# ROBUST TRACKING WITH MOTION ESTIMATION AND KERNEL-BASED COLOR MODELLING

*R. Venkatesh Babu* *Patrick Pérez* *Patrick Bouthemy*

IRISA/INRIA
Campus de Beaulieu
35042 Rennes Cedex, France
E-mail: {venkat, perez, bouthemy}@irisa.fr

**ABSTRACT**

Visual tracking is still a challenging problem in computer vision. The applications of Visual Tracking are far-reaching, ranging from surveillance and monitoring to smart rooms. In this work, we propose a new method to track arbitrary objects using both sum-of-squared differences (SSD) and color-based mean-shift (MS) trackers in the Kalman filter framework. The SSD and the MS trackers complement each other by overcoming their respective disadvantages. The rapid model change in SSD tracker is overcome by the MS tracker module, while the inability of MS tracker to handle large displacements and occlusions is circumvented by the SSD module. In addition, rapid scale changes of the object generated by camera ego-motion or zooming are measured by a global affine motion estimation. Finally, the global appearance model on which MS relies is updated, based on the Bhattacharyya distance between this target model and current candidate model. This permits to tackle global appearance changes of the object. The performance of the proposed tracker is better than the individual SSD and MS trackers.

## 1. INTRODUCTION

Visual tracking in a cluttered environment remains one of the challenging problems in computer vision for the past few decades. Various applications like surveillance and monitoring, video indexing and retrieval require the ability to faithfully track the objects in a complex scene involving appearance and scale change. Though there exists many techniques for tracking objects, color-based tracking with kernel density estimation, introduced in [1, 2], has recently gained more attention among research community due to its low computational complexity and its robustness to appearance change. The former is due to the use of a deterministic gradient ascent (the "mean shift" iteration) starting at location in previous frame. The latter relies in the use of a global appearance model, usually in terms of colors, as opposed to very precise appearance models such as pixel-wise intensity templates.

Though mean shift (MS) tracker performs well for the sequences with relatively small object displacement, its performance is not guaranteed for objects that get occluded, move fast (inter-frame displacement larger than their size), exhibit large scale changes, or are subject to global appearance (e.g., color) changes.

In this paper, we try to improve the performance of MS tracker against i) large displacements and occlusions, ii) global appearance changes, and iii) large scale changes due to camera operation. For each of these problems, solutions have been considered within pure MS trackers: incorporation of a dynamical model (e.g., using Kalman filter in [1, 3] or particle filter in [4, 5]) to cope with large displacements, occlusions and, to some extent, with scale changes; simple linear histogram updates with fixed forgetting factor [5] for on-line adaptation of reference model; rather complex procedures [6, 7] for addressing the generic problem of scale changes (no matter their origin).

The originality of the proposed approach is to address the three problems within a single and simple approach which exploits the complementarity of global reference color model and instantaneous motion estimation based on pixel-wise intensity conservation. The latter is provided by greedy minimization of the intensity sum-of-squared differences (SSD), which is classic in point tracking and motion field estimation by block matching.

In our approach, the problem with large displacements is tackled by cascading this SSD tracker with a MS tracker. In order to adapt to the current global appearance of the object, the reference model is carefully updated at every frame. Finally, scale changes of the object that are due to the camera zoom effect or ego-motion, are estimated by approximating the dominant apparent image motion by an affine model.

## 2. PROPOSED ALGORITHM

In this work the tracking is done in the Kalman filter framework. The object to be tracked is specified by center location and scale (for a fixed aspect ratio) in the image plane. The objective of the tracking algorithm is to find the correct location in the future frames. An SSD tracker based on frame-to-frame appearance matching is useful in finding the location of the objects in the future frame. However, the problem with the SSD tracker is its short-term memory which can cause drifting problems or even complete loss in worse cases. On the other hand, MS trackers, which rely on persistent global object properties such as color, can be much more robust to detailed appearance changes caused by shape and pose modification. But MS tracker, due to its gradient ascent nature, has problems with large displacements and occlusions. It would be fruitful if we could combine the advantages of the aforementioned two trackers. In this work, we cascade the two trackers to get a better tracking performance. The measurement obtained by this combined tracker module is used for estimating the states of the Kalman filter.

The state-space representation of the tracker to be plugged in

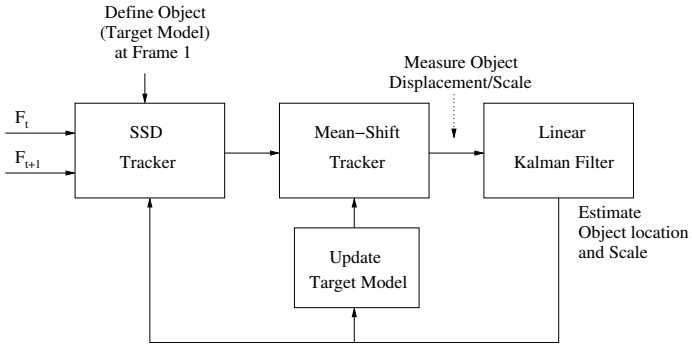


**Fig. 1**. Overview of the proposed tracking system

the Kalman filter is:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ x_t \\ y_t \\ \frac{s_{t+1}}{s_t} \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ \frac{s_t}{s_{t-1}} \end{bmatrix} + \mathbf{w}_t \quad (1)$$

where $\mathbf{x}_t = (x_t, y_t)$ indicates the location of the object center at time $t$, $s_t$ is the scale at time $t$ and $\mathbf{w}_t$ is a white Gaussian noise with diagonal variance $Q$. The measurement equation relates the states and measurements at time $t$ as follows:

$$\begin{bmatrix} u_t \\ v_t \\ \xi_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ \frac{s_t}{s_{t-1}} \end{bmatrix} + \mathbf{z}_t \quad (2)$$

where $\mathbf{u}_t = (u_t, v_t)$ is the measured velocity (displacement) of the object, $\xi_t$ is the measured zoom factor (ratio of scale at time $t$ over scale at time $t-1$), and $\mathbf{z}_t$ is a white Gaussian noise with diagonal variance $R_t$. The displacement measurement $\mathbf{u}_t$ is obtained through the SSD-MS tracker module, whereas scale measurement is provided by global parametric motion estimation. The overview of the proposed system is illustrated in Fig. 1. The following subsections explain each of the module in detail.

### 2.1. SSD-MS motion measurement

The SSD tracker localizes the object in the given search window of the next frame based on minimum distance between the target and candidate object images. SSD tracker works well even for large displacement as long as the object appearance changes only slightly between the two consecutive frames. In reality, the appearance of the object often changes considerably with time. In a typical SSD tracker, the winning candidate becomes the new target for the next time instance. This process makes the SSD to forget the original model rapidly with time though for a given target it performs well between any two consecutive frames.

Given the state estimate $(\hat{\mathbf{x}}_{t-1}, \hat{s}_{t-1})$ at previous instant, the SSD-based displacement estimate is

$$\mathbf{u}_t^{ssd} = \arg\min_{\mathbf{u}\in W} \sum_{\mathbf{d}\in D} [F_t(\mathbf{u} + \hat{\mathbf{x}}_{t-1} + \hat{s}_{t|t-1}\mathbf{d}) - F_{t-1}(\hat{\mathbf{x}}_{t-1} + \hat{s}_{t-1}\mathbf{d})]^2 \quad (3)$$

where $F_{t-1}$ and $F_t$ are the two consecutive intensity images, $\hat{s}_{t|t-1} = \hat{s}_{t-1}$ is the scale prediction, $W$ is the search window, and $D$ is the normalized sub-image support (rectangle of original object's size and origin at its center).

This first displacement estimate is used for initializing the MS tracker. The target color model $\mathbf{q} = (q_i)_{i=1\cdots m}$, with $\sum_{i=1}^m q_i = 1$, is composed of $m$ bins in some appropriate color space (e.g., RGB or Hue-Saturation). It is gathered at the initialization of the overall tracking. The candidate histogram $\mathbf{p}(\mathbf{x}, s)$, at location $\mathbf{x}$ and scale $s$ in the current frame is given by:

$$p_i(\mathbf{x}, s) = \frac{\sum_{\mathbf{d}\in s\cdot D} k(s^{-2}|\mathbf{d}|^2)\delta[b(\mathbf{x}+\mathbf{d}) - i]}{\sum_{\mathbf{d}\in D} k(s^{-2}|\mathbf{d}|^2)} \quad (4)$$

where $k(x)$ is a convex and monotonic decreasing kernel profile, almost everywhere differentiable and with support $D$, which assigns smaller weights to pixels far away from the center, $\delta$ is the Kronecker delta function, and function $b(\mathbf{x}) \in \{1...m\}$ is the color bin number at pixel $\mathbf{x}$ in the current frame. One seeks the location whose associated candidate histogram is as similar as possible to the target one. When similarity is measured by Bhattacharyya coefficient, $\rho(\mathbf{p}, \mathbf{q}) = \sum_i \sqrt{p_i q_i}$, convergence towards the nearest local minima is obtained by the iterative mean-shift procedure [1]. In our case, this gradient ascent at time $t$ is initialized at $\mathbf{y}_0 = \hat{\mathbf{x}}_{t-1} + \mathbf{u}_t^{ssd}$ and proceeds as follows:

1. Given current location $\mathbf{y}_0$ and scales $s$, compute candidate histogram $\mathbf{p}(\mathbf{y}_0, s)$ and Bhattacharyya coefficient $\rho[\mathbf{p}(\mathbf{y}_0, s), \mathbf{q}]$.
2. Compute candidate position

$$\mathbf{y}_1 = \frac{\sum_{\mathbf{d}\in s\cdot D} w(\mathbf{y}_0+\mathbf{d}) k'(s^{-2}|\mathbf{d}|^2)(\mathbf{y}_0+\mathbf{d})}{\sum_{\mathbf{d}\in s\cdot D} w(\mathbf{y}_0+\mathbf{d}) k'(s^{-2}|\mathbf{d}|^2)}$$

with weights at location $\mathbf{x}$

$$w(\mathbf{x}) = \sum_{i=1}^m \sqrt{\frac{q_i}{p_i(\mathbf{y}_0, s)}}\delta[b(\mathbf{x}) - i].$$

3. While $\rho[\mathbf{p}(\mathbf{y}_1, s), \mathbf{q}] < \rho[\mathbf{p}(\mathbf{y}_0, s), \mathbf{q}]$ do $\mathbf{y}_1 \leftarrow \frac{1}{2}(\mathbf{y}_1 + \mathbf{y}_0)$
4. If $\|\mathbf{y}_1 - \mathbf{y}_0\| < \varepsilon$ stop, otherwise set $\mathbf{y}_0 \leftarrow \mathbf{y}_1$ and repeat Step 2.

The final estimate provides the displacement measurement $\mathbf{u}_t = \mathbf{y}_1 - \hat{\mathbf{x}}_{t-1}$. Finally, the two entries associated to this measurement in the covariance matrix $R_t$ of the observation model (2) are chosen as

$$\sigma_u^2 = \sigma_v^2 = \alpha e^{-\beta\kappa(\mathbf{y}_1)}, \quad (5)$$

where $\kappa(\mathbf{y}_1)$ is the curvature of the SSD function around $\mathbf{y}_1$ and $\alpha$ and $\beta$ are two parameters set to $10^2$ and 50 respectively in the experiments.

### 2.2. Scaling measurement

Scaling is another important parameter in visual tracking. Often the scale change of the objects are due to the camera zoom operation or camera ego-motion. The scale change in our work is measured (to be plugged in Kalman Filter) through the affine motion parameters of the global (dominant) image motion between the current and next frame. Such parameters can be estimated in a fast and robust way [8]. If the $2 \times 2$ matrix $A_t$ stands for the

linear part of the affine motion model thus estimated at time $t$, the measured zoom factor is

$$\xi_t = 1 + 0.5\,\mathrm{trace}(A_t). \tag{6}$$

The entry associated to this measurement in the covariance matrix $R_t$ of the observation model (2) is set to a small constant (1 in the experiments).

### 2.3. Target model update

Updating the target model is one of the crucial issues in tracking. The performance of the mean-shift algorithm decreases considerably when the global appearance of the objects changes with time. In this case, the color histogram obtained from the target definition from the first frame correlates less with the current view of the tracked object. In order to maintain the effectiveness of the mean-shift tracker in this scenario, it is essential to update the target model while tracking. This model update helps the tracker to perform well in a cluttered background condition and appearance changes. In our system the Bhattacharyya distance, which measures the distance between target model and model at current location estimate provided by the Kalman filter, is used to update the reference. This model update is a trade-off between adaptation to rapid changes and robustness to changes due to occlusion. In order to be on safer side, in our system candidate models close to the target contributes more than farther ones. The update procedure used is defined as:

$$\mathbf{q}_{t+1} \propto \mathbf{q}_t + e^{-\alpha[1-\rho(\mathbf{q}_t, \mathbf{p}(\hat{\mathbf{x}}_t, \hat{s}_t))]}\mathbf{p}(\hat{\mathbf{x}}_t, \hat{s}_t), \tag{7}$$

where $\alpha$ is a real positive scalar, which determines the model update rate. Typical value of $\alpha$ used in our experiments is set to 10.

### 2.4. Algorithm summary

The complete algorithm is summarized below. Given previous reference color model $\mathbf{q}_{t-1}$ and previous state estimate $(\hat{\mathbf{x}}_{t-1}, \hat{s}_{t-1})$ with error covariance $P_{t-1}$:

1. Obtain SSD-based displacement measurement $\mathbf{u}_t^{ssd}$ according to (3).
2. Correct this measurement with MS iterative search, initialized at $\mathbf{u}_t^{ssd}$ and with reference color model $\mathbf{q}_{t-1}$, to obtain final measurement $\mathbf{u}_t$.
3. Estimate global affine motion over the image and derive new scale measurement $\xi_t$ according to (6).
4. Using displacement and scale measurement $\mathbf{u}_t$ and $\xi_t$, update state estimate with Kalman filter, providing $(\hat{\mathbf{x}}_t, \hat{s}_t)$ and associated error covariance $P_t$.
5. Update target color model according to (7) to get $\mathbf{q}_t$.

Initial state $(\hat{\mathbf{x}}_1, \hat{s}_1 = 1)$ in frame 1 is obtained either by manual interaction or by detection, depending on the scenario of interest. Initial reference color model is then $\mathbf{q}_1 = \mathbf{p}(\hat{\mathbf{x}}_1, \hat{s}_1)$.

## 3. RESULTS AND DISCUSSION

The proposed algorithm has been tested on several videos and the proposed tracking system, which uses both SSD and mean-shift tracker, works better than individual SSD or mean-shift tracker.

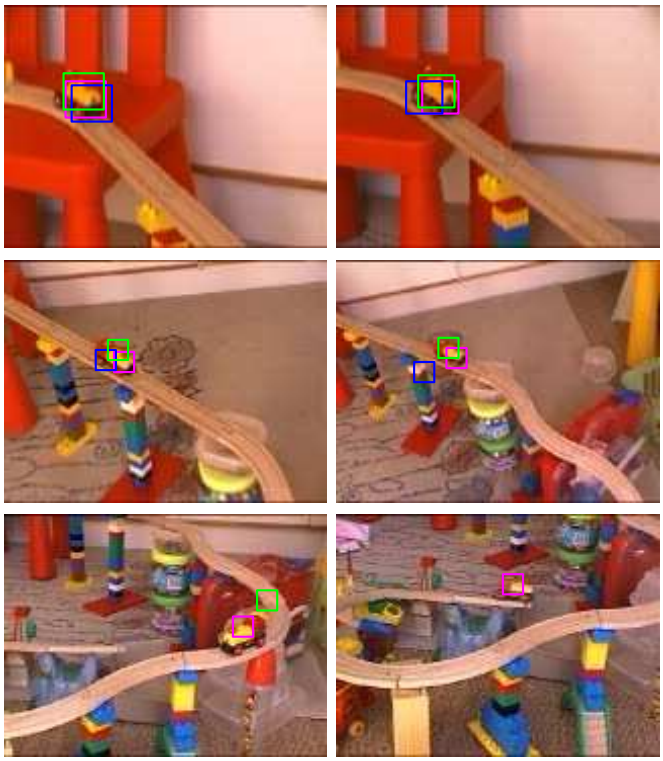

**Fig. 2**. Tracking result of proposed system (magenta) against SSD (green) and MS (blue) tracker for 'train' sequence. Frames shown 20, 50, 200, 270, 620 and 1150.

Tracking results for three personal videos of low quality taken by a hand-held camera are presented in Figs. 2, 3 and 4.

In the first sequence, there is a lot of shaking which was challenging for the trackers to follow the objects. The proposed algorithm was able to track the toy train throughout the sequence. It is observed that the mean-shift tracker just oscillates about the object whenever there is a heavy shake and gets lost later, whereas the SSD performs well approximately till 600th frame and collapses after a heavy shake. It can be seen that the toy train turns around almost 180 degrees from starting to end, and experiences substantial scale changes due to camera zooming in and out. The model update helps here to learn the object while tracking.

In the second sequence, the fast movements of the camera and of the racing go-carts result in large displacements in the image and dramatic motion blur. In addition, go-carts get briefly occluded. Despite all these difficulties, the combined tracker manages to track successfully one go-cart, whereas the SSD and MS trackers get lost.

In the third sequence, the results were presented for the frames that are temporally subsampled by 3. In this sequence, the rapid camera movement makes the MS and SSD tracker fail. The combined tracker tracks the object correctly with proper zooming.

## 4. CONCLUSION

In this paper, we have proposed an efficient visual tracker by coupling SSD and mean-shift algorithms, which have complementary properties. The approach also includes scale adaptation ac-

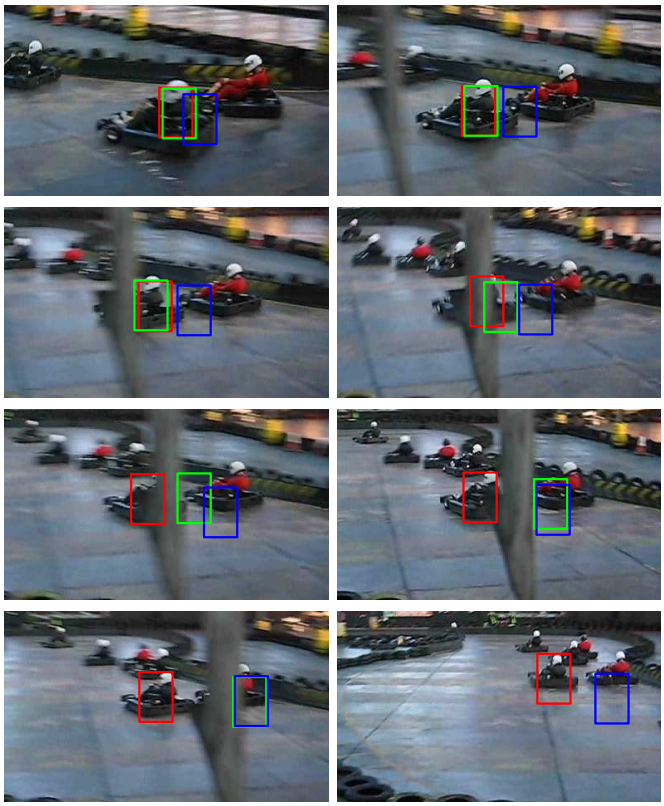

**Fig. 3**. Tracking result of proposed system (red) against SSD (green) and MS (blue) tracker for 'go-carts' sequence. Frames shown 35, 42, 45, 46, 47, 48, 50, 58.

counting for camera zoom operations and ego-motion, and on-line adaptation of the kernel-based reference color model. The resulting tracker performs well even for objects that move fast in cluttered background, get occluded, change appearance over time and change scale due to camera operations. Since both SSD and M-S trackers have real-time computational complexity, the proposed compound tracker is suitable for real time tracking of objects.

## 5. REFERENCES


[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. Conf. Comp. Vision and Pattern Recog.*, Hilton Head, SC, 2000.

[2] G. Bradski, "Computer vision face tracking as a component of a perceptual user interface," in *Workshop on App. of Comp. Vision*, Princeton, NJ, 1998.

[3] Z. Zhu, Q. Ji, and K Fujimura, "Combining kalman filtering and mean shift for real time eye tracking under active ir illumination," in *Proc. Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002.

[4] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Europ. Conf. Computer Vision*, Copenhagen, Denmark, 2002.

[5] K. Nummiaro, E. Koller-Meier, and L.J. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.


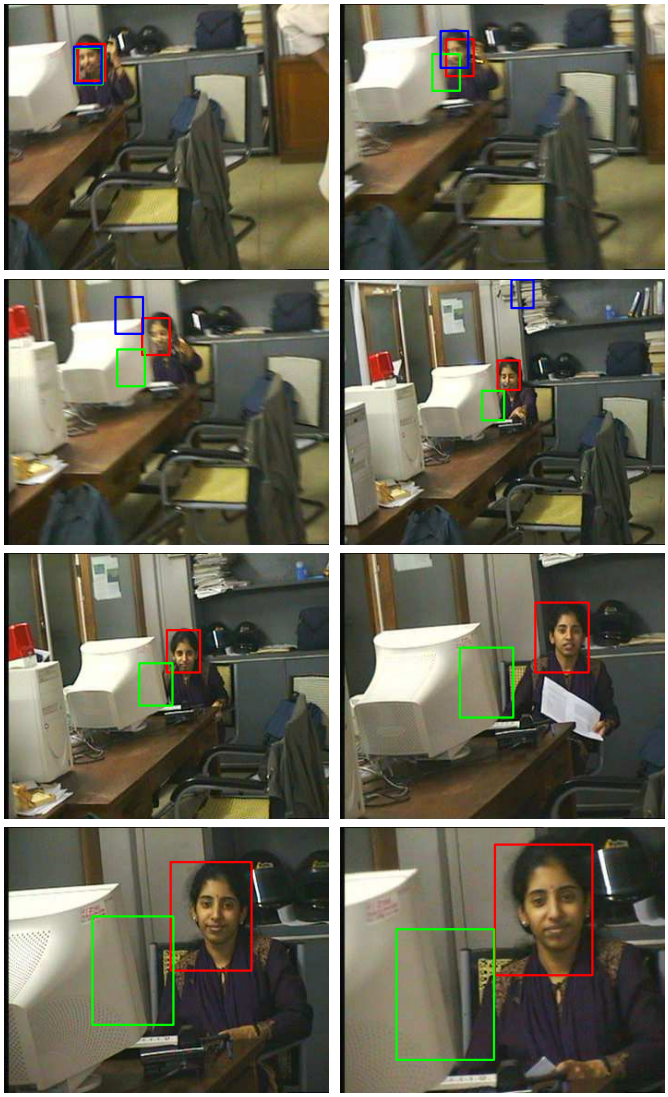

**Fig. 4**. Tracking result of proposed system (red) against SSD (green) and MS (blue) tracker for 'lab' sequence. Frames shown 53, 56, 59, 101, 149, 200, 251, 296.


[6] R. Collins, "Mean-shift blob tracking through scale space," in *Proc. Conf. Comp. Vision Pattern Rec.*, Madison, Wisconsin, 2003.

[7] Z. Zivkovic and B. Kröse, "An EM-like algorithm for color-histogram-based object tracking," in *Proc. Conf. Comp. Vision Pattern Rec.*, Washington, DC, 2004.

[8] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Visual Com. Image Repr.*, vol. 6, no. 4, pp. 348–365, 1995.