Improved Tracking by Decoupling Camera and Target Motion

Shawn Lankton and Allen Tannenbaum

Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332, USA

ABSTRACT

Video tracking is widely used for surveillance, security, and defense purposes. In cases where the camera is not fixed due to pans and tilts, or due to being fixed on a moving platform, tracking can become more difficult. Camera motion must be taken into account, and objects that come and go from the field of view should be continuously and uniquely tracked. We propose a tracking system that can meet these needs by using a frame registration technique to estimate camera motion. This estimate is then used as the input control signal to a Kalman filter which estimates the target's motion model based on measurements from a mean-shift localization scheme. Thus we decouple the camera and object motion and recast the problem in terms of a principled control theory solution.

Our experiments show that using a controller built on these principles we are able to track videos with multiple objects in sequences with moving cameras. Furthermore, the techniques are computationally efficient and allow us to accomplish these results in real-time. Of specific importance is that when objects are lost off-frame they can still be uniquely identified and reacquired when they return to the field of view.

Keywords: Algorithms, Tracking, Mean Shift, Registration, Kalman Filter, Control

1. INTRODUCTION

Video tracking is widely used for surveillance, security, and defense purposes.[1] In cases where the camera is not fixed due to pans and tilts or due to being fixed on a moving platform tracking can become more difficult. This is especially true when the object or objects being tracked represent only a small portion of the area of each frame. In these circumstances, camera motion must be taken into account. When significant camera motion is present, it is possible that objects will come and go from the field of view. For example, consider the sample frames shown in Figure 1. The highlighted targets should be continuously and uniquely tracked if and when they reappear.

Figure 1. Frames 20, 30, 40, 50, 60, 70, 80, and 90 of the Aerial Traffic sequence. In this sequence the two cars highlighted by the ellipse disappear from view for nearly 50 frames until the camera shifts to show them again.

Further author information: (Send correspondence to :)
Shawn Lankton : E-mail: slankton@ece.gatech.edu, Allen Tannenbaum : tannenba@ece.gatech.edu

Real-Time Image Processing 2008, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6811, 681112, © 2008 SPIE-IS&T · 0277-786X/08/$18

Other authors have considered the problem from an automatic detection standpoint.[2] In this work, we will take a different approach. By modeling the motion of the targets as well as using the motion of the camera it is possible to maintain the approximate location of an object even when it is not in view. Then, when it returns to view it can be re-acquired. This requires localization onto the target, frame registration, and a way to combine these components.

There are various approaches for frame-to-frame localization in video tracking. Essentially all of these approaches perform a local search in order to optimize some matching energy. The location where the matching energy is optimal is the place where the target is most likely to be. Many such localization schemes exist such as active contours,[3–5] localized graph cuts,[6] and covariance tracking.[7] In this work, we have chosen to use the popular mean shift localization procedure.[8] This method has received a great deal of attention in recent years[9–12] and is popular because of its easy implementation, efficient computation, and reasonably good results.

A problem with all such localization schemes is the inherent assumption that target motion from one frame to the next will be small enough that the position of the target in frame n is close to the position of the target in frame $n-1$. If this is not the case due to fast moving objects, occlusion, or large camera motion it may be difficult to reacquire the object. Motion models can be used to address this problem in some cases. For instance, fast moving objects and occlusions are often handled by incorporating dynamics so that the localization is initialized close to the new position of the object based on its modeled movement and previous position. Typically linear motion models are used for simplicity and because motion of real-world objects is often too complex to be modeled completely.

Motion models can fail, however, when camera motion is coupled with object motion. Specifically, consider the case of a moving camera following a moving object. It is possible for large non-linearities in motion models to occur. This makes the motion model less able to match the targets movement accurately. By decoupling the camera's motion and the target's motion it is possible to obtain much better tracking results, and estimate a better model of the target's actual motion.

In order to decouple camera motion it is necessary to have a good estimate of how the camera moves. This amounts to an image registration problem between the current frame and the previous frame. There are a host of image registration techniques,[13] but due to our demand for real-time tracking performance we looked to fast registration methods available in the video compression community.[14–16] These use multi-resolution and gradient methods to compute a quick, robust estimate of global motion parameters.

To reap the benefits of the camera motion compensation we must combine it with the localization information from the mean shift. The Kalman filter is sometimes used to incorporate a motion model into a tracking system.[8] However, when this is done the Kalman filter is often simplified by eliminating a signal related to the control input. This signal is the ideal place to incorporate camera motion into the system. Incorporation of camera motion into the filter improves performance and frames the task as a well posed control problem.

The decoupling of the target motion and camera motion via the Kalman filter allows the system to predict the motion of the object, independent of the camera motion, based on measurements made while the object is in view. When an object leaves view its position can be continuously updated using its estimated motion model. Simultaneously, the knowledge of camera motion allows the system to keep the target's position relative to the current camera position. Thus, targets can be uniquely reacquired when they return to view.

The remainder of the note is organized as follows. Section 2 gives a brief overview of the mean shift procedure and the implementation used by the authors. Section 3 gives a summary of the frame registration algorithm incorporated in this work. Section 4 shows how these tools are combined into a Kalman framework to allow decoupling of the object and camera motion. In Section 5 we show experiments on actual video sequences. Finally we make concluding remarks and discussions in Section 6.

2. MEAN-SHIFT LOCALIZATION

Mean shift methods have emerged as a very popular localization method since they were introduced in 2003.[8] The algorithm provides an efficient, gradient ascent method to find a portion of the image whose local histogram matches some known target histogram. We will briefly cover the algorithm here, using the notation of Hager *et al.*[9] who reinterpreted the original formulation of the algorithm into matrix notation.

We begin with a target in an image, I specified by a center point, $\mathbf{c}$ and all points $\{\mathbf{x}_i\}_{i=1\ldots n}$ in a local window around $\mathbf{c}$. We intend to create a kernel-weighted histogram of the target region. To do this consider a feature vector $\mathbf{f} \in \mathscr{F}$ that exists at each pixel in I, and $\mathscr{U} = 1 \ldots m$ which represents m feature bins. Let $b : \mathscr{F} \rightarrow \mathscr{U}$ be a binning function that takes every $\mathbf{f}$ and puts it into one of the m bins. Finally, consider a kernel function $K : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ that weights each pixel location in $\mathbf{x}_i$ according to its distance to $\mathbf{c}$. K is always chosen such that $\sum_{i=1}^{n} K_i = 1$ and $K'(x_i) = 1 \ \ \forall i$.

Now, in order to construct a weighted histogram, $\mathbf{q} = (q_1, q_2, \ldots, q_m)$, at time t we create an n by m sifting matrix $\mathbf{U}$. To do this we first create m sifting vectors $\mathbf{u}_i(t) = \delta(b(x_i, t), u)$, where δ is the Kronecker delta function. We then form $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$. Further, we rewrite $\mathbf{K}$ as a vector by reshaping the matrix $\mathbf{K}_i(\mathbf{c}) = \mathbf{K}(x_i, \mathbf{c})$. Then, we write the formula for $\mathbf{q}$ very simply as

$$\mathbf{q} = \mathbf{U}^\top \mathbf{K}(\mathbf{c}) \tag{1}$$

Now, that the target histogram is known, we perform the mean shift operation to move a candidate region so that the histogram in that region is best matched with the target histogram. We create the histogram of the candidate region, $\mathbf{p}$ at time t' with

$$\mathbf{p}(\mathbf{c}) = \mathbf{p}(\mathbf{c}, t') = \mathbf{U}^\top(t')\mathbf{K}(\mathbf{c}) \tag{2}$$

With these two histograms, we measure their similarity using the Bhattacharyya metric, $\mathscr{B} = \sqrt{\mathbf{p}(\mathbf{c})} \cdot \sqrt{\mathbf{q}}$. Here, the square root applies component-wise to the vectors. The Bhattacharyya metric is iteratively increased by moving the center point towards the local maximum of $\mathscr{B}$ according to

$$\triangle\mathbf{c} = \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{c})w_i}{\sum_{i=1}^{n} w_i} \tag{3}$$

$$\mathbf{w} = \mathbf{U}\left(\frac{\sqrt{\mathbf{q}}}{\sqrt{\mathbf{p}(\mathbf{c})}}\right) = [w_1, \ldots, w_n] \tag{4}$$

The new center becomes $\mathbf{c} + \triangle\mathbf{c}$ and the process is repeated until convergence. In most cases the center point will converge in one to five iterations. When the target contains a small number of pixels each iteration is computationally inexpensive. This combined with the short convergence time allows one to easily track multiple targets in real-time.

The downside of all localization schemes is that they must be initialized nearby the target within a basin of attraction in order to converge at the correct local optimum. The Bhattacharyya metric maximized in the mean-shift procedure is relatively smooth, and so the mean shift can usually recover from initializations that put its center as far as one target's length from the desired center. It is because of this local convergence property that frame registration and motion prediction are so important to the proposed method.

3. FRAME REGISTRATION

Tracking camera motion is equivalent to tracking the apparent motion of fixed objects in the scene. The camera's motion can be considered to be the inverse of the apparent motion of these non-moving objects. When the camera motion is dominant to the motion of the targets in the frame, it is possible to view the problem of camera motion estimation as an image registration problem.

By registering each frame to the previous frame, we can compensate for global motion in the scene caused by the moving camera. Because speed is paramount in tracking applications, we propose to use frame registration techniques taken from video compression literature.[14–16] Specifically, we investigated the use of gradient-based motion estimation models (GM). These models estimate the transform parameters needed to register two images to each other. Consider two images $I_1(x, y)$ and $I_2(x, y)$ related to each other by an affine transform such that $I_1(x_1, y_1) = I_2(x_2, y_2)$ where

$$x_2 = a \cdot x_1 + b \cdot x_2 + c \tag{5}$$

$$y_2 = d \cdot y_1 + e \cdot y_2 + f \tag{6}$$

GM algorithms attempt to find the parameter vector $\mathbf{p} = \{a, b, c, d, e, f\}$ that minimizes $\sum_S (I_1(x_1, y_1) - I_2(x_2, y_2))^2$ where $\mathbf{S}$ is the set of all points common to both images. This is solved iteratively by linearizing the Gauss-Newton nonlinear minimization and solving a least squares problem.[14] Typically, a multi-resolution Gaussian pyramid is used to speed convergence. However, but these early algorithms remain very computationally expensive.

Standard algorithms perform intensity matching on every pixel in the joint image domain. This creates a huge number of equations which slow computation time and cause the system to be highly overdetermined. Even in a pair of small images, there will be tens of thousands of equations to solve for just six variables in $\mathbf{p}$. Keller *et al.*[15,16] propose a much faster means for computing these registration parameters using a variant of the same algorithm.

In order to reduce complexity, only a small percentage of image pixels are used for comparison. Because the most influential pixels are those near strong image gradients, these are the pixels used. A new set of pixels $\hat{\mathbf{S}}$ are identified at the coarsest level of a multi-resolution Gaussian pyramid. This group of pixels are used to compute $\mathbf{p}$ at this level and are then interpolated to the next level and so on until convergence is reached at the finest level. This method is capable of producing speed ups between 70 and 170 times compared to standard GM algorithms with very little accuracy difference. These methods are therefore capable of performing frame registration in real-time.

4. COMBINING IN A KALMAN FRAMEWORK

In order to combine the localization result from mean shift and the camera motion estimation from the GM frame registration we incorporate both measurements into the Kalman filter framework. The Kalman filter is often used in tracking to add dynamics to a system. In the simplest form, the discrete Kalman filter describes the state $\mathbf{x} \in \mathbb{R}^n$ at iteration k in terms of the state at the previous iteration, $k-1$ and a control signal $\mathbf{u}$. This relationship is given by the linear equation

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{7}$$

supplemented by a measurement $\mathbf{z} \in \mathbb{R}^m$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{8}$$

Here, $\mathbf{w}_k$ and $\mathbf{v}_k$ are the process and signal noise respectively and are considered to be independent zero mean Gaussian noise sequences. Covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are associated with the process and signal respectively. In our implementation, we set

$$\mathbf{Q} = q\mathbf{I}, \qquad \mathbf{R} = r\mathbf{I}. \tag{9}$$

where $\mathbf{I}$ is the identity matrix and $q < r$ such that measurements are weighted more heavily than estimations. Measurements are made by using the mean shift localization scheme described in Section 2. $\mathbf{H}$ is taken as the identity since $\mathbf{z}$ is a measurement of the exact parameters in $\mathbf{x}$. When measurements are unavailable such as when the object is out of frame we take $\mathbf{z} = \mathbf{x}$ thereby giving the estimate full weight.

In most tracking computer vision tracking applications, the second part of Equation 7 is omitted by assuming that $\mathbf{u}$ is always 0. In this work, we keep the control signal, $\mathbf{u}$, and use it to incorporate the global motion estimate into the filter. We define $\mathbf{x}$, $\mathbf{u}$, $\mathbf{A}$, and $\mathbf{B}$ as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_1' \\ x_2' \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{10}$$

The result is that estimated target motion (x_1' and x_2') is combined with estimated translation component of the camera motion (u_1 and u_2) to create the overall predicted motion for iteration $k+1$. This separation of motion models allows a linear model to fit much better to target motion and therefore allows for better prediction. Better prediction due to decoupling camera and object motion allows us to improve tracking performance in several ways. Targets are able to be tracked accurately at lower frame rates, and their location relative to the frame can be predicted accurately when they leave view all together. Using this improved prediction, the targets can be reacquired when they return to view.

5. EXPERIMENTS

In this section, we show three experiments. The first demonstrates the proposed algorithm's ability to maintain track despite large jumps caused by reducing the frame rate of the tracked sequence. Next, we demonstrate the ability of the system to recover from loosing an object off frame.

First, consider the IR Traffic sequence in Figure 2. The four targets are cars whose motion is approximately linear. There is a simultaneous large camera motion in the form of rotation and translation. Using solely the mean shift localization technique from Section 2, track is maintained at 30fps. When linear target dynamics are incorporated using a Kalman filter to predict the coupled motion of the camera and the targets the frame rate can be dropped to 15fps without losing track. However, when the camera motion is incorporated as a control signal we can maintain track even at 5fps. In Figure 2 we demonstrate successful tracking at 5fps with the proposed algorithm despite a large camera motion. Also shown is the failure to track when camera and target motion models remain coupled.

The second major benefit of this technique is that as objects leave the field of view, their position can be continuously estimated and maintained. This is facilitated by the fact that once camera motion has been decoupled from target motion, the motion model attached to the target's motion is accurately based on the target's movement and not the combined movement of the target-camera system. This is demonstrated in the experiments shown in Figure 3.

In this sequence two of the tracked cars leave the frame all together. At that time, their position estimate is updated based on the perceived camera motion and the motion model that had been obtained when the target left view. Based on these two motion estimates, the system predicts where and when the target will come back into view. This allows the system to reacquire the target and resume tracking it as soon as it is visible again.

The ability to track off frame and with low frame rate are complemented by the speed of this technique. Because the frame registration and localization are both completed with efficient techniques, the overall system is capable of tracking in real time. Below is a table showing the tracking frame rate acquired speeds of in the three sequences shown.

Table 1. Frame rates achieved with the proposed algorithm on various sequences

Sequence	Figure	Frame Rate
IR Traffic	2	15.15fps
Aerial Traffic	3	16.87fps

In all cases, the frame rate is above 15fps and can be considered real-time. However, note that our implementation performs at real-time speed despite being implemented solely in Matlab. Significant speedups could result from reimplementation in C or C++ or from implementation into a dedicated hardware system. Furthermore, because the properties of the system allow frame rate to be reduced and still maintain track, fewer frames are needed. This can further reduce computational load in a real-life application.

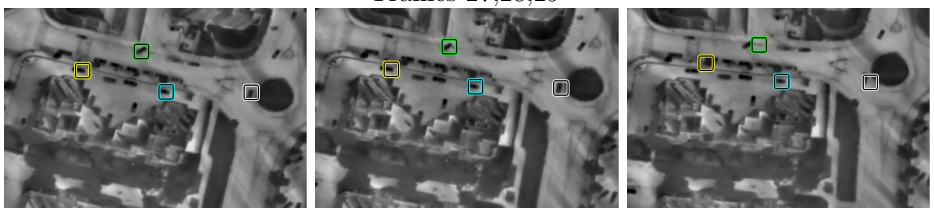
6. DISCUSSION

We have shown a method of including frame registration information into a standard computer vision tracking system using the often ignored control input of the Kalman filter. Doing so has drastically improved performance beyond localization methods alone or even localization methods with motion models that are affected jointly by camera and target motion. By selecting efficient algorithms such as fast GM frame registration, mean shift localization, and Kalman filtering to combine the two, we have constructed a system that is capable of delivering these benefits in real-time.

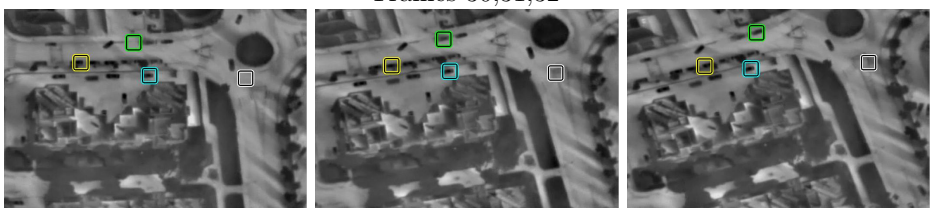
Of course, the method is only well-suited for certain types of imagery like the ones demonstrated in Section 5. For video sequences where object motion is the dominant motion in a video sequence, or where the object takes up the majority of the frame, this method will not be as effective because the frame registration algorithm

(a) Failure to track using coupled motion model

Frames 27,28,29

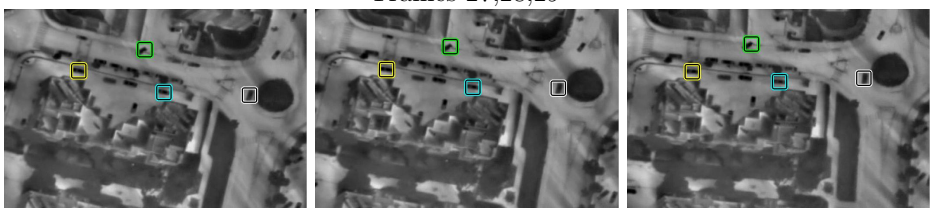


Frames 30,31,32



(b) Successful tracking with proposed method

Frames 27,28,29



Frames 30,31,32

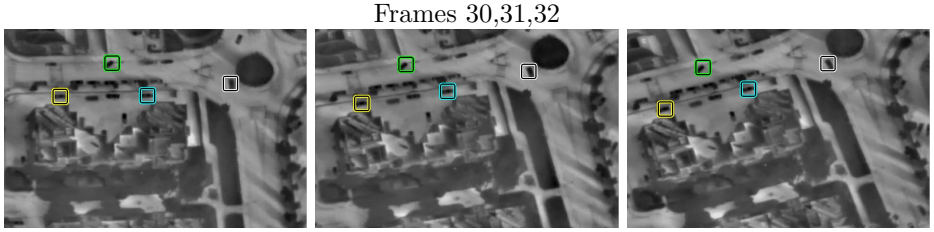


Figure 2. Tracking results from two algorithms on six consecutive frames from the IR Traffic sequence. These videos were tracked at a reduce frame rate of 5fps. These frames were selected because of the large rotational camera motion present. (a): Failure to track by mean-shift localization with a Kalman filter modeling coupled motion. The targets found in the final frame are not the original targets. (b): Successful tracking by proposed method with decoupled camera and target motion models.

will be impaired by the low percentage of pixels showing fixed background objects. However, for applications such as surveillance where targets take a small portion of the frame and camera motion accounts for a significant part of the motion, this technique improves tracking performance.

Future work will focus on improving prediction by using non-linear models so that as objects disappear from view their approximate position is estimated more accurately. Additionally the use of particle filters[17] may improve the ability of the system to reacquire targets when they return to the frame if the estimated motion model is incorrect.

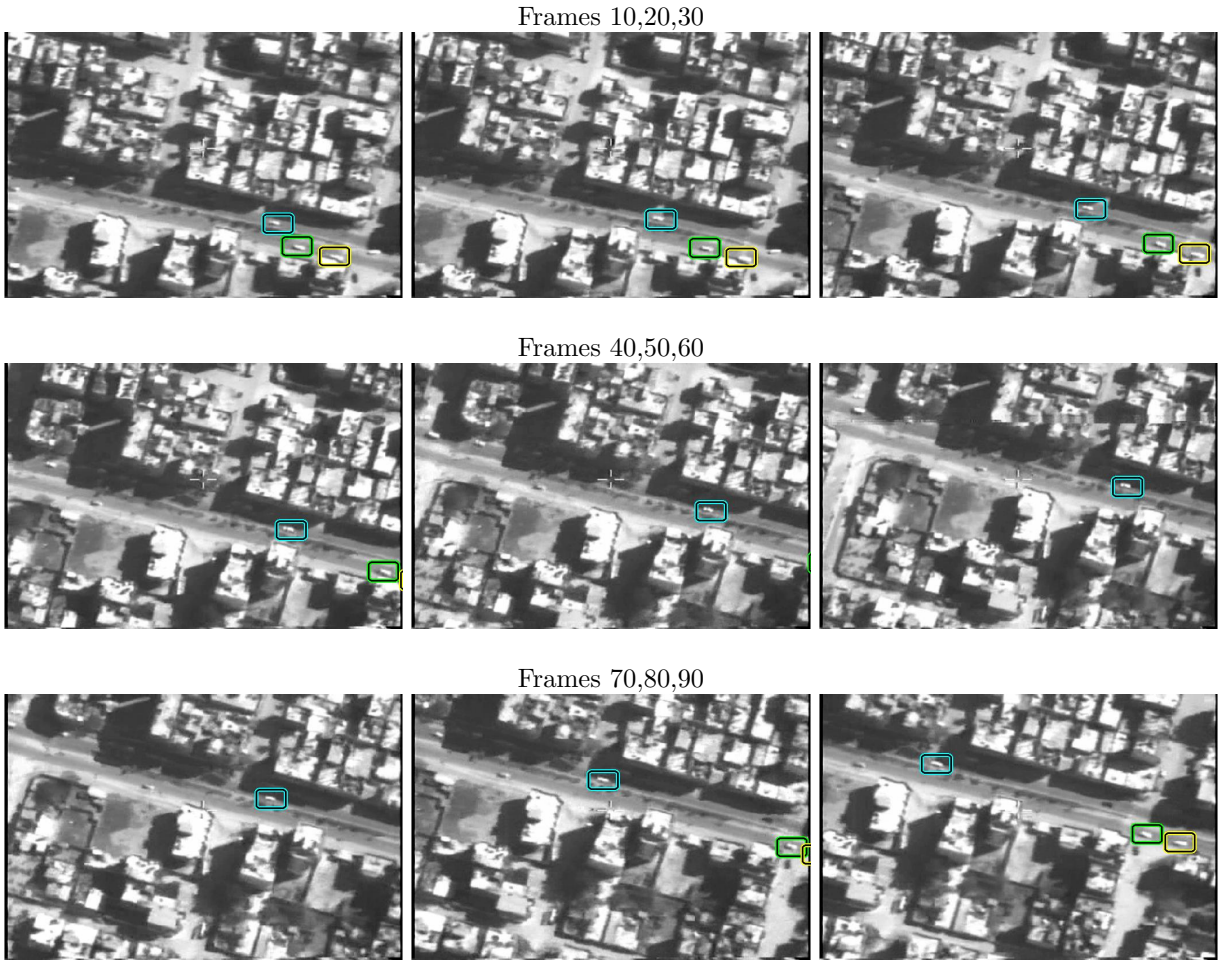


Figure 3. Selected frames from the Aerial Traffic sequence showing three cars simultaneously tracked. Note that two of the cars go out of view on the lower right for approximately 50 frames as the camera pans left and right. The cars are reacquired correctly when they return to view.

ACKNOWLEDGMENTS

This work was supported in part by grants from NSF, AFOSR, ARO, MURI, as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from `http://nihroadmap.nih.gov/bioinformatics`.

REFERENCES

1. R. Kumar, H. Sawhney, S. Samaraseker, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, H. Hansen, and P. Burt, "Aerial video surveillance and exploitation," in *Proceedings of the IEEE*, **89**(10), pp. 1518–1539, 2001.
2. Y. Guo, S. Hsu, Y. Shan, H. Sawhney, and R. Kumar, "Vehicle fingerprinting for reacquisition and tracking in videos," in *Conference on Computer Vision and Pattern Recognition*, **2**, pp. 761–768, 2005.
3. N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 266–280, March 2000.

4. S. Dambreville, Y. Rathi, and A. Tannenbaum, "Tracking deformable objects with unscented kalman filtering and geometric active contours," in *Proceedings of American Control Conference*, **1**, pp. 601–606, 2006.
5. Y. Rathi, N. Vaswani, and A. Tannenbaum, "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Trans. on Image Processing* **16**, pp. 1370–1382, May 2007.
6. J. Malcolm, Y. Rathi, and A. Tannenbaum, "Multi-object tracking through clutter using graph cuts," in *Non-rigid Registration and Tracking Through Learning (ICCV)*, 2007.
7. F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 728–735, 2006.
8. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **25**, pp. 564–577, May 2003.
9. G. Hager, M. Dewan, and C. Stewart, "Multiple kernel tracking with ssd," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 790–797, 2004.
10. C. Chang, R. Ansari, and A. Khokhar, "Multiple object tracking with kernel particle filter," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 566–573, 2005.
11. M. Dewan and G. Hager, "Toward optimal kernel-based tracking," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 618–625, 2006.
12. W. Qu and D. Schonfeld, "Efficient object tracking using control-based observer design," in *Conference on Multimedia and Expo*, pp. 1001–1004, July 2005.
13. B. Zitov, "Image registration methods a survey," in *Image and Vision Computing*, **21**, pp. 977–1000, October 2003.
14. F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for videocoding," *IEEE Trans. on Image Processing* **9**, pp. 497–501, March 2000.
15. Y. Keller and A. Averbuch, "Fast gradient methods based on global motion estimation for video compression," *IEEE Trans. on Circuits and Systems for Video Technology* **13**, pp. 300–309, April 2003.
16. Y. Keller and A. Averbuch, "Fast motion estimation using bidirectional gradient methods," *IEEE Trans. on Image Processing* **13**, pp. 1042–1054, August 2004.
17. M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing* **50**, pp. 174–188, Feb 2002.