Report Number: WUCS-94-22

1994-01-01

# Cell Tracking using a Distributed Algorithm for 3D Image Segmentation

Vikas Awasthi, Keith W. Doolittle, Guru Parulkar, and James G. McNally

We have developed and tested an automated method for simultaneous 3D tracking of numerous, flourescently-tagged cells. The procedure uses multiple thresholding to segment individual cells at a starting timepoint, and then iteratively applies a template-matching algorithm to locate a particular cell's position at subsequent time points. To speed up the method, we have developed a distributed implementation in which template matching is carried out in parallel on several different server machines. The distributed implementation showed a monotonic decrease in response time with increasing number of servers (up to 15 tested), demonstrating that the tracking algorithm is well suited to... Read complete abstract on page 2.

# Cell Tracking using a Distributed Algorithm for 3D Image Segmentation

Vikas Awasthi, Keith W. Doolittle, Guru Parulkar, and James G. McNally

Complete Abstract:

We have developed and tested an automated method for simultaneous 3D tracking of numerous, flourescently-tagged cells. The procedure uses multiple thresholding to segment individual cells at a starting timepoint, and then iteratively applies a template-matching algorithm to locate a particular cell's position at subsequent time points. To speed up the method, we have developed a distributed implementation in which template matching is carried out in parallel on several different server machines. The distributed implementation showed a monotonic decrease in response time with increasing number of servers (up to 15 tested), demonstrating that the tracking algorithm is well suited to parallelization, and that nearly real-time performance could be expected on a parallel processor. Of four different template matching statistics tested for 3D tracking of amebae from the cellular slime mold Dictyostelium discoideum, we found that the automated procedure performed best when using a correlation statistic for matching. Using this statistic, the method achieved a .985% success rate in correctly identifying a cell from one timepoint to the next. This method is now being used regularly for 3D tracking of normal and mutant cells of D. discoideum, and as such provides a means to quantify the motion of many cells within a three-dimensional tissue mass.

# Cell Tracking using a Distributed Algorithm for 3D Image Segmentation

Vikas Awasthi, Keith W. Doolittle, Guru Parulkar and James G. McNally

WUCS-94-22

July 1994

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899

To appear in Journal of Bioimaging.

# Cell Tracking using a Distributed Algorithm for 3D Image Segmentation

Vikas Awasthi[†], Keith W. Doolittle[‡], Guru Parulkar[†] and James G. McNally[‡•]
[‡]Dept. of Biology, [†]Dept. of Computer Science, [•]Institute for Biomedical Computing
Washington University, St. Louis MO. 63109

## Abstract

We have developed and tested an automated method for simultaneous 3D tracking of numerous, fluorescently-tagged cells. The procedure uses multiple thresholding to segment individual cells at a starting timepoint, and then iteratively applies a template-matching algorithm to locate a particular cell's position at subsequent timepoints. To speed up the method, we have developed a distributed implementation in which template matching is carried out in parallel on several different server machines. The distributed implementation showed a monotonic decrease in response time with increasing number of servers (up to 15 tested), demonstrating that the tracking algorithm is well suited to parallelization, and that nearly real-time performance could be expected on a parallel processor. Of four different template matching statistics tested for 3D tracking of amebae from the cellular slime mold *Dictyostelium discoideum*, we found that the automated procedure performed best when using a correlation statistic for matching. Using this statistic, the method achieved a .985% success rate in correctly identifying a cell from one timepoint to the next. This method is now being used regularly for 3D tracking of normal and mutant cells of *D. discoideum*, and as such provides a means to quantify the motion of many cells within a three-dimensional tissue mass.

key words: 3D microscopy, cell tracking, image segmentation, *Dictyostelium*

## Introduction

Cell motion plays a critical role in morphogenesis in virtually all animals, but the molecular methods which govern this motion are not well understood. One reason for our ignorance is that morphogenesis typically occurs within a 3D mass of tissue, and so in many cases it has been difficult to visualize cell movement in vivo. This is now possible by using either 3D confocal microscopy [1] or 3D computational optical sectioning microscopy [2]. Because of its sensitivity to low light levels, we are using the latter method to obtain time-lapse 3D images of cell motion during morphogenesis in a model developmental system, the cellular slime mold *Dictyostelium discoideum*. By determining how cells move normally during morphogenesis, and then comparing this motion to the behavior in a variety of different mutants, we hope to identify the particular molecules and cell-cell interactions that are important for regulating cell movement during multicellular development in *Dictyostelium*. To quantify the behavior of normal and mutant cells, we require an automated method for fast and accurate 3D tracking of moving objects within a living specimen.

A successful automated-tracking scheme should be able to simultaneously track many moving objects from a biological scene, since this is often the norm in a tissue or within a cell. Simultaneous tracking of multiple objects can generate enough data for statistical analysis, and also help reveal any global patterns of movement. Ideally, such tracking should be computable in real time so as to provide immediate feedback to the biologist, who often must screen numerous specimens before identifying a few suitable for detailed scrutiny.

Several different schemes have been developed previously to track objects in biological scenes. Among the simplest are nearest-neighbor methods which first identify objects by a segmentation applied at each observation time or "time point", and then construct paths by linking nearest-neighbor objects at successive time points [3,4]. Such methods are often effective for *in vitro* studies of cell motility [5-7], where the density of cells can be kept low and temporal sampling kept high.

In addition to the nearest-neighbor method, a few other methods have also been developed for biological tracking. These other procedures have been used because the nearest-neighbor method fails when the objects are close enough together and the observation times far enough apart, such that the nearest neighbor at the next timepoint is not necessarily the same object. Under these conditions, which arise for example in *in vivo* analyses of particle or cell motility, tracking has been accomplished by using some form of matching statistic [8-13]. In these approaches, the most likely position of an object at the next timepoint has been determined by a probability measure which selects among several candidate objects located in the neighborhood of the object's former position. The best match seeks to conserve some combination of the object's features, including its size, shape, intensity and/or expected displacement. When the features are based on the object's size, shape and intensity, the matching procedure is referred to as template matching [10-12].

Of all of the preceding tracking procedures for biological data, all but one were designed for 2D tracking. The exception is a nearest-neighbor method designed for tracking cells moving *in vitro* within a 3D collagen lattice [6]. Unfortunately, as noted above this nearest-neighbor approach is generally not suitable for 3D tracking of objects within a living specimen, where it is often difficult to limit the number and density of labeled objects. Even when this is possible, it is frequently preferable to label many objects to assess if neighboring trajectories are correlated. Moreover, in living specimens, temporal sampling must be kept low enough to avoid photodamage to the organism and/or bleaching of a fluorescent label. Thus, under these conditions, matching methods which discriminate among neighboring objects would appear more suitable, but these are more computationally demanding, and so far have been implemented in only 2D. More elaborate 3D tracking schemes have been developed for tracking planes or missiles, but virtually all of these methods incorporate assumptions about the kinematics of the object's motion. Such assumptions are potentially risky in biological tracking where little is known about the the laws governing an object's motion, and where the very purpose of tracking is in fact to learn about these kinematics.

Here, we describe and evaluate an automated 3D tracking scheme that uses multiple thresholding to identify a number of different cells at a starting timepoint, and then iteratively applies template matching to track cells from one timepoint to the next. We have evaluated this method's tracking accuracy for four different template-matching statistics, and obtained the most reliable performance with a correlation statistic.

Our 3D implementation of template matching introduces significantly more computational complexity compared to previous 2D template-matching schemes [10-12]. In these schemes, a 2D template of side $n$ requires $n^2$ pixel-by-pixel calculations to compute a matching statistic. To find the location of best match at the next time point, this matching statistic is then computed at all possible locations in a search volume that is larger than the template volume. If one side of the search volume is of length $n+k$, then in a 2D matching scheme, $k^2$ matching statistics must be computed. Since each statistic require $n^2$ calculations, the complexity is of the order $n^2k^2$ for 2D template matching. For 3D template matching, this increases to $n^3k^3$. For the values of $n$ and $k$ that we have used in this study, this corresponds roughly to a ~100 fold increase. To address this added complexity, we have developed a distributed implementation for the automated tracking method to evaluate its potential for speed up by parallel computing.

The preceding methods have been implemented and tested for the specific problem of 3-D time-lapse analysis of fluorescently labeled *Dictyostelium discoideum* cells [14,15] imaged by computational optical-sectioning microscopy. The biological problem is to understand the mechanisms controlling cell movement within the *D. discoideum* tissue mass, which consists of approximately $10^5$ cells, of which typically about 200 are fluorescently labeled. A complex assortment of motile behaviors has been observed within this tissue mass [16], and so a thorough quantitative analysis of cell trajectories is required both to categorize the types of motion, and to rigorously compare normal motion to the abnormal motion observed in a variety of different mutants.

In what follows, we first discuss information about the *D. discoideum* images that influenced some of the specific features of our tracking procedure. Then we briefly describe a manual method for 3D tracking that we implemented to generate reliable data for testing the automated procedure. We then describe this automated procedure and evaluate its accuracy. We also report data on the extent of speed up by a distributed implementation, and conclude with a discussion of how the methods described here have been used in *D. discoideum* tracking, and also how the methods could be improved.

## Image Characteristics

Individual *D. discoideum* cells are labeled with a fluorescent dye that becomes concentrated primarily in one or several intracellular vesicles, and at a much lower level throughout the cells' cytoplasm. The labeled cells are mixed with unlabeled cells and the mixture allowed to form the tissue mass or *mound*, which is observed on a conventional fluorescence microscope equipped with a CCD camera cooled to -40$^{o}$C. 3D images of the mound are collected at 2 minute intervals to generate time-lapse data sets as large as 256x256x64x60 in $(x,y,z,t)$, which, at 2 bytes per pixel, yield ~500 Mbytes of data. To remove out-of-focus light, 3D images from each time point are processed with a regularized linear least-squares restoration procedure [17]. After processing (Figure 1), the labeled cells appear either as a single bright spot or as a cluster of several bright spots. The bright spots correspond to the labeled intracellular vesicles. The size of a bright spot varies from around 3-10 voxels. Sometimes the labeled vesicle or vesicles are surrounded by a uniformly weaker region of fluorescence intensity, which presumably represents dye distributed more extensively throughout the cell's cytoplasm. Because of limitations [18] in the linear least-squares restoration procedure, the labeled vesicles, which are essentially spherical, appear elongated along the optical axis $(z)$, and also possess residual cones of out-of-focus light (due to the microscope's point-spread function) that are faintly visible in $xz$ cross sections of an image.

The image histogram for such 3D image data is Gaussian in shape and lacks bi- or multimodal features that could be used to segment the image based on cell vs. background intensities. The cell intensities occupy about 90% of the image's dynamic range, with individual cells varying widely in their average intensity, due to the fact that different cells accumulate different amounts of dye. As a consequence, the residual out-of-focus cones from bright cells are frequently brighter than the actual cell intensities from dim cells. A further complication, for the purpose of tracking a cell containing several labeled vesicles, is that the cell's image changes over time because the labeled vesicles move around within the cell's cytoplasm.

## Manual Tracking

In order to generate reliable trajectory data for testing a fully automated tracking method, we first developed a method requiring extensive user interaction (Figure 2). In this manual procedure, a user may view a 3D image at a starting time point in any $xy$ and $xz$ cross section. A specific $xy$ and $xz$ cross section may be selected either by their numbered position within the image stack, or by stepping sequentially through the stack of images (by changing $z$ in $xy$ sections or changing $y$ in $xz$ sections). Typically, the cross sections selected for viewing are those where the cell to be tracked appears brightest. The images are displayed with a linear gray scale, whose maximum and minimum may be set interactively. In the displayed image, intensities above the maximum are set to white, and intensities below the minimum are set to black. Normally, a user sets the maximum so that the brightest cells are saturated and many dimmer cells are visible, and sets the minimum so that background light is black.

To segment the image, the user adjusts a binary threshold, such that the cell to be tracked is above threshold, but no larger than 5-10 voxels (the expected size of the labeled portion of a cell). To assess the choice of threshold, the binary image is displayed in both $xy$ and $xz$ cross section next to the unthresholded $xy$ and $xz$ cross sections. When a satisfactory threshold has been set, the user may move a cursor over the binary image of the cell to be tracked, and then click a mouse button. The voxel so identified by the cursor position is used as a seed to initiate a recursive algorithm for region-growing by flood filling [19]. The volume so identified is then displayed in red in the $xy$ and $xz$ cross sections of the binary-thresholded image. If the user is not satisfied with the region selected, then he or she may repeat the process of binary thresholding. When the user indicates a satisfactory region has been identified, then a center of mass for that region is automatically calculated and stored.

To track a cell, the user then displays the 3D volume at the next timepoint adjacent to the image of the preceding timepoint, which can be used as a frame of reference to help locate the cell at the new timepoint. The preceding steps of binary thresholding and region growing can then be applied at the new point, and a new center of mass computed and stored. In this way, a trajectory for the cell is generated.

To verify these trajectories, the center-of-mass point from each timepoint is displayed as a red dot and superimposed on successive time-lapse images of the cells. Typically these time-lapse images are viewed as an xy projection of the 3D volume, and then displayed rapidly in sequence to generate a movie of cell motion. When a cell has been tracked correctly, then a red dot consistently overlays and follows the movement of the cell image throughout the time-lapse movie. If the red dot and cell diverge, then the trajectory can be deleted from the data set. The reliability of tracking depends largely on the skill of the user, but with experience users can manually track dozens of cells without having to discard any trajectories after the verification step. Unfortunately, the manual method is quite tedious. A typical cell moving over 60 timepoints may require an hour or longer to track.

## Fully-Automated Tracking

This method is a two-step process. At a starting time point, cells are identified by multiple thresholding. For each thresholded cell, a template is extracted around that cell's center of mass. In the second step, that template is used to find the position of best match at the next time point. That position then determines a new template, and template matching is applied to the next time point.

### Identification of cells at a starting time point

We used a combination of multiple thresholding, dilation, region growing and binary object forests [20] to automatically identify cells at a starting timepoint (Figure 3). Multiple thresholds (Figure 3a-c), as opposed to just a single threshold, were necessary because of the wide variance among average cell intensities. A single high threshold would have identified only the brightest cells, whereas a single low threshold while identifying dimmer cells, would also have merged neighboring bright cells into one larger object.

In practice, we set four threshold levels using a fixed-percentage thresholding method [19]. In this approach, thresholds are calculated based on the ranked ordering of intensities in the image data. In our implementation, the four thresholds corresponded to the intensities encompassing either the top 0.04%, 0.08%, 0.12% or 0.16% of the brightest voxels. These four levels led to the identification on average of 200 cells, and this was deemed sufficient for generating enough quantitative information from a single time-lapse data set.

Automated multiple thresholding leads to several problems in identifying cells. First, it inevitably fragments contiguous labeled regions of some cells into two parts: one above and one below a particular threshold level. To correct this problem, we used the same region-growing scheme that we had used for manual tracking (see above), but augmented by a dilation algorithm [19], which converts *off* voxels to *on* voxels if at least two of the voxel's six adjacent voxels are *on*. The dilation process helps smooth the edges of a cluster of *on* voxels. In most cases, dilation successfully connected fragmented cell volumes (Figure 3d-f). Following dilation, regions were discarded if they did not lie within a certain size range. The lower limit (15 voxels) was set to prevent selection of noise, and the upper limit (850) was set to prevent selection of unincorporated dye particles.

Two other problems are also introduced by multiple thresholding. The first of these is that a cell must only be identified once, even though once above threshold it will remain so at all lower thresholds. Second, nearby bright cells can merge into larger objects when thresholded at lower intensities. To address these problems, we first identified regions at each threshold level using the region-growing routine described above for manual tracking. Then, we constructed binary object forests [20] that encoded a geneology of all the regions identified at different thresholds. In creating such a geneology, region C was defined as child and region P as parent, if region C (present at a higher threshold) is contained in region P (present at a lower threshold). We then computed cell centers only for those regions, P, having no children, C. In this way, we ignored cells present at lower thresholds or regions containing several cells, because in both cases these regions had children at higher thresholds. For example, when two separate cells are close to each other, they will form a single parent region at a lower threshold, but split to form two separate child regions at a higher threshold. At the next highest threshold, these child regions may disappear, or simply shrink in size. In the latter case the new regions would be grandchildren of the first parent region. In any event, cell centers would only be computed at the threshold level where the regions in question had no children at the next highest threshold.

To test the accuracy of this procedure for identifying cells at a starting timepoint, we applied the method to over 100 3D images, and then examined the centers identified by the algorithm. Many different images were tested yielding a total of about 10,000 different cells. In all cases, the method accurately identified cell locations for both bright and dim cells (Figure 3g).

**Tracking of cells by template matching**

Once a cell has been identified at a starting timepoint, a volume surrounding the cell is used as a template for finding the location of best match at the next timepoint. The center of the best-match position defines the cell's center of mass at that timepoint, and that best-matching template provides the new template for searching at the next timepoint, and so on.

*Sizes of template and search volumes*

For template matching, two parameters must be set empirically, since they depend on the size of the object being tracked and the maximum distance that the object might move in a single time interval. The size of the object depends on the magnification used on the microscope. The distance moved by the object depends on the time interval between 3D snapshots, and possibly on the particular strain of cells under examination. All of these variables typically remain the same for a large number of experiments. The two parameters affected by these variables are the size of the template volume, defined as the volume centered about the cell's centroid at timepoint $n$, and the size of the search volume, defined as the volume at timepoint $n+1$ centered about the cell's centroid from timepoint $n$.

For the analysis reported here, the size of the template volume was 11x11x17 voxels, and the size of the search volume was normally 19x19x23. The template volume size was determined by inspection of the images, and by limited trial and error testing in the automated tracking. This volume is larger than the labeled portion of a cell (about 5-10 voxels), and longer in the $z$ direction because the image processing produces an elongation of the object in $z$ and also leaves a residual cone extending beyond the object (see Figure 1, and McNally *et al.*, 1993). The search volume size was determined by examining cell-motion data obtained from the manual tracking method, and by trial and error in actual tracking (see below). The search volume is 8x8x6 larger than the template volume. The search volume's smaller width in $z$ relative to $x,y$ was used because cells were observed to move less in this direction.

*Template - matching statistics*

The template volume at timepoint $n$ is scanned over the search volume at timepoint $n+1$, and at each position in the search volume, a matching statistic is computed. Then, the position in the search volume yielding the best match is defined as the new cell location at timepoint $n+1$.

Four different template-matching statistics were evaluated:

difference: $$S(\mathbf{d}) = \sum_{\mathbf{r} \in \mathbf{T}} \left| I_{n+1}(\mathbf{r}+\mathbf{d}) - I_n(\mathbf{r}) \right|$$

squared difference: $$S(\mathbf{d}) = \sum_{\mathbf{r} \in \mathbf{T}} \left( I_{n+1}(\mathbf{r}+\mathbf{d}) - I_n(\mathbf{r}) \right)^2$$

normalized cross correlation: $$S(\mathbf{d}) = \frac{\sum_{\mathbf{r} \in \mathbf{T}} I_{n+1}(\mathbf{r}+\mathbf{d}) \cdot I_n(\mathbf{r})}{\sqrt{\sum_{\mathbf{r} \in \mathbf{T}} I_{n+1}^2(\mathbf{r}+\mathbf{d})} \cdot \sqrt{\sum_{\mathbf{r} \in \mathbf{T}} I_n^2(\mathbf{r})}}$$

correlation coefficient: $$S(\mathbf{d}) = \frac{\sum_{\mathbf{r} \in \mathbf{T}} \left( I_{n+1}(\mathbf{r}+\mathbf{d}) - \bar{I}_{n+1} \right) \cdot \left( I_n(\mathbf{r}) - \bar{I}_n \right)}{\sqrt{\sum_{\mathbf{r} \in \mathbf{T}} \left( I_{n+1}(\mathbf{r}+\mathbf{d}) - \bar{I}_{n+1} \right)^2} \cdot \sqrt{\sum_{\mathbf{r} \in \mathbf{T}} \left( I_n(\mathbf{r}) - \bar{I}_n \right)^2}}$$

where $S$ is the value of the matching statistic, $I_n$ the intensity function at timepoint $n$, $\mathbf{r}$ a vector in 3-space specifying location in the template volume $\mathbf{T}$, and $\mathbf{d}$ a displacement vector ranging over all possible locations in the search volume. $\bar{I}_n$ represents the mean intensity over the template volume.

These different statistics were tested by using them to track cells whose trajectories had already been determined by the manual tracking procedure, and verified as described above by superimposing the cell centroid over a movie of cell motion. To provide an unbiased data set for testing, we used a sampling of tracks from three different time-lapse data sets collected at different points over a six-month period. In total, 15 different cell tracks spanning 409 timepoints were analysed. For each path and for each statistic, we determined the number of timepoints correctly tracked until the first miss. A miss was identified by comparing the manual and automated tracks and determining if and when they diverged. When the separation between the two consistently exceeded 5 pixels for at least 3 timepoints, then a miss was declared. This was checked by superimposing the tracks on the movies of cell motion, where misses were seen as the automatically tracked position either jumping incorrectly to a neighboring cell or wandering away from the correct cell into background noise surrounding it. By summing the correctly tracked lengths for each of the 15 tracks, we then calculated for each statistic the percentage of the total track length that was correctly matched before the first miss. The results suggested that the correlation statistic would provide the best tracking (Table 1). This was supported by a visual analysis of many other tracked cells using, as before, *xy* projections of time-lapse movies superimposed on the computed cell centroid at each timepoint. This analysis also showed that the correlation statistic yielded accurate tracking for more timepoints than any of the other three statistics.

*Additional tests of template matching by the correlation statistic*

The preceding analysis is a realistic evaluation of actual tracking capability, but if a miss happens early in a track, the remaining data from the verified track are never used. To make use of all of the data available from the verified tracks, and to provide a second test of the tracking procedure, we manually corrected the automated tracking procedure whenever it failed, and then allowed automated tracking to continue from that point. Then we counted the total number of misses over all of the tracks examined. For the data set described above consisting of 15 different tracks, 6 misses occurred over 409 timepoints. This corresponds to a probability of .985 that a cell is accurately tracked over one time step, or a probability of $.985^n$ that the cell is correctly tracked for $n$ timepoints. We also evaluated tracking with the correlation statistic on a second strain of cells (myosin-II-null). For this strain, we examined 16 tracks spanning 411 timepoints obtained from two different time-lapse data sets. This strain shows much less motion than the preceding strain (Ax-2), and so the automated tracking was more accurate with only one miss per 411 timepoints.

The second way we evaluated the correlation statistic's capacity for automated tracking was by visual comparison of automated and manual tracks. When these two different tracks were never separated by more than a cell diameter, tracking was considered successful. Successful tracking occurred over a range of cell behaviors and labeling patterns, including cells which showed little change in their labeling pattern as they traversed either small (Figure 4) or large (Figure 5) distances, and other cells which showed considerable change in their labeling pattern as they traversed large distances (Figure 6).

These visual comparisons were also used to assess why the tracking method failed at certain timepoints. Many failures were either due to a tracked cell moving out of the search volume at the next timepoint, or the entry into the search volume of a cell resembling the tracked cell (Figure 7). These two deficiencies are related. When the search volume size is increased to ensure that the largest jumps can be tracked, then the risk of enclosing similar cells in the volume increases. The search volume size used here was a compromise between these two extremes.

**Distributed Implementation**

The preceding automated tracking method consists of essentially two compute-intensive steps: cell-center recognition and template matching. The former is executed only once (at the starting timepoint), while the latter is executed for every cell at every subsequent timepoint. In our scheme, template matching becomes even more compute-intensive because, of the four template matching statistics evaluated, we opted for the most accurate one which was also the most computationally complex. Clearly by speeding up template matching, the overall cell tracking could be made much faster. One way to reduce computation times for template matching is by parallel computing on several processors. The template matching step is well suited to parallelization, because tracking of one cell is independent of other cells.

As one approach to parallelization, we developed a distributed implementation for template matching. This was accomplished using a single client - multiple server model (Figure 8). The client was a single process running on a machine with a disk large enough to store the full time-lapse data set (~500 Mb). The client stored the image data from the $n^{th}$ and $n+1^{th}$ timepoints in memory, and for each cell being tracked, extracted a template volume from the $n^{th}$ timepoint and a search volume from the $n+1^{th}$ timepoint. These extracted volumes comprised the request packets, which were sent to the servers running on multiple machines. Upon arrival of a request packet, the server process computed the matching statistic for all possible locations in the search volume, and then returned the location of the best match to the client. After receiving all responses for the $n+1^{th}$ timepoint, the client proceeded to examine the $n+1^{th}$ and $n+2^{th}$ timepoints, and so on. Interprocess communication among the client and server processes was accomplished using socket connections over TCP/IP running over Ethernet.

To estimate some of the network overhead incurred by the distributed implementation, we performed a tracking test on a DECstation 5000 acting as both client and server, and then repeated the test on the same machine without networking protocols. To eliminate the possibility of load variations over time, the machine was operated throughout in a single-user mode. The tracking test consisted of 77 search requests. Each search request transferred 20.7 Kb of data and all requests required equal processor time. The 77 requests required 498 sec. without client and server protocols, and 534 sec. with them, thus yielding a 7% overhead for networking (standard deviations on these measurements were less than 1 sec).

We then added machines as servers, and evaluated the potential for speed up in a distributed implementation. First, we measured response times for a tracking test that used either one, two or four homogeneous servers (DECstation 5000's). We found that the distributed implementation yielded a monotonic reduction in response time with an increasing number of up to four servers, the total number of homogeneous servers available to us (Figure 9a). We also measured response times for a heterogeneous array of up to 15 servers (the total number available) that included DECstation 5000's, Sun workstations, a Sparcstation 1041 and a Stardent Titan. We made two changes to the distributed implementation to accomodate heterogeneous servers. First, because the byte-ordering convention differed among these machines, we incorporated byte-ordering conversions into the implementation. Second, because the computing power and loads varied considerably among these machines, we modified the distributed implementation so that the client, rather than allocating requests to the servers in a round-robin loop, instead distributed requests dynamically based on the server's performance. This was achieved by the client first posting two requests to each server, and then posting the next request to whichever server responded first, and then repeating this strategy for additional requests. We also found that this distributed implementation yielded a monotonic reduction in response time as additional servers were added (Figure 9b).

## Discussion

We have described and tested a manual and an automated method for 3D cell tracking of *D. discoideum* cells visualized by computational optical-sectioning microscopy. The manual tracking method appears virtually 100% accurate, but tedious (~1 min to track 1 cell per 1 time step). The automated method is less accurate (.985 probability of accurate tracking per time step), but faster (~7 sec. to track one cell per one time step). We are now using the automated method regularly to track labeled *D. discoideum* amebae. In practice, the tracking procedure is started in the middle of the timelapse data, and a forward and a backward track are computed from that point so as to double the average length of an accurate track. The method has been used to track normal *Dictyostelium* cells as well as cells from one mutant. The results of this analysis have demonstrated that cells exhibit a diversity of motile behaviors, thereby suggesting that cells may be responding to a variety of different cues which affect their guidance and locomotion [16].

The automated method was intentionally designed for flexibility to facilitate adaptation to other 3D tracking problems. There are two steps in the automated procedure, either one of which could be modified to suit other images. For the first step, we found that multiple thresholding was an effective technique for identifying *D. discoideum* cell positions at a starting time point. In other images where multiple thresholding may not prove so advantageous, other segmentation techniques could be substituted. For the second step of template matching at successive timepoints, we found a correlation coefficient to be the most effective matching statistic. This may be a particulary useful statistic for biological tracking, since several 2D tracking schemes have also employed it for template matching [10-12], but other statistics are easily substituted should they prove advantageous. To assess further the method's robustness, we plan to evaluate its performance on other 3D tracking problems, as data become available.

To improve the accuracy of automated tracking of *D. discoideum* several modifications could be made. The tracking method was observed to fail because a cell's displacement exceeded the search volume size, or because two cells, similar in shape and intensity, were present in the same search volume. These deficiencies are related, and could be addressed by first enlarging the search volume and then applying better strategies for distinguishing among several different cells within that volume. One such approach would be to recognize the presence of two or more cells in the search volume (for example, by applying multiple thresholding within the search volume). On those occasions, a second matching statistic, more sensitive to cell shape, could be applied to choose between the two (or more) possible locations.

Unfortunately, matching statistics more sensitive to cell shape cannot be used routinely because cells change shape as they move. To a lesser degree, a cell's intensity also changes over time (primarily because of flicker in the excitation light and bleaching of the dye). These complications highlight a basic limitation in our tracking strategy. Because a cell's image changes over time, any tracking method that relies exclusively on shape and intensity information is likely to fail from time to time.

To address this complication, other information besides the cell's intensity distribution would have to be incorporated into the tracking procedure. One possibility might be to include *a priori* information about how a cell is likely to move. This strategy has been successfully employed for multiple target tracking of fluid particles [21] or missiles [22,23], for instance. Unfortunately, at present we know very little about the "kinematics" governing cell motion. We do know, of course, that cells cannot move large distances in a single time step, and so we have limited the size of the search volume over which template matching is performed. As we gather more data about cell trajectories, we may be able to add other constraints on cell movement that could be incorporated into an estimation method for tracking.

An alternative and complementary approach for incorporating more than just a single cell's shape and intensity distribution in tracking would be interdependent determination of neighboring cell trajectories. At present, cells are tracked entirely independently. But in principle, when several cells enter the same search volume, the tracking method could seek to determine the most likely set of trajectories consistent with each of the cell images (and any other relevant information) from preceding timepoints. Such an approach could yield an improvement in tracking accuracy by simultaneously incorporating shape and intensity information from many cells.

One advantage of the present cell-independent tracking method is that the algorithm is easily parallelized. We took advantage of this feature and created a distributed implementation that yielded significant speed-ups. We tested up to 15 servers, and over this range observed a monotonic increase in performance. This suggests that still further improvement is possible, up to the point at which the number of servers becomes so large that the client can no longer post requests fast enough.

These results suggest that if the algorithm were implemented on a parallel processor, tracking could be fast enough to permit a microscopist to visualize the tracks as they are being obtained. Typically, several hundred cells are labeled in a specimen and 3D snapshots are collected every few minutes, so if a new cell position could be computed on the order of minutes, (i.e. the time between snapshots) then feedback to the experimenter would be fast enough to determine whether continued tracking is desirable. This would require approximately a 100-fold speed up of the tracking per time step achieved on a single DECstation 5000/200. This goal seems achievable, because if we subtract estimated network delays from the response times measured for increasing number of servers (Figure 9), then the distributed implementation tested here appears to provide nearly a linear speed-up. Thus increasing the number of processors should yield further speed ups. An additional favorable factor is that each processor could be several fold faster than the processor (MIPS R3000) used in the current study. Thus there is reason to be optimistic that, at least with the current method, computerized 3D tracking of cells on for example a Silicon Graphics Challenge or a DEC MPP computer could be fast enough to be interactive with data collection.

## REFERENCES

1. Lemasters J J, Chacon E, Zahrebelski G, Reece J M and Nieminen A-L 1993 Laser scanning confocal microscopy of living cells *Optical Microscopy: Emerging Methods and Applications* ed Herman B and Lemasters J J (San Diego:Academic Press) 339-354

2. Agard D A, Hiraoka Y, Shaw P, and Sedat J W 1989 Fluorescence microscopy in three dimensions *Meth. Cell Biol.* **30** 353-377

3. Geerts H, De Brabander M, Nuydens R, Geuens S, Moeremans M, De Mey J and Hollenbeck P 1987 Nanovid tracking: a new atuomatic method for the study of mobility in living cells based on colloidal gold and video microscopy *Biophys. J.* **52** 775-782

4. Lee G M, Ishihara A and Jacobson K A 1991 Direct observation of Brownian motion of lipids in a membrane *Proc. Nat. Acad. Sci.* **88** 6274-6278

5. Dow J A T, Lackie J M and Crocket K V 1987 A simple microcomputer-based system for real-time analysis of cell behaviour *J. Cell Sci.* **87** 171-182

6. Noble P B 1987 Extracellular matrix and cell migration. Locomotory characteristics of Mos-11 cells within a three-dimensional hydrated collagen lattice *J. Cell Sci.* **87** 241-247

7. Soll D R 1988 "DMS," a computer-assisted system for quantitating motility, the dynamics of cytoplasmic flow and pseudoppod formation: its application to *Dictyostelium* chemotaxis *Cell Motil. Cytoskel.* **10** 91-106

8. Thurston G, Jaggi B and Palcic B 1988 Measurement of cell motility and morphology with an automated microscope system *Cytom.* **9** 411-417

9. Askey D B and Herman I M 1988 Computer-assisted analysis of the vascular endothelial cell motile response to injury *Comp. & Biomed. Res.* **21** 551-561

10. Gelles J, Schnapp B J and Sheetz M P 1988 Tracking kinesin-driven movements with nanometre-scale precision *Nature* **331** 450-453

11. Schnapp B J, Gelles J and Sheetz M P 1988 Nanometer-scale measurements using video light microscopy *Cell Motil. Cytoskel.* **10** 47-53

12. Salmon E D, Magers S, Skibbens R and Gliksman N R 1991 Video-enhanced differential interference contrast light microscopy: a method for semi-automatic object tracking in Proceedings of the 49th Annual Meeting of the Electron Microscopy Society of America p. 238-239.

13. Anderson C M, Georgiou G N, Morrison I E G, Stevenson G V W and Cherry R J 1992 Tracking of cell surface receptors by fluorescence imaging microscopy using a charge-coupled device camera *J. Cell Sci.* **101** 415-425

14. Bonner J T 1967 The Cellular Slime Molds Princeton University Press Princeton, N J

15. Devreotes P N *Dictyostelium discoideum*: a model system for cell-cell interactions in development *Science* **245** 1054-1058

16. Doolittle K W, Reddy I, McNally J G 1994 3D analysis of cell movement during normal and myosin-II-null cell morphogenesis in *Dictyostelium* (submitted)

17. Preza C, Miller M I, Thomas L J and McNally J G 1992 Regularized linear method for reconstruction of three-dimensional microscopic objects from optical sections *J. Opt. Soc. Amer. A.* **9** 219-228

18. McNally J G, Preza C, Concello J A and Thomas L J 1994 Artifacts in computational optical sectioning microscopy *J. Opt. Soc. Amer. A* (in press)

19. Russ J C 1990 Computer Assisted Microscopy, Plenum Press New York

20. Nichol D and Fiebig M 1991 Image segmentation and matching using the binary object forest *Imag. and Vis. Comp.* **9** 139-149

21. Wernet M P 1991 Two-dimensional particle displacement tracking in particle imaging veloci-

metry *Appl. Opt.* **30** 1839-1846

22. Sastry C R Simaan M and Kamen E W 1991 An efficient algorithm for tracking the angles of arrival of moving targets *IEEE Transactions on Acoustics, Speech and Signal Processing* **ASSP-39** 242-246

23. Uhlmann J K 1992 Algorithms for multiple-target tracking *Amer. Sci.* **80** 128-141

## TABLES

Table 1: Average % length accurately tracked as a function of matching statistic. The percentages were obtained from 15 different tracks spanning a total of 409 time points.

| Difference | Squared Difference | Cross-correlation | Correlation |
|---|---|---|---|
| 35% | 47% | 55% | 70% |

## FIGURE LEGENDS

Figure 1. *(Note that Figures 1 and 4 are on the same print. Figure 1 is on the bottom.)* XY and XZ cross sections of a 3D image of fluoroscently labeled *Dictyostelium discoideum* cells after processing by a regularized linear-least squares estimation procedure. The horizontal white line indicates the slice plane in either the XY or XZ section from which the corresponding XZ or XY section is viewed. Each cluster of bright pixels represents a labeled vesicle within a cell. The XY section illustrates the variation in the spatial pattern and intensity of labeling. The XZ section demonstrates the residual elongation of labeled cells along the Z axis (compare the cell's length to its width), as well as the residual cones of out-of-focus light emanating from above and below each labeled cell. Scale bar = 10 µm.

Figure 2. A portion of the display for the manual tracking procedure. XY cross sections (a,b) and XZ cross sections (c,d) of the 3D image are shown. The white horizontal line in c and d corresponds to the z position of the focal plane shown in a and b. The cell to be tracked is indicated by the red arrows in a,c. In b,d, a threshold has been selected such that this cell is above threshold and isolated from surrounding cells. The red region in b,d shows the result of the flood-fill algorithm initiated by pointing to any one pixel within the region. In addition to the images shown here, the actual manual-tracking program also displays an xy image from the preceding timepoint, so as to facilitate location of cells at the next timepoint. Scale bar = 12 µm.

Figure 3. An illustration of several steps in the abstraction of cell centers, including multiple thresholding (a-c), region growing (d-f) and the final assignment of cell centers (g). In the first-step of cell segmentation, a gray-scale image (a) was typically thresholded at four different levels, two of which are shown here for illustration (b,c). Note that different objects are visible at different threshold levels. The region-growing process is shown in d-f (the images in a and d are different), where a gray-scale image (d) after thresholding (e) is then expanded by dilation (f). The object represented by A appeared to move coherently in the movie of cell motion, and so presumably corresponded to several labeled vesicles within one cell. The effect of dilation is to merge these two separate domains into one. In contrast, the object represented by B is simply expanded uniformly in all directions. The two isolated single bright pixels are not expanded at all by the dilation algorithm because these isolated pixels lack neighboring bright pixels. The result of these steps and others (see text) is the identification of a center-of-mass point for those cells whose intensities lay within the four thresholds. An example is shown in g, an *xy* cross section of a gray scale image in which cell centers are indicated by open circles. Note that some centers consist of several closely spaced bright spots which were merged by dilation. The large bright blob at the lower right edge of the screen is an example of a region that was not selected because it was too large (see text). Scale bar = 10 µm

Figure 4. *(Note that Figures 1 and 4 are on the same print. Figure 4 is on the top.)* Analysis of automated tracking for a cell with little movement. Gray-scale *xy* cross-sections of the cell at selected time points (1,5,7,30) are shown. The position of the xy subimage displayed is not fixed, but rather shifts so that the tracked cell remains roughly in the center. Cell position as determined by manual tracking is indicated by an open square, and by automated tracking by an open circle. Observe that for this cell at these timepoints the manual and automated tracks are in complete agreement. Also shown (leftmost image) is the superposition of the complete 30-timepoint trajectories determined by each method. Here the view remains fixed so that the actual cell trajectory can be discerned. The trajectories are displayed in a 3D volume viewed by looking along the z axis (the microscope's optical axis). White squares represent the manual track, white circles the automated track, and white lines connect cell positions at successive timepoints. This particular cell moves very little, and so the white circles, squares and lines from the 30 timepoints are virtually superimposed. (Compare to Figure 1 - Figure 1). Scale bar = 10 μm.

Figure 5. Analysis of automated tracking for a cell with significant movement. Format of the figure is as in Figure 1. The centers of mass identified by both tracking methods are close, and always located within the bright region of pixels that defines the cell. This particular cell moves actively as indicated by the trajectories in the topmost image. (Due to the cell's motion, the gray-scale images at timepoints 16,17,20 and 23 are not all centered at the same position). Scale bar = 10 μm.

Figure 6. Analysis of automated tracking for a cell with both significant movement and changing intensity pattern. Format of the figure is as in Figure 1. Observe from the gray-scale images that this cell exhibits a complex intracellular staining pattern, presumably reflecting labeling of several intracellular vesicles which jostle as the cell itself moves. Examination of the movie of cell motion demonstrates that this cluster of bright voxels moves coherently throughout the period of observation. In the course of this locomotion, the manual and automated tracking methods at some timepoints identify the same vesicle (e.g. timepoints 1,13,15), and at other timepoints identify different vesicles (e.g. timepoint 6,12). These occasional disparities are reflected in the trajectories generated by each method (topmost image). However, the centers of mass identified by the two methods never differ by more than 10 μm, the approximate diameter of a cell, and so the automated tracking is deemed successful. (Not all timepoints can be seen in the trajectory image because some motion occurs exclusively in z, which is the viewing axis in the topmost image. Note also that background can vary considerably in the gray-scale images (e.g. timepoint 6). This reflects slow temporal fluctuations in the intensity of excitation light from the mercury arc lamp). Scale bar = 10 μm.

Figure 7. Example of automated tracking failure. Format of the figure is as in Figure 1. The manual and automated tracks coincide through the first 14 timepoints (12-14 are shown here), but then diverge. Examination of the movie of cell motion suggests that the manual track is correct. The automated track errs at timepoint 15 because the intensity distributions of the tracked cell and a nearby cell become very similar. Subsequent analysis of these cells at later timpoints shows that the automated track proceeds to successfully follow the new cell (e.g. timepoints 16,19). Note that the competing cell was also in the vicinity of the tracked cell at timepoints 12-14, but was then less similar in appearance and so not mistaken for the tracked cell. Scale bar = 10 μm.

Figure 8. Single client - multiple server configuration. The client machine has a large disk to store all of the time-lapse 3D data. For a cell being tracked, this machine sends a template and a search volume to each of the servers. A server computes the position of best match of the template in the search volume, and sends this information back to the client. The client continues to submit search requests to each of the servers until new cell positions have been found for all of the cells at that time point. Once this is accomplished, the entire process is repeated at the next time point.

Figure 9. Response time improvements with homogeneous and heterogeneous servers. The tracking test contained 77 search requests. The homogeneous servers were DECstation 5000's and the heterogeneous servers also included Sun workstations, a Sparcstation 1041 and a Stardent Titan. ("0" servers indicates that the program was run on a single machine acting as both client and server). Monotonic decreases in response time with increasing servers are observed in both cases.

Fig. 4

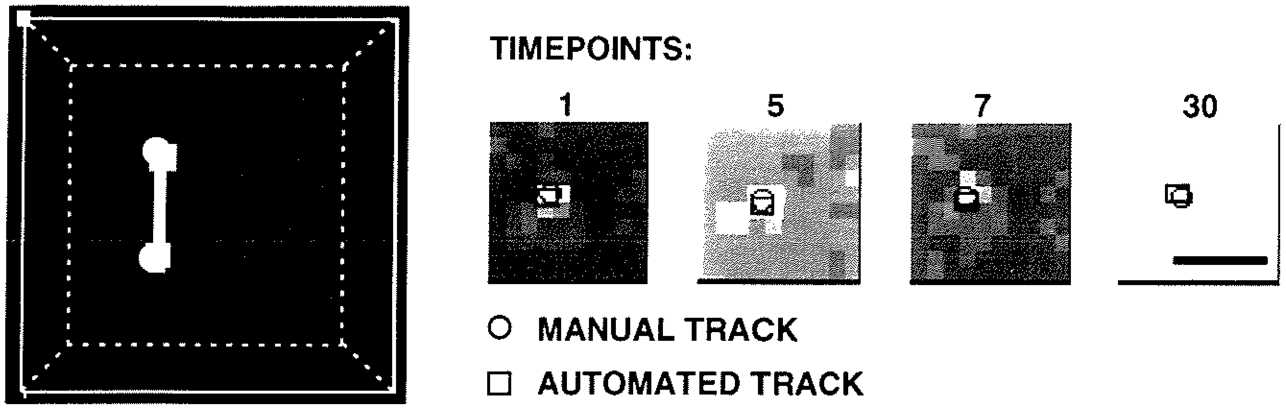


Fig. 1

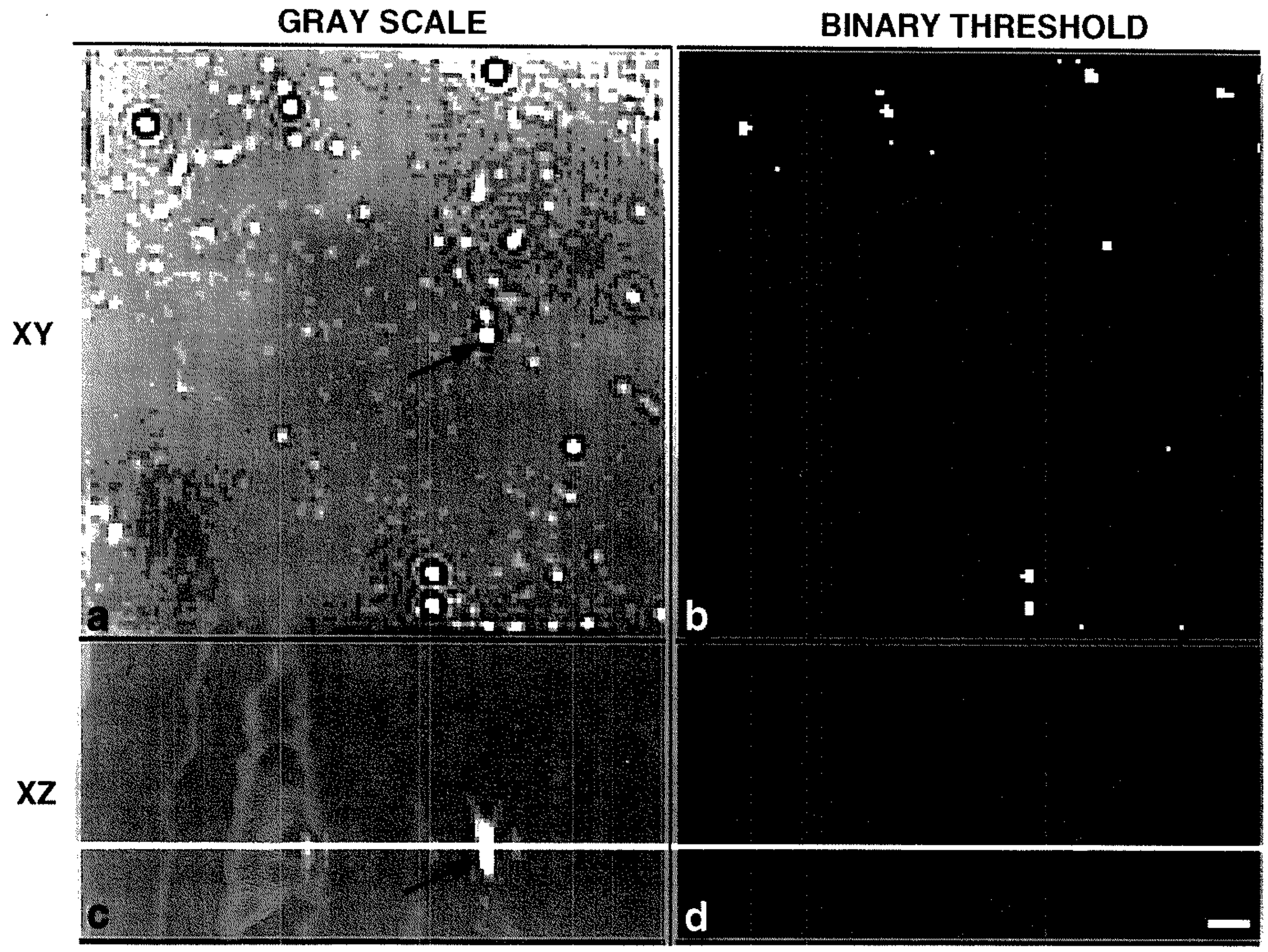


Fig. 2

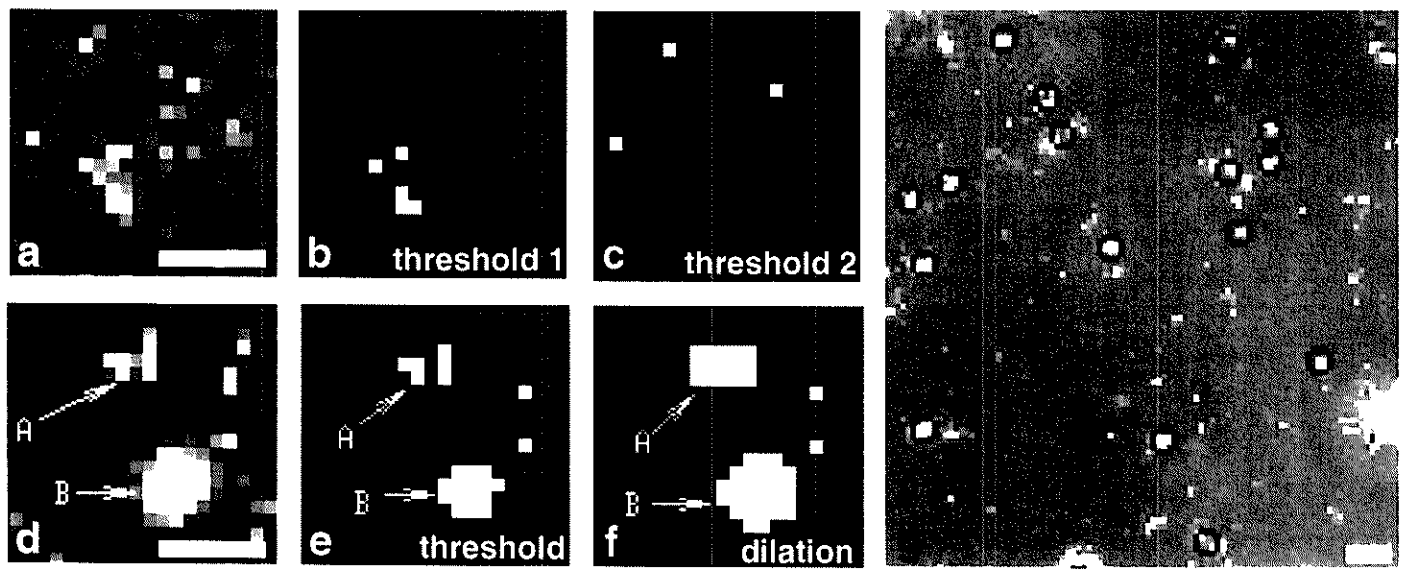
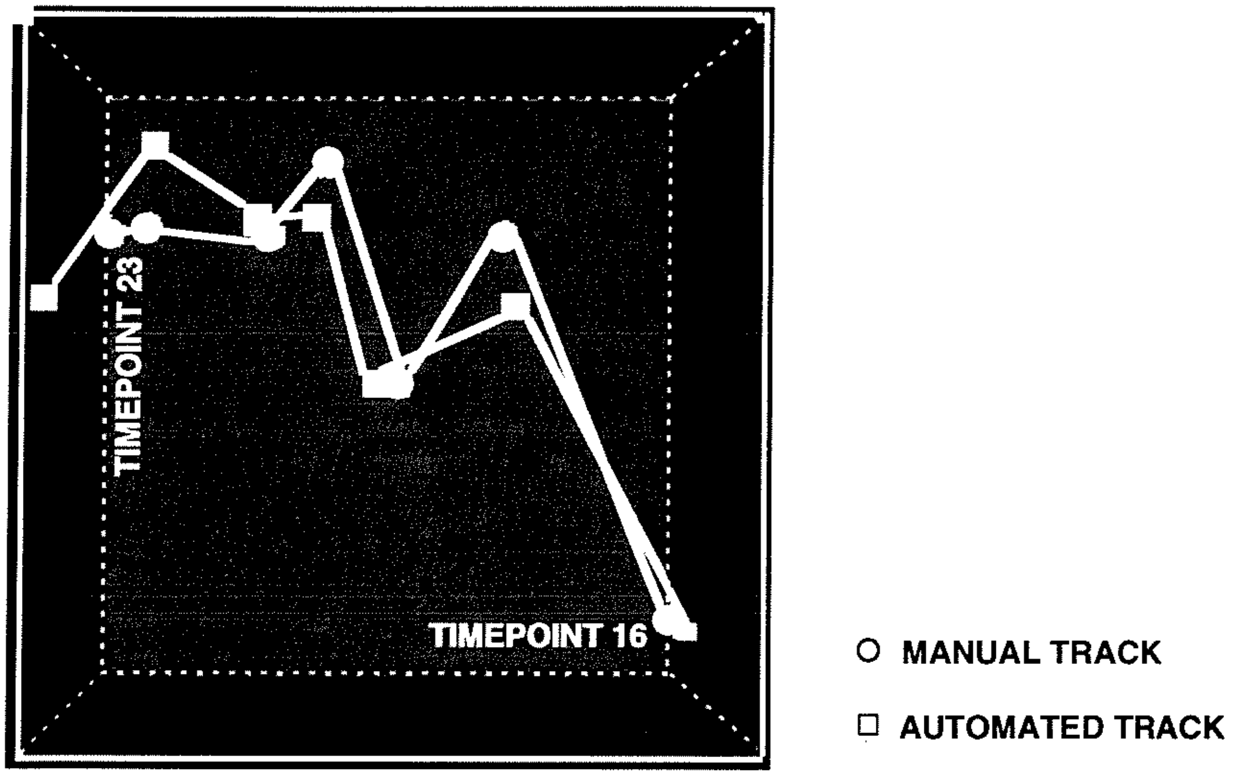


Fig. 3


TIMEPOINT 23
TIMEPOINT 16
○ MANUAL TRACK
□ AUTOMATED TRACK

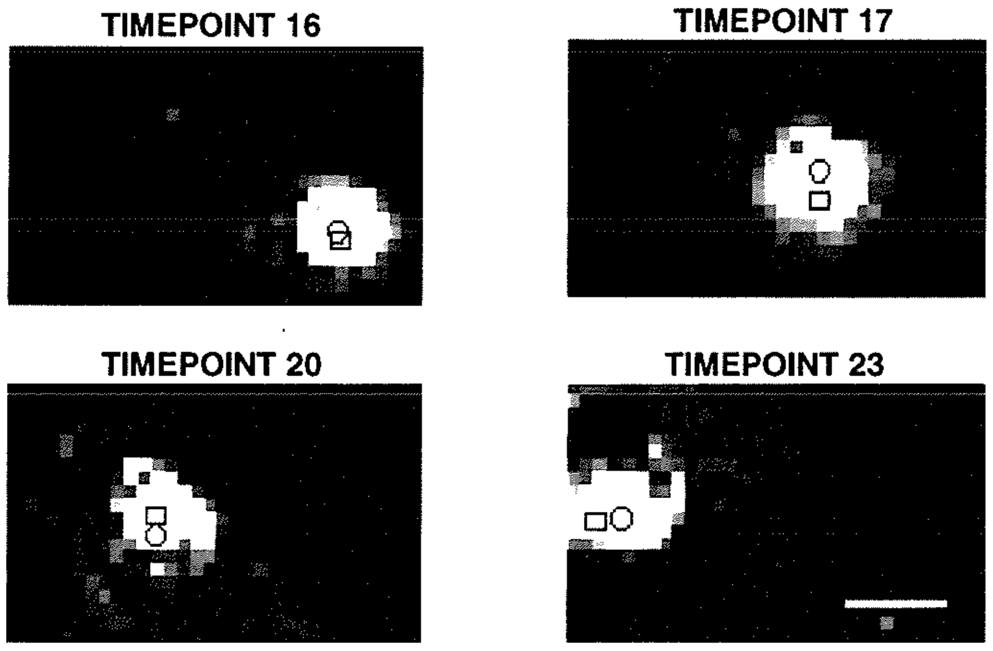

TIMEPOINT 16
TIMEPOINT 17
TIMEPOINT 20
TIMEPOINT 23

Fig. 5

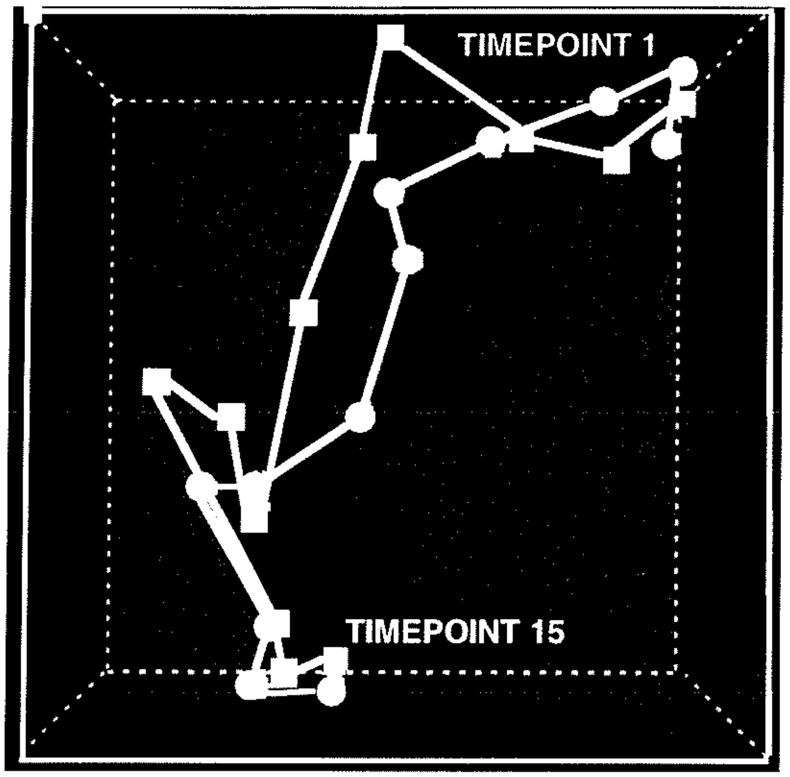
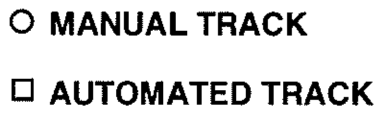
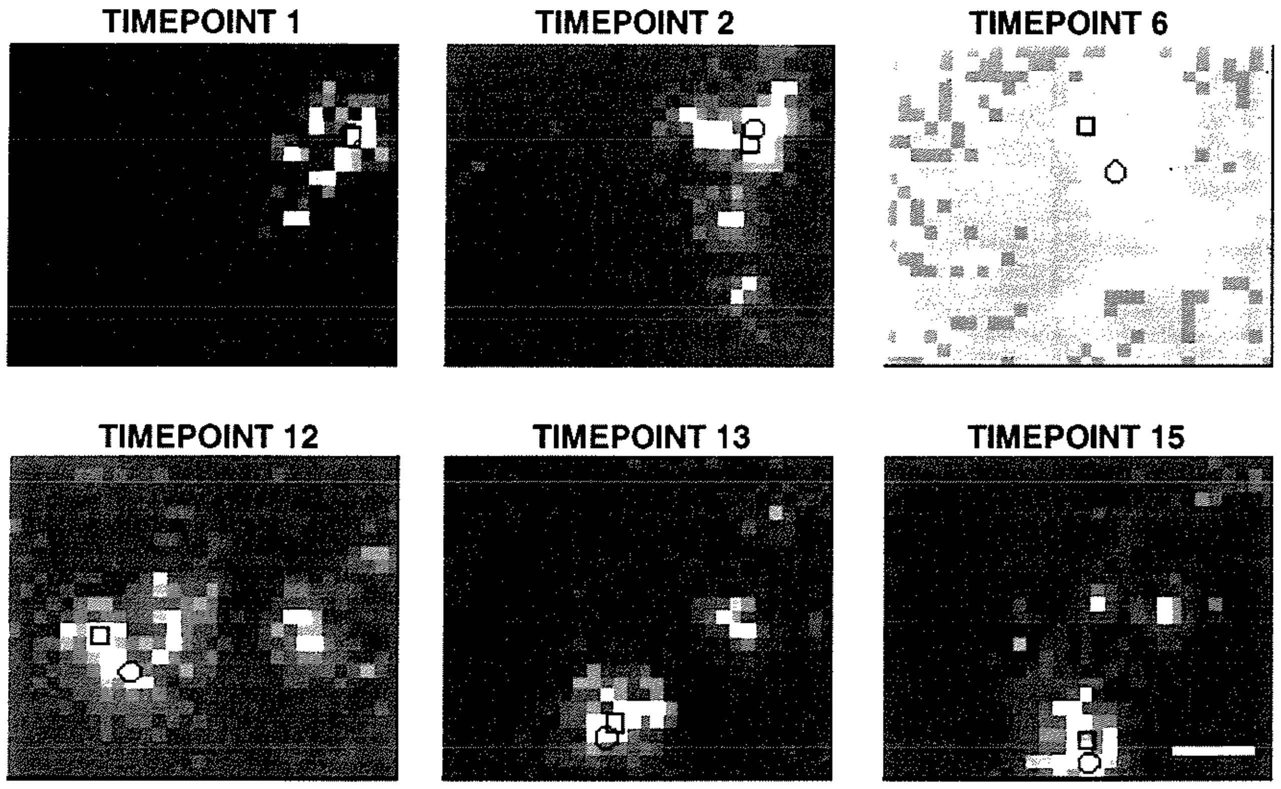
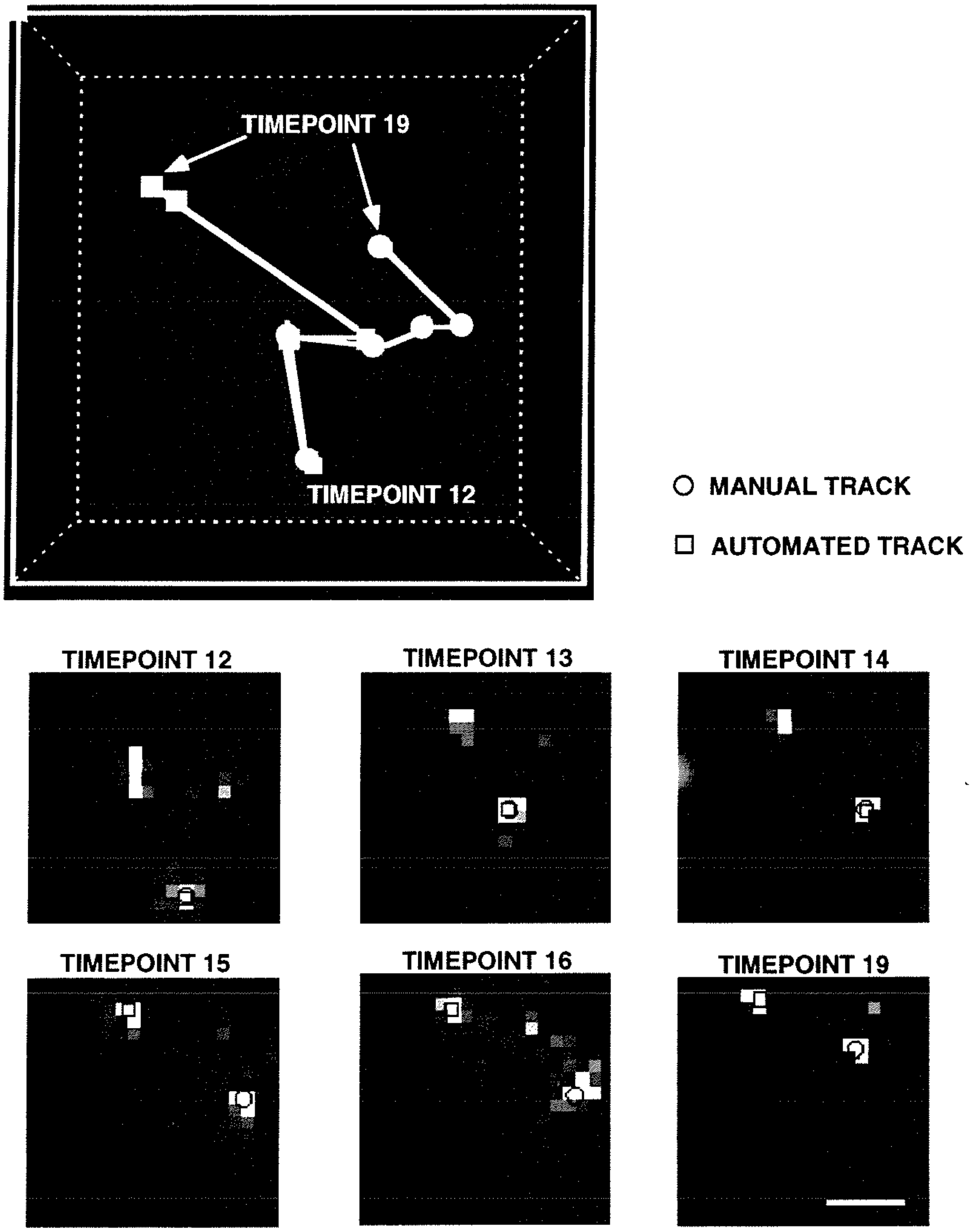

TIMEPOINT 1
TIMEPOINT 15
○ MANUAL TRACK
□ AUTOMATED TRACK
TIMEPOINT 1
TIMEPOINT 2
TIMEPOINT 6
TIMEPOINT 12
TIMEPOINT 13
TIMEPOINT 15



Fig. 6



TIMEPOINT 19
TIMEPOINT 12
○ MANUAL TRACK
□ AUTOMATED TRACK
TIMEPOINT 12
TIMEPOINT 13
TIMEPOINT 14
TIMEPOINT 15
TIMEPOINT 16
TIMEPOINT 19

Fig. 7

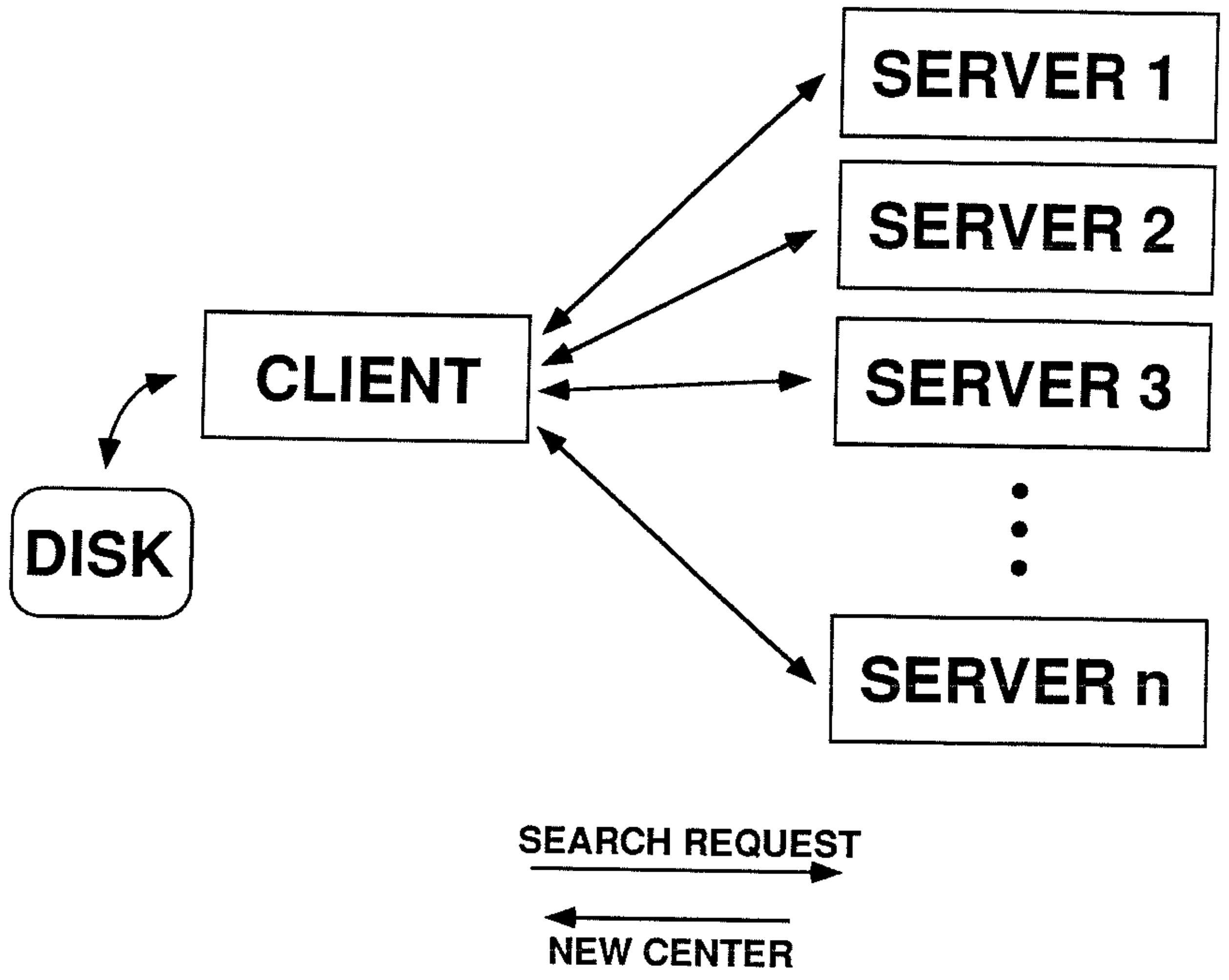
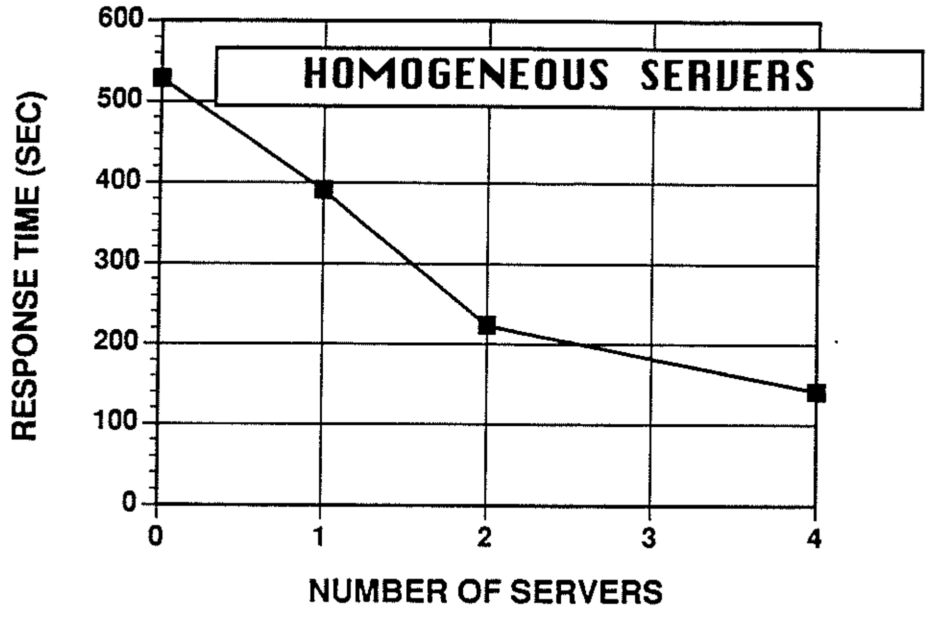

SERVER 1
SERVER 2
SERVER 3
SERVER n
CLIENT
DISK
SEARCH REQUEST
NEW CENTER

Fig. 8


HOMOGENEOUS SERVERS
RESPONSE TIME (SEC)
600
500
400
300
200
100
0
0
1
2
3
4
NUMBER OF SERVERS

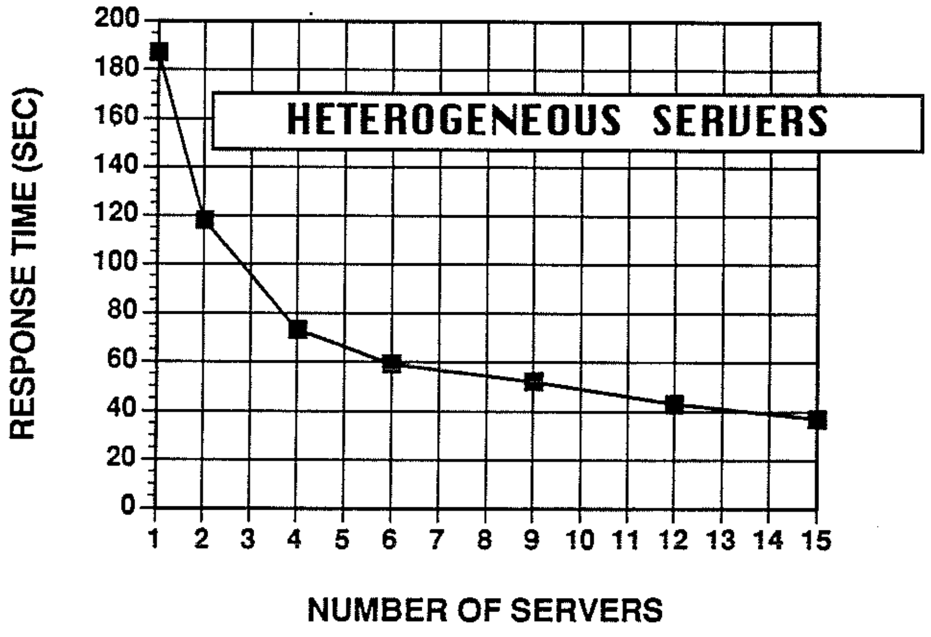

HETEROGENEOUS SERVERS
RESPONSE TIME (SEC)
200
180
160
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
NUMBER OF SERVERS

Fig. 9