Human Motion Tracking in video: a practical approach

Tony Tung
Kyoto University, Japan

Takashi Matsuyama
Kyoto University, Japan

ABSTRACT

This chapter presents a new formulation for the problem of human motion tracking in video. Tracking is still a challenging problem when strong appearance changes occur as in videos of humans in motion. Most trackers rely on a predefined template or on a training dataset to achieve detection and tracking. Therefore they are not efficient to track objects whose appearance is not known in advance. A solution is to use an online method that updates iteratively a subspace of reference target models. In addition, we propose to integrate color and motion cues in a particle filter framework to track human body parts. The algorithm process consists of two modes, switching between detection and tracking. The detection steps involve trained classifiers to update estimated positions of the tracking windows, whereas tracking steps rely on an adaptative color-based particle filter coupled with optical flow estimations. The Earth Mover distance is used to compare color models in a global fashion, and constraints on flow features avoid drifting effects. The proposed method has revealed its efficiency to track body parts in motion and can cope with full appearance changes. Experiments were performed on challenging real world videos with poorly textured models and non-linear motions.

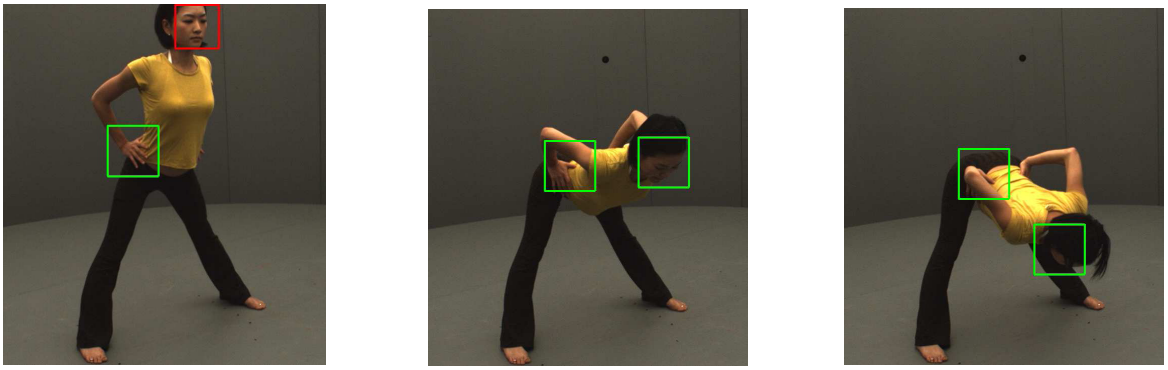
1. INTRODUCTION

Human motion tracking is a common requirement for many real world applications, such as video surveillance, games, cultural and medical applications (e.g. for motion and behavior study). The literature has provided successful algorithms to detect and track objects of a predefined class in image streams or videos. Simple object can be detected and tracked using various image features such as color regions, edges, contours, or texture. On the other hand, complex objects such as human faces require more sophisticated features to handle the multiple possible instances of the object class. For this purpose, statistical methods are a good alternative. First, a statistical model (or classifier) learns different patterns related to the object of interest (e.g. different views of human faces), including good and bad samples. And then the system is able to estimate whether a region contains an object of interest or not. This kind of approach has become very popular. For

example, the face detector of (Viola, & Jones, 2001) is well known for its efficiency. The main drawback is the dependence to prior knowledge on the object class. As the system is trained on a finite dataset, the detection is somehow constrained to it. As a matter of fact, most of the tracking methods were not designed to keep the track of an object whose appearance could strongly change. If there is no a priori knowledge on its multiple possible appearances, then the detection fails and the track is lost. Hence, tracking a head which turns completely, or tracking a hand in action remain challenging problems, as appearance changes occur quite frequently for human body parts in motion .

We introduce a new formulation dedicated to the problem of appearance changes for object tracking in video. Our approach integrates color cues and motion cues to establish a robust tracking. As well, an online iterative process updates a subspace of reference templates so that the tracking system remains robust to occlusions. The method workflow contains two modes, switching between detection and tracking. The detection steps involve trained classifiers to update estimated positions of the tracking windows. In particular, we use the cascade of boosted classifiers of Haar-like features by (Viola, & Jones, 2001) to perform head detection. Other body parts can be either detected using this technique with ad-hoc training samples, or chosen by users at the initialization step, or as well can be deduced based on prior knowledge on human shape features and constraints. The tracking steps rely on an adaptative color-based particle filter (Isard, & Blake, 1998) coupled with optical flow estimations (Lucas, & Kanade, 1981; Tomasi, & Kanade, 1991). The Earth Mover distance (Rubner, Tomasi, & Guibas, 1998) has been chosen to compare color models due to its robustness to small color variations. Drift effects inherent to adaptative tracking methods are handled using optical flow estimations (motion features).

Our experiments show the accuracy and robustness of the proposed method on challenging video sequences of human in motion. For example, videos of yoga performances (stretching exercises at various speed) with poorly textured models and non-linear motions were used for testing (cf. Fig. 1).



#1 #30 #70

Fig. 1. Body part tracking with color-based particle filter driven by optical flow. The proposed approach is robust to strong occlusion and full appearance change. Detected regions are denoted by red squares, and tracked regions by green squares.

The rest of the chapter is organized as follows. The next section gives a recap of work related to the techniques presented in this chapter. Section 3 presents an overview of the algorithm (initialization step and workflow). Section 4 describes the tracking process based on our color-based particle filter driven by optical flow. Section 5 presents experimental results. Section 6 concludes with a discussion on our contributions.

2. STATE OF THE ART

In the last decade, acquisition devices have become even more accurate and accessible for non-expert users. This has lead to a rapid growth of various imaging applications. In particular, the scientific community has shown a real interest to human body part detection and tracking. For example, face detection in images is nowadays a popular and well explored topic (Viola, & Jones, 2001; Hjelmas, & Low, 2002; Choudhury, Schmid, & Mikolajczyk, 2003). In (Viola, & Jones, 2001), the authors proposed a cascade of boosted tree classifiers of Haar-like features. The classifier is first trained on positive and negative samples, and then the detection is performed by sliding a search window through candidate images and checking whether a region contains an object of interest or not. The technique is known to be fast and efficient, and can be tuned to detect any kind of object class if the classifier is trained on good samples.

Tracking in video is a popular field of research as well. Recognition from video is still challenging because frames are often of low quality, and details can be small (e.g. in video surveillance). Various approaches were proposed to track image features (Lucas, & Kanade, 1981; Tomasi, & Kanade, 1991; Lowe, 2004; Lucena, Fuertes, & de la Blanca, 2004; Tola, Lepetit, & Fua, 2008). Lucas, Tomasi and Kanade first select the good features which are optimal for tracking, and then keep the tracks of these features in consecutive frames. The KLT feature tracker is often used for optical flow estimation to estimate the deformations between two frames. As a differential method, it assumes that the pixel intensity of objects is not significantly different between two frames.

Techniques based on prediction and correction as Kalman filter, and more recently particle filters have become widely used (Isard, & Blake, 1998; Doucet, Godsill, & Andrieu, 2000; Perez, Hue, Vermaak, & Gangnet, 2002; Sugimoto, Yachi, & Matsuyama, 2003; Okuma, Taleghani, de Freitas, Kakade, Little, & Lowe, 2004; Dornaika, & Davoine, 2005; Wang, Chen, & Gao, 2005; Li, Ai, Yamashita, Lao, & Kawade, 2007; Ross, Lim, Lin, & Yang, 2007; Kim, Kumar, Pavlovic, & Rowley, 2008). Particle filters (or sequential Monte Carlo or Condensation) are Bayesian model estimation techniques based on simulation. The basic idea is to approximate a sequence of probability distributions using a large set of random samples (called particles). Then the particles are propagated through the frames based on importance sampling and resampling mechanisms. Usually, the particles converge rapidly to the distributions of interest. The algorithm allows robust tracking of objects in cluttered scene, and can handle non-linear motion models more complex than those commonly used in Kalman filters. The major differences between the different particle filter based approaches rely on the design of the sampling strategies, which make particles having higher probability mass in regions of interest.

In (Black, & Jepson, 1998; Collins, Liu, & Leordeanu, 2005 ; Wang, Chen, & Gao, 2005; Ross, Lim, Lin, & Yang, 2007; Kim, Kumar, Pavlovic, & Rowley, 2008), linear dimension reduction methods (PCA, LDA) are used to extract feature vectors from the regions of interest. These

approaches suit well for adaptative face tracking and can be formulated in the particle filtering framework as well. Nevertheless they require a big training data set to be efficient (Martinez, & Kak, 2001), and still cannot cope with unpredicted change of appearance. On the other hand, color-based models of regions can capture larger appearance variations (Bradski, 1998; Comaniciu, Ramesh, & Meeh, 2000). In (Perez, Hue, Vermaak, & Gangnet, 2002), the authors integrate a color-based model tracker (as in the Meanshift technique of Comaniciu, Ramesh, and Meeh) within a particle filter framework. The model uses color histograms in the HSV space and the Bhattacharyya distance for color distribution comparisons. Nevertheless these methods usually fail to track objects in motion or have an increasing drift on long video sequences due to strong appearance changes or important lighting variations (Matthews, Ishikawa, & Baker, 2004). Indeed most algorithms assume that the model of the target object does not change significantly over time. To adapt the model to appearance changes and lighting variations, subspace of the target object features are extracted (Collins, Liu, & Leordeanu, 2005; Wang, Chen, & Gao, 2005; Ross, Lim, Lin, & Yang, 2007; Kim, Kumar, Pavlovic, & Rowley, 2008). In (Ross, Lim, Lin, & Yang, 2007), a subspace of eigenvectors representing the target object is incrementally updated through the tracking process. Thus, offline learning step is not required and tracking of unknown objects is possible. Recently, (Kim, Kumar, Pavlovic, & Rowley, 2008) proposed to extend this approach with additional terms in the data likelihood definition. In particular, the drift error is handled using an additional dataset of images. However, these approaches are particularly tuned for face tracking, and still require training datasets for every different view of faces.

The core of our approach divides into two steps which are detection and tracking, as (Sugimoto, Yachi, & Matsuyama, 2003; Li, Ai, Yamashita, Lao, & Kawade, 2007). Switching between the two modes allows to dynamically update the search window to an accurate position whenever the detection is positive. In this work, we propose to run a color-based particle filter to achieve the tracking process. Our tracker uses a subspace of color models of regions of interest extracted from the previous frames, and relies on them to estimate the position of the object in the current frame. The subspace is iteratively updated through the video sequence, and dynamically updated by the detection process. The detection is performed by a cascade of boosted classifiers (Viola, & Jones, 2001) and thus can be trained to detect any object class. We also propose to use the Earth Mover distance to improve the robustness of tracking with lighting variations, and constraints based on optical flow estimations to cope with drift effects.

3. ALGORITHM OVERVIEW

This section describes the algorithm workflow. The proposed approach combines two modes, switching between detection mode and tracking mode. The tracking process can be run independently if no detector is available for the class of the object of interest. Besides the tracking process, the detection improves the response accuracy online, and is used as well as initialization step. A subspace of color-based models is used to infer the object of interest location.

3.1. Initialization

The initialization step consists in defining the objects to track. In our framework, we focused on human body parts because of the wide rang of possible applications. Basically, there are three straightforward ways to define the regions of interest:

1. Automatic: this can be achieved by a detection process using statistical machine learning method (e.g. the face detector of Viola and Jones).

2. Manual: regions of interest are defined by the user (e.g. by picking regions in the first frame). This allows to track any body part without having any prior knowledge.

3. Deduction: as the human body has self-constrained motions, its structure can be deduced using a priori knowledge and fuzzy rules. For example, a detected face gives some hints to deduce the torso position, etc.

In some of our experiments (cf. Sect. 5), we have combined the three approaches (e.g. head is automatically detected, torso is deduced, and hands are picked). Afterwards, the regions of interest are used as reference templates on which the tracker relies on to process the next frames.

3.2. Workflow

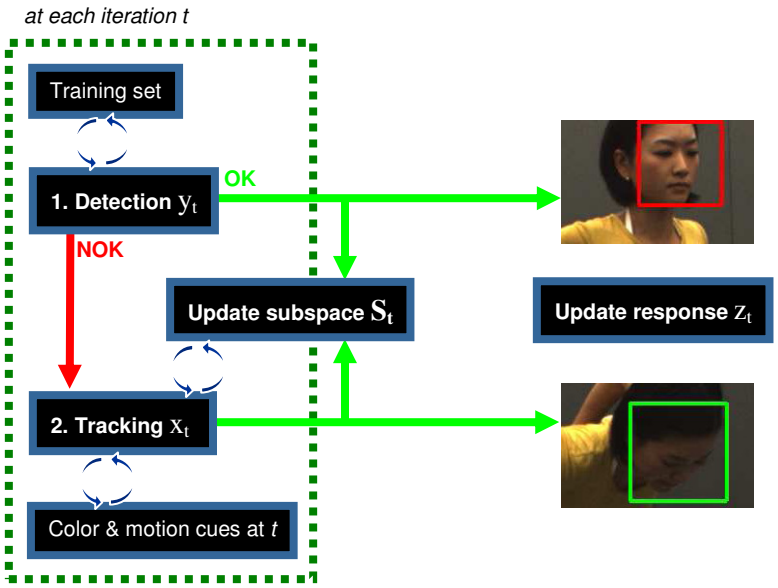


Fig. 2. Algorithm workflow. If the detection process y_t at time t is positive, then the algorithm response z_t and the subspace of color models S_t are updated with y_t. If the detection fails, then z_t and S_t are updated by the tracking process x_t. Note that S_t is used to infer the state x_t.

Assuming the initialization occurs at time t_0, then for every frame at $t, t > t_0$, the tracker estimates the positions of M objects of interest $\{A_i\}_{i=1...M}$ based on the color-model subspace

$S_t^i = \{h_{t-k}^i, ..., h_{t-1}^i\}$, where h_j^i denotes the color-model of A_i at time j, and k is the size of the subspaces (which in fact can be different for every object). Assuming a Bayesian framework (cf. Sect. 4), the hidden state x_t^i corresponding to the estimated position of A_i at time t by the tracker, is inferred by S_t^i and x_{t-1}^i. We denote by y_t^i the data corresponding to the detection of A_i at time t, and z_t^i the response of the algorithm. Thus, if the detection of A_i at t is positive, then $z_t^i = y_t^i$, else $z_t^i = x_t^i$. Indeed if the detection of A_i at t is positive, then S_{t+1}^i will be updated with the color model corresponding to y_t^i. And if not, then S_{t+1}^i will be updated with the color model corresponding to x_t^i. The workflow is illustrated on Figure 2 with $M = 1$ and $k = 1$.

4. PARTICLE FILTERING DRIVEN BY OPTICAL FLOW

In this section we present our algorithm formulation based on color-based particle filtering (Isard, & Blake, 1998; Perez, Hue, Vermaak, & Gangnet, 2002] and optical flow estimations (Tomasi, & Kanade, 1991). We propose to use the Earth Mover Distance (Rubner, Tomasi, & Guibas, 1998) to compare color models, and extracted motion features to improve tracking accuracy. Moreover our method updates iteratively a subspace of template models to handle appearance changes and partial occlusions.

4.1. Particle filtering

We denote by x_t a target state at time t, z_t the observation data at time t, and $Z_t = \{z_1, ..., z_t\}$ all the observations up to time t. Assuming a non-Gaussian state space model, the prior probability $p(x_t \mid Z_{t-1})$ at time t in a Markov process is defined as:

$$p(x_t \mid Z_{t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1}) dx_{t-1}, \quad (1)$$

where $p(x_t \mid x_{t-1})$ is a state transition distribution, and $p(x_{t-1} \mid Z_{t-1})$ stands for a posterior probability at time $t-1$. The posterior probability whose the tracking system aims to estimate at each time is defined as:

$$p(x_t \mid Z_t) \propto p(z_t \mid x_t) p(x_t \mid Z_{t-1}), \quad (2)$$

where $p(z_t \mid x_t)$ is the data likelihood at time t. According to the particle filtering framework, the posterior $p(x_t \mid Z_t)$ is approximated by a Dirac measure on a finite set of P particles $\{x_t^i\}_{i=1...P}$ following a sequential Monte Carlo framework (Doucet, Godsill, & Andrieu, 2000). Candidate particles are sampled by a proposal transition kernel $q(\tilde{x}_t^i \mid x_{t-1}^i, z_{t-1})$. The new filtering distribution is approximated by a new sample set of particles $\{\tilde{x}_t^i\}_{i=1...P}$ having the importance weights $\{w_t^i\}_{i=1...P}$, where

$$w_t^i \propto \frac{p(z_t \mid \tilde{x}_t^i)\, p(\tilde{x}_t^i \mid x_{t-1}^i)}{q(\tilde{x}_t^i \mid x_{t-1}^i, z_{t-1})} \qquad \textit{and} \qquad \sum_{i=1}^{P} w_t^i = 1. \tag{3}$$

The sample set $\{x_t^i\}_{i=1\ldots P}$ can then be obtained by resampling $\{\tilde{x}_t^i\}_{i=1\ldots P}$ with respect to $\{w_t^i\}_{i=1\ldots P}$. By default, the Bootstrap filter is chosen as proposal distribution: $q(\tilde{x}_t^i \mid x_{t-1}^i, z_{t-1}) = p(\tilde{x}_t^i \mid x_{t-1}^i)$. Hence the weights can be computed by evaluating the corresponding data likelihood. Finally, x_t is estimated upon the Monte Carlo approximation of the expectation $\hat{x}_t = \frac{1}{P}\sum_{i=1}^{P} x_t^i$.

We denote by E, the overall energy function: $E = E_s + E_m + E_d$, where E_s is an energy related to color cues (cf. Sect. 4.2), E_m and E_d are energies related to motion features (cf. Sect. 4.4). E has lower values as the search window is close to the target object. Thus, to favor candidate regions whose color distribution is similar to the reference model at time t, the data likelihood $p(z_t \mid x_t)$ is modeled as a Gaussian function:

$$p(z_t \mid \tilde{x}_t^i) \propto \exp\left(-\frac{E}{\sigma^2}\right), \tag{4}$$

where σ is a scale factor, and therefore a small E returns a large weight.

4.2. Color-based model

The efficiency of color distributions to track color content of regions that match a reference color model has been demonstrated in (Bradski, 2000; Comaniciu, Ramesh, & Meeh, 2000; Perez, Hue, Vermaak, & Gangnet, 2002]. They are represented by histograms to characterize the chromatic information of regions. Hence they are robust against non-rigidity and rotation. In addition, the Hue-Saturation-Value (HSV) color space has been chosen due to its low sensitivity to lighting condition. In our approach, color distributions are discretized into three histograms of N_h, N_s, and N_v bins for the hue, saturation, and value respectively.

Let α be h, s, or v, $q_t(x_t) = \frac{1}{3}\sum_{\alpha} q_t^{\alpha}(x_t)$, and $q_t^{\alpha}(x_t) = \{q_t^{\alpha}(i, x_t)\}_{i=1\ldots N_\alpha}$. $q_t(x_t)$ denotes the kernel density estimate of the color distribution in the candidate region $R(x_t)$ of the state x_t at time t, and is composed by:

$$q_t^{\alpha}(i, x_t) = K_\alpha \sum_{u \in R(x_t)} \delta[h_\alpha(u) - i], \tag{5}$$

where K_α is a normalization constant so that $\sum_{i=1}^{N_\alpha} q_t^{\alpha}(i, x_t) = 1$, h_α is a function assigning the pixel color at location u to the corresponding histogram bin, and δ is the Kronecker delta function.

At time t, $q_t(x_t)$ is compared to a set of reference color model templates $S_t = \{h_{t-k},...,h_{t-1}\}$, where k is the number of templates. The templates are extracted iteratively from the detected regions at each frame. We recall that color model subspaces help to handle appearance changes and partial occlusions, and we define the energy function:

$$E_s[S_t, q_t(x_t)] = \min_{h \in S_t}(D^2[h, q_t(x_t)]), \tag{6}$$

where D is a distance between color distributions (cf. Sect. 4.3).

4.3 Earth Mover distance

We propose to use the Earth Mover distance (EMD) (Hillier, & Lieberman, 1990; Rubner, Tomasi, & Guibas, 1998) to strengthen the property of invariance to lighting of the HSV color space. EMD allows to make global comparison of color distributions relying on a global optimization process. This method is more robust than approaches relying on histogram bin-to-bin distances that are more sensitive to quantization and small color changes. The distributions are represented by sets of weighted features called *signatures*. The EMD is then defined as the minimal amount of *work* needed to match a signature to another one. The notion of work relies on a metric (e.g. a distance) between two features. In our framework we use the L_1 norm as distance, and histogram bins as features.

Assuming two signatures to compare $P = \{(p_1, w_1),...,(p_m, w_m)\}$ and $Q = \{(q_1, u_1),...,(q_n, u_n)\}$, P having m components p_i with weight w_i, and Q having n components q_j with weight u_j. The global optimization process consists in finding the amount of data f_{ij} of a signature to be transported from the component i to the component j that minimizes the work W:

$$W = \min_{f_{ij}}(\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}), \tag{7}$$

where d_{ij} is the distance between the components p_i and q_j assuming the following constraints:

$$f_{ij} \geq 0 \qquad 1 \leq i \leq m, 1 \leq j \leq n,$$

$$\sum_{j=1}^{n} f_{ij} \leq w_i \qquad 1 \leq i \leq m,$$

$$\sum_{i=1}^{m} f_{ij} \leq u_j \qquad 1 \leq j \leq n,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} w_i, \sum_{j=1}^{n} u_j).$$

The first constraint allows only the displacements from P to Q. The two following constraints bound the amount of data transported by P, and the amount of data received by Q to their respective weights. The last constraint sets the maximal amount of data that can be displaced.
The EMD distance D between two signatures P and Q is then defined as:

$$D(P,Q) = \frac{W}{\mathrm{N}} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}, \tag{8}$$

where the normalization factor N ensures a good balance when comparing signatures of different size (N is the smallest sum of the signature weights). Note that EMD computation can be approximated in linear time with guaranteed error bounds (Shirdhonkar, & Jacobs, 2008).

4.4 Motion cues

We propose to use motion features to guide the search window through the tracking process. Motion features are extracted using the KLT feature tracker (Lucas, & Kanade, 1981; Tomasi, & Kanade, 1991). The method detects feature windows and matches the similar ones between consecutive frames (cf. Fig. 3).

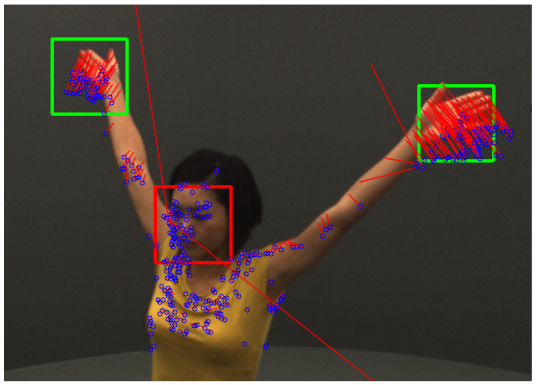


Fig. 3. Feature tracking. The tracking process is driven by motion features. Blue dots denote feature positions in the previous frame. Red lines show the estimated motion flows.

Assuming the set $Y_{t-1} = \{y_{t-1}^{j}\}_{j=1\ldots m}$ of m motion features detected in the neighborhood region of the state x_{t-1} (cf. Sect. 4) at time $t-1$, and the set $Y_t = \{y_t^{j}\}_{j=1\ldots m}$ of matching features extracted at time t, then $(Y_{t-1}, Y_t) = \{(y_{t-1}^{j}, y_t^{j})\}_{j=1\ldots m}$ forms a set of m motion vectors (optical flow field) between the frames at time $t-1$ and t. As well, we denote by $\tilde{Y}_t^{i}$ the set of features detected in the neighborhood region of the particle $\tilde{x}_t^{i}$, and $\tilde{y}_t$ the position of the search window estimated

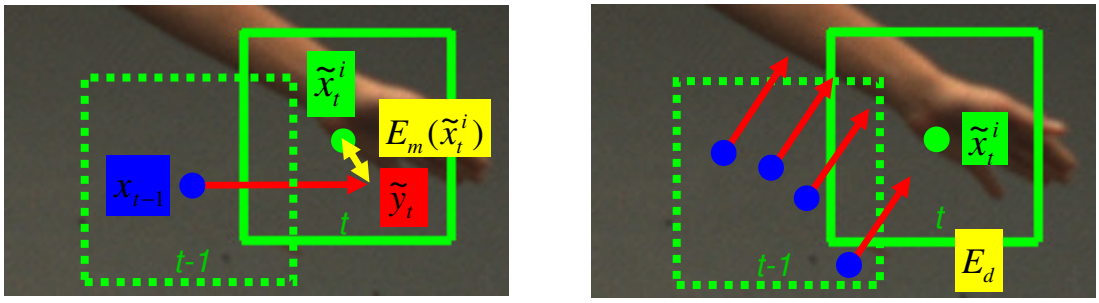
by optical flow as: $\tilde{y}_t = x_{t-1} + \text{median}(\{y_{t-1}^j - y_t^j\}_{j=1\ldots m})$. Thus we define the following energy functions:

$$E_m(\tilde{x}_t^i) = \alpha \cdot \left\| \tilde{x}_t^i - \tilde{y}_t \right\|_2 \qquad \textit{and} \qquad E_d = \beta \cdot \mathrm{C}(\tilde{Y}_t^i, Y_t), \tag{9}$$

where α and β are two constant values, and C is the following function:

$$\mathrm{C}(\tilde{Y}_t^i, Y_t) = 1 - \frac{\text{card}(\tilde{Y}_t^i \cap Y_t)}{\text{card}(Y_t)}. \tag{10}$$

The data energy E_m aims to favor the particles located around the object target position estimated by optical flow, whereas E_d aims to prevent the drift effect. E_d works as a constraint which attracts the particles near the estimated search window (cf. Fig. 4). E_m and E_d are introduced in the overall energy formulation as described in Sect. 4.1.



(a) (b)

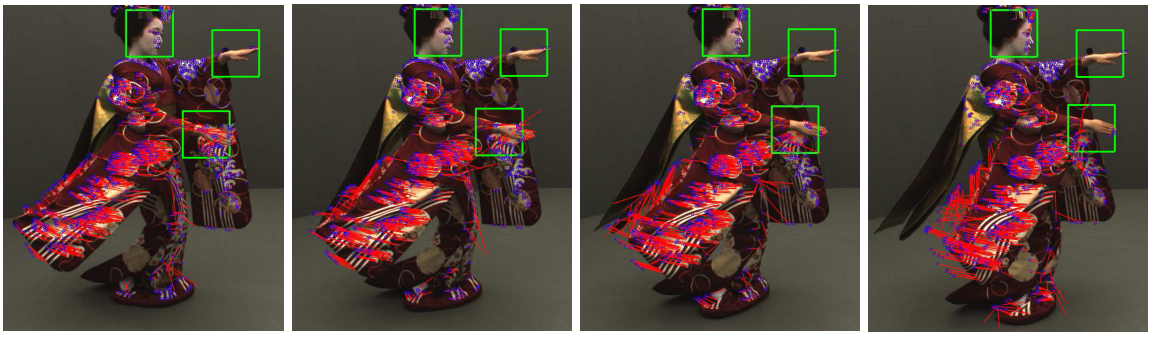
Fig. 4. Motion cues. Motion cues are formulated in term of energy to minimize. (a) E_m measures the distance between the estimated position $\tilde{x}_t^i$ by particles and the estimated position by optical flow $\tilde{y}_t$. (b) E_d maximizes the number of features detected in the previous frame.

5. EXPERIMENTAL RESULTS

Our algorithm has been tested on various real video sequences. For example, we have tracked the body parts of a lady practicing yoga (head, hands, torso, and feet) in different video sequences and from different viewpoints. The model wears simple clothes with no additional features (cf. Fig. 1 and Fig. 7). As well, we have tested the tracker on a model wearing traditional Japanese clothes which are more much complex and contain a lot of features (cf. Fig. 5). In this study case,

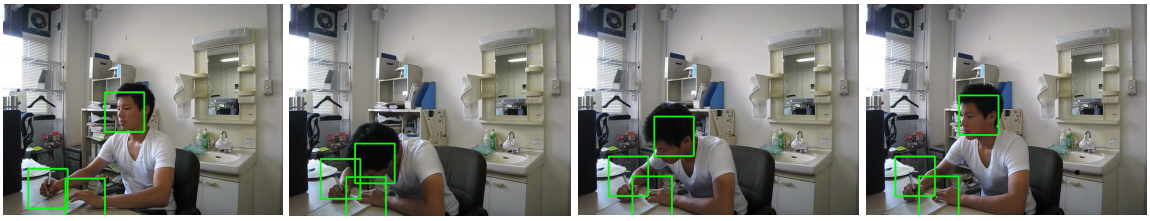
the video frame sizes are 640x480 and 720x576 pixels and were acquired at 25 fps. The algorithm was run on a Core2Duo 3.0 GHz with 4GB RAM.

The following parameters were identical for all the experiments: we have used $N_h = 10$, $N_s = 10$ and $N_v = 10$ for the quantization of color models, $P = 200$ particles, $k = 5$ for the color model subspace size, and $\sigma^2 = 0.1$ as scale factor of the likelihood model. The constant values α and β weight the contribution of the motion cues, and are tuned regarding to the frame size. He have defined a square window size of 40 pixels to determine the regions of interest. The proposed formulation has shown promising results even in uncontrolled environments. The Figures 1 and 6 illustrate the robustness to appearance change, lighting variation and partial occlusion, thanks to the online update of the color-based model subspace combined with the Earth Mover distance and motion cues. For example, the system can track a head even if the face in no more visible (e.g. hidden by hair or due to changing viewpoint). Figure 5 illustrates an accurate tracking with free-drift effect of a hand with a varying background under the guidance of optical flow as motion cues. Figure 7 illustrates the robustness of our approach in comparison to a color-based particle filter (Condensation of Perez, Hue, Vermaak, and Gangnet) that does not include our features. We show that the Condensation mixes regions having the same color shape and distribution whereas our tracker is not confused by the similar regions. This is due in particular to the addition of motion cues.



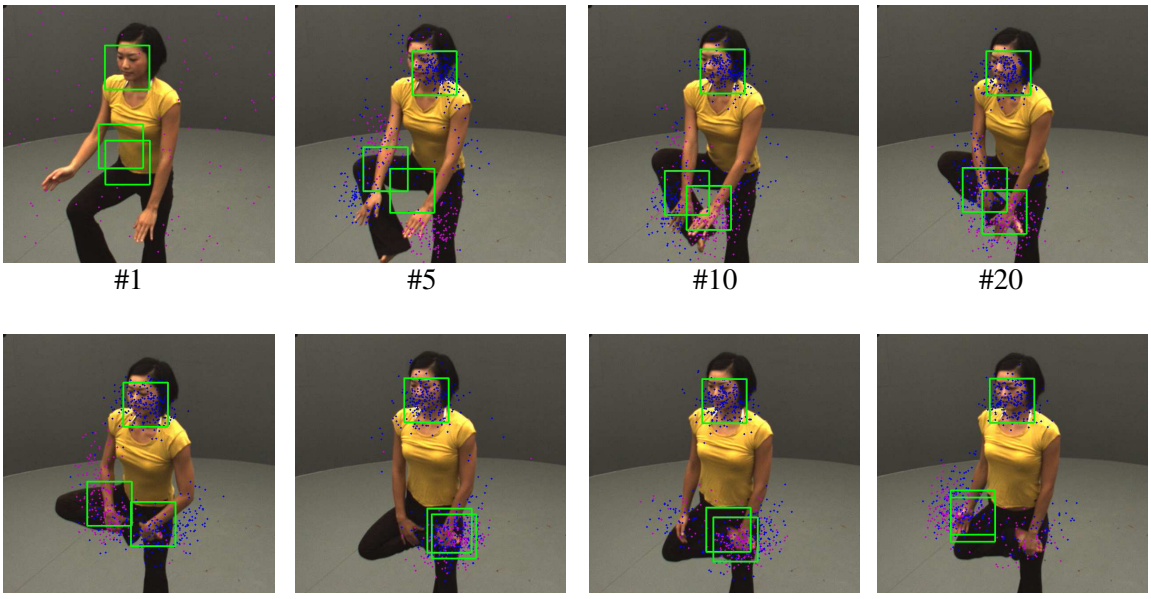
#1 #2 #3 #4

Fig. 5. Using optical flow to improve tracking. The combination of color cues and motion cues allows to perform robust tracking and prevent drift effects. The tracking of hands is efficient even with a changing background.

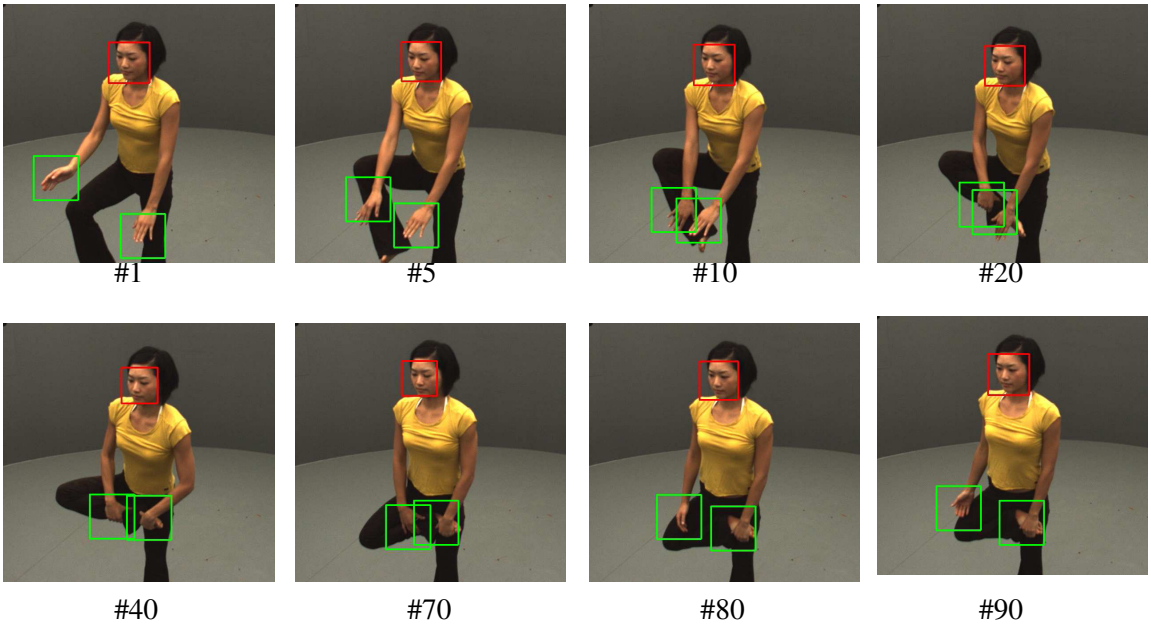


#1 #45 #145 #165

Fig. 6. Tracking with appearance change. The proposed approach integrates motion cues and a subspace of color models which is updated online through the video sequence. The system can track objects in motion with appearance change.



(a) Condensation.



(a) Particle filter driven by optical flow.

Fig. 7. Robust body part tracking. (a) Classical Condensation methods (Isard, & Blake, 1998; Perez, Hue, Vermaak, & Gangnet, 2002) are confused by regions with similar color and shape content. (b) In frame #20, both hands are almost included in a same tracking window, but afterwards motion cues have helped to discriminate the different tracks.

6. CONCLUSION

Human motion tracking in video is an attractive research field due to the numerous possible applications. The literature has provided powerful algorithms based on statistical methods especially dedicated to face detection and tracking. Nevertheless, it is still challenging to handle complex object classes such as human body parts whose appearance changes occur quite frequently while in motion.

In this work, we propose to integrate color cues and motion cues in a tracking process relying on a particle filter framework. We have used the Earth Mover distance to compare color-based model distribution in the HSV color space in order to strengthen the invariance to lighting condition. Combined with an online iterative update of color-based model subspace, we have obtained robustness to partial occlusion. We have also proposed to integrate extracted motion features (optical flow) to handle strong appearance changes and prevent drift effect. In addition, our tracking process is run jointly with a detection process that dynamically updates the system response. Our new formulation has been tested on real videos, and results on different sequences were shown. For future work, we believe our approach can be easily extended to handle a online manifold learning process. This would improve both detection and tracking modes.

REFERENCES

Black, M., & Jepson, A. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26, 63–84.

Bradski, G. (1998). Computer vision face tracking as a component of a perceptual user interface. *In Workshop on Applications of Computer Vision*. 214–219.

Choudhury, R., Schmid, C., & Mikolajczyk, K. (2003). Face detection and tracking in a video by propagating detection probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 1215-1228.

Collins, R., Liu, Y., & Leordeanu, M. (2005), Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1631-1643.

Comaniciu, D., Ramesh, V., & Meeh, P. (2000). Real-time tracking of non-rigid objects using mean shift. *IEEE Conference on Computer Vision and Pattern Recognition*, 2, 142–149.

Dornaika, F., & Davoine, F. (2005). Simultaneous facial action tracking and expression recognition using a particle filter. *IEEE International Conference on Computer Vision,* 1733-1738.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3), 197–208.

Hillier, F.S., & Lieberman, G.J. (1990). Introduction to mathematical programming. *McGraw-Hill*.

Hjelmas, E., & Low, B.K. (2002). Face detection: a survey. *Computer Vision and Image Understanding*, 83, 236–274.

Isard, M., & Blake, A. (1998). Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1), 5–28.

Kim, M., Kumar, S., Pavlovic, & V., Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Y., Ai, H., Yamashita, T., Lao, S., & Kawade, M. (2007). Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. *IEEE Conference on Computer Vision and Pattern Recognition*.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* , 60(2), 91–110.

Lucas, B., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *International Joint Conferences on Artificial Intelligence*, 674–679.

Lucena, M., Fuertes, J.M., & de la Blanca, N.P. (2004): Evaluation of three optical flow based observation models for tracking. *International Conference on Pattern Recognition*, 236–239.

Martinez, A.M., & Kak, A.C. (2001): PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 228–233.

Matthews, I., Ishikawa, T., & Baker, S. (2004): The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 810–815.

Okuma, K., Taleghani, A., de Freitas, N., Kakade, S., Little, J., & Lowe, D. (2004). A boosted particle filter: multitarget detection and tracking. *European Conference on Computer Vision*, 28–39.

Perez, P., Hue, C., Vermaak, J., & Gangnet, M. (2002). Color-based probabilistic tracking. *European Conference on Computer Vision*, 661–675.

Ross, D., Lim, J., Lin, R., & Yang, M. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1), 125-141.

Rubner, Y., Tomasi, C., & Guibas, L.J. (1998). A metric for distributions with applications to image databases. *IEEE International Conference on Computer Vision*, 59–66.

Shirdhonkar, S., & Jacobs, D.W. (2008). Approximate Earth mover's distance in linear time. *IEEE Conference on Computer Vision and Pattern Recognition.*

Sugimoto, A., Yachi, K., & Matsuyama, T. (2003). Tracking human heads based on interaction between hypotheses with certainty. *The 13th Scandinavian Conference on Image Analysis*.

Tola, E., Lepetit, V., & Fua, P. (2008). A fast local descriptor for dense matching. *IEEE Conference on Computer Vision and Pattern Recognition*.

Tomasi, C., & Kanade, T. (1991). Detection and tracking of point features. *Technical Report CMU-CS-91-132*, Carnegie Mellon University.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 511–518.

Wang, J., Chen, X., & Gao, W. (2005). Online selecting discriminative tracking features using particle filter. *IEEE Conference on Computer Vision and Pattern Recognition*, 2, 1037-1042.