Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues

Vijay Badrinarayanan*, Patrick Perez†, Francois Le Clerc*, Lionel Oisel*

*Thomson R&D France
1 Avenue de Belle-Fontaine
35576 - Cesson Sevigne, France
{vijay.badrinarayanan}@thomson.net

† INRIA Rennes-Bretagne Atlantique
Campus Universitaire de Beaulieu
35042 Rennes Cedex - FRANCE
perez@irisa.fr

Abstract

We present a probabilistic multi-cue tracking approach constructed by employing a novel randomized template tracker and a constant color model based particle filter. Our approach is based on deriving simple binary confidence measures for each tracker which aid priority based switching between the two fundamental cues for state estimation. Thereby the state of the object is estimated from one of the two distributions associated to the cues at each tracking step. This switching also brings about interaction between the cues at irregular intervals in the form of cross sampling.

Within this scheme, we tackle the important aspect of dynamic target model adaptation under randomized template tracking which, by construction, possesses the ability to adapt to changing object appearances. Further, to track the object through occlusions we interrupt sequential resampling and achieve relock using the color cue.

In order to evaluate the efficacy of this scheme, we put it to test against several state of art trackers using the VIVID online evaluation program and make quantitative comparisons.

1. Introduction

The problem of combining visual cues, predominantly color, shape and templates, to achieve robust object tracking has been prevalent in the research community for some time now. Of the most recent contributions we find: A Kalman filter framework for combining geometric templates, color tracking and blob detection in [13] primarily focussed on tracking vehicles on highways; An adaptive particle filtering technique due to Emilio *et al*. [9] employing color and orientation information to formulate a likelihood as a function of derived cue uncertainties; The "Blackboxes" approach [7] to combining intra and inter state space measurements relying on assumptions like conditional independency of measurements given the state and deterministic mapping between state spaces; The flocks of features and color integration by Mathias *et al*. [6] for hand tracking; And fusion of visual cues with regard to their measurement consistency by Hua *et al*. [4].

The common trend in all of these approaches has been to arrive at a single fused target state distribution at every tracking step and then estimate the target state from it. Two points need to be pondered upon here.

First, statistical measures like covariances, derived to quantify the benefits of this kind of fusion by comparing marginal uncertainties and fused uncertainty, are consistent if the underlying marginal distributions of the cues are Gaussian in nature and even so, only if all possible cross correlations are considered. This is not true in practice and theoretically evaluating cross covariances involves influencing one cue by another until all possible cue combinations are exhausted. The same is true for more general quantities like conditional entropies and mutual information. They are themselves difficult to approximate on irregularly sampled distributions generated from Monte Carlo methods [10]. Further finite sample approximations necessitate weight-variance control techniques like sequential resampling [1] which compound difficulties in analysing the time evolution of these quantities.

Secondly, it is somewhat unclear in many cases as to what the role of the fused distribution is, in terms of influencing future tracking for each of the cues. Template tracking is a clear case in point. How does the fused distribution influence, say, the search space of the templates?

Apart from the points enlisted above it is also difficult to draw comparative conclusions about any of these techniques owing to their niche applications and varying test sequences. Bearing all this in mind, we set forth to investigate a simpler strategy of online switching between color based filtering and template tracking for state esti-

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

mation. Thereby, we make no attempt to produce a fused distribution at every tracking step and consequently derive the state estimate from one of the two state distributions associated to each of the modalities. This implies a priority based system using binary confidence measures.

We begin with template tracking in section 2, wherein we introduce a novel randomized template tracker. This tracker possesses the desirable ability to adapt to continuous target appearance changes. In section 3, we discuss the elements of the color based particle filter and study the influence of interrupting sequential resampling on occlusions. In section 4, we delve into multi-cue tracker and the interactions between the cues therein. We discuss the experimental setup, third party evaluations on the PETS database and results in section 5. Following this, in section 6 we discuss the limitations of the presented approach and prospective work. We conclude thereafter.

2. Randomized Template Tracking

The defining aspect of the randomized template tracking is that all the template (rectangular patch) locations throughout the tracking sequence are chosen from random draws from the target state distribution at each instant and have randomly varying dimensions. No conventional feature selection procedure like corner detectors [3] or SIFT-features [8] is made use of in choosing the templates.

Given a hand marked or detected object in the starting frame of a video sequence, we randomly draw a small set, say N, of template locations using a uniform number generator and associate each of these locations to a template of predefined maximum size. The templates are trimmed to lie within the marked object boundary, which may probably result in varying their dimensions. At this point we structure the explanation of this tracker into the following three parts which are sequentially repeated in the same order at every tracking step unless a tracking failure is signaled.

1. **Template Tracking:** Each template is tracked using the common normalized cross-correlation tracker in a predefined rectangular search space centered on the location of the template at the previous instant. This associates every template with a correlation surface at each tracking step. The grid location corresponding to the maximum of the correlation surface is the estimated new location of the template. Consequently, a translation vector is derived for each template in the set. We therefore have a set of templates, associated two dimensional motion vectors and correlation surfaces, denoted by $\{T_i, V_i, \chi_i\}_{i=1}^{i=N}$.
   We then discard a subset of the these templates using a robust clustering scheme in motion space.

2. **Probabilistic Rejection Control:**
   Clustering:
   We augment the set in step 1 to $\{T_i, V_i, B_i\}_{i=1}^{i=N}$, where B_i is termed the bin count for the i^{th}. The bin count is set to 1 initially for all the templates.
   Each motion vector in the set $\{V_i\}_{i=1}^{i=N}$ is compared with the remaining vectors using an Euclidean distance measure and a corresponding bin-count is incremented for every vector that lies within a small predefined clustering radius r of this vector.
   A subset $\{V_i\}_{i\in I}, |I| \leq N$ of $\{V_i\}_{i=0}^{i=N}$ is formed by selecting all the motion vectors with associated bin-count $B_i \geq \frac{N}{2}$. If N is odd we define the relation as $B_i \geq \frac{N+1}{2}$.
   Rejection Control:
   We compute the two-dimensional mean of the subset $\{V_i\}_{i\in I}, |I| \leq N$ and the resulting covariance matrix, denoted as $\{\mu, \Sigma\}$ respectively. For computation sake, we assume the cross variances to be zero.
   We setup a Gaussian distribution $g(V)$ in motion space with these parameters and assign a weight $w_i = g(V_i)$ to each motion vector in the set $\{V_i\}_{i=1}^{i=N}$. We now form the set $\{T_i, w_i\}_{i=1}^{i=N}$.
   We sort this set $\{T_i, w_i\}_{i=1}^{i=N}$ in a descending order of weights and choose the weight of the element with index $\frac{N}{2} - 1$ if N is even or $\frac{N-1}{2}$ if N is odd. We denote this weight as C.
   Each template is accepted with a probability

$$p_i = \min\left\{1.0, \frac{w_i}{C}\right\} \quad (1)$$

   We then discard templates based on a threshold on this probability. At this juncture, we consider tracking to be successful if a minimum of $\frac{N}{2}$ if N is even or $\frac{N+1}{2}$ is odd, templates are retained.
   In the event of successful tracking, the correlation surfaces of the retained templates are combined into a probability distribution, from which new templates are resampled to replenish the template set. We elaborate below.

3. **Residual Resampling:**
   Let $\{T_i\}_{i=0}^{i=M}$, where $M \leq N$, be the retained templates at the end of rejection control.
   With $\{T_i\}_{i=0}^{i=M}$, we estimate the target state x_{est} (throughout this paper we assume the target state is the location of the target object on the image grid) to be the mean of these template locations. We also construct the minimum bounding rectangle, denoted B_{min}, around these templates, which acts as the spatial bound for the state distribution computed below.
   Given x_{est}, we *temporarily* place the centre of the correlation surfaces $\{\chi_i\}_{i=0}^{M}$ associated with $\{T_i\}_{i=0}^{i=M}$ to this location, sum and normalize, within B_{min}, to re-

sult in a probability distribution $p(x)$ as indicated below.

$$p(x) \propto \Sigma_{i=1}^{M} \chi_i(x) \tag{2}$$

We consider $p(x)$ as the distribution of target state generated by the randomized template tracker. From this distribution we sample $N - M$ template locations from $p(x)$ as follows.

$$x_k \sim p(x), 1 \leq k \leq (N - M) \tag{3}$$

To each sampled location in the set $\{x_k\}$, we add Gaussian noise to increase sample diversity;

$$\hat{x}_k = x_k + \eta(\left[\begin{array}{c} 0 \\ 0 \end{array}\right], \sigma^2 \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right]). \tag{4}$$

Finally, to each sample location $\hat{x}_k$ we associate an image patch around it as described below.

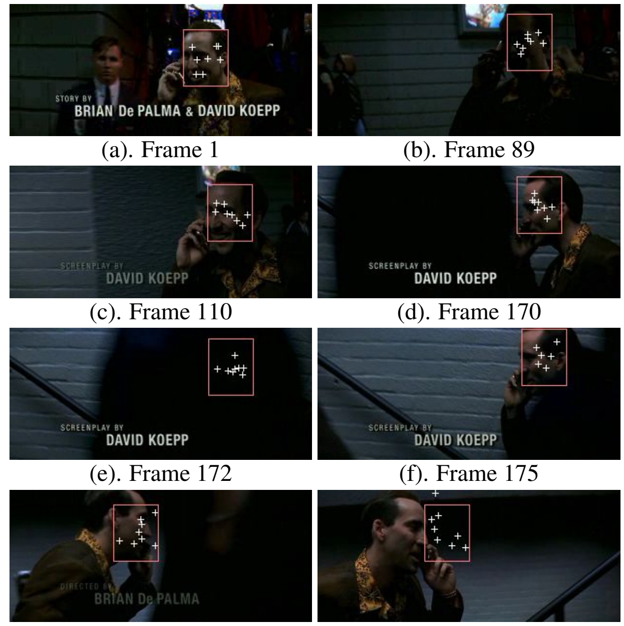
4. **Template setting:**
In absence of a natural scale estimation scheme here, we are forced to remain conservative in choosing the dimensions of the new template. The chosen template is trimmed to fit $\alpha B_{min}, \alpha \geq 1$. The value $\alpha = 1$ is the most conservative estimate which we employ when tracking very small sized objects in aerial videos and the like. Here α must not be misconstrued to be a scale estimate of the tracked object.

From the procedure delineated above, we see that residual resampling favors new template locations at proximity to the template locations which have survived for a longer duration when $\alpha = 1$ and η is very low in magnitude (both due to lack of scale information). This results in lack of diversity of templates over long periods of time and biases the state estimate towards the location of the longest surviving template. Some knowledge of the evolution of the target object shape is necessary to offset this drift. This is currently beyond the scope of this paper.

Thus far the discussion of this scheme concentrated on the procedure when the tracking was signaled as successful at the end of the probabilistic rejection control step. In the event of a failure, the control passes to the following step.

Interrupting resampling: We retain all templates in the set $\{T_i\}_{i=0}^{i=N}$ and extrapolate their positions by the last successfully registered object translation, meaning the translation of the target state at the step in the past when tracking was signaled as successful. The control is then passed to step 1 when the next frame arrives. Such a scheme is found to be useful in handling very short occlusions of a few frames.

A series of snapshots of head tracking results for the "snakeeyes" test sequence are presented in Fig. 1. The total length of the sequence is about 1000 frames and among these we display only a small selection. The full video of



(a). Frame 1 (b). Frame 89
(c). Frame 110 (d). Frame 170
(e). Frame 172 (f). Frame 175
(g). Frame 318 (h). Frame 330

Figure 1. Randomized Template Tracking on the Snakeeyes sequence.

this tracking result can be found in the supplemental data.

In each selected frame the template centres are marked by small crosses. The ability to track through extreme illumination changes, Figs. 1(b), 1(c) and short occlusions Figs. 1(d), 1(e), 1(f) are highlighted in this sequence. The loss of diversity in templates too is discernible in Fig. 1(b). Finally, tracking loss due to a long occlusion can be seen in Fig. 1(h).

In Fig. 2 we present a graphical plot of the number of templates retained after each execution of rejection control for the snakeeyes sequence in Fig. 1. The occlusion frames eliminate most of the templates as expected, otherwise only a few or none are eliminated including the duration of extreme illumination changes.

For the snakeeyes sequence in Fig. 1, we also plot the age in frames of the oldest living template at each frame of the sequence in Fig. 3. We see that average age of the longest living template is about 30 frames. This provides some insight into the stability of tracking via templates chosen at *random*.

3. Color based Tracking

The tracker we employ is the color based particle filter proposed by Perez *et al.* [11], with resampling at each tracking step. The filtering scheme utilises the concept of importance sampling and arrives at a probability distribution of the target state at each tracking step, described by a set of sample locations and associated weights, which we denote by $\{x_i, \pi_i\}_{i=0}^{i=K}$ (See [1] for further details). The weights

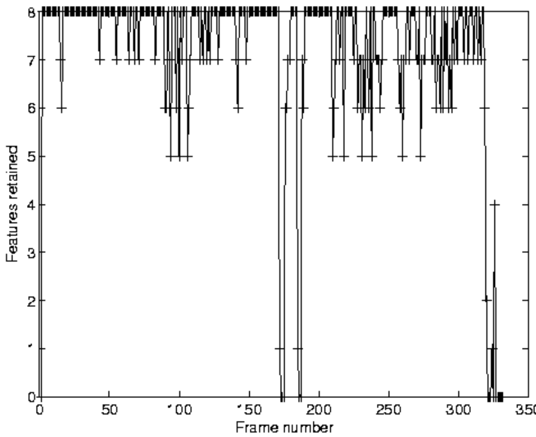


Figure 2. A plot of the number of templates retained at each frame versus the frame number.

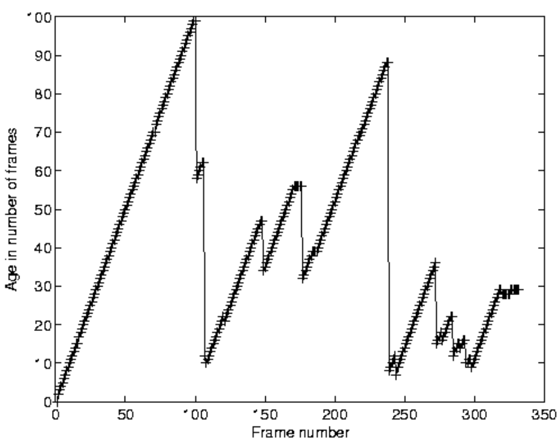


Figure 3. A plot of the age of the oldest template versus the frame number for the snakeeyes sequence.

$\{\pi_i\}$ sum to unity. We then estimate the target state as the *mean* of this distribution.

At each tracking step we aim to signal a successful tracking by computing, what we term the acceptance probability. This may equivalently be viewed as a rejection probability to keep consistency with the rejection control in the randomized feature tracker. We compute this probability as follows.

Computing the rejection probability: Given the set $\{x_i, \pi_i\}_{i=0}^{i=K}$ we compute the covariance matrix C_π of this distribution. We also compute the covariance matrix C_s of the set $\left\{x_i, \frac{1}{K}\right\}_{i=0}^{i=K}$, which is essentially the covariance of the filtering distribution before weighting by the likelihood.

We now compute the determinants of C_π and C_s. From the property of determinants we know that this scalar quantity measures the effective volume of the parallelopiped constructed by the row vectors of the matrix [12] (This is also equivalent to analysing the (covariance) matrix as a function of its eigen values [9, 12]). This motivates us to employ this quantity as a scalar confidence measure.
With the above notations, we define the rejection probability p_r as follows:

$$p_r = \min\left\{1.0, \frac{\det[\mathrm{C}_\pi]}{\det[\mathrm{C_s}]}\right\} \tag{5}$$

p_r tends to 1 as the uncertainty in the distribution increases and tends towards 0 as the distribution becomes more peaked. It is interesting to note that it can be inconsistent to analyse the performance of the filter based solely on evolution of its covariance over time. This is because the spread (covariance) of the samples at each tracking step is not constant and even with resampling there is bound to be some fluctuations. Therefore, it is necessary to account for this variable spread via a factor like C_s. Finally, we signal a tracking success (or equivalently failure) if p_r is less (or greater) than an empirical threshold. In the event of a success at the end of the preceding analysis the filtering distribution is resampled (See [1] for details) and the filtering recommences when the next frame arrives. In the opposite event we adopt the following procedure.

Interrupting Resampling: We temporarily arrest resampling the filtering distribution and this causes the samples to be more and more spread at each tracking step. In the absence of clutter, the sample weights tend to be uniform which results in p_r tending to 1. But once a subset of the samples gains distinctly more weight (say in a relock scenario after a few frames of occlusion) a few frames later, the rejection probability p_r tends towards 0 leading to a success signal.

In Fig. 4 we present snapshots of color based tracking on the snakeeyes sequence (See Fig. 1) . The instability of the color tracker under strong illumination changes, Figs. 4(b), 4(c) and clutter, Figs. 4(c), 4(f) can be observed here. On the other hand its ability to relock onto the target after occlusions (of about 10 frames) can also be clearly seen here in Figs. 4(e), 4(h).The full video of this tracking result can be found in the supplemental data.

In Fig. 5, we provide a plot of the evolution of the ratio of determinants against the frame number. As may be expected the ratio is close to 1.0 when the illumination changes (compare with Fig. 4(b)), during occlusions (compare with Fig. 4(d)) and clutter (compare with Fig. 4(f)).

4. Multi-Cue Tracking

At the initialization stage of tracking, we assign priority to one of the two trackers, either the randomized template based or color based tracker. This priority is switched between the two depending on their current signs of success or failure (See section 2 and 3) and controls the execution

(a). Frame 1 (b). Frame 89
(c). Frame 110 (d). Frame 169
(e). Frame 175 (f). Frame 223
(g). Frame 319 (h). Frame 326

Figure 4. Color Based Tracking on the Snakeeyes sequence

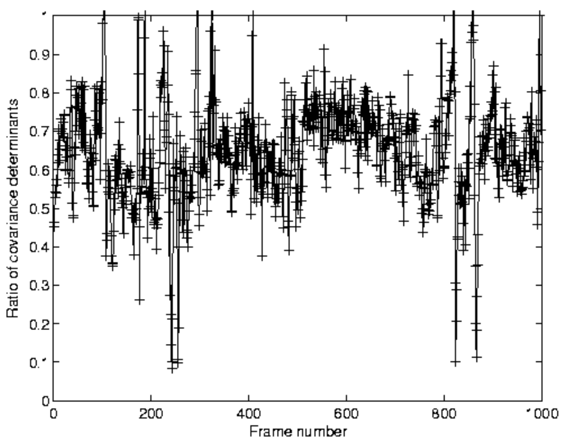


Figure 5. A plot of the ratio of covariance determinants versus the frame number for the snakeeyes sequence.

order of algorithms at each tracking step. Furthermore, the priority algorithm at each instant makes an estimation of the target state.

For the rest of this discussion we will refer to the two algorithms as tracker **T** and tracker **C**, where **T** stands for template tracking and **C** for color. We also assume that at the initialization stage tracker **T** has been given the priority. The trackers then are run independently, **T** first at each instant and **C** second, without any interaction or information exchange. The state estimate as stated earlier is made from tracker **T**. Now say at some later instant tracker **T** fails but **C** is successful, possibly when there is partial occlusion and/or jerky out of the image plane rotation. We then proceed to the following interaction step.

Cross Sampling Templates: The state estimate is now made from tracker **C** and the priority switched to tracker **C**.

The entire template set $\{T_i\}_{i=0}^{i=N}$ is discarded and a new set of N template locations are sampled from the color filtering distribution at the corresponding instant and each location assigned a image template. The templates are then trimmed, if necessary, to fit the bounds of the object as decided by tracker **C**. It is also to be noticed that the past object model composed by the templates is totally discarded and updated at the current instant from the color distribution.

Now, say we arrive at a scenario where tracker **C** fails and **T** is successful, typically when there are extreme illumination changes and/or nearby clutter. We then adopt the following procedure.

Cross Sampling Particles: The state estimate is now made from the tracker **T** and the priority switched to tracker **T**. The current samples which compose the filtering distribution are replaced by new samples drawn from the target state distribution output by tracker **T** and each sample weighted equally. The color model is however *not* updated.

We have until now described the procedures adopted when one of the two trackers fail. The remaining case is when both the trackers signal their respective failure at the same tracking step, typically a complete occlusion scenario where the occluding object shares no commonality with the target reference model. We deal with this situation as follows.

Interrupting Resampling: We temporarily arrest the resampling step in both the trackers **T** and **C** and resort to *interrupting resampling* step in both the trackers (See Section 2 and 3 for more details on this procedure). This is continued until one or both of the trackers are signaled as successful.

A graphical synopsis of the entire algorithm is presented in Fig. 6 for the readers convenience.

In Fig. 7 we present snapshots of multi-cue tracking on the snakeeyes sequence (See Fig. 1). The blue bounding boxes indicate the priority for state estimation lies with the template tracker at that instant and the yellow bounding boxes indicate color priority. The template centres are also marked by crosses for visual evaluation. The ability to handle several occlusions, Figs. 7(e), 7(i), illumination changes, Figs. 7(b), 7(g) and drastic orientation changes, Figs. 7(f), 7(l), 7(m), 7(n) can all be seen here.The full video result is also included in the supplemental data for visual evaluation.

Before terminating the discussions of this section, we wish to comment upon the ability of the multicue tracker to adapt to varying target appearances.

Target Model Adaptation: The target model which is em-

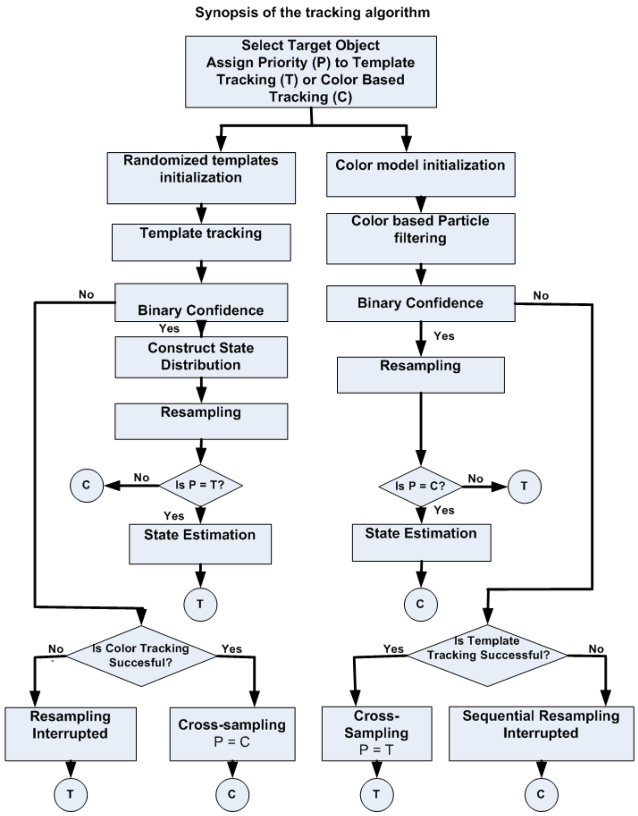
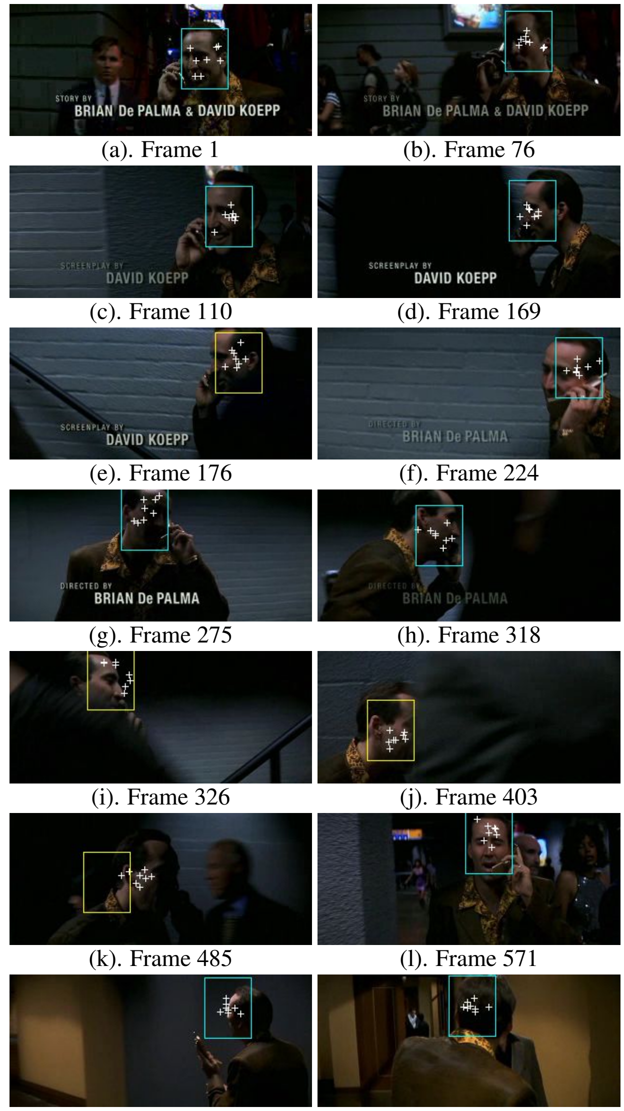


Figure 6. Flow Chart of the Multi Cue Tracking algorithm

ployed in this tracker is a multi-part model. The first part is composed of the gray level templates. Due to the sequential resampling procedure, at any tracking instant, the age of each element in the template set is possibly different from the rest. Therefore, this set consists of templates having lifespans of a few to several tens of frames and thus plays the role of the dynamically adapted part of the entire target model.

The second part of the model is the constant color histogram of the target object. This is the static part of the two part target appearance model and does not interact with the first part. The histogram is deliberately kept constant to avoid false adaptation due to illumination, orientation and size changes.

It is interesting to note that this two part model bears similarity to the scheme proposed by Jepson *et al.* [5]. Their scheme involves explicit construction of a fused object model with static, slowly varying and fast varying components. In our scheme, the extreme static part is the color histogram and the other two components can be mapped to the age variant template set.



(a). Frame 1 (b). Frame 76
(c). Frame 110 (d). Frame 169
(e). Frame 176 (f). Frame 224
(g). Frame 275 (h). Frame 318
(i). Frame 326 (j). Frame 403
(k). Frame 485 (l). Frame 571
(m). Frame 790 (n). Frame 874

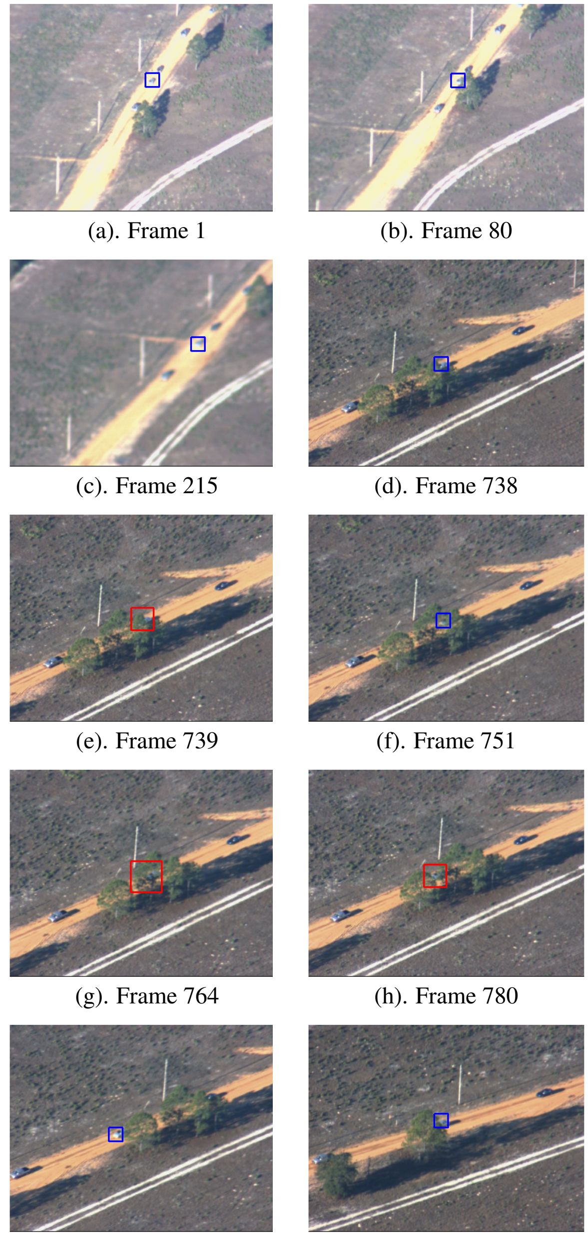
Figure 7. Multi-Cue Tracking on the Snakeeyes sequence.

5. Experiments and results

In all of the conducted experiments, the template tracker used 8 templates. The initial clustering radius was set at 5. The acceptance probability threshold was empirically set at a value of 0.8, which in simple terms means a template was retained if its acceptance probaility exceeded this value.

The color based tracker was run with 200 particles. The state dynamics used was a first order Markov model with random walk of standard deviation 4.0. The rejection probability threshold was empirically set at 0.9, meaning the color tracker was declared unsuccessful if the computed rejection probability exceeded this threshold. It is to be noted that this threshold is inversely proportional to the standard deviation of the random walk.

We tested our algorithm on all ground truthed color se-



(a). Frame 1 (b). Frame 80
(c). Frame 215 (d). Frame 738
(e). Frame 739 (f). Frame 751
(g). Frame 764 (h). Frame 780
(i). Frame 810 (j). Frame 943

Figure 8. Particle Filtering with Interrupted Resampling on PETS-2005 Egtest04 sequence. The bounding box is turned red when the resampling is interrupted temporarily. The size of the red bounding box is also increased when the resampling is interrupted over a large number of frames.

quences provided on the VIVID-PETS database [2]. The sequences were all aerial videos of moving cars or military vehicles. The sequences provide wide variety of challenges like tracking very small size of the objects, variable motions, defocus blurs, scale changes due to camera zoom variations and object orientation changes, extreme illumination changes due to sunlight glare and partial to complete occlusions. The database provides an online third party evaluation system to analyse the performance of the algorithm against several evaluation parameters, primary of which we study is the percentage of total frames tracked. Other parameters like shape matches are beyond the scope of this paper. The evaluation system also provides comparative results against several fundamental/ state of the art trackers. The results of this evaluation are produced in Table 1 for all the experimented sequences.

Table 1 compares the performance of our algorithm when the initial priority is switched between the two cues. Apart from the PETS sequences the table also contains results for the snake eyes sequence and the Fleet sequence [5]. These sequences however have not been evaluted against any ground truth.

From the results in Table 1, we see the algorithm outperforms most state of art trackers and in cases where it does not it only lags by a negligible percentage. The fact that the overall performance on any test sequence does not change when the initial priority is varied confirms its insensitivity to the initial priority and the robustness that is achieved due to multi cue tracking.

The fact that multi-cue tracking can clearly outperform a solo cue tracker can be clearly observed from the results of the snakeeyes sequence. Each of the two trackers is unable to track through the entire sequence, but their combination achieves this goal.

6. Limitations and Prospective work

The proposed algorithm does not estimate scale changes over time. Although it is possible to estimate scale changes due to camera zoom using an affine motion model, this procedure would lead to unstable results when scale varies due to changes in object orientation or object distance to the camera. Therefore we abstain from estimating any global motion model and instead aim to tackle this issue by including a shape estimation framework. Another interesting prospect for scale estimation resulting from the current experiments is that the large scale variations of the target object does not entirely affect the template tracker due to the presence of randomly varying dimensions of the templates. A comparitive investigation into the dimension of the surviving templates versus change in target scale may shed some light on this problem.

The second issue is that the tracking fails when there is gradually increasing occlusion over a few frames, in which case the resampled templates have a tendency to accrue parts of the occluding object. This causes the template tracker to lock onto another object and therefore causing the entire scheme to fail. The test sequences Egtest04 and Egtest05 earlier presented such a situation. It is however interesting to note that the color based tracker alone with interrupted resampling can provide effective tracking through

	Test sequences and % of total frames tracked in each							
Candidate Algorithms	Egtest01	Egtest02	Egtest03	Egtest04	Egtest05	RedTeam	Snakeeyes	Fleet
Proposed Algorithm (C†)	100.0%	100.0%	100.0%	39.89%	13.64%	100.0%	98.7%	100.0%
Color proportion*	99.4%	19.9%	11.1%	99.6%	60%	27.5%	18%	48%
Proposed Algorithm (T†)	100.0% ()	100.0%	100.0%	39.89%	13.64%	100.0%	98.7%	100%
Template proportion*	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	78%	100%
PeakDiff	100.0%	30.77%	12.06%	3.83%	13.64%	98.43%		
Mean Shift	17.58%	39.23%	20.62%	9.84%	13.64%	84.29%		
FgBg Ratio	100.0%	39.23%	17.90%	8.74%	13.64%	100.0%		
Adaptive Shape Color	100.0%	100.0%	20.23%	9.84%		100.0%		
Variance Ratio	29.12%	27.69%	12.06%	9.84%	13.64%	100.0%		
Particle Filter	99.45%			40.44%		100.0%		
Template Match	87.91%	21.54%	12.84%	2.73%	17.61%	16.23%		
GraphCut	2.20%	15.38%	3.11%	0.55%	2.27%	34.03%		

Table 1. The figures indicate the percentage of total frames tracked in each sequence. C†, T† indicates the initial priority was assigned to the color based tracker and template tracker respectively. * indicates the proportion of frames in which this cue was used for target state estimation. The blank boxes indicate non availability of results. Courtesy: VIVID Evaluation [2].

such situations.

The snapshots of this tracking result on the Egtest04 sequence is presented in Fig. 8. The bounding box has been purposefully exaggerated for visual comfort. The bounding box is also represented in red at instances where the resampling has been temporarily interrupted, Figs. 8(e), 8(g), 8(h). The prospect of handling long occlusions can be observed here in Figs. 8(f), 8(i) . For comparisons sake it is interesting to note that more than 80% of the Egtest04 sequence can be tracked by this method (compare with table 1) .

The results in Fig. 8 provide a good prospect to track through complex occlusions if the template tracker drift can be effectively handled.

7. Conclusions

The goal of the work presented here was twofold. The first being the introduction of the randomized template based tracking which by itself acts a multi-part model of the object, with age-variant templates. This removes all necessity to construct a separate adaptive model of the target by an exclusive technique. Moreover since the template selection is random there is no extra feature selection technique necessary. The second goal was to experiment multi-cue tracking with binary confidence measures and minimal interaction between the cues. The end result of this experiment was a successful algorithm which outperforms most state of the art trackers. However, the approach is still sensitive to drifts due to gradual occlusions and insensitive to changes in the target size. The solutions to these problems would constitute extensions of this work in the near future.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/ non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.

[2] R. T. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2005.

[3] C. Harris and M. J. Stephens. A combined corner and edge detector. In *In Alvey Vision Conference*, 1988.

[4] G. Hua, Y. Wu, and Z. Fan. Measurement integration under inconsistency for robust tracking. In *CVPR, New York City, NY*, 2006.

[5] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on PAMI*, 25(10):1296–1311, 2003.

[6] M. Kolsch and M. Turk. Fast 2d tracking with flocks of features and multi-cue integration. In *CVPR Workshop*, 2004.

[7] I. Leichter, M. Lindenbaum, and E. Rivlin. A generalised framework for combining visual trackers - the "black boxes approach". *IJCV*, 67(3):343–363, 2006.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[9] E. Maggio, F. Smeraldi, and A. Cavallaro. Combining colour and orientation for adaptive particle filter-based tracking. In *BMVC*, 2005.

[10] I. Nemenman. Entropy estimation: A general overview. In *NIPS Entropy Estimation Workshop*, 2003.

[11] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. IEEE*, 92(3):495–513, 2004.

[12] G. Strang. *Linear Algebra and its Applications*. Brooks Cole, 1988.

[13] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos. Robust target detection and tracking through integration of motion, color and geometry. *CVIU*, 103:121–138, 2006.