## Improved Tracking by Decoupling Camera and Target Motion

**Shawn Lankton  Allen Tannenbaum**  
Georgia Institute of Technology, Atlanta, GA

### Summary

**Challenges**

- **Moving Camera:** Camera motion due to pans, tilts, or a moving platform can introduce large motion not associated with the object.
- **Disappearing Objects:** Especially when camera motion is present, objects may come and go from the field of view. These objects should be uniquely reacquired when they return.

**OUR CONTRIBUTION**

- **Decoupling Motion:** We estimate object and camera motion separately.
- **Completing Kalman Filtering:** Estimating object motion with a Kalman filter and using camera motion as a control input completes the Kalman filter and creates a more principled system.

### Combining in a Kalman 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 $x \in \mathbb{R}^n$, at iteration $k$, in terms of the state at the previous iteration, $k - 1$, and a control signal $u$. This relationship

$$x_k = A x_{k-1} + B u_{k-1}$$

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

$$z_k = H x_k$$

In most tracking computer vision tracking applications, the $B$ term is omitted by assuming that $u$ is always 0. In this work, we use the control signal, $u$, to incorporate the global motion estimate into the filter.

**RESULT**

The predicted target position at each frame is the previous location plus a decoupled combination of target motion and camera motion estimates.

### Experiment: Frame Rate

The framerate sequence demonstrates the technique’s ability to maintain track despite large camera motion. Notice the shifting position of the background throughout these six consecutive frames.

![Frame Rate Experiment](image)

**Efficiency**

The chosen frame registration algorithm and mean shift are both efficient techniques. Thus the overall system is capable of tracking in real time. Our prototype Matlab implementation achieved speeds of at least 15 fps.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Frame Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAMERATE</td>
<td>15.15fps</td>
</tr>
<tr>
<td>DISAPPEAR</td>
<td>16.87fps</td>
</tr>
</tbody>
</table>

### Inspiration

- **Mean Shift:** Mean shift is a popular localization method which provides an efficient way to find a local image region whose histogram best matches some target histogram.
- **Basin of Attraction Dependence:** This local method must be initialized near the object of interest in order to converge to the correct place. This is a major limitation of mean shift.

**Global Motion Estimation**

- **Equivalence:** The global motion is equivalent to the inverse of camera motion when the object motion is relatively small.
- **Efficient Computation:** We employ an efficient frame registration technique that uses only a fraction of image pixels to increase speed with almost no loss in accuracy.

---

**Frames 20, 30, 40, 50, 60, 70, 80, and 90 of the actual sequence.**

**Combining in a Kalman 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 $x \in \mathbb{R}^n$, at iteration $k$, in terms of the state at the previous iteration, $k - 1$, and a control signal $u$. This relationship

$$x_k = A x_{k-1} + B u_{k-1}$$

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

$$z_k = H x_k$$

In most tracking computer vision tracking applications, the $B$ term is omitted by assuming that $u$ is always 0. In this work, we use the control signal, $u$, to incorporate the global motion estimate into the filter.

**RESULT**

The predicted target position at each frame is the previous location plus a decoupled combination of target motion and camera motion estimates.

**Experiment: Disappearing Objects**

In the disappear sequence two of the tracked cars leave view. While out of sight, their position estimate is updated based on camera motion and the objects’ motion model. Based on these two motion estimates, the system predicts where and when the target will come back into view, and reacquires track.

**Frames 10, 20, and 30**

**Frames 40, 50, and 60**

**Frames 70, 80, and 90**

The cars tracked in green and yellow disappear for approx. 50 frames.

**Efficiency**

The chosen frame registration algorithm and mean shift are both efficient techniques. Thus the overall system is capable of tracking in real time. Our prototype Matlab implementation achieved speeds of at least 15 fps.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Frame Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAMERATE</td>
<td>15.15fps</td>
</tr>
<tr>
<td>DISAPPEAR</td>
<td>16.87fps</td>
</tr>
</tbody>
</table>

---

**Frames 20, 30, 40, 50, 60, 70, 80, and 90 of the actual sequence.**