Visual Tracking with Deformation Models

James M. Rehg[1] Andrew P. Witkin[2]

August 1990
CMU-CS-90-156

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]Department of Electrical and Computer Engineering, Carnegie Mellon University

[2]School of Computer Science, Carnegie Mellon University

The research described in this report was supported in part by Apple Computer and Silicon Graphics, Inc

Keywords: Machine Vision, Motion Analysis, Visual Tracking

Abstract

Visual tracking is a class of correspondence-based motion analysis concerned with describing the evolution of image features through a motion sequence. This paper presents a novel solution to the tracking problem with two major components: a 2D *deformation model* that constrains the interpretation of motion, and a set of energy-based *match criteria* that specify image features to be used in tracking. An important property of our formulation is that is *interactive*— the user guides the system to a starting point, from which it solves a series of local optimization problems. This results in fast and flexible operation over a wide class of image motions. We give experimental results for two real-world image sequences.

1 Introduction

This report presents a flexible, interactive system for model-based image tracking. It is based on an underlying deformation model that captures the change in shape and position of a region of pixels over a sequence of images. A variety of energy-based match criteria are employed to connect the deformation model to image data. We develop an estimation algorithm for deformation parameter recovery and demonstrate its performance on two real-world image sequences. Our algorithm has been implemented in a system that runs at interactive speeds on a low end graphics workstation.

Their are four major sections in this paper. First, we describe and motivate the tracking problem. Second, we present the mathematical framework and implementation of our solution. In the third section we contrast our approach to previous vision research in motion. In the fourth, we present the results of applying our system to two real image sequences.

2 Motivation

One of the most basic characteristics of our world is the way it changes with time. Computer vision has a rich history of techniques for analyzing the time-varying image field produced by a dynamic scene. We address a particular type of correspondence-based motion analysis, that we term *visual tracking*. Fundamentally, the tracking problem consists of identifying regions in an image which correspond to interesting structures in the imaged scene, and following them across an image sequence. The trajectories of the image features provide basic information about the kinematic behavior of structures in the scene.

Two practical examples of tracking problems considered in this paper are *3D object tracking* and *cell structure kinematic analysis*. In Section 5 we describe the application of our system to sample problems in these two domains. In the paragraphs that follow, we give a brief overview of these two problem areas and the motivation for our research.

In a typical 3D object tracking scenario, a robot employs a camera to monitor the position of moving objects in its environment. This requires identifying a group of pixels in an image that correspond to an interesting object, and following these pixels through a motion sequence. Two sample

problem domains are autonomous navigation of vehicles in traffic [5], in which object tracking of adjacent vehicles is vital for collision avoidance, and visual robot servoing [17], in which robot end-effector velocity is controlled to match that of a reference object in the scene.

A second application of visual tracking occurs in medical imaging. The dynamic behavior of cell structures gives important clues about their biological function. *Quantitative fluorescence microscopy* is a promising technology for measuring chemical and molecular dynamics within living cells, that draws on the fields of fluorescence chemistry, biochemistry, spectroscopy, and image processing [16]. The basic approach is to engineer flourescent biological materials that can be incorporated into the normal structure of a living cell. Dynamic cell behavior can then be recorded by illuminating the cell with specific wavelengths of light and capturing the emitted light with a video camera. Sophisticated image processing techniques are necessary to turn this raw video data into measurements of specific cell parameters [3]. As an initial approach to this important problem, we are employing tracking techniques to measure cell structure kinematics.

3 Tracking Problem

The essential components of the tracking problem are the image features selected for tracking, the model employed to describe their motion, and the algorithm that computes model parameters from an image sequence. Our choices in these areas reflect our four major system objectives:

1. **Flexibility** In order to address the markedly different types of image motion that result from the two tracking problems described above, our formulation must encompass a variety of image features and motion models, and allow the user to tailor his choice to the application. Flexibility in specification is achieved by separating the model of image deformation from the match criteria, and developing a general framework for each.

2. **Interactive Capability** In order to be truly flexible, our system must allow a user to interact with the model on the fly and adapt its behavior to the problem. Thus we leave the issue of initial feature selection to the user, who can employ his special knowledge of the

application. The energy-based formulation we present allows us to include the user interface to the model in the same framework as the match criteria.

3. **Speed** Our ultimate goal is to track image features at frame rate (30 frames/sec for NTSC video), thereby following the change in the image at the rate at which it is occurring. The system we describe in this paper is operating at interactive speeds on a Silicon Graphics Personal Iris.

4. **Focus of Attention** One important benefit of an interactive approach is that the user can focus the attention of the system to specific areas in the image, in contrast to general motion techniques which must perform computations at every pixel. The corresponding reduction in computational overhead is the primary reason we believe we can reach frame rate speeds without specialized hardware.

Our solution to the tracking problem consists of three parts: motion model, match criteria, and estimation algorithm. The motion model, called a *patch*, is a deformable sheet with mass, that is allowed to change shape in a constrained manner. It is connected to the images through the match criteria, a set of energy functions on the image that exert forces on the sheet, causing it to follow the motion. The user initializes the tracking system by positioning a patch on an area of the image and selecting the match criteria. As each new image is acquired, the estimation algorithm computes the patch model parameters that best explain the motion in the new frame. Therefore, a sequence of images generates a sequence of optimization problems. In the subsections that follow, we describe each of the above parts in more detail and explain how they succeed as a whole in achieving our performance objectives.

3.1 Patch Model and Optimization Framework

The function of the patch model is to describe the deformation of a variable-sized region of an image. The patch itself is a two dimensional sheet, whose shape is described by a parametric function $\mathbf{w}(\mathbf{q}, \mathbf{r})$ that maps *material* coordinates, $\mathbf{r} = [u, v]^1$, to *image* coordinates. $\mathbf{q}$ is a vector of state variables

[1] $\mathbf{b} = [a1, a2, \ldots, an]$ defines an n-element column vector, and $\mathbf{b}^t$ denotes its transpose

that determine the shape and position of the patch. If $\mathbf{x} = [x, y]$ is a point in image coordinates, then the relation $\mathbf{x} = \mathbf{w}(\mathbf{q}, \mathbf{r}) \quad 0 \leq (u, v) \leq 1$ defines a region of pixels in the image. Furthermore, if the states are functions of time, then $\mathbf{x}(t) = \mathbf{w}(\mathbf{q}(t), \mathbf{r})$ describes a time-varying pixel deformation that can be used to model image motion. Therefore, the patch serves two purposes: it defines the set of pixels being tracked, and it constrains the interpretation of their motion. The advantage of the patch framework is that a motion model can be specified independently of the choice of features for recovering the motion.

We restrict ourselves to deformations that can be described by polynomials in material coordinates. In this case, the state variables are polynomial coefficients. To obtain a mapping that is linear in these variables, we write:

$$\mathbf{x} = \mathbf{R}\mathbf{p} \tag{1}$$

where $\mathbf{R}$ is a matrix formed from the elements of $\mathbf{q}$, and $\mathbf{p}$ is a vector function of material coordinates. In a first order model, for example, $\mathbf{R}$ is a homogeneous transformation matrix and $\mathbf{p} = [u, v, 1]$. In this case, the patch is an affine transformation of the unit square, and its state variables consist of four coefficients of rotation/scaling/shearing and two of translation. Alternatively, a second order model can be easily obtained by letting $\mathbf{p} = [u^2, v^2, uv, u, v, 1]$ and adding six more state variables.

Within this framework, the estimator's goal is to generate a sequence of state vectors, $\hat{\mathbf{q}}_i$, one for each image, that best describe the motion in the sense of being the local minimum of a set of objective functions. In the next subsection, we describe some sample match criteria and their associated objective functions. In the remainder of this subsection, we develop the optimization framework that leads to the estimation algorithm.

Consider the problem of finding the vector $\hat{\mathbf{q}}$ that is a local minimum of some function $V(\mathbf{q})$. A standard class of numerical solutions to this problem, know as Steepest Descent Methods [4], involve iteratively solving the equation

$$\dot{\mathbf{q}} = -\frac{\partial \mathbf{V}}{\partial \mathbf{q}} \tag{2}$$

The above equation can be viewed as a dynamic model for the patch, driven by the forcing function $-\partial V/\partial \mathbf{q}$. For many optimization problems, including our patch model, Equation 2 exhibits a *scaling problem* [4]. The difficulty is

that the forcing function is allowed to affect all of the parameters equally, in spite of the fact that a differential change in some of the parameters, like rotation, may have a much stronger effect on image coordinates than a change in others, such as translation. Stability may be improved by introducing a weighting matrix that normalizes the generalized force by adjusting it to the scales of its parameters. The desired normalization can be accomplished using a *mass matrix*, which in physical systems expresses the relationship between force and acceleration. By assigning mass to points in the patch model, we can apply standard methods of classical mechanics to derive its mass matrix and equations of motion.

A physical patch model with mass possesses both kinetic and potential energy. The potential energy, $V(\mathbf{q})$, of the patch consists of the objective functions that constrain its shape and location, and insure that the equilibrium state of the patch dynamic system represents a local solution to the minimization problem. The kinetic energy, $T(\mathbf{q})$, describes the physical behavior of the patch in response to forces, and influences the path it follows to the minimum. The physical patch model bears a strong resemblance to the physical animation models described in [19], although the energy functions have a different interpretation. An additional side effect of providing a "physical" patch model is that human intuition about the behavior of everyday physical systems can be exploited in designing the user interface to the model [18].

To derive the dynamic equations of the patch, we use a modified form of Lagrange's equations [7], in which the normal time derivative of $\partial T/\partial \dot{\mathbf{q}}$ is ommitted, leading to a first order, rather than second order, dynamic system. The equation of motion is given by:

$$\frac{\partial T}{\partial \dot{\mathbf{q}}} - \frac{\partial (T - V)}{\partial \mathbf{q}} = 0$$

To obtain a discrete dynamic model, material coordinates are sampled over a grid and a mass m_i is assigned to each discrete image point $\mathbf{x}_i$. We then have

$$T = \frac{1}{2} \sum_i m_i \dot{\mathbf{x}}_i^t \dot{\mathbf{x}}_i$$

Now, writing Equation 1 in a more useful form, we obtain

$$\mathbf{x}_i = \mathbf{A}_i^t \mathbf{q} \tag{3}$$

where

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{p}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_i \end{bmatrix}$$

and $\mathbf{q}$ is the usual state vector. Note that if $\mathbf{p}$ is an n-vector, then $\mathbf{A}$ is a $2n \times 2$ matrix. Since $\mathbf{A}_i$ is not a function of time, we have point velocity

$$\dot{\mathbf{x}}_i = \mathbf{A}_i^t \dot{\mathbf{q}}$$

and kinetic energy

$$T = \frac{1}{2} \sum_i m_i \dot{\mathbf{q}}^t \mathbf{A}_i \mathbf{A}_i^t \dot{\mathbf{q}}$$

or

$$T = \frac{1}{2} \dot{\mathbf{q}}^t \mathbf{M} \dot{\mathbf{q}}$$

where $\mathbf{M}$ is a constant symmetric block diagonal mass matrix

$$\mathbf{M} = \sum_i m_i \mathbf{A}_i \mathbf{A}_i^t = \begin{bmatrix} \mathbf{m} & \mathbf{0} \\ \mathbf{0} & \mathbf{m} \end{bmatrix}$$

with submatrix

$$\mathbf{m} = \sum_i m_i \mathbf{p}_i \mathbf{p}_i^t$$

It follows trivially that

$$\frac{\partial T}{\partial \dot{\mathbf{q}}} = \mathbf{M}\mathbf{q}$$

and

$$\frac{\partial T}{\partial \mathbf{q}} = 0$$

leading to the dynamic equation

$$\mathbf{M}\dot{\mathbf{q}} = -\frac{\partial \mathbf{V}}{\partial \mathbf{q}} \tag{4}$$

Comparing Equation 4 to Equation 2, we see that the only difference is the presence of the mass matrix, $\mathbf{M}$. To obtain the conditioning matrix, we invert $\mathbf{M}$ and move it to the right hand side of Equation 4.

3.2 Match Criteria

The right hand side of Equation 4 can be viewed as a generalized force, acting in parameter space to drive the system to an operating point at which the potential energy, or alternatively the objective function, is at a minimum. The objective function specifies the match criteria, and determines how the patch responds to the image data. In general, the total potential energy V is a sum of objective functions, each of which derives from a specific image feature. The main advantage of this formulation is that the user can tailor his choice of image features to the problem without altering the deformation model.

In our implementation of this framework, we have focused on two energy functions, which we describe below. The first is termed *blob energy*, and it measures squared intensity error between groups of pixels in two images. A blob is a window on the patch that connects it to a group pixels in the underlying image, and provides a basic region tracking capability. Each blob contains a set of stored pixels, and can compute the intensity error between these pixels and the set of pixels it "sees" as a result of the current patch state. Specifically, if $\mathbf{q}_0$ denotes some initial position of the patch in image I_0, then

$$V_b(q) = \sum_i [I_0(\mathbf{w}(\mathbf{r}_i, \mathbf{q}_0)) - I(\mathbf{w}(\mathbf{r}_i, \mathbf{q}))]^2 \tag{5}$$

is the blob objective function for the current image. The state estimate that minimizes V_b gives the deformation of the patch in image I which is locally best in an RMS sense in tracking a group of pixels between frames.

A second useful objective function is the simple spring potential defined by

$$V_s = k\|\mathbf{x}_a - \mathbf{x}_b\|^2 \tag{6}$$

If $\mathbf{x}_a$ is a function of material coordinates, then Equation 6 describes the scenario in which the spring is affixed to the patch at one end, and either the image or a mouse at the other. If $\mathbf{x}_b$ is a function of a second set of material coordinates, then V_s models a spring constraint between two patchs, that can be used for simple coordinated motion tracking.

The above two examples are by no means exhaustive. A large variety of image features can be employed for motion interpretation, and our energy-based formulation provides a simple framework for incorportating them into

our estimator. We are currently investigating the use of contour-based objective functions related to those described in [9].

3.3 Estimation Algorithm

The primary value of Equation 4 is that it reduces the optimization problem for the patch to the simulation to equilibrium of a first order dynamic system. The numerical methods literature is filled with techniques for performing this simulation [10]. For our implementation, we obtained a simple estimation algorithm by applying Euler's method to Equation 4, resulting in the update equation:

$$\mathbf{q}^{j+1} = \mathbf{q}^{j} - \rho \mathbf{M}^{-1} \frac{\partial V}{\partial \mathbf{q}} \tag{7}$$

where ρ is the step size. As stated earlier, this is equivalent to a steepest descent algorithm with conditioning matrix $\mathbf{M}^{-1}$.

The algorithm is applied to image sequences in the following manner: in the first image, the user positions the patch over some region of interest, thereby initializing the patch state to some $\mathbf{q}_0$. The user then selects from a menu of image features, such as blobs and springs, and constructs the objective function. As the images are processed sequentially, the final state estimate from the ith image is taken as the initial starting point in tracking the $(i+1)$th image. For each new image, Equation 7 is simulated to equilibrium, resulting in an estimate $\hat{\mathbf{q}}_i$ of patch state that represents the best local match to the given initial region of pixels. The result of applying this algorithm to m images is a set of state vectors, $\mathbf{q}_0$ to $\mathbf{q}_{m-1}$, that describe the evolution of the patch with time.

4 Previous Work

The most important characteristic of the tracking problem is its description of motion in terms of *correspondences* between image features. Concern for the *perceptual identity* of features in motion [15] is what distinguishs correspondence-based motion problems, of which tracking is an example, from gradient-based approachs such as optical flow [8]. A second difference is the temporal scale of the problem. Optical flow computation, for example, is a local operation in time. It is useful because the instantaneous velocity

in an image at a single time instant has a meaningful interpretation. The position function for an image feature, on the other hand, is only meaningful over a time *interval,* and so the tracking problem is *global* in time.

Our definition of the tracking problem and its proposed solution differ from conventional correspondence-based motion work [1, 14] in two important ways: the use of an explicit 2D model for constraining motion, and the interactive nature of our solution. Our deformable patch is clearly related to the 3D object models presented in [11, 12]. However, most motion techniques employ implicit constraints on motion, rather than explicit models. An example is the smoothness constraint for optical flow, introduced in [8]. The only exception in the literature is in the case of rigid body motion, where 3D motion of world points between image frames can be characterized by eight paramters of rotation and translation [14]. Two other examples of the use of explicit 3D models in rigid body motion are [6, 13].

There are many cases, however, in which the rigid body motion assumption is not appropriate. There are even cases, such as the actin experiments we discuss in Section 5, in which an explicit 3D model is undesirable. In general, our 2D deformation model has two main advantages over conventional motion techniques: robustness and speed. By describing the motion of a *region* of pixels with a small number of parameters, we obtain robustness. This is in contrast to the optical flow problem, in which the number of unknowns (two at each pixel) far outweighs the available measurements. The second advantage of our model is that it restricts our focus to a region of pixels, allowing us to perform fewer computations and track at higher frame rates. Other systems employing a window-based approach are described in [2, 5]. An advantage of our method, however, is that it does not require specialized hardware to run at interactive speeds.

The second difference between our work and previous attempts at motion analysis is the *interactive* nature of our solution. As described in [9], one of the main advantages of an interactive approach is that it frees the system from making the arbitrary high-level decisions often required for successful low-level interpretation. In our framework, for example, if the user knows that a scene contains only translational motion, he can easily add a constraint to the model causing it to prefer pure translation. Or if he knows that a scene produces strong intensity contours, he can incorporate them into the objective function and improve tracking. It can be very difficult for a general vision system to infer these image properties, and failure to do so may lead

to incorrect low-level interpretations.

5 Experiments

We have implemented our system on a Silicon Graphics Personal Iris 4D-20 workstation, and applied it to two sets of real-world motion data. In the first sequence, some frames of which are given in Figure 1, a single patch model with four smaller blobs tracks a coffee mug which is rotating, translating, and zooming. Each of the blobs tries to track the region of pixels it covered in the first frame of the sequence. It is interesting to note the graceful failure of the blob in the lower left corner of the patch, when the pixels it was following dissapeared from the scene in the final few frames. The patch as a whole was able to track the scene correctly.

The second set of data, shown in Figure 2, was taken from a growth stimulation experiment performed at the Center for Flourescence Research (CRF) at Carnegie Mellon University [16]. It depicts the formation and transport of actin fibers in a single Swiss 3T3 cell. The patch model successfully tracks the cell nucleus, in spite of the fact that it fades into the background.

6 Conclusion and Future Work

We have described an interactive approach to visual tracking that exhibits the desirable properties of flexibility, speed, and robustness. At the heart of our solution is an energy-based deformable model that constrains motion interpretation, but permits user-directed application to a variety of scenarios.

Extensions of our work could proceed along two lines. First, as discussed in Section 4, estimator performance is linked to the local intensity structure of the image. It would be desirable to find an analytic relationship between the intensity pattern and estimator convergence. Also, the numerical methods we currently employ should be augmented to include second order techniques [4] and analyzed for stability and convergence. In addition to further analysis, it would be interesting to experiment with additional match criteria, such as contour information, and extensions of the implementation to second order deformations.

We would like to thank the Image Understanding group at CMU for use

of the Calibrated Imaging Laboratory, in which the first motion sequence was obtained. We are also grateful to Michel Nederlof and Kevin Ryan of the CFR for providing the actin data.

References

[1] **S. Barnard and W. Thompson**, "Disparity Analysis of Images", *IEEE Trans. PAMI*, **2**(4), 1980, 333-340.

[2] **P. Burt, C. Yen, and X. Xu**, "Multiresolution Flow-through Motion Analysis", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1983, Washington, DC, 246-252.

[3] **P. Conrad, M. Nederlof, et. al.**, "The Correlated Distribution of Actiun Myosin and Microtubules at the Leading Edge of Migrating Swiss 3T3 Fibroblasts: Analytical Immunofluorescence", *Journal of Cell Biology*, submitted for publication.

[4] **J. Dennis and R. Schnabel**, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[5] **E. Dickmans and A. Zapp**, "Autonomous High Speed Road Vehicle Guidance by Computer Vision," *Preprint, IFAC-Congress 1987*, **4**, Munich, 1987, 232-237.

[6] **D. Gennery**, "Tracking Known Three-Dimensional Objects", *Proc. AAAI-82*, 13-17.

[7] **H. Goldstein**, *Classical Mechanics*, Addison-Wesley, 1980.

[8] **B. K. P. Horn and B. Schunck**, "Determining Optical Flow," *Artificial Intelligence*, **17**, 1981, 185-203.

[9] **M. Kass, A. Witkin, and D. Terzopoulos**, "Snakes: Active Contour Models", *Int. J. of Computer Vision*, **1**(4), 1987, 321-331.

[10] **W. Press, B. Flannery, et. al.**, *Numerical Recipes in C*, Cambridge University Press, 1988.

[11] **D. Terzopoulos, A. Witkin, and M. Kass**, "Energy Constraints on Deformable Models: Recovering Shape and Non-Rigid Motion", *AAAI-87*, 755-760.

[12] **D. Terzopoulos, A. Witkin, and M. Kass**, "Symmetry-Seeking Models and 3D Object Reconstruction", *Int. J. of Computer Vision*, **1**, 1987, 211-221

[13] **D. Thompson and J. Mundy**, "Model-Based Motion Analysis- Motion from Motion", *Fourth Int. Symp Robotics Research*, 299-309.

[14] **R. Tsai and T. Huang**, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE Trans. PAMI*, **6**(1), 1984, 13-26.

[15] **S. Ullman**, *The Interpretation of Visual Motion*, The MIT Press, 1979.

[16] **A. Waggoner, R. DeBiasio, et. al.**, "Multiple Spectral Parameter Imaging," *Methods in Cell Biology*, **30**, 1989, 449-478

[17] **L. Weiss, A. Sanderson, and C. Neuman**, "Dynamic Sensor-Based Control of Robots with Visual Feedback," *IEEE Robotics and Automation*, **RA-3**, 5, October, 1987.

[18] **A. Witkin, M. Gleicher, and W. Welch**, "Interactive Dynamics", To be published in *Computer Graphics*.

[19] **A. Witkin and W. Welch**, "Fast Animation and Control of Nonrigid Structures", *CMU Tech Report CS-90-103*, 1990.

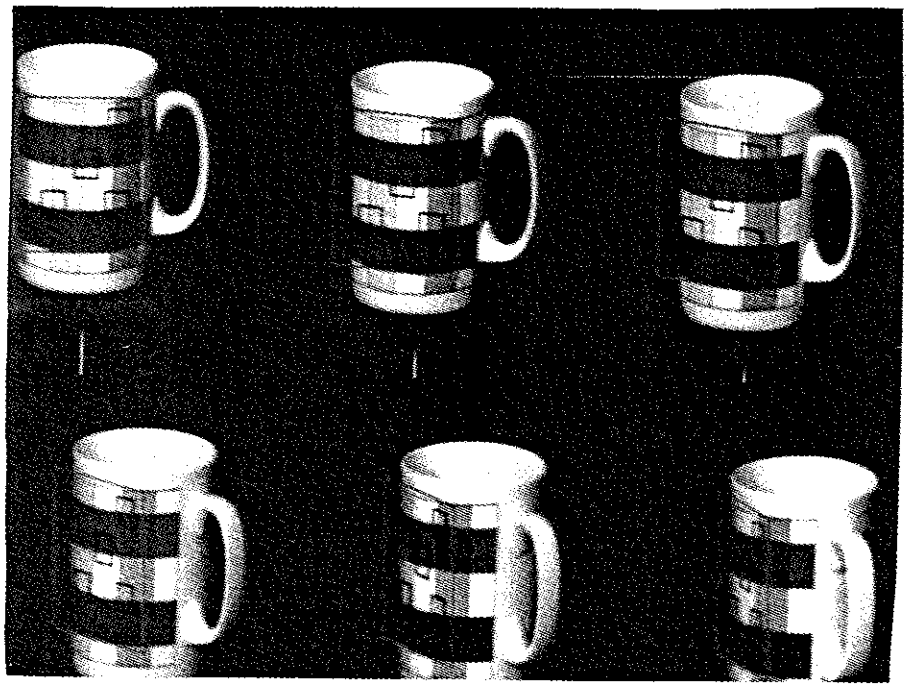

Figure 1: Mug Sequence: a patch model with four blobs was applied to a 13 frame sequence consisting of a rotating, translating, and zooming mug.

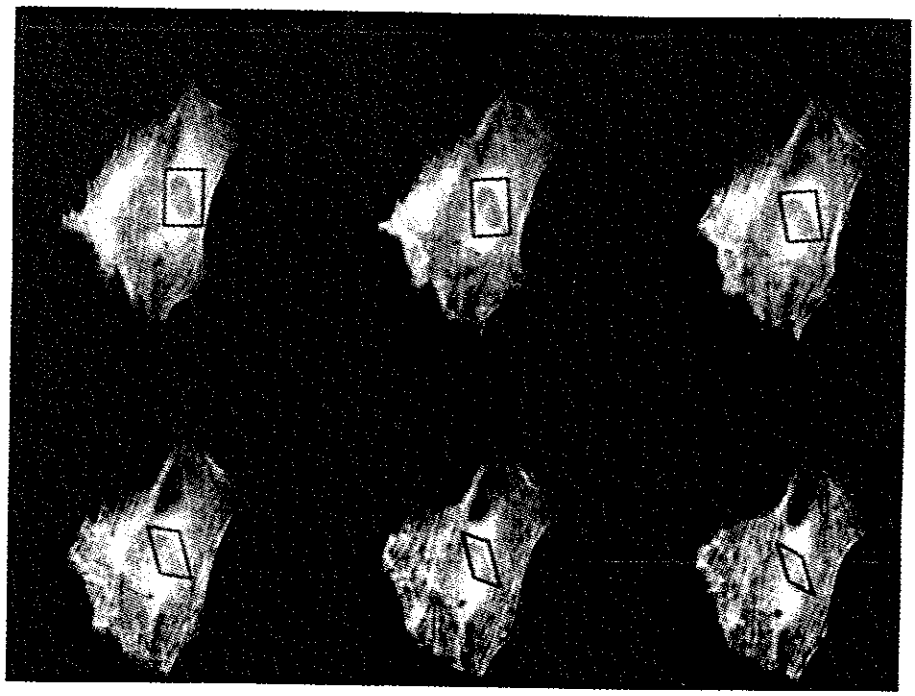

Figure 2: Actin Sequence: a Swiss 3T3 cell, injected with a flourescent analog of actin, responds to growth stimulus over a 22 frame sequence. A patch with a single blob (not shown) tracks the nucleus.