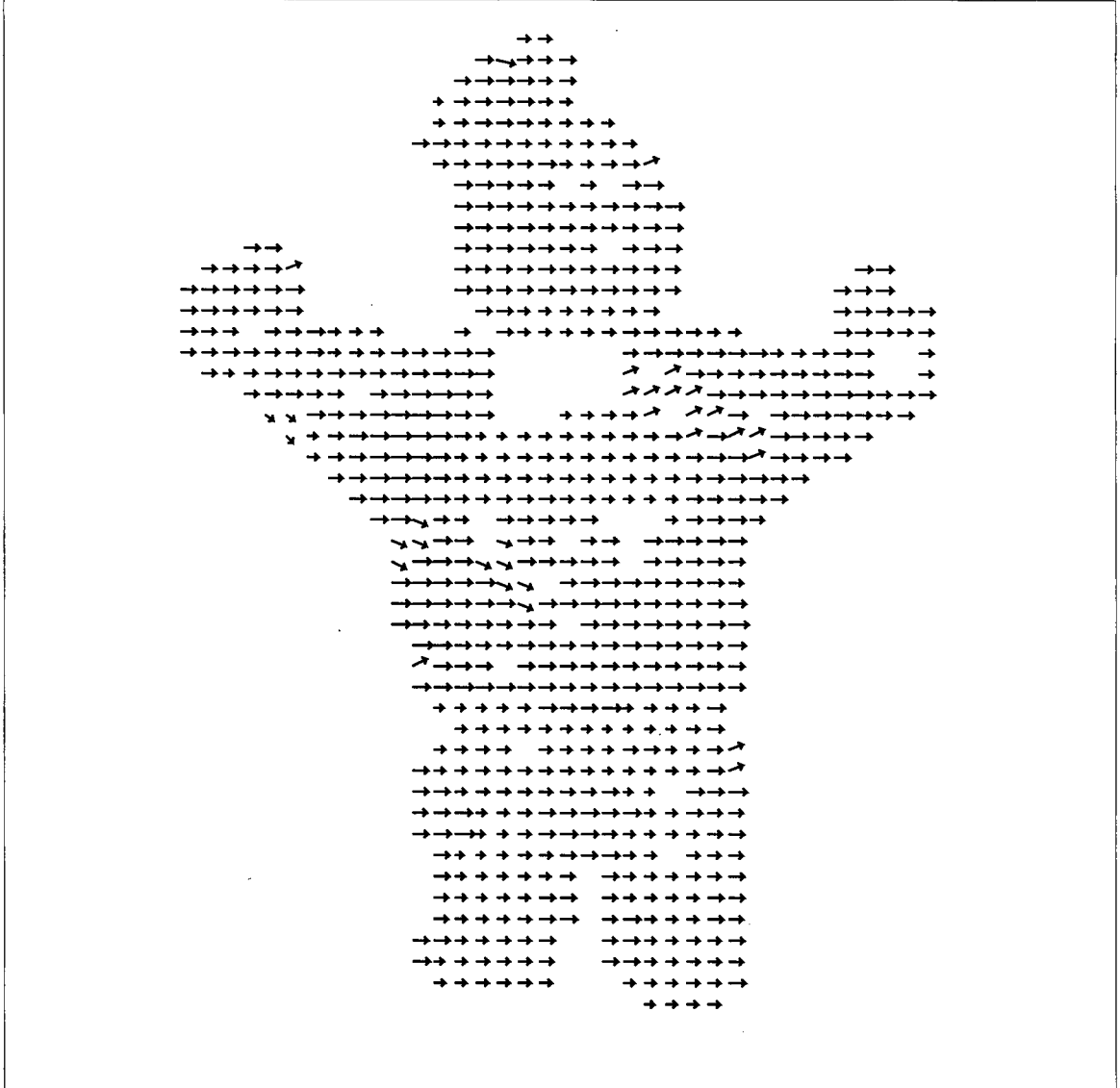


Figure 5.41: Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. The image was convolved with a DOG filter and the window function was a Gaussian of $\sigma = 3$, $\|\nabla E\| \geq 2.5$. Red light source.

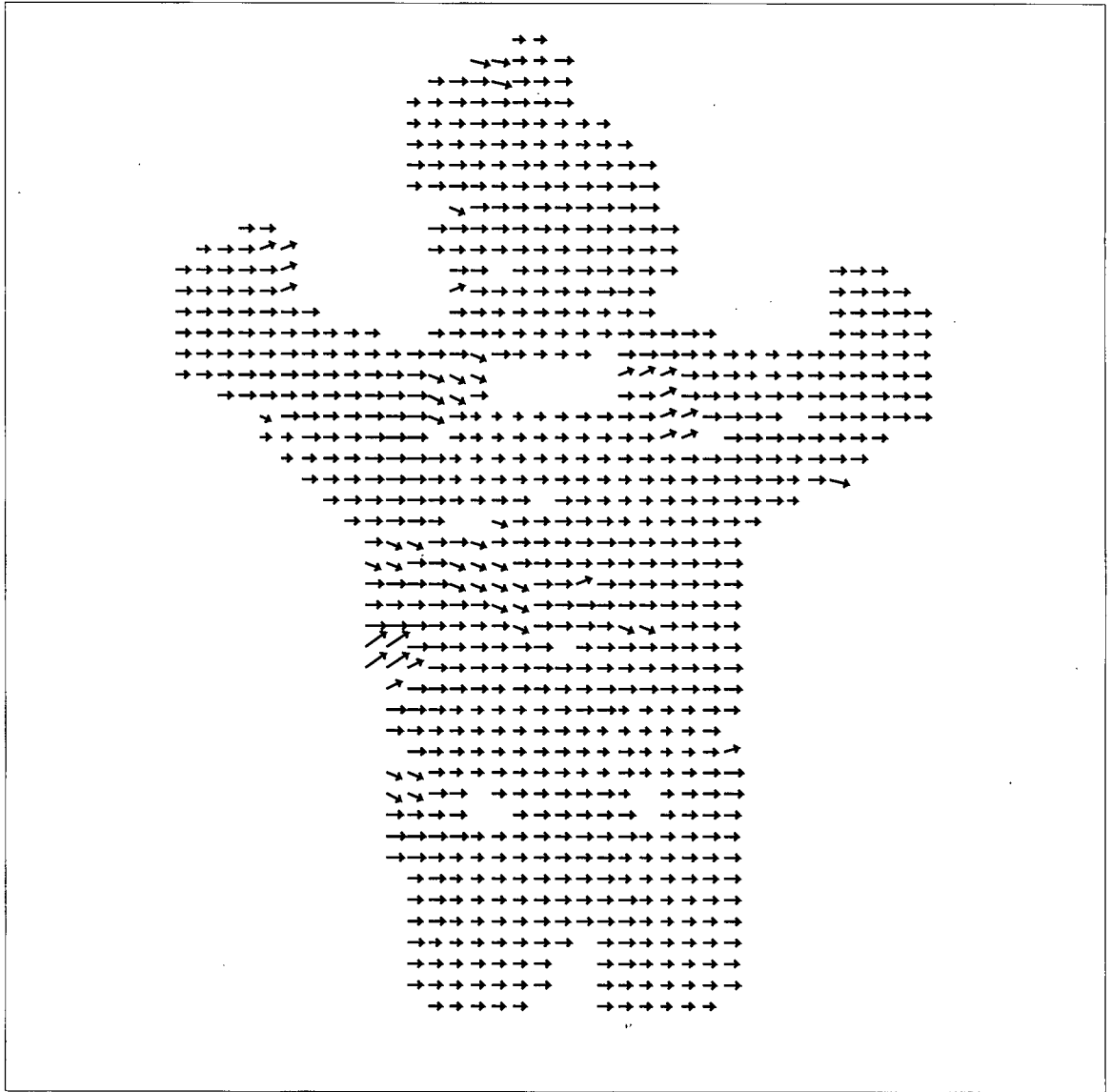


Figure 5.42: Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. The image was convolved with a DOG filter and the window function was a Gaussian of $\sigma = 3$, $\|\nabla E\| \geq 2.5$. Green light source.

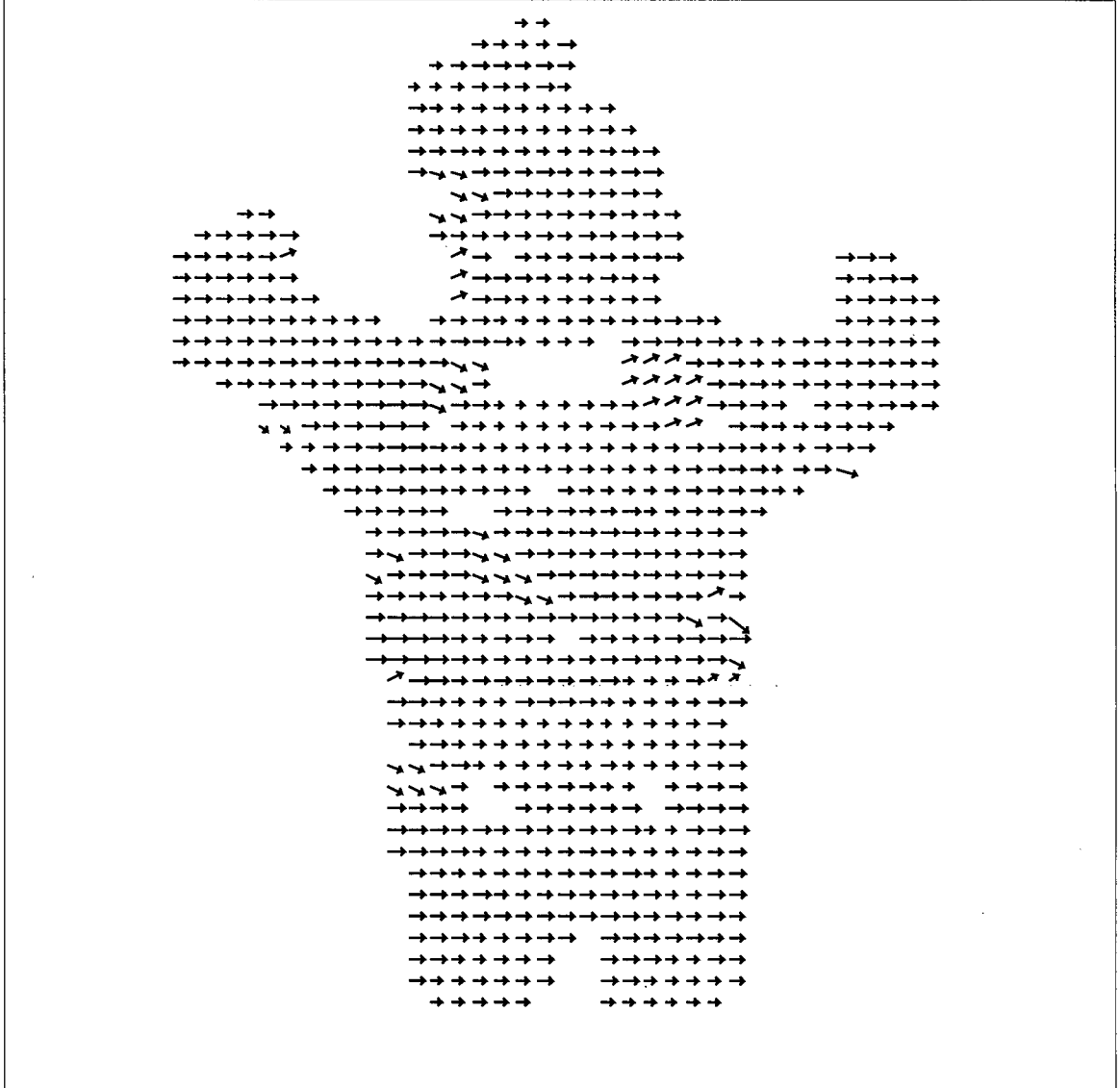


Figure 5.43: Translating Stay-Puft Marshmallow Man, Little, Bülthoff and Poggio technique. The image was convolved with a DOG filter and the window function was a Gaussian of $\sigma = 3$, $\|\nabla E\| \geq 2.5$. Blue light source.

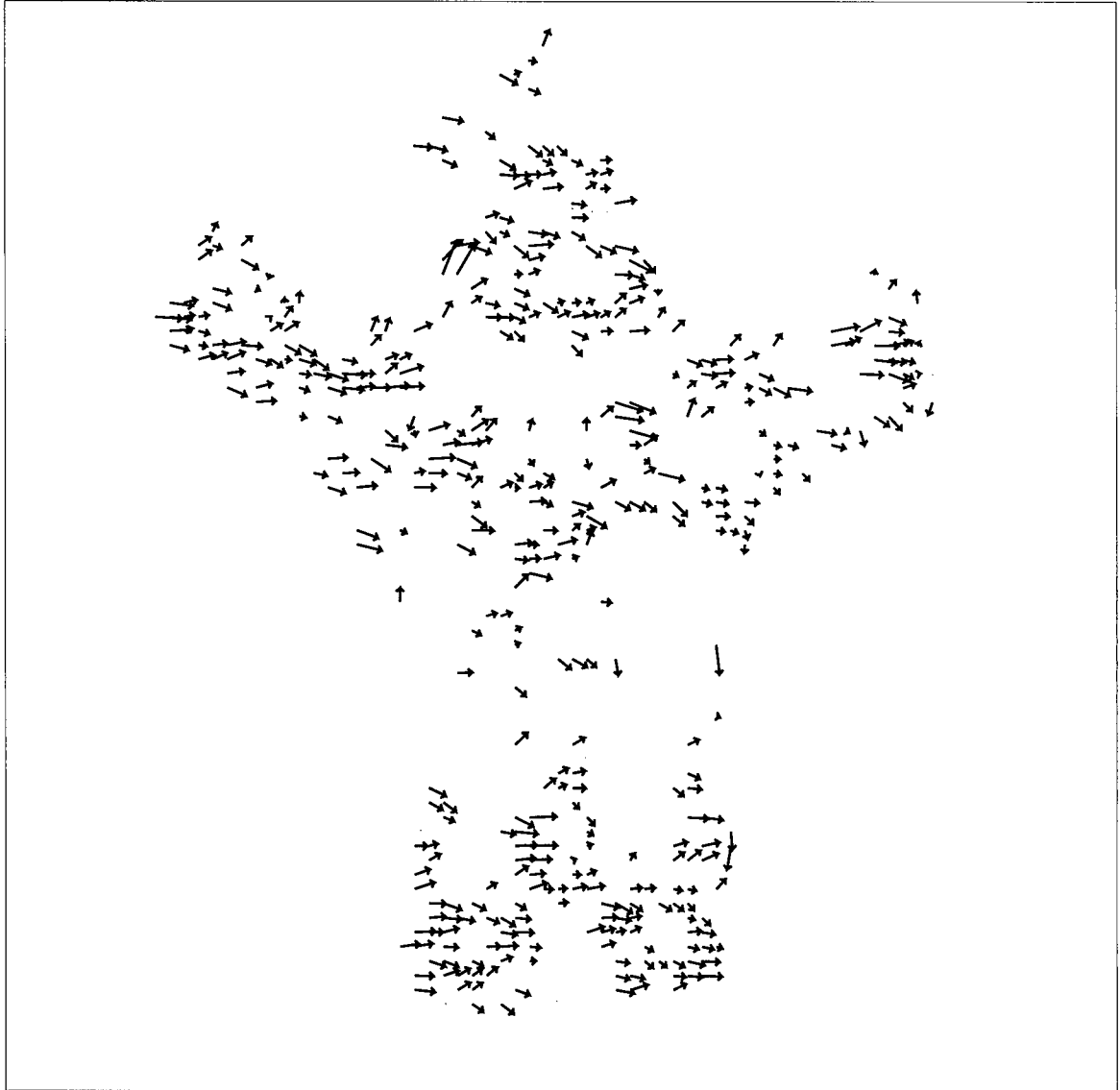


Figure 5.44: Translating Stay-Puft Marshmallow Man, Anandan's technique, $c_{min} \geq 0.01$, 10 iterations, window size 5x5. Red light source.

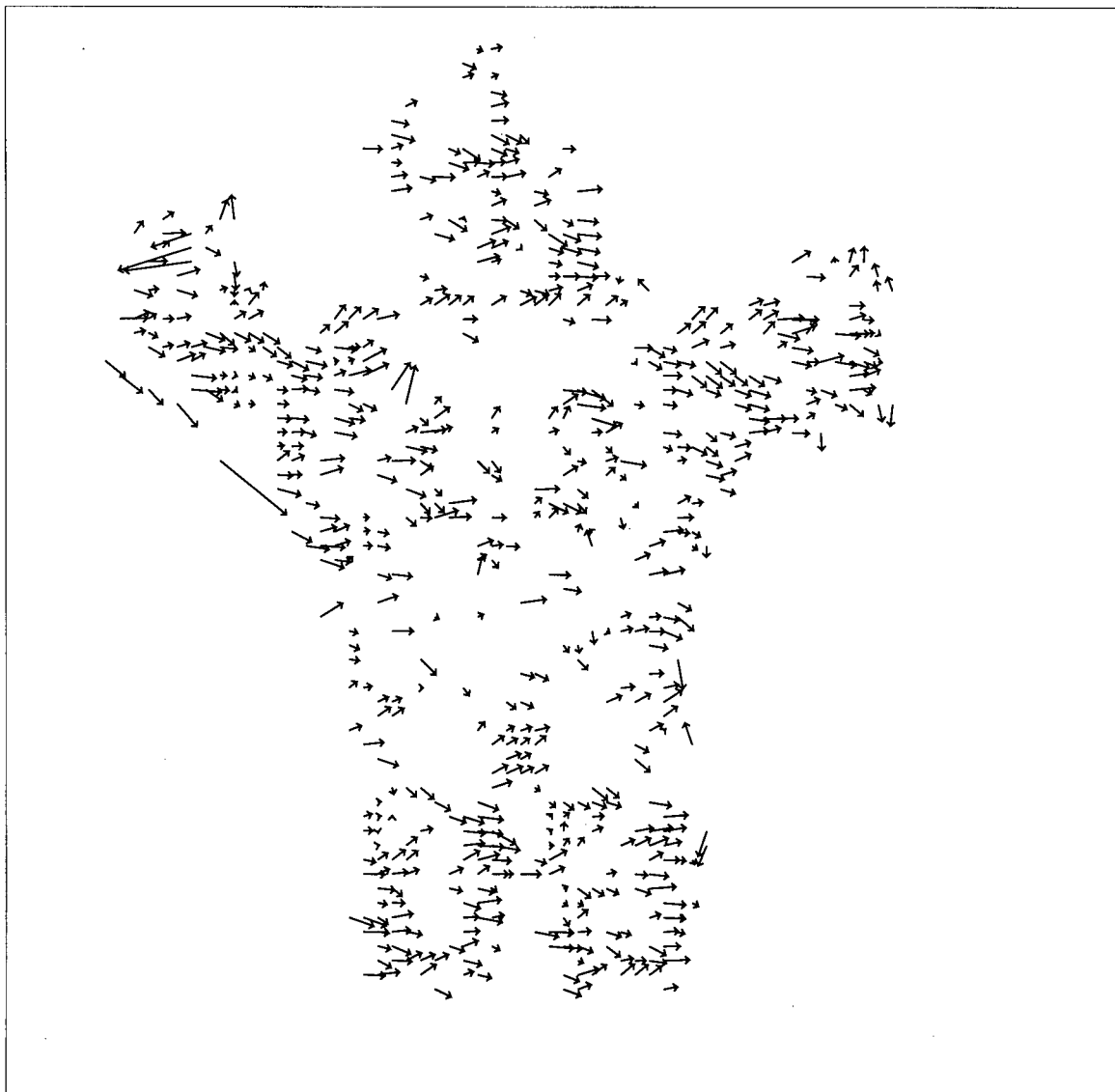


Figure 5.45: Translating Stay-Puft Marshmallow Man, Anandan's technique, $c_{min} \geq 0.01$, 10 iterations, window size 5x5. Green light source.

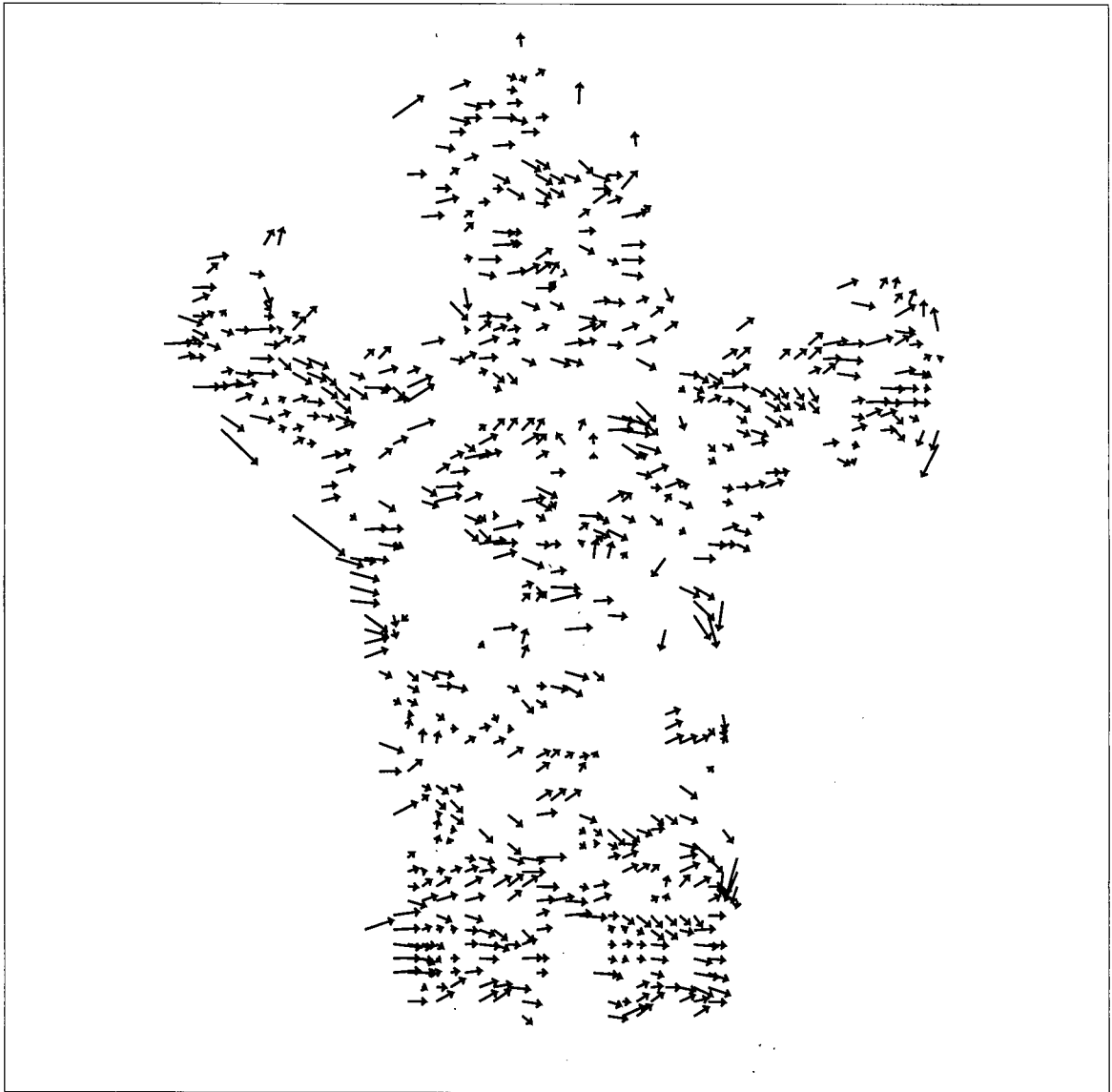


Figure 5.46: Translating Stay-Puft Marshmallow Man, Anandan's technique, $c_{min} \geq 0.01$, 10 iterations, window size 5x5. Blue light source.

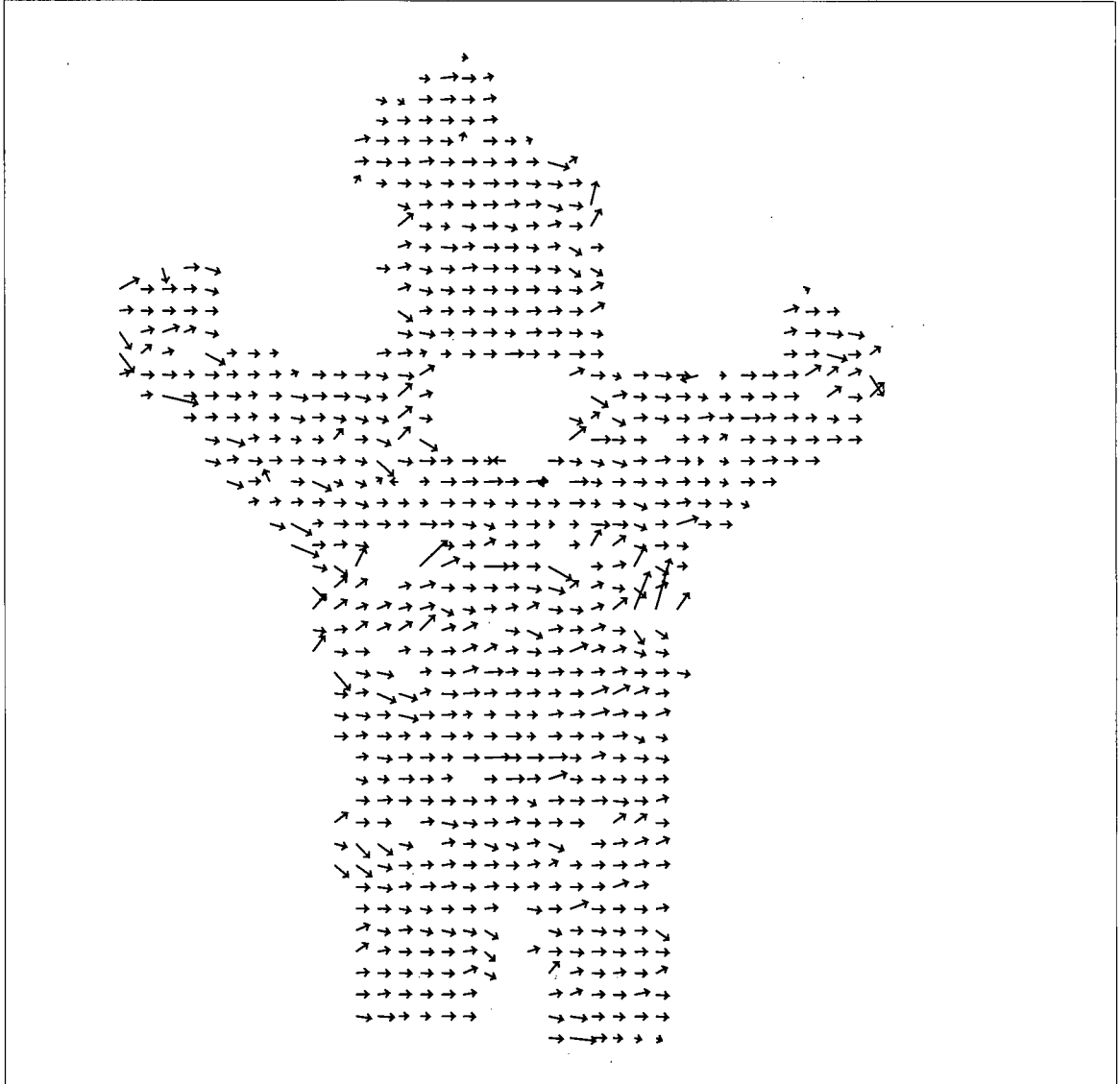


Figure 5.47: Translating Stay-Puft Marshmallow Man, multiple light source optical flow, $\|\nabla E\| \geq 2.5$, $\sigma = 1.5$.

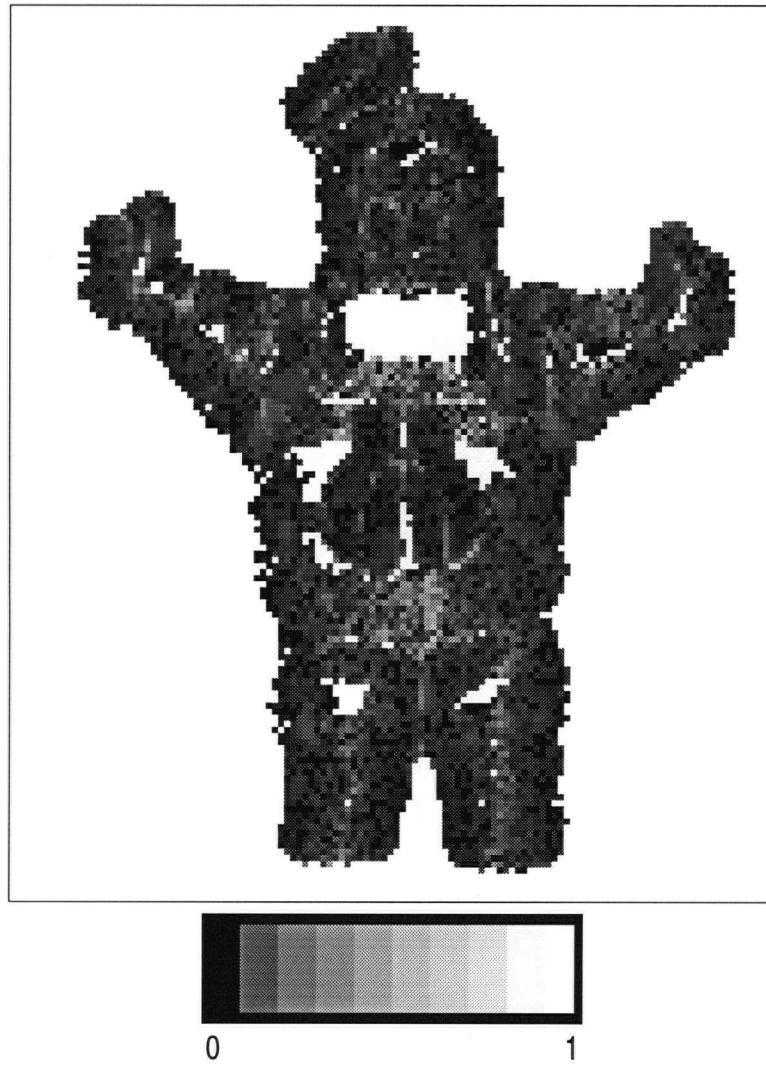


Figure 5.48: Translating Stay-Puft Marshmallow Man. Relative error for multiple light source optical flow.



Figure 5.49: Translating Stay-Puft Marshmallow Man. Condition number for multiple light source optical flow.

Chapter 6

Conclusions

The measurement of optical flow is the first step in the analysis of sequences of time varying images. The second step is to use the optical flow field, for example, to recover the 3-D structure and the relative motion of the objects in the scene. Multiple light source optical flow is a method that allows the computation of a dense local representation of the flow field. A comparison with other methods shows that multiple light source optical flow performs very well when the surfaces exhibit significant shading, away from boundaries and surface markings. In that sense, multiple light source optical flow is complementary to matching techniques; it works well when matching methods fail and vice-versa. The method fails at surface markings because brightness changes at these points are dominated by local features and are independent of the direction of illumination. At image points where the redundancy of the three light sources can be exploited, the accuracy and robustness of the method can be assessed by the use of confidence measures. The locality of the calculations allows the estimation of velocities for a deforming surface. The inflating and deflating balloon is one example.

There are many methods for computing optical flow but very few of them run at frame rates on conventional hardware. This thesis describes an implementation of multiple light source optical flow that allows the processing of 3.5 to 4.0 frames per second and gives a dense optical flow field. The method is particularly suited for real time processing for it does not involve any iterative technique, it is easy to parallelize and performs a uniform, local, point-by-point calculation. Even though not implemented in the real-time system, spatial smoothing poses no special problems. Data storage capability is the main limitation for performing temporal smoothing. A study of the response of the system for different speeds shows that the behaviour is linear for velocities up to 1.5 pixels/frame. This implementation has used three spectral channels in the visible spectrum. The use of other wavelengths is also possible. However, this would imply the use of a different camera.

A method related to multiple light source optical flow is photometric stereo[17]. Multiple images taken under different conditions of illumination are used in this case to solve the shape-from-shading problem. Multiple light source optical flow makes no assumptions about the reflectance properties of the surface undergoing motion. However, if the reflectance properties are known multiple light source can be applied simultaneously with photometric stereo to recover the 3-D structure and 3-D motion of the surface. Integrating these methods would be an important contribution to vision systems.

Appendix A

Least squares solution of linear equations

Following reference [27], let us consider a linear system given by

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (\text{A.1})$$

where $\mathbf{A} \in \mathbf{R}^{m \times n}$, $\mathbf{x} \in \mathbf{R}^{n \times 1}$ and $\mathbf{b} \in \mathbf{R}^{m \times 1}$. If we define an error term given by

$$\mathbf{e} = \mathbf{b} - \mathbf{A} \mathbf{x} \quad (\text{A.2})$$

then

$$\begin{aligned} \text{trace}(\mathbf{e}^T \mathbf{e}) &= \text{trace}((\mathbf{b} - \mathbf{A} \mathbf{x})^T (\mathbf{b} - \mathbf{A} \mathbf{x})) \\ &= \text{trace}(\mathbf{b}^T \mathbf{b} - \mathbf{b}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}) \end{aligned} \quad (\text{A.3})$$

and using that

$$\frac{d}{d\mathbf{A}}\text{trace}(\mathbf{A}\mathbf{B}) = \mathbf{B}^T \quad (\text{A.4})$$

we obtain

$$\frac{d}{d\mathbf{x}}\text{trace}(\mathbf{e}^T\mathbf{e}) = -2\mathbf{A}^T\mathbf{b} + 2\mathbf{A}^T\mathbf{A}\mathbf{x} \quad (\text{A.5})$$

If we set this derivative to be zero, then

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \quad (\text{A.6})$$

Bibliography

- [1] E.C. Hildreth, "The Measurement of Visual Motion", MIT Press, Cambridge, Mass., 1984.
- [2] A. Verri, T. Poggio, "Motion field and optical flow: qualitative properties", IEEE Trans. Pattern Analysis and Machine Intelligence (11), pp. 490-498, 1989.
- [3] R.J. Woodham, "Multiple Light Optical Flow", Proceedings of the Third International Conference in Computer Vision, pp. 42-45, Osaka, Japan, 1990
- [4] B.K.P.Horn, B.G. Schunck, "Determining Optical Flow", Artificial Intelligence(17), pp. 185-203, 1981.
- [5] B.Lucas, T.Kanade, "An Iterative Image Registration Technique with Application to Stereo Vision", Proc. DARPA Image Understanding Workshop, pp. 121-130, 1981.
- [6] J.L.Barron, D.J.Fleet, S.S.Beauchemin, "Performance of Optical Flow Techniques", IJCV, 12(1), pp. 43-77, 1994.
- [7] S.S.Beauchemin and J.L.Barron, "The Computation of Optical Flow Techniques", ACM Computing Surveys, 27(3), pp. 433-467, September 1995.

- [8] David S. Walkins, "Fundamentals of Matrix Computations", John Wiley & Sons, 1991.
- [9] B.K.P.Horn, R.W.Sjoberg, "Calculating the Reflectance Map", *Appl. Opt.*, 18, pp. 1770-1779, 1979.
- [10] R.Battiti, E.Amaldi, C.Koch, "Computing Optical Flow Across Multiple Scales: An Adaptive Coarse-to-Fine Strategy", *IJCV*, 6:2, pp. 133-145, 1991.
- [11] J.Kearney, W.Thompson, D.Boley, "Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization", *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-9*, pp. 229-244, 1987.
- [12] E.Simoncelli, E. Adelson, D. Heeger, "Probability distributions of Optical Flow", *Proc. Conf. Comput. Vis. Patt. Recog.*, Maui, pp. 310-315, 1991.
- [13] H.-H.Nagel, "On the Estimation of Optical Flow: Relations between Different Approaches and Some New Results", *Artificial Intelligence*, 33, pp. 299-324, 1987.
- [14] S.Uras, F.Girosi, A. Verri, V.Torre, "A Computational Approach to Motion Perception", *Biol.Cybern.* 60 ,pp. 79-87, 1988.
- [15] A.Verri, F.Girosi, V.Torre, "Differential Techniques for Optical Flow", *J.Opt.Soc.Am A*, 7(5), pp. 912-922, 1990.
- [16] David Hsu, "Multiple light source estimation of the 2-D motion field when reflectance properties are known", Bachelor's Thesis, Computer Science, UBC, April, 1995.

- [17] R.J. Woodham, "Gradient and Curvature from Photometric Stereo Including Local Confidence Estimation", J.Opt.Soc.Am. A, vol 11, pp. 3050-3068, 1994.
- [18] V.Markandey, B.E.Flichbaugh, "Multispectral Constraints for Optical Flow Computation", Proc. 3rd International Conference in Computer Vision, Osaka, Japan, pp. 38-41, 1990.
- [19] N.Ohta, "Optical Flow Detection by Color Images", Proc. IEEE Conf. Image Processing, Singapore, pp. 801-805, 1989.
- [20] J. Little, H. H. Bülthoff, T. Poggio, "Parallel optical flow using local voting", Proc. 2nd intern. Conf. Comput. Vis., Tampa, pp. 454-459, 1988.
- [21] P.Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion", IJCV, 2, pp. 283-310, 1989.
- [22] A.Singh, "An Estimation-Theoretic Framework for Image-Flow Computation", Proc. 3rd Intern. Conf. Comput. Vis., Osaka, pp. 168-177, 1990.
- [23] E.H.Adelson, J.B.Bergen, "Spatiotemporal Energy Models for the Perception of Motion", J.Opt.Soc.Am. A, 2, pp. 284-299, 1985.
- [24] D.J.Heeger, "Optical Flow Using Spatiotemporal Filters", IJCV, 1, pp. 279-302, 1988.
- [25] D.J.Fleet, A.D.Jepson, "Computation of component image velocity from local phase information", IJCV, 5, pp. 77-104, 1990.
- [26] "User's Guide, TMS320C4x", Texas Instruments, 1993.
- [27] B.K.P.Horn, "Robot Vision", The MIT Press, 1986.