Integrating robust likelihoods with Monte-Carlo filters for multi-target tracking

Giorgio Panin, Thorsten Röder, Alois Knoll

Technische Universität München, Fakultät für Informatik
Boltzmannstrasse 3, 85748 Garching bei München, Germany
Email: {panin,roeder,knoll}@in.tum.de

Abstract

In this paper, a dynamic multi-modal fusion scheme for tracking multiple targets with Monte-Carlo filters is presented, with the goal of achieving robustness by combining complimentary likelihoods based on color and foreground segmentation. The generality of the proposed approach allows defining the measurements on different levels (pixel-, feature- and object-space) through dynamic data fusion. We demonstrate the approach in a people tracking context, by using a multi-target MCMC particle filter.

1 Introduction

Integrating complimentary visual modalities can be crucial for multiple object tracking, because of an improved robustness and adaptivity to variable conditions such as background clutter, lighting and mutual occlusions of the targets.

Several examples of multi-modal fusion for people tracking are already well-known in the computer vision literature [17, 5, 15, 9], with the common requirement of using generic offline models of the person shape and appearance, while building and refining more precise models (with color, edges, background information) during the on-line tracking task.

In a Bayesian framework [16, 2], the multi-target state update is performed through the observation process, which results in a more or less large set of target-associated *measurements*.

Monte-Carlo filters [8, 11] in particular show to be well-suited in presence of non-Gaussian likelihoods, dealing with uncertain data association arising from clutter background and target interactions. In this context, as shown in [11] and [7], simultaneous tracking of multiple targets can also be obtained without exponentially increasing of the number of particles (state hypotheses).

Apart from the huge variety of visual modalities that can be defined for tracking (color, motion, texture, edges etc.), here we also distinguish between three main *processing levels*, defined after the standard data fusion terminology [4]:

- *Pixel-level*: any measurement resulting in a dense or sparse pixel-wise response map (e.g. color segmentation, edge detection, optical flow field, etc.)
- *Feature-level*: detection and matching of target-associated primitives, of the most variable nature (shape blobs, contour points or lines, local keypoints, etc.)
- *Object-level*: a local (or global) maximum-likelihood procedure, directly resulting in a state-space estimate

When multiple modalities are employed for tracking, data fusion can be performed in static or dynamic ways.

Static fusion techniques compute a combined measurement out of N for a single likelihood evaluation, which can be obtained in several ways (weighted average, voting, fuzzy, etc.), also depending on the level and type of measurements involved.

Dynamic fusion, instead, basically amounts to separately compute and multiply all independent likelihoods during the state update. In a Kalman filter setting, the latter corresponds to stack together individual measurements in a global vector, whereas for Monte-Carlo filters the product of all likelihoods is explicitly evaluated.

Dynamic fusion has a number of advantages, since it optimally integrates the object dynamics within the fusion process [1], it is performed in a unique way for every combination of modalities (Sec. 2.3), and it can integrate data of different na-

ture and abstraction level in a uniform way.

In this paper, we present a multi-level fusion methodology using robust multi-hypothesis likelihoods, and we apply it to a people tracking task in outdoor environments, with an MCMC particle filter.

The paper is organized as follows: Section 2 reviews the Bayesian tracking problem, and formulates the multi-level robust likelihoods; Section 3 proposes the multi-target MCMC filter, and Section 4 provides experimental results related to the people tracking task; finally, Section 5 concludes the paper and proposes future developments.

2 Problem formulation

The aim of our system is to follow multiple objects in time, by integrating past information with the current measurements, in order to update the posterior state estimate. In the *Bayesian tracking* framework, knowledge about the system state is represented and propagated in a probabilistic way [16, 2], with two main steps:

1. *Prediction* (Chapman-Kolmogorov equation):

$$P\left(s_t|\, Z^{t-1}\right) = \int_{s_{t-1}} P\left(s_t|\, s_{t-1}\right) P\left(s_{t-1}|\, Z^{t-1}\right) \quad (1)$$

2. *Correction* (Bayes' rule):

$$P\left(s_t|\, Z^{t}\right) = kP\left(z_t|\, s_t\right) P\left(s_t|\, Z^{t-1}\right) \quad (2)$$

where current state statistics s_t integrate the associated measurements z_t together with the set $Z^{t-1} \equiv z_{1\ldots t-1}$ of all past measurements, up to time $t-1$.

In this scheme, a dynamical model $P\left(s_t|\, s_{t-1}\right)$ and a measurement likelihood $P\left(z_t|\, s_t\right)$ need to be specified, and the two steps will be implemented according to the chosen estimation filter.

Concerning the measurement model, one of two equivalent forms for may be provided

- Explicit form: $z_t = h(s_t, e_t)$
- Implicit form (likelihood): $P(z_t|\, s_t)$

where the measurement *residual* e_t is supposed to be a discrete, zero-mean white process with covariance matrix R_t.

The second form is the *likelihood* of the observation z given s, and alone can be used for a maximum-likelihood (ML) parameter estimation; for Gaussian models, this corresponds to an LSE optimization (possibly nonlinear).

When the first form is available, with Gaussian noise, then a Kalman Filter (or EKF, UKF [10] for nonlinear cases) can be used.

However, often in computer vision a *multi-hypothesis* data association is a more realistic and robust model, in presence of clutter background and missing detections; this requires a non-Gaussian density with possibly multiple peaks (*modes*).

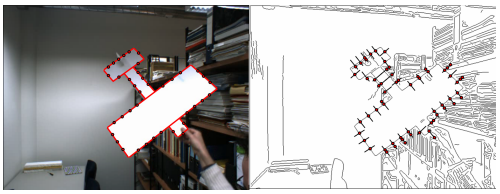

Figure 1: Example of feature-level measurement. Left: expected contour points under a given pose hypothesis; Right: multiple corresponding points, detected on the Canny edge map.

2.1 Feature-level likelihoods

Generally speaking, a feature-level measurement is obtained by defining one or more *measurement probes* on each projected model primitive (contour points, local keypoints, etc.), which are supposed to be statistically independent one another; each one provides a single or multiple data association hypotheses $(h_i, z_{ij}), j = 0, ..., J_i$, where h_i is the expected value and z_{ij} are the observed, associated data (in a variable number J_i, possibly also $J = 0$).

This results in a set of residual values e_{ij} and covariances r_i (with time index omitted for sake of clarity)

$$(e_{ij}, r_i), j = 1, ..., J_i \quad (3)$$

A well-known example is given by contour points (Fig. 1), which provide a multi-hypothesis likelihood [8] where h_i are projected model points, and z_{ij} the corresponding image edges, detected along the normals.

The corresponding likelihood function has typically the form of a product of Gaussian Mixtures,

Figure 2: Example of pixel-level measurement. Left: expected object shadow, Middle: foreground segmentation on the real image; Right: residual image.

plus uniform clutter noise [8]

$$P(z|s) \propto \prod_{i=1}^{N} \left[\alpha + \frac{1-\alpha}{\lambda\sqrt{2\pi} r_i} \sum_{j=1}^{J_i} \exp\left(-\frac{e_{ij}^2}{2r_i^2} \right) \right] \quad (4)$$

with α the *missing detection* rate $P(J=0)$, and λ the *false alarm* density (average rate of clutter-originated measurements).

This model is usually employed in Monte-Carlo filters, that can also deal with non-Gaussian densities. The purely Gaussian likelihood can be obtained as a special case of (4), by imposing $\alpha = 0$ and $J_i = 1$ for all i, typically with a *nearest-neighbor* approach

$$(e_i, r_i) = \left(e_{ij^*}, r_i : j^* = \arg\min_j \|e_{ij}\| \right) \quad (5)$$

2.2 Pixel-level likelihoods

When the measurement is obtained pixel-wise, h and z are, respectively, expected and observed pixel maps of any kind.

We mention here a few examples: foreground segmentation (Fig. 2) provides a binary map that can be matched to the expected object *shadow* (left), that represents an ideal segmentation, without shape errors or clutter; an optical flow field [6] can also be matched with the projected surface flow, at a given pose and velocity. Finally, when a fully textured model is available, the surface can be directly rendered and matched against the underlying image values.

Pixel-level measurements have the advantage of being more informative and generally more precise than feature-level ones; on the other hand, they involve an expensive and highly nonlinear computation that, however, can be nowadays performed by exploiting the parallelism of graphics hardware, via the OpenGL shader language [14].

In the case of binary images, we can consider each pixel x as a measurement probe, with only two possible values $z_x = \{0, 1\}$. Therefore, we can define the following situations

- False alarm: $h_x = 0, z_x \neq 0$, occurring with probability λ
- Missing detection: $h_x \neq 0, z_x = 0$, with probability α
- Correct match: $h_x = z_x$, with probability $(1 - \alpha - \lambda)$

By taking into account some numerical issues, we propose a scale-normalized likelihood

$$P(z|s) = \frac{1}{K} \cdot \quad (6)$$
$$\cdot \alpha^{\frac{|MD|}{N_{obj}}} \lambda^{\frac{|FA|}{N_{obj}}} (1-\alpha-\lambda)^{\frac{N-|MD|-|FA|}{N_{obj}}}$$

In this formula N is the image size, $MD(s)$ is the subset of undetected pixels and $FA(s)$ the number of false alarms, observed under the state hypothesis s. The term N_{obj} is the average object size (area of the predicted shadow $\hat{h}$), which provides scale normalization in conjunction with the coefficient K

$$K = \lambda^{\frac{FA_{max}-N_{obj}}{N_{obj}}} (1-\alpha-\lambda)^{\frac{N-FA_{max}+N_{obj}}{N_{obj}}} \quad (7)$$

where the maximum number of false alarms FA_{max} is given by the overall number of foreground pixels (worst case).

Basically, the scale-normalization coefficient $1/N_{obj}$ acts by modifying the covariance of the three error terms with respect to the translational displacement: in fact, an object with half the size would normally exhibit a half variation in both the MD and FA errors, resulting in a likelihood with smaller variance.

Instead, the K coefficient ensures to maintain a roughly unitary maximum likelihood value, otherwise decreasing very fast with the object size. Both K and N_{obj} must be updated from frame to frame.

Furthermore, in the case of pixel-level measurements, the two parameters λ and α can be updated during time as well, providing more robustness and adaptivity to challenging conditions (e.g. a variable clutter density, or partial occlusions).

For this purpose, we propose a simple adaptation scheme: under the current pose estimate, the missing detection α and false alarm λ rates can be estimated under the two regions defined by the object

shadow (foreground and background)

$$\begin{aligned} \alpha &= \frac{\#\text{missing pixels under the shadow}}{\text{shadow area}} \\ \lambda &= \frac{\#\text{detected pixels in the background}}{\text{shadow area}} \end{aligned} \quad (8)$$

While modeling the likelihood as in (6), a further issue may arise because of the spatial closeness of neighboring pixels, which somehow invalidates the mutual independence assumption. However, to our experience neglecting this effect does not degrade significantly performances of tracking, while keeping the likelihood function well shaped, and peaked around the correct pose.

2.3 Multi-target and multi-modal fusion

In a dynamic fusion scheme, individual measurements z_m from different modalities are not directly combined, but rather the *tracks* are fused together, by providing individual likelihoods to the Bayesian filter for the state update (2).

Given M visual modalities, possibly defined at different levels, the integrated likelihood is given by

$$P(z|s) = \prod_m P(z_m|s) \quad (9)$$

This scheme assumes a basic independency between modalities, which should be intuitively achieved when using complimentary features (such as color, edges, motion, etc.). Although in some situations this assumption may be less realistic, the resulting likelihood (9) is again correctly behaving around the target pose.

When multiple, simultaneous targets are concerned, a state hypothesis for the system is made up of I target states, $s \equiv (s_1, ..., s_I)$.

By assuming also independence between targets, both at the dynamic and measurement level, the overall system likelihood becomes

$$P(z|s) = \prod_{m=1}^{M} \prod_{i=1}^{I} P(z_m|s_i) \quad (10)$$

In order to adress and even to avoid two tracks from being locked onto the same target [11], the likelihood function can be improved to include penalty terms $\psi \leq 1$ for overlapping target states (s_i, s_j), decreasing with the common areas of the respective bounding boxes $B(s_i), B(s_j)$

$$\psi(s_i, s_j) \equiv \exp\left(-\beta \frac{|B(s_i) \cap B(s_j)|}{\min(|B(s_i)|, |B(s_j)|)}\right) \quad (11)$$

where the intersection area is normalized to $[0, 1]$, and $\beta \geq 0$ is a coefficient which regulates its behavior ($\beta = 0$ gives no penalty term).

This is computed for all target pairs with overlapping regions under the hypothesis s, giving

$$\Psi(s) = \prod_{i,j>i} \psi(s_i, s_j) \quad (12)$$

that corresponds to a Markov Random Field (MRF) in the global state-space.

Finally, the likelihood for the multi-target hypothesis becomes

$$P(z|s) = \Psi(s) \prod_{m,i} P(z_m|s_i) \quad (13)$$

3 Multi-target tracking with the MCMC particle filter

In order to apply the Bayesian tracking equations to a multi-target and multi-modal problem, we employ a MCMC particle filter, which, compared to other traditional Monte-Carlo methods, also allows an efficient modeling of multi-target interactions (penalty term) [11]. Although with this filter a variable number of targets can also be handled, for sake of simplicity here we limit ourselves to a fixed and known number of hypotheses I, detected at the beginning of the sequence.

In a Monte-Carlo context, state statistics s_t are represented by a set of N weighted particles

$$\{s_t^n, \pi_t^n\};\ n = 1 \dots N \quad (14)$$

where $\sum_n \pi_t^n = 1$. For MCMC filters, in particular, the sampling procedure provides no weights $\pi^n = 1/N$ and the equivalent statistics for the posterior density are given by possibly duplicated particles, so that

$$P(s|Z^t) \approx \{s_t^n, 1/N\};\ n = 1 \dots N \quad (15)$$

Following [11] (Sec. 4), the prior distribution can be defined as

$$P\left(s_t|\, Z^{t-1}\right) \approx \frac{1}{N}\sum_n \left[\prod_i P\left(s_{i,t}|\, s_{i,t-1}^n\right)\right] \tag{16}$$

which is a discrete approximation of the integral in (1); however, the last quantity can be computationally very expensive, and therefore we consider a further approximation more suitable for real-time tasks

$$\begin{aligned} P\left(s_t|\, Z^{t-1}\right) &\approx \prod_i P\left(s_{i,t}|\, \hat{s}_{i,t-1}\right) \qquad (17)\\ \hat{s}_{i,t-1} &= \frac{1}{N}\sum_n s_{i,t-1}^n;\ i=1,...,M \end{aligned}$$

where $\hat{s}_{t-1}$ are the average states of the last frame. Therefore, the posterior distribution is given by

$$P\left(s_t|\, Z^t\right) \propto P\left(z_t|\, s_t\right)\prod_i P\left(s_{i,t}|\, \hat{s}_{i,t-1}\right) \tag{18}$$

The MCMC filter generates each time a new particle set distributed according to (18), as a Markov chain. The chain is generated from an arbitrary initial state (*seed*) by using the Metropolis-Hastings algorithm:

Algorithm 1. Metropolis-Hastings sampling

Starting from an initial state s^0, for each particle n do

1. Propose a new state s_t' from the previous one s_t^{n-1} by sampling from a proposal density $Q\left(s_t'|\, s_t^{n-1}\right)$
2. Compute the acceptance ratio

$$a = \frac{P\left(s_t'|\, Z^t\right)}{P\left(s_t^{n-1}\middle|\, Z^t\right)}\frac{Q\left(s_t'|\, s_t^{n-1}\right)}{Q\left(s_t^{n-1}\middle|\, s_t'\right)} \tag{19}$$

3. If $a \geq 1$ accept the proposed state $s_t^n \leftarrow s_t'$. Otherwise, accept it with probability a; in case of rejection, the old state is kept $s_t^n \leftarrow s_t^{n-1}$

The proposal distribution Q can be to some extent arbitrary, and we choose to use the dynamical model itself $P\left(s_t|\, s_{t-1}\right)$, which is symmetric and therefore the second ratio in (19) cancels out.

In the MCMC formulation, a big computational saving is obtained by updating a single target i at a time (randomly chosen), which results in a proposal ratio only for this target $P\left(s_{i,t}|\, s_{i,t-1}\right)$.

Under the assumption of independent measurements for each target, also the two likelihoods $P\left(z_t|\, s_t'\right)$ and $P\left(z_t|\, s_t^{n-1}\right)$ differ only for a single target, as well as the penalty terms (12).

By substituting the posterior model (18) in (19), with likelihood given by (13), we finally get

$$\begin{aligned} a &= \prod_j \frac{\psi\left(s_{i,t}', s_{j,t}'\right)}{\psi\left(s_{i,t}^{n-1}, s_{j,t}^{n-1}\right)}\cdot\prod_m \frac{P\left(z_t^m|\, s_{i,t}'\right)}{P\left(z_t^m|\, s_{i,t}^{n-1}\right)}\\ &\quad\cdot\frac{P\left(s_{i,t}'\middle|\, \hat{s}_{i,t-1}\right)}{P\left(s_{i,t}^{n-1}\middle|\, \hat{s}_{i,t-1}\right)} \end{aligned} \tag{20}$$

In this formula, the first term is the *penalty ratio*, which favors state motions towards less overlapping areas; the second term is the *likelihood ratio*, which privilege motions that increase the multimodal likelihood; and the last term is the *prior ratio*: it accepts any motion which is dynamically consistent with the previous target state $\hat{s}_{i,t-1}$.

Therefore, as expected from the Bayesian tracking framework, new particles will be generated for states which are consistent with both (measurement and dynamics) models, as well as with the MRF model for interacting targets.

Concerning the startup phase of the MCMC estimation, also the initial state can be chosen arbitrarily, provided it is not too far from the real targets, and that the first particles, generated before the chain reaches its stationary distribution, are discarded (*burn-in* sample).

For this purpose, we choose the old average states $\hat{s}_{i,t-1}$ as the seed, and discard the first 20% particles from the final sample; the latter procedure introduces some computational overhead to the overall algorithm, that however is well compensated by the fact, that an MCMC filter requires much less particles with respect to a standard particle filter, especially for multi-target problems [12, 9].

4 Experimental results

We tested our methodology on a people tracking task using sequences from the CAVIAR project (`http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`). For this task, we model people shapes by generic rectangles and dynamics by a simple Brownian model

$$s_t = s_{t-1} + w_t \tag{21}$$

For tracking, we employ two complementary visual modalities: color statistics (feature-level)

and foreground segmentation (pixel-level). In the described scenario the color statistics modality is responsible for maintaining the target identity, whereas the foreground segmentation modality is used to increase the tracking robustness and accuracy in cases of simple motion dynamics.

Color histogram matching is performed with the Bhattacharyya coefficient [13, 3] as residual between 2D color histograms in Hue-Saturation space

$$e_{col}\left(f\left(s\right), f^{*}\right) = \left[1 - \sum_{b} \sqrt{f_b^{*} f_b\left(s\right)}\right]^{\frac{1}{2}} \quad (22)$$

where f^{*}, f are expected and observed color histograms over the underlying image region for each target, and the sum is performed over $(B \times B)$ bins (with $B = 20$ in our implementation).

The color modality provides a single feature-per-target (an expected-observed histogram pair), with standard deviation r_{col} and a single association hypothesis ($\alpha = 0$), so that its likelihood (4) reduces to a Gaussian

$$P_{col}\left(z|\, s\right) \propto \exp\left(-\frac{e_{col}\left(f\left(s\right), f^{*}\right)^{2}}{2r_{col}^{2}}\right) \quad (23)$$

In the color modality, adaptation to changing light or viewpoint conditions is provided by combining the off-line reference histogram f^{*} with an on-line version f_{on}^{*}, updated every time from the image data at the estimated pose. The off-line reference histogram is gathered once at initial target detection time (track creation) and remains fixed during the whole tracking process, whereas the on-line histogram is used to adapt to varying illumination conditions. On-line and off-line histograms are sampled in HSV color space in order to further increase the robustness in varying illumination conditions. The on-line histogram provides an additional likelihood term $P_{col,on}\left(z|\, s\right)$ for the dynamic fusion context.

For the described people tracking task, we restrict ourselves to hypotheses with a fixed size of 16 by 62 pixels. Using more degrees of freedom (e.g. 1D or 2D scale) typically involves also an increasing number of particles for the posterior approximation. In order to increase the overall robustness of the color modality, we vertically divide the hypothesis in three subregions, and collect the color statistics for each of the subregions separately. By doing this, we inherently keep track of the subregions spatial layout and the related histograms.

Finally, the pixel-level modality is given by foreground segmentation (Fig. 2), which provides a binary map to be matched with the predicted object shadows; missing detection α and false alarm rate λ are estimated and updated on-line, as in (8).

Initialization for this system is obtained by detecting foreground blobs, in a generic and target-independent way.

During tracking, dynamic fusion is performed through the MCMC filter and the acceptance ratio (20) for the Metropolis-Hastings algorithm.

Results are shown in Fig. 6, where we can see the individual modalities and the pose estimation result, also in presence of mutual occlusions.

For ground truth comparison, we use a subsequence of the CAVIAR *OneShopOneWait1front* sequence (frames 440 to 562) in order to show the tracking performance for three different combinations: foreground segmentation alone, color statistics alone, and dynamic fusion. This subsequence was selected in order to have a fixed number of targets, while exhibiting multiple occlusions.

For the three cases, Fig. 7 shows selected frames of the tracking behaviour, and Fig. 3, 4 and 5 show the observed error distances with respect to the ground truth.

Since the degrees of freedom are limited to 2D translations for this scenario (without addressing scale and orientation for sake of simplicity), we use the centers of the bounding boxes for computing the error distances, and limit the number of particles for estimating the posterior state $\hat{s}_t$ to 400.

A short description of the results follows:

- *Initialization*: Initial detection of all three targets without occlusions (Frame 445), and initial gathering of the reference color statistics f_{*} are performed.
- *First occlusion*: Shortly after the first partial occlusion of targets 1 and 2 (Frame 477), the foreground segmentation modality is not able to maintain target identities, thus the likelihoods of targets 1 and 2 lock onto the same target. Because of the penalty terms, one of the two targets gets lost, as can be seen in Fig. 3.

  However, by using the color modality or the dynamic fusion, identities are successfully maintained. Looking at the error plots for the fusion case (Fig. 5), the first peak of target 2 is caused by the impact of mutual occlusion and

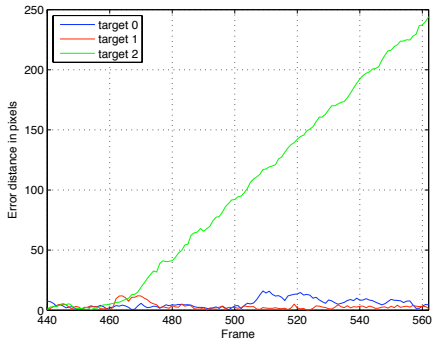

Figure 3: Error distance using foreground segmentation only.

the consequent penalty term.

- *Second occlusion*: The second partial occlusion of target 0 and 2 (from frame 503 to 521) results in a local error peak for target 0 shown in Fig. 5. Another peak (around frame 540) occurs because of the lack of scaling freedom in our estimate, so that the target is becoming smaller and the shape matching more imprecise.
  This occlusion is successfully solved (Frame 554), both for the color statistics modality and for the dynamic fusion.

After frame 557, it can be seen that the tracking system starts loosing target 2, which is going to walk in front of a white background. In this case neither the target itself, nor the background wall show a relevant difference in color statistics, with respect to the main torso region and the chosen bin size of the color histogram; this leads to a non-distinguishable situation for both modalities.

Finally, table 1 provides numerical evaluations with rms values for the three cases. As we can see, although color statistics on their own can give quite good results, we can see from table 1, that combining both complementary modalities by means of dynamic fusion can increase the tracking performance with respect to precision and jittering effects, leading to smoother trajectories, even in presence of coarse models and simple dynamics.

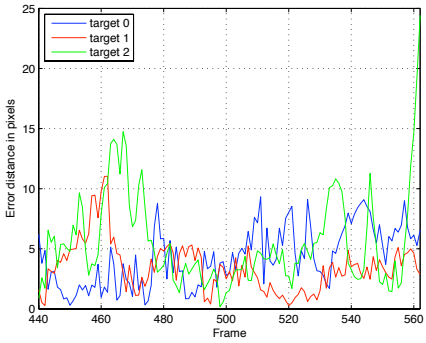

Figure 4: Error distance using color statistics only.

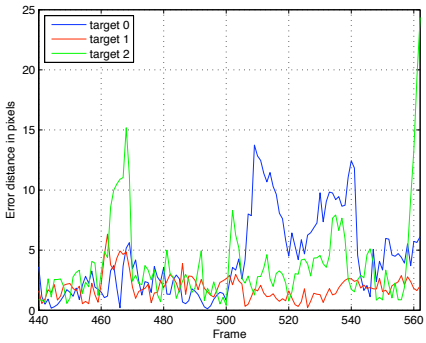

Figure 5: Error distance using dynamic fusion of both, foreground segmentation and color statistics.

	Modalities	Target 0	Target 1	Target 2(*)	Target 2
Median	Fg. Seg.	4.53	2.54	4.22	94.39
	Color	4.14	3.41	4.23	4.49
	Fusion	**3.46**	**1.81**	**2.76**	**2.81**
Variance	Fg. Seg.	14.47	6.76	14.69	6221.88
	Color	**5.98**	4.39	10.44	15.68
	Fusion	11.61	**1.04**	**6.62**	**12.99**

Table 1: Median and variance of the observed error distances with respect to each target and modality combination. Target 2(*): In addition, to allow a more feasible comparison, errors of target 2 are also considered only until it was lost (=minus the last 4 frames) or the estimated pose locked onto the wrong target. Bold values: minimum values per target for each of the 3 configurations.

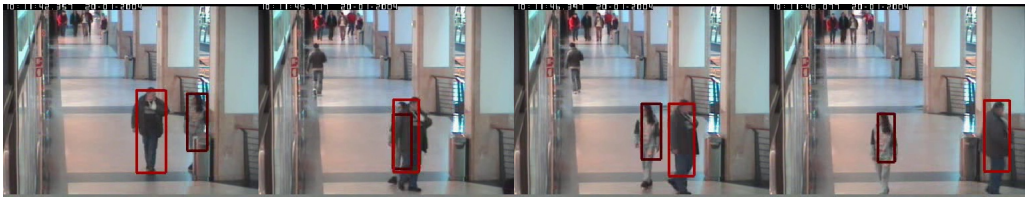

Figure 6: Tracking results on a sequence from the CAVIAR database, showing the successful handling of a temporary occlusion.

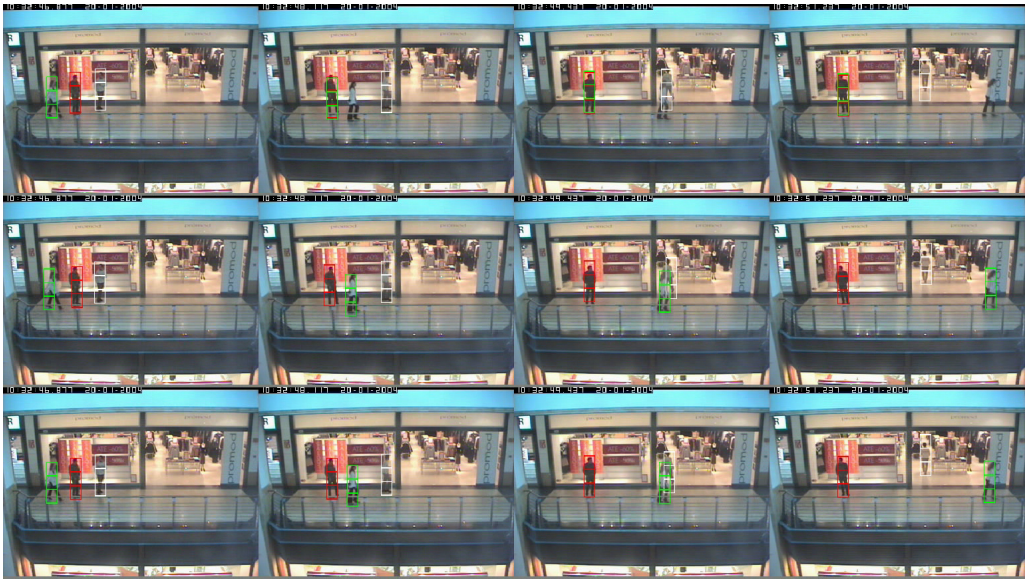

Figure 7: First row: matching done by using only foreground segmentation, second row: matching done by using only color statistics, third row: matching done by using cross-level dynamic data fusion of both modalities. Columns from left to right: frame no. 445, no. 477, no. 509, no. 554 of the CAVIAR sequence (OneShopOneWait1front) used for error computation. (Figures 3, 4, 5). Encoding: target 0 (white), target 1 (red), target 2 (green)

5 Conclusions

We developed an integrated, multi-modal and multi-level data fusion framework for object tracking, in the context of Monte-Carlo filters for multiple interacting targets. Furthermore, we proposed a simple scale-invariant likelihood model for binary pixel-level residuals. The benefits of complimentary modalities at different processing levels, with on-line adaptive models, have been shown through experimental results concerning people tracking in outdoor environments.

Future developments for this system include exploiting the computational power of GPUs for pixel- and feature-level likelihood evaluation, possibly in conjunction with GPU-assisted particle filters.

References

[1] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[2] Samuel S. Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library, 1999.

[3] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.

[4] David L. Hall and James Llinas. *Handbook of Multisensor Data Fusion*. CRC Press, June 2001.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: A real time system for detecting and tracking people. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 962, Washington, DC, USA, 1998. IEEE Computer Society.

[6] Berthold K. P. Horn and Brian G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] C. Hue, J.-P. Le Cadre, and P. Prez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Processing*, 50(2):309–325, February 2002.

[8] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998.

[9] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *ICCV*, pages 34–41, 2001.

[10] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[11] Zia Khan. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(11):1805–1918, 2005. Member-Tucker Balch and Member-Frank Dellaert.

[12] Zia Khan, Tucker Balch, and Frank Dellaert. Efficient particle filter-based tracking of multiple interacting targets using an MRF-based motion model. Las Vegas, 2003.

[13] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK, 2002. Springer-Verlag.

[14] Randi J. Rost. *OpenGL(R) Shading Language*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[15] Nils T. Siebel and Stephen J. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 373–387, London, UK, 2002. Springer-Verlag.

[16] L. D. Stone, T. L. Corwin, and C. A. Barlow. *Bayesian Multiple Target Tracking*. 1st. Artech House, Inc., 1999.

[17] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.