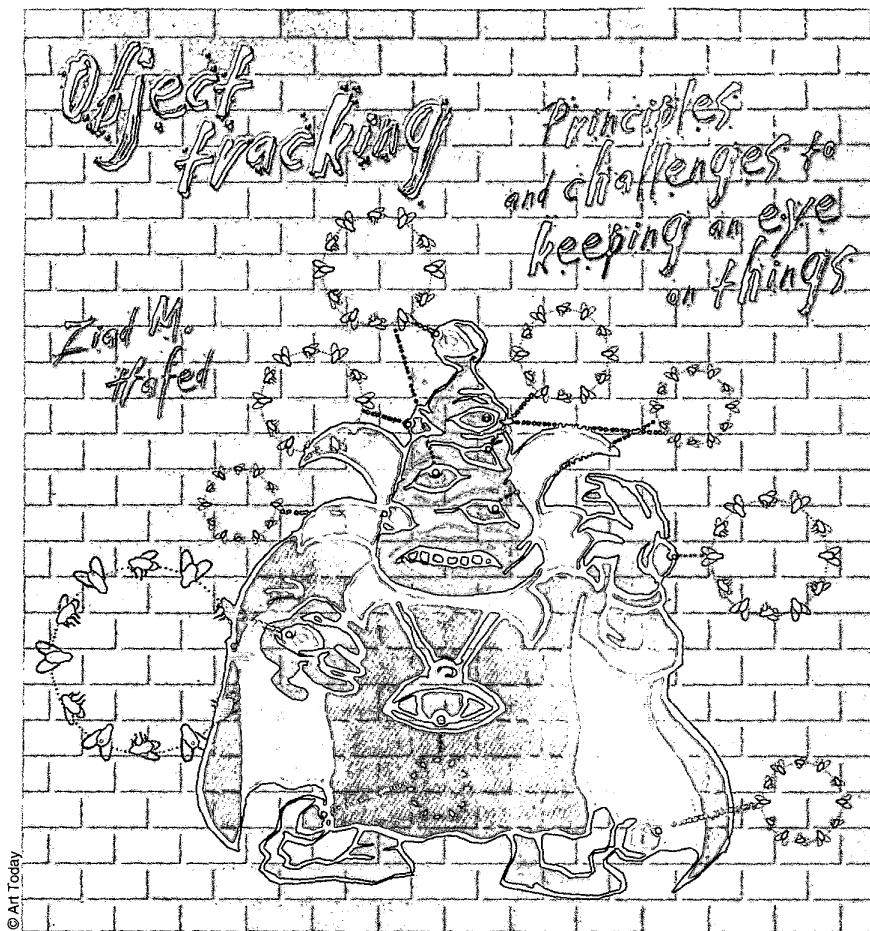




© Art Today

Object tracking means tracing the progress of objects (or object features) as they move about in a visual scene. Object tracking, thus, involves processing spatial (space) as well as temporal (time) changes. This processing is usually done digitally on a general or a special purpose computer.

Figure 1 shows the schematic of a generic object tracking system. As can be seen, visual input is usually achieved through digitized images obtained from a camera connected to a digital computer. This camera can be either stationary or moving depending on the application. Beyond image acquisition, the computer performs the necessary tracking and any higher-level tasks using the tracking results.

Applications of object tracking are numerous and they span a wide range of domains. For example, object tracking may prove useful in missile guidance, vehicle tracking and aerospace applications. It is of paramount importance in autonomous vehicle navigation.

Object tracking could easily assist in video data communication. For instance, it could allow low-bit-rate coding of video with complex backgrounds; or, in indexing video databases, it could be a preprocessing stage for an application such as face recognition.

Other interesting applications of object tracking exist. The work of Ferrie et al. (see Read more about it) attempts to track the progress of living cells and their interactions. This would assist scientists in deciphering experimental data about these cells and their life cycles. Also, the algorithm by Intille et al. tracks young children in order to implement an interactive, narrative play space called KidsRoom.

All these applications pose significant challenges to any object-tracking algorithm. Some, such as real-time constraints, seem obvious, whereas others are not.

For example, in order to track objects, certain features of those objects have to be selected. However, because of object (and possibly camera) motion, these features have to be identifiable

under varying poses. Furthermore, these features need to be matched over different frames. This matching is not necessarily easy to do.

The challenges of tracking include occlusion, which occurs when a tracked object is temporarily hidden by other objects. The following quote from the book, *Machine Vision*, by Jain, Kasturi and Schunck summarizes this difficulty succinctly:

"One of the main difficulties with the use of multiple frames is the problem of occlusion. When working with a large sequence of frames, it is possible that some objects may disappear totally or partially. Similarly, some new objects may appear in the frame sequence from some intermediate frame onward. In addition, the changing geometry of objects due to motion and the changes in scene illumination over the frame sequence can cause significant changes in the feature point data that is to be matched over the frames."

General approaches

Looking at the research literature, one finds a wealth of work dealing with the problem of object tracking. However, in most studies, there is a recurrent use of one or more of the following main concepts:

- 1) *Correlation*: the process of assessing relationships between sets of data;
- 2) *Edge information*: high contrast information extracted from images;
- 3) *Color information*: red, green and blue components of the color of an object (Color can also be characterized in other formats—hue, saturation and intensity, for example.);

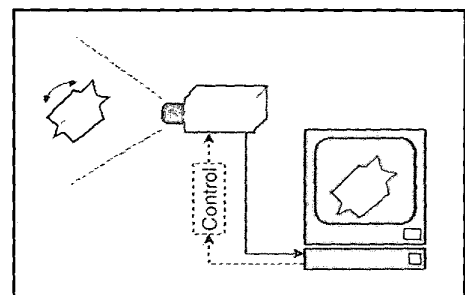


Fig. 1 Schematic of a generic object tracking system. The camera obtains visual images, and the computer tracks the seen object(s). The camera can also be mobile, in which case tracking would involve controlling a robotic manipulator to keep objects in the center of the visual field.

4) *Spatio-temporal information*: information about objects from single (spatial) as well as multiple (temporal) frames/images;

5) *Kalman prediction*: a special kind of filtering that allows for state estimation and that is often used in predicting object trajectories and speeds over frames;

6) *Hypothesis testing*: a technique whereby certain hypotheses about tracked objects are made and then tested with "goodness" metrics.

In addition, approaches to object tracking can be further characterized from a different perspective. In particular, one can categorize object-tracking algorithms into different classes. This categorization not only assists in understanding the principles of various object-tracking algorithms, but it also allows for sound comparisons among such algorithms. For example, some algorithms are model-based while others are not. We will describe three different types.

Knowledge-base

Some algorithms present in the literature require a priori information to be available about the objects being tracked. Other algorithms do not require such information. The advantages or disadvantages to having or not having such prior knowledge really depend on the application domain.

As an example, consider Rabiner's and Jacquin's work in tracking human faces and bodies. In this case, the object-tracking algorithm assumes prior knowledge about the objects to be tracked. In particular, Rabiner and Jacquin model faces and bodies using ellipses of various sizes. They then look for the best elliptical fit of the targets in each frame (see Fig. 2).

Initially, Rabiner and Jacquin performed motion based image segmentation in which they remove background clutter. They then combine edge data with motion data to get a motion-and-edge image of the tracked face/body. An elliptical fit algorithm is applied on this image and the best fit is chosen based on various thresholds.

Rabiner and Jacquin use knowledge about the ellipse's size and position from prior frames to predict current properties. Thus, their algorithm uses adaptive modeling of the scene content. They then accept an ellipse based on two criteria.

The first criterion involves passing a

threshold on a fitness or goodness metric of the form:

$$F = d_{border} + d_{motion} + P_{motion} \quad (1)$$

where d_{border} is the density of edge data on a candidate ellipse border; d_{motion} is the density of motion data inside the ellipse, and P_{motion} is the percentage of motion data inside the ellipse (relative to some window centered on the ellipse).

The second criterion restricts how far a candidate ellipse can fall compared to its position in previous frames. The authors report an average success rate of



Fig. 2 An illustration of the modeling technique used by Rabiner and Jacquin in which faces are modeled by ellipses.

94% for tracking heads. But, that rate drops for scenes with complex backgrounds.

Other approaches to object tracking assume no prior knowledge about the scenes being processed. Moscheni et al. present such an algorithm in which they first perform spatio-temporal segmentation of image sequences. This is done by first removing camera motion through global motion estimation. The algorithm then iteratively merges and splits regions based on motion information extracted from an affine motion model. Regions with similar motions are merged together, and scenes are finally described as sets of coherent moving entities.

To better localize moving objects and track them, additional operations are performed. In particular, a Kalman filter is used to predict object locations. Also, the correspondences between features are found by hypothesis testing. In particular, hypotheses about the spatial and temporal information regarding tracked objects are checked independently. Depending on the various cases of spatial/temporal hypotheses being

accepted, or not, feature correspondences are established.

Camera motion

Another algorithm category for object tracking involves the use of camera motion. Rabiner and Jacquin's algorithm (previously described) uses a static camera. In such algorithms, the tracking system traces the locations of moving objects in a stationary visual field.

Figure 3 shows an example tracking result using a static camera. The figure shows the final image in a video sequence. It superimposes the moving object's tracked position throughout the length of the sequence. As can be seen, the system is able to follow the object's trajectory.

But, if a moving camera is employed, then the goal is to always center the tracked object in the image. An example of tracking with a moving camera is shown in Fig. 4. Although the object moves through space, its relative position in the camera image is maintained at roughly the same position.

A similar approach is used by Papanikolopoulos et al. Their work shows how tracking can provide control signals to robots as well as create model robot environments. The approach combines vision and control techniques in an interesting manner.

Specifically, a cross-correlation technique is used to determine displacements between subsequent frames (that is, optical flow). These displacements are inputted to a controller in order to

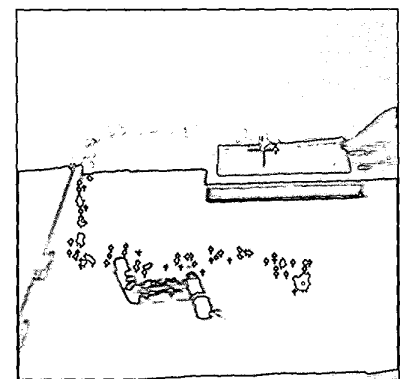


Fig. 3 An example tracking result with a stationary camera. The image shows the final frame in a video sequence in which a book is tracked as it is moved about in the scene. The superimposed points show the inter-frame book positions.

move a robotic manipulator. The main goal of the system is to manipulate a robotic arm to keep a target within a specified neighborhood in the image plane. The controller could be a simple PI (proportional-integral) controller or a more sophisticated controller that uses a Kalman filter for state estimation. The main drawback of this approach is that optical flow computations may be too intensive.

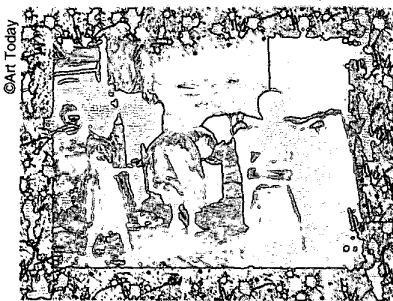
A moving camera can be used in navigation applications. This is because the vehicle's inherent motion causes changes in captured images even if targets are stationary. This situation could arise when an autonomous vehicle uses a stationary landmark to progress in its navigation task. If the targets to be tracked are mobile, then the tracking problem becomes even more complex.

Rigid body vs. non-rigid body

The ability to track non-rigid bodies is a very important property of an algorithm, for non-rigid bodies are often encountered in real-life. For example, walking humans or playing children translate into non-rigid targets to be tracked.

This is exactly the case in the KidsRoom system mentioned previously. In particular, the system attempts to track multiple, non-rigid objects. The system is real-time and model-based. That is, it uses contextual information, knowledge about the objects being tracked and their current relationships to one another.

"No matter how good you think you are you don't have the capability to predict who are the people who are going to bloom."



Mussolini's youth army, etc. often under ghastly conditions. He became severely malnourished as a result. (His Mother had been incarcerated at the Dachau concentration camp.)

et al. argue that they can use contextual knowledge by using a closed-world assumption. They define a closed-world as a specific region of space and time. Also, the specific context of what is in the

region is assumed to be known.

Their algorithm basically performs background differencing and clustering to obtain blobs that may represent individual or close objects. It uses object size, color, velocity, current position and past position to match objects in previous frames to blobs in the current frame. Blobs are computed for each frame, and a local closed-world description of each blob is provided. This description would include, for example, two nearby objects that appear in the same blob.

Thus, the closed-world assumption helps in solving the correspondence problem in the presence of non-rigid targets. The algorithm performs well for single objects and for multiple objects with distinctive color features.

Some algorithms can only work with rigid objects. However, this again may be all that is needed for certain applications. For example, in an industrial setting, where robots only manipulate manufactured parts, visual tracking may only have to deal with rigid bodies. If this is the case, then certain techniques—not well suited for non-rigid bodies—can be used.

Dealing with the challenges

We categorize the major problem areas of object tracking as follows:

- feature selection,
- correspondence,
- occlusion,
- real-time constraints and
- noise.

Feature selection

Object tracking often involves selecting certain object features and trying to match them in successive frames.

This done in order to describe their motion in the image plane. The matching problem is a major hurdle in itself; but, the chosen features have to be found first before they can be matched.

Thus, the appropriate selection of features is fairly important. Bad features would undergo significant changes from frame to frame due to object/camera motion. Thus, these would be extremely hard to localize. Now, it should be evident that the features of interest are directly related to the changing regions

in scenes. Thus, temporal, as well as spatial, variations are usually used to select features.

We have already mentioned several approaches to feature selection. Both edge data and motion data are used by Rabiner and Jacquin to localize moving heads/bodies. Color information is used almost exclusively in the KidsRoom system. In the work of Moscheni et al., spatial features are chosen to be affine invariant moments.

Correspondence

Feature correspondence refers to matching certain features across multiple frames. It is similar to the problem of matching features in pairs of stereo images. In the example systems investigated here, correspondence is achieved by an explicit set of operations.

For example, the goodness of fit function of equation (1) is used to accept a suitable ellipse for face/head tracking. Also, using closed-world assumptions the KidsRoom system greatly aids the correspondence process. Specifically, the assumptions allow for continued tracking of objects that get fused together into single blobs in the motion segmentation and background removal stages.

Occlusion

The occlusion problem arises naturally because of camera and/or object motion. Thus, this is a problem that cannot be completely avoided and must be dealt with in tracking systems. Many tracking algorithms are not robust with respect to occlusion, unless provisions for it are made in the algorithm design.

For example, color based tracking is not inherently occlusion robust. Imagine a scenario where a tracked object is partially or fully occluded by another object of a similar color. Clearly, a color based tracking system would encounter difficulty in this scenario. One solution is to model the occlusion process and include this model in the object tracking algorithm design.

Real-time constraints

Real-time operation is a major requirement in many object-tracking applications. In addition, many algorithms rely heavily on temporal correlation and smooth object motion (inertia). This is because these algorithms predict, in one way or the other, future object trajectories based on past behavior. The assumption of temporal correlation is

—Mario R. Capecchi, 1996 winner of Japan's Inamouri Foundation's Kyoto Prize for research on inserting/deleting genes in mice. Capecchi born in 1937 lived from 1942 to 1946 on the street, in orphanages,

valid as long as near video frame rates are achieved. This combination of application-related need and algorithm-related dependence results in requirements to run at acceptable frame rates.

Having said that, it is not easy to achieve frame rates of, say, 30 frames/sec. The frame rates observed in the algorithms reviewed here range from three to seven frames/sec in general. This suggests that object tracking is a really computationally intensive operation. It may, therefore, cost a significant amount of money to implement efficiently.

Noise

Noise surely plays a role in affecting the performance of an object-tracking algorithm. Yet, little work seems to have been done to quantify its effects.

Siebert and Engelhardt have attempted to do such an analysis on two representative tracking algorithms. They present an interesting analysis that provides lower bounds on the performance of tracking algorithms relative to camera noise, quantization levels, object size, and contrast between objects and backgrounds. Despite using statistical methods and making some assumptions about the scene (e.g., one object), their analysis shows how performance may be affected by issues such as noise.

The two algorithms studied by Siebert and Engelhardt are a *product correlator* and an *edge detector*. These two algorithms are representative of generic classes of tracking algorithms. The authors use the following equation to relate mean object brightness, B_T , to mean background brightness, B_B :

$$B_T = B_B + k\sigma_N \quad (2)$$

where k is a contrast factor and σ_N is the standard deviation of the noise. They then evaluate a quantity, σ_D , which is defined as the standard deviation of the tracking error.

A threshold value of 0.25 is defined for this quantity to delimit the range of "good" performance. That is, situations in which this standard deviation is less than 0.25 are acceptable situations. In their work, Siebert and Engelhardt argue that the following parameter ranges allow for "good" object tracking performance:

$$k \geq 3$$

$$\text{object size} \geq 4 \times 4 \quad (3)$$

quantization level ≈ 7 (or the characteristic camera noise parameter in bits).

This basically means that the contrast factor between object and back-



Fig. 4 An example tracking result with a mobile camera. The superimposed points on the final frame show the interim object positions. The object is always centered in the image.

ground should be greater than 3; that the object size should be bigger than 4×4 pixels, and that the quantization level should be greater than 7 bits.

At suboptimal object sizes or quantization levels, it is argued that one can still achieve good performance with higher contrast. Also, the correlation algorithm is found to be better at small quantization levels. Whereas, the edge detector is better at small object sizes.

Final comments

In general, there is a wide range of object tracking algorithms, with some being quite efficient and robust. However, major hurdles still need to be jumped. For example, tracking multiple objects remains a major problem for many algorithms. Also, real-time operation (with 30 frames/sec rates) at reasonable costs has yet to be realized. Finally, more work on quantifying the limiting performance boundaries of more recent object tracking algorithms is needed.

Read more about it

• Belanger, P., 1995, *Control Engineering: A Modern Approach*, Saunders: Orlando, FL.

• Ferrie, F., Levine, M. and Zucker, S., 1982, "Cell tracking: a modeling and minimization approach," *IEEE Trans. on Pattern Anal. And Mach. Intell.*, Vol. PAMI-4, No. 3, pp. 277-291.

• Intille, S., Davis, J. and Bobick, A., 1997, "Real-time closed-world tracking," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 697-703.

• Jain, R., Kasturi, R. and Schunck, B., 1995, *Machine Vision*, McGraw-Hill: New York, NY.

• Moscheni, F., Dufaux, F. and Kunt, M., 1996, "Object tracking based on temporal and spatial information," *Proc. IEEE ICASSP*, Vol. 4, pp. 1914-1917.

• Papanikolopoulos, N., Khosla, P. and Kanade, T., 1991, "Vision and control techniques for robotic visual tracking," *Proc. 1991 IEEE Int'l Conf. on Robotics and Automation*, pp. 857-864.

• Rabiner, W. and Jacquin, A., 1996, "Object tracking using motion-adaptive modeling of scene content," *IEEE Global Telecommunications Conference*, Vol. 2, pp. 877-881.

• Siebert, H. and Engelhardt, W., 1988, "Performance data and limiting conditions of object tracking algorithms in the presence of camera noise," *IEE Proc., Part F: Communications, Radar & Signal Processing*, Vol. 135, February, pp. 111-113.

About the author

Ziad M. Hafed obtained his B. Eng. (Hons) and M. Eng. degrees in Electrical Engineering from McGill University (Montreal) in 1997 and 1999, respectively. He is currently a PhD candidate in the Visual-Motor Systems Group at McGill University's Center for Intelligent Machines. His research interests include human and machine vision, virtual reality and telepresence, and he is particularly interested in the problem of human-machine interaction. Ziad M. Hafed received several honors and awards including the British Association Medal for Great Distinction (1997) and Master's and Doctoral scholarships from the Natural Sciences and Engineering Research Council of Canada (1997 and 1999).