Combining Colour and Orientation for Adaptive Particle Filter–based Tracking

Emilio Maggio[1], Fabrizio Smeraldi[2], Andrea Cavallaro[1]

[1]Dept. of Electronic Engineering - Multimedia and Vision Laboratory
[2]Dept. of Computer Science - Vision Group
Queen Mary, University of London – Mile End Road, London, E1 4NS
Email: fabrizio.smeraldi@dcs.qmul.ac.uk,
{emilio.maggio,andrea.cavallaro}@elec.qmul.ac.uk

1 Abstract

We propose an accurate tracking algorithm based on a multi-feature statistical model. The model combines in a single particle filter colour and gradient-based orientation information. A reliability measure derived from the particle distribution is used to adaptively weigh the contribution of the two features. Furthermore, information from the tracker is used to set the dimension of the filters for the computation of the gradient, effectively solving the scale selection problem. Experiments over a set of real-world sequences show that the adaptive use of colour and orientation information improves over either feature taking separately, both in terms of tracking accuracy and of reduction of lost tracks. Also, the automatic scale selection for the derivative filters results in increased robustness.

2 Introduction

Colour histograms are widely used for target representation because of their invariance to scaling and rotation and robustness to partial occlusions [5, 10]. Moreover colour histograms allow for significant data reduction, and can be computed efficiently. Nevertheless their descriptiveness is limited by the lack of spatial information, which makes difficult discriminating between targets with similar colour properties. To complement colour information, gradient information has been recently used. Starting from the work of Nishihara [9], Birchfield in [4] proposed a face tracker based on colour histograms and on the projection of the gradient onto the perpendicular to the target perimeter. Similarly in [7], edge density is calculated near the same perimeter using a binary Laplacian map. These representations use the edge information near the target border discarding that of the interior; our proposal is to code this part of information as well. Starting from the work of Freeman in [6] for hand gesture recognition, we propose to create a model using the histogram of the gradient orientation. The intrinsic problem of the derivative operators used in [6] to estimate the gradient, is the noise amplification. In order to enhance robustness, we substitute the point-wise estimate of the gradient with a least squares estimate based on the structure tensor, which is known to be more resilient to noise [3]. The orientation and colour histograms are then combined together in a Particle Filter approach [1].

In Particle Filter, the likelihood of the tracked target, having the position, orientation and size (i.e. the state) specified by each particle, is calculated separately based on the orientation and colour information. However, the decision about the target state should be performed by adaptively combining the two features, hence estimating their reliability. Our proposal is to extend to Particle Filter the measure of uncertainty based on the covariance matrices of the target state [8]. The peculiarity of this approach is that the uncertainty is not based on the likelihood itself but on the variability of the likelihood in the state space. For each feature, a high variance corresponds to a high uncertainty in the localization of the target, which indicates that the system should not rely on it. This method has been used recently in [5, 8] to estimate the uncertainty in a Kalman filter tracker. The extension to particle filter is not straightforward and is part of our contribution.

The paper is organized as follows. Sec. 3 introduces the target representation. Particle filter and the adaptive integration of the two representations are described in Sec. 4. Experimental results are presented in Sec. 5, followed by conclusions in Sec. 6.

3 Target representation

The target area is approximated by an ellipse centred in $\mathbf{y} = (x, y)$, with length of the minor axis h, eccentricity e, and rotation θ. These parameters specify the state of the target $\mathbf{x}_t = [\mathbf{y}, h, e, \theta]$ at time t.

3.1 Colour Histograms

Colour information is represented by a normalised colour histogram $p(\mathbf{x}) = \{p_u(\mathbf{x})\}_{u=1,\ldots m}$, where m is the number of bins. More specifically $p(\mathbf{x})$ can be obtained as

$$p_u(\mathbf{x}) = B \sum_i K_{e,\theta} \left(\left\| \frac{\mathbf{y} - \mathbf{w}_i}{h} \right\|^2 \right) \delta \left[b(\mathbf{w}_i) - u \right], \tag{1}$$

where the $\mathbf{w}_i$ are the pixels of the target and $b(\mathbf{w}_i)$ associates each $\mathbf{w}_i$ to its histogram bin [5]. The elliptic kernel $K_{e,\theta}(.)$ is used to lower the weight of the pixels that are closer to the border of the target. The normalization factor B ensures that the sum of the bins is one.

Given a reference histogram q defining the target model, and the histogram $p(\mathbf{x})$ extracted from a candidate state $\mathbf{x}$, we measure their similarity according to the metric

$$d\left[p(\mathbf{x}), q\right] = \sqrt{1 - \sum_{u=1}^{m} \sqrt{p_u(\mathbf{x}) \cdot q_u}}, \tag{2}$$

which is based on the Bhattacharyya coefficient [5].

3.2 Orientation Histograms and the Structure Tensor

The orientation histogram is created by estimating the gradient over the frame. Then for each pixel in the region of interest the magnitude of the gradient is cumulated on the bin selected by the orientation. However, the intrinsic problem of the gradient operator is

noise amplification, since derivation enhances the high frequencies of an image. Furthermore, the orientation histograms thus defined are not invariant to image rotations.

To reduce the noise Bigun et al. in [3] proposed to evaluate the gradient using a least square estimate obtained from the structure tensor

$$J(\mathbf{w}) = \int \omega(\mathbf{w}-\mathbf{w}') \left(\nabla I(\mathbf{w}')^t \nabla I(\mathbf{w}')\right) d\mathbf{w}', \tag{3}$$

where ω averages the estimate around the $\mathbf{w}$. The eigenvector $\mathbf{k}_{max}$ of J associated to the largest eigenvalue λ_{max} is the best local fit to the direction of the gradient. The two eigenvalues λ_{max} and λ_{min} carry information about the associated local neighbourhood. The value of λ_{min} is zero in the presence of a clear edge, while $\lambda_{max} \approx \lambda_{min}$ if the grey values change in all the directions. Starting from these considerations a measure of edge certainty has been defined in [2] as

$$C = (\lambda_{max} - \lambda_{min})(\lambda_{max} + \lambda_{min}). \tag{4}$$

C highlights neighbourhoods corresponding to strong straight edges, and penalizes neighbourhoods with $\lambda_{min} \neq 0$. Experimental results in [3] show that the eigenvectors and the certainty measure are accurate and reliable in the presence of additive noise.

The structure tensor based orientation histogram is obtained by dividing the range $[-\pi/2, \pi/2]$ into the desired number of bins. For each position, the value of C is cumulated in the bin corresponding to the orientation ϕ defined by $\mathbf{k}_{max}$ as

$$\phi = \arctan(k_{max}(y)/k_{max}(x)). \tag{5}$$

Note that vectors with opposite directions are cumulated on the same bin to force invariance with target moving through regions with different background. A triangular kernel is used to smooth the estimated histogram. To increase robustness, a threshold T is applied to the certainty measure C. For each frame, T is set to include in the histogram only strong edges according to the following procedure. The probability distribution $p(C)$ is approximated by the histogram of C, computed over the target in the previous frame. Then the cumulative probability $P(C)$ is derived . Finally using $P(C)$ the threshold T is set to retain only a fixed percentage of pixels.

3.2.1 Resilience to target Rotations

Resilience to rotations has been addressed in [6] by blurring the histogram with a kernel; the main problem is that the invariance is still bounded by the kernel width. Furthermore a large kernel results in an excessive loss of information. To avoid these drawbacks we propose two different models. One is more general and can be used in any application of the orientation histogram where complete invariance is required. The other instead can be applied only when an estimate of the target orientation is part of the state $\mathbf{x}$.

The first solution achieves complete invariance by computing the Fourier Transform of the orientation histogram (Fourier Orientation Histograms: FOH). The representation is made invariant to rotations by discarding the phase coefficients. Now the remaining spectrum coefficients contain an orientation-invariant description of the target shape (i.e. rectangular, circular) and internal edges.

Our second solution consists in using the estimate $\hat{\theta}$ of the target orientation provided by its state vector (as sampled by the Particle Filter algorithm, see Sec. 4.1). In our

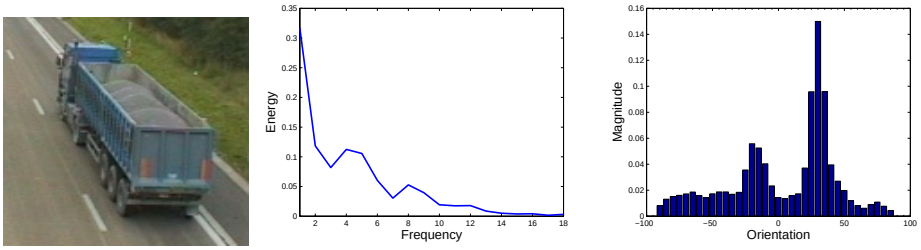


Figure 1: Orientation histogram representations of a truck. Centre: FOH. Right: ROH.

tracker the orientation is modelled by the rotation of the ellipse bounding the target area; the idea is to shift the coefficients of the histogram according to the ellipse rotation angle $\hat{\theta}$ (Rotated Orientation Histograms: ROH). Note that the alignment is *not* based on the dominant orientation in the histograms themselves.

Fig. 1 shows the two histograms (FOH and ROH) representing a truck.

3.2.2 The scale selection problem

The orientation histograms defined above share the scale invariance properties of normalized histograms. However, a problem arises under large scale changes in connection with the computation of the structure tensor. The derivative filters used in the computation of the gradient and the smoothing kernel ω of the structure tensor (see Eq. (3)) both have a scale parameter that determines the effective level of detail. For the representation to be truly invariant, this should be adapted to the varying dimensions of the target. We propose to adapt the scale parameter of the filters by making it proportional to the area of the target, as estimated by the bounding ellipse.

4 Multi-Feature Adaptive Particle Filter

In this section we present a generic solution to adaptively combine different target representations in a single Particle Filter framework. This is done through an extension of the covariance–based uncertainty measure described in [8]. After giving an overview of the Particle Filter (PF) algorithm, we introduce our uncertainty measure in Sec. 4.2.

4.1 Particle Filter Overview

Particle Filter [1] solves the tracking problem by finding the sequence of states $\mathbf{x}_t$ defined in Sec. 3, based on the previous observations $\mathbf{z}_{1:t}$ (in our case, the observation vector $\mathbf{z}_{1:t}$ represent the image pixels observed up to time t). In a Bayesian approach, the problem consists in calculating the conditional density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$.

The main characteristic of a Particle Filter is that the posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of the status of the target is approximated with a sum of N_s Dirac functions (the "particles") centred in $\{\mathbf{x}_k^i\}_{i=1,\dots N_s}$, where ω_t^i are the weights associated to the particles. Given a state transition model that defines the probability of finding the target in state $\mathbf{x}_t$ at time t given that it was in state $\mathbf{x}_{t-1}$ at time $t-1$ (i.e. $p(\mathbf{x}_t|\mathbf{x}_{t-1})$), this can be used as proposal

Figure 2: Example of wrong track using colour histograms.

distribution to propagate the particles toward new states. If a re-sampling algorithm is also applied to discard the particles with lower weights [1], this leads to $\omega_t^i \propto p(\mathbf{z}_t|\mathbf{x}_t^i)$ (that is, the weights are proportional to the likelihood of the observation vector). The likelihood is then calculated using the distance defined in Eq. (2) from the model histogram, as in

$$p(\mathbf{z}_t|\mathbf{x}) = e^{-\left(\frac{d[p(\mathbf{x}),q]}{\sigma}\right)^2}. \tag{6}$$

where the histogram $p(\mathbf{x})$ defined by the status $\mathbf{x}$ is calculated over the pixels of the observation vector (the image) $\mathbf{z}_t$. The best state at the time t is derived based on the discrete approximation created by the weighted particles. The most common solution is the Monte Carlo approximation of the expectation $\mathbb{E}(\mathbf{x}_t|\mathbf{z}_{1:t})$ calculated as the weighted average of the particles $\mathbf{x}_t^i$.

This is the PF solution to the tracking problem in case of a single-feature representation. In the multiple-feature approach the likelihood $p(\mathbf{z}_t|\mathbf{x})$ should be dependent on the distance from the model calculated for each feature.

4.2 Uncertainty in Particle Filter

Similarity measures like the one defined in Eq. 2 can lead to unreliable matches, particularly in regions with similar information. For example Fig. (2) (b) shows a wrong result returned by the tracker based on colour histograms. In this case the tracker is uncertain about the position and dimension of the target. Measuring the spatial uncertainty is a possible solution to set the influence of different models in the overall tracking process [8].

Suppose that each feature (histogram type) has a comparable measure to determine the likelihood (i.e. Eq. (6)). Hence a likelihood vector

$$\mathbf{l}_t(\mathbf{x}_t^i) = \left[p^j(\mathbf{z}_t|\mathbf{x}_t^i)\right]_{j=1\ldots N_f}, \tag{7}$$

with dimension N_f equal to the number of features, is associated to each particle. For each feature j at the frame t we calculate the covariance matrix C_t^j of the particles $\mathbf{x}_t^i$ weighted by the likelihood. If the state has only two dimensions $\mathbf{x} = (u, v)$, the normalised covariance matrix is calculated as

$$C_t^j = \begin{bmatrix} \frac{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)(u_t^i - E[u_t])^2}{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)} & \frac{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)(u_t^i - E[u_t])(v_t^i - E[v_t])}{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)} \\ \frac{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)(u_t^i - E[u_t])(v_t^i - E[v_t])}{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)} & \frac{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)(v_t^i - E[v_t])^2}{\sum_{i=1}^{N_s} l_t^j(u_t^i,v_t^i)} \end{bmatrix}. \tag{8}$$

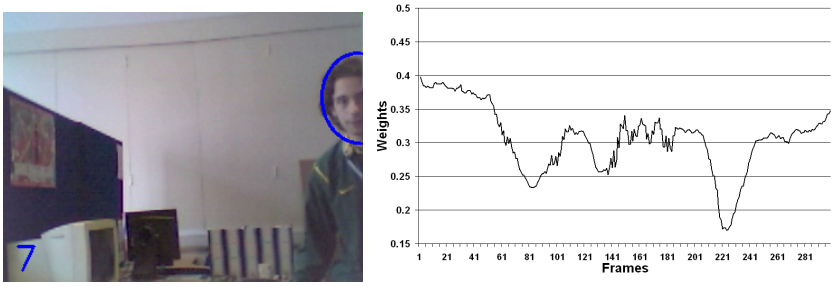


Figure 3: Sequence *Emilio*. Colour and orientation histograms are combined adaptively. Left: frame 226. Right: plot of the weights associated to the orientation histogram.

The feature uncertainty is estimated by analysing the eigenvalues of C_t^j. We define the uncertainty U^j as

$$U^j = \prod_{k=0}^{D} \lambda_k^j = \det(C_t^j). \tag{9}$$

This measure is proportional to the volume of the hyper-ellipse having the eigenvalues as semi-axes. If the volume is large, then the selected feature does not return a precise information about the state of the target. The final step is to derive $p(\mathbf{z}_t|\mathbf{x}_t^i)$ using the uncertainty coefficients as weights:

$$p(\mathbf{z}_t|\mathbf{x}_t^i) \propto \frac{\sum_j \frac{l_t^j(\mathbf{x}_t^i)}{U^j}}{\sum_j \frac{1}{U^j}}, \tag{10}$$

the contribution of each feature is inversely proportional to its uncertainty.

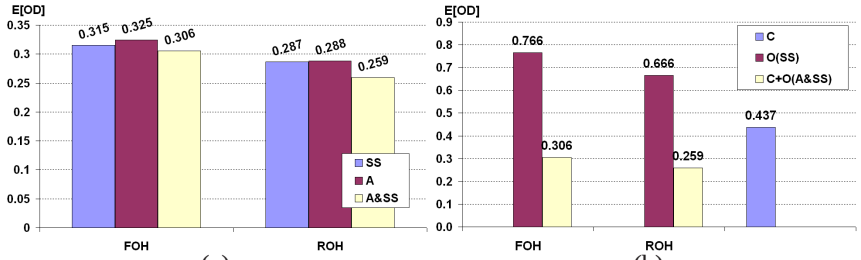
A plot of the weights over time is drawn in Fig. 3 for the sequence *Emilio*. A low weight for the orientation histogram is associated to the frame showed (N. 226). Since the target is partially occluded, the gradient representation is not complete and this results in a large uncertainty for that feature. Moreover between frame 140 and frame 200 the target performs several abrupt left-right shifts, in this case the weight is lower when the target is affected by camera blur.

5 Experimental Results

The results presented in this section are obtained on a dataset of targets extracted from 6 different test sequences (Fig. 4). They are divided into three classes: (i) PEOPLE: two pedestrians from the PETS2001 DATA-SET1 (P1: man with backpack, P2: man with grey pull), and a person walking (P3) from the sequence *Gabin*. (ii) FACES: two from high quality sequences *Toni* (F2) and *Nikola* (F3), and one from low quality sequence *Emilio* (F1). (iii) VEHICLE: a truck (V4) is selected from the CIF sequence *Highway*.

The parameters of the tracker are described in the following. The colour histograms are calculated in RGB space with 10x10x10 bins, while the orientation histograms are calculated using 36 bins. PF uses a zero-order motion model (i.e. $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{n}$) where $\mathbf{n}$ is a multivariate Gaussian random variable, with $\sigma_x = \sigma_y = 5.5$ for all the targets except

Figure 4: Target initialization. Three pedestrians (P1,P2,P3), a truck from the MPEG-7 test sequence (V4), and faces on cluttered background (F1,F2,F3).



(a) (b)

Figure 5: Tracking accuracy results (average distance from the ground truth). Comparison of FOH versus ROH. (a) Colour and orientation adapting the weights (A), selecting the filter scale (SS), and both (A&SS). (b) Orientation (O(SS)), Colour (C), and colour + orientation (C+O(A&SS)).

than F1 and F2 where $\sigma_x = \sigma_y = 10$; $\sigma_h = 0.05$, $\sigma_e = 0.021$, and $\sigma_\theta = 10^o$. PF uses 150 samples per frame. The filters for the computation of the gradient are implemented with first derivatives of Gaussian. The evaluation of the automatic Scale Selection (SS) is done comparing our solution with the performance obtained by fixing the scale parameter at the value it has in the first frame. The multi-feature adaptive Particle Filter (A) is compared with a non-adaptive one that fixes the importance of colour and orientation to 66% and 34% respectively. These are the values that return the best average result on the dataset.

The performance evaluation is based on a metric using true positive pixels $TP(t)$ in each frame t. The number of true positives is the number of pixels belonging both to the ground truth ellipse, as well as to the tracker output. The metric is defined as

$$OD(t) = 1 - \frac{2 \times TP(t)}{Card(A_c(t)) + Card(A_{gt}(t))}. \tag{11}$$

where $A_{gt}(.)$ and $A_c(.)$ are the ground truth and the candidate area, respectively. This normalized metric rewards candidates with a high percentage of true positive pixels, and with few false positives and false negatives. Using Eq. (11) a Lost Track (LT) is declared at the frame t when $OD(t) > 0.8$.

The charts of Fig. 5 show the comparison between the two orientation histogram models (Fourier Orientation Histogram (FOH) and Rotated Orientation Histograms (ROH), see Sec. 3.2.1). These results suggest that on average ROH outperforms FOH. This could be expected, since the ROH representation is more descriptive than FOH. However we have to stress that ROH is a possible solution only when an estimate of the target orientation is available (i.e. the rotation of the ellipse). In the absence of this, FOH remains a good solution to improve the performance of the classic colour tracker (see Fig. 5 (b)).

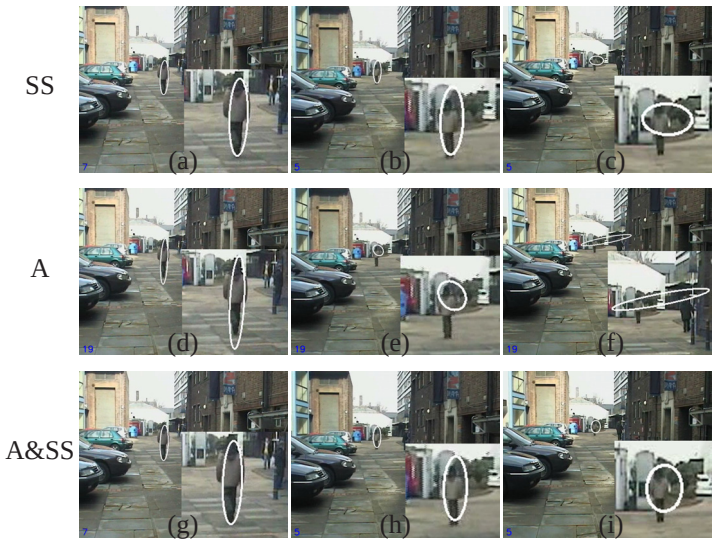


Figure 6: Examples of tracking results (Target P3 frames 160, 465, 659) using automatic scale selection (SS), adaptive weighting for Particle Filter (A), and both (A&SS).

Fig. 5 (a) shows the effect of the automatic scale selection (A&SS versus A). Changing the scale of the filters according with the target size improves the performance of the tracker. SS is particularly effective in presence of large scale changes (P3, V4), where the error is significantly reduced. Visual results for P3 are presented in Fig. 6: the lost track obtained with a fixed scale (Fig. 6 (d)-(f)) is avoided (Fig. 6 (g)-(i)).

The average result of adapting the contribution of colour and orientation, as described in Sec. 4.2, is showed in Fig. 5 (a) (A&SS versus SS). Adapting the weights improves the performance of Particle Filter and results in a lower error. By analysing further the values of Tab. 1 we notice that a good improvement is obtained on the targets P2, F2, and V4. In V4 the importance of the orientation histogram decreases together with the object size, due to the insufficient detail available. For other targets like F1 the results are similar, however we should consider that the weights fixed for the comparison are optimized for this dataset, and therefore are data dependent. Hence our adaptive solution reduces the free parameters in the multi-feature PF, increasing in average the quality of the track.

Fig. 7 shows sample frames of pedestrian tracking. The target (P1) is modelled with colour histograms (C) (Fig. 7 (a)-(d)) and with the adaptive combination of colour and orientation (C+O(A&SS)) (Fig. 7 (e)-(h)). In this case the information introduced by the orientation histogram makes the difference (Fig. 7 (e)-(h)), and the lost track returned by the colour only is avoided. In Fig. 8 sample frames of F1 face tracking are showed. As for P1 the orientation histogram model improves the performance of the tracker.

Finally Fig. 5 (b) summarises the results of the multi-features tracker compared with the single-feature ones. The model based only on the Orientation histogram (O(SS)) suffers from low descriptiveness; however the combination with the colour histogram (C+O(A&SS)) outperforms the classical colour based solution (C) improving the quality and reducing the number of lost tracks (see also Tab. 1).

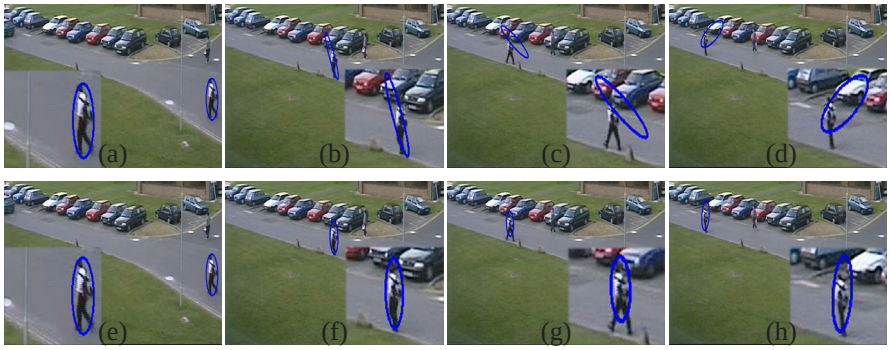


Figure 7: Examples of tracking results (Target P1 frames 0, 146, 230, 293) using colour histograms (top row), adaptive colour and orientation histograms (bottom row).

Figure 8: Examples of tracking results (Target F1, frames 85, 364, 1120) using colour histograms (top row), and adaptive colour and orientation histograms (bottom row).

Table 1: Comparison of tracking accuracy results by using different features. C: colour, O: orientation, C+O(SS): colour and orientation with fixed weights, C+O(A&SS): as C+O(SS) but with adaptive weights.

	C	**O**			**C+O(SS)**		**C+O(A&SS)**	
		FOH	ROH		FOH	ROH	FOH	ROH
P1	0.535(LT)	0.932(LT)	0.513(LT)		0.322(LT)	0.190	0.294	0.183
P2	0.861(LT)	0.944(LT)	0.940(LT)		0.433(LT)	0.403(LT)	0.414(LT)	0.351(LT)
P3	0.457(LT)	0.896(LT)	0.828(LT)		0.333(LT)	0.304(LT)	0.339(LT)	0.268
F1	0.281	0.742(LT)	0.710(LT)		0.228	0.213	0.240	0.222
F2	0.397(LT)	0.633(LT)	0.519(LT)		0.376(LT)	0.337(LT)	0.355(LT)	0.320(LT)
F3	0.279	0.369	0.305		0.235	0.237	0.244	0.229
V4	0.249	0.845(LT)	0.844(LT)		0.275	0.322	0.253	0.239

6 Conclusions

We introduced a novel multi-feature target tracker that employs Particle Filter over a combination of colour and orientation histograms. Colour histograms are calculated in the RGB colour space. Orientation histograms are obtained from eigenvalues of the structure

tensor, providing a least-square estimate of the gradient which increases robustness to noise. The two representations are combined at the Particle Filter level. The weighed distribution of the particles is analysed frame by frame, and a reliability measure is derived for each feature based on the dispersion of the corresponding particles. This is used to balance the contribution of the two features adaptively. The scale selection problem for the computation of the gradient and of the structure tensor is solved by iteratively using the output of the tracker for setting the scale of the filters.

Experimental results over a set of real-world sequences show that the multi-feature representation is more descriptive and leads to better results than the standard colour-based histograms. Also, the adaptive Particle Filter algorithm improves the flexibility of the representation by exploiting the complementarity of the failure modes in an efficient way. This flexibility will allow for the integration of other features in the representation.

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 Transactions on Signal Processing*, 50(2):174–188, February 2002.

[2] J. Bigun, T. Bigun, and K. Nilsson. Recognition by symmetry derivatives and the generalized structure tensor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(12):1590–1605, 2004.

[3] J. Bigun and G. H. Granlund. Optimal orientation detection of linear symmetry. In *Proceedings of the IEEE First International Conference on Computer Vision*, pages 433–438, London, Great Britain, June 1987.

[4] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog.*, pages 232–237, June 1998.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[6] W.T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proc.of Workshop on Autom. Face and Gesture Recognition*, pages 296–301, 1995.

[7] T.L. Liu and H.T. Chen. Real-time tracking using trust-region methods. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):397–402, March 2004.

[8] K. Nickels and S. Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image Vision Comput.*, 20(1):47–58, 2002.

[9] H.K. Nishihara, H.J. Thomas, and E. Huber. Real-time tracking of people using stereo and motion. In *SPIE Proceedings*, pages 266–273, 1994.

[10] K. Nummiaro, E. Koller-Meier, and L. Van Gool. A color-based particle filter. In *Proc.of Workshop on Generative-Model-Based Vision*, pages 53–60, June 2002.