Tracking objects in videos with texture features

John Klein
University of Rouen, LITIS
Rouen, France
Email: john.klein@etu.univ-rouen.fr

Christele Lecomte
University of Rouen, LITIS
Rouen, France
Email: christele.lecomte@univ-rouen.fr

Pierre Miche
INSA of Rouen, LITIS
Rouen, France
Email: pierre.miche@insa-rouen.fr

Abstract— **In this article, we address the problem of object tracking in videos for multimedia applications. To produce a reliable and robust tracking algorithm, several visual characteristics of the target object must be examined. The different sources of information must be carefully chosen so as to select only the most informative ones. We argue that color and texture cues can be both quickly handled by cooccurrence matrices. Yet these matrices are too sensitive to illumination changes occurring in natural scenes. By adding weights, drawn from kernels centered on representative colors of the object, the feature can cope with this matter. We provide experimental result obtained from different kinds of natural scenes.**

I. INTRODUCTION

In the wake of the exponential growth of the multimedia market, more and more intelligent systems are needed. In the field of video processing, one of these needs is the possibility to identify, track and process objects within videos. For example, this is also one of the MPEG7 research topics. This matter has been investigated by the computer vision community for a long time, and some results are already available. One of the first approach to catch the attention of the whole computer vision community was the mean shift algorithm [1], because it achieved much more robust tracking in real-time. More recently this algorithm was outperformed by sequential Monte Carlo methods, i.e. particle filters [2]. Indeed these filters can better deal with occlusions and clutters. Yet these tracking techniques usually cannot overcome any event occurring in a natural scene, most notably illumination changes, or particular movements.

To achieve this, reliable information from different sources must be introduced within the algorithms. Information quality is enhanced when non redundant sources are aggregated, but before using fusion process, one must carefully select the sources. In this article we argue that textural and color information are redundant and can be jointly extracted from raw image data. To produce such a feature, some authors firstly thought about applying texture extraction techniques to each color plan. Jung [3] computes wavelets on RGB plans, and so does Palm [4] with cooccurrence matrices. It is also possible to use texture extraction by substituting gray-level with colors, if you consider color as a continuous quantity. These approaches must be carefully designed, because they induce closeness between colors, which may not be relevant. Chang [5] calculates color cooccurrences, by counting occurrences of neighbor colors, instead of neighbor gray levels, and thereby induces no sense of closeness. Other methods do not belong to the two previously cited schemes, like color coherence vectors, spatial chromatic histograms [6], color density [1], as well as the works of Paschos [7].

Color cooccurrences were retained as a first process, because spatial and color information are equally rendered by this feature. Our contribution is the addition of weights for each color, using kernels centered on colors of the target object. This new feature is by essence more adapted to tracking issues, because it will be able to cope with soft illumination changes by its own. Our new feature will judged on two aspects: its capacity to combine color and texture without loss of information, and its robustness to soft natural illuminations changes.

This article is organized as follows, part II will present our contribution. Calculation of weighted color cooccurrences will be detailed and justified. In part III, a particle filter using our color-texture feature for visual tracking will be presented. In section IV some experiments will be presented and discussed. Finally results and procedure are summarized in a conclusion.

II. WEIGHTED COCCURRENCE MATRICES

A. Cooccurrences

Cooccurrence matrices constitute one of the most popular texture characterization techniques. They were introduced by Haralick [8] for gray-level texture. The formula used for their computation is the following:

$$M_{\vec{d}}(i,j) = \frac{\#\left\{p \in D/I\left(p\right)=i, I\left(p+\vec{d}\right)=j\right\}}{\#\left\{D\right\}} \quad (1)$$

I is an image defined on D, p a pixel of this image, $\vec{d}$ a translation vector, (i,j) a couple of gray levels, and $\#$ the cardinal of a set. It simply consists in storing statistics about gray levels topology. Note that in addition the computation of the matrices is fast. The vectors $\vec{d}$ are generally chosen so as to draw statistics in the 8-neighborhood of the pixel. This leads to four directions imposed by the discrete space D, ie $\vec{d} = \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} et\ \theta \in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$. Examining a texture at different scales may impose to adapt the norm of $\vec{d}$.

Adaptation to color is natural, the couple of gray levels can be replaced by a couple of colors. Thus matrices characterize colors, because if a color is absent from the processed texture, its line and column within the matrix will be empty. Spatial information is depicted by the number of occurrences of a given couple of color.

B. Adding weights

As part of tracking application, cooccurrence matrices from an image subwindow containing the object will constitute an *a priori* model. To perform the track in the driving sequence, we will afterwards need to judge the resemblance of any subwindow from the current image with our model. Such judgment can be obtained by comparing matrices from the local subwindow to those of the model, therefore a metric for the matrices is required. Cooccurrence matrices can be considered as histograms, because the storing order of the statistics has no importance. Many existing distances, or pseudo-distances, can be used for histograms: classical L_1, L_2 or L_p Minkowski distances, Kullback-Leibler divergence, χ^2 distance, Hellinger distance or Bhattacharyya distance. The latter one was retained for our study, because of its well-known noise robustness. Distance d is computed from two normalized histograms h and h', whose bins are indexed by i:

$$d\left(h, h'\right) = 1 - \sum_{i=1}^{M} \sqrt{h_i h'_i} \quad (2)$$

The two histograms must have the same number of bins M. Such distance can be applied to the cooccurrence matrices, after concatenation. Using this distance, it is clear that only identical matrices will produce a null distance. However, imagine that the matrices are computed upon two different images of the same objects, then only a slight change of illumination will produce very different matrices, therefore leading to a distance close to 1. In other words, cooccurrences precision is a drawback for tracking purpose, because one object can produce some matrices at some times of the sequence, which cannot be matched to the reference ones.

Noticing this problem, we came up with the idea of changing the occurrences increment. Indeed the goal to reach is that colors of the same kind fall in the same bins, so as to cope with slight illumination changes, which are unavoidable in natural scenes. A naive solution would be to reduce the number of bins M, but if so, the grid of the matrices is not representative of the object color-texture. The proposed solution is to give different voting weights to each color c_i, depending on whether this color is close to a representative color of the object, denoted as $\tilde{c_j}$. The sets of representative colors can be limited to a very few colors. We propose to compute the weights W_i using kernels centered on the representative colors:

$$W_i = \sum_{j=1}^{N_K} K_j\left(\|\tilde{c}_j - c_i\|\right) \quad (3)$$

N_K is the number of kernels equal to the number of representative colors, and $K_j()$ is a kernel. Kernels are adequate for our problem, because colors very close to a chosen representative color will be almost as important as the chosen one, colors a little far from the chosen ones will influence the increment and finally colors too far from the model will all be silent. In our study we have considered the use of three different kernels displayed over one dimension on figure 1. These kernels mix

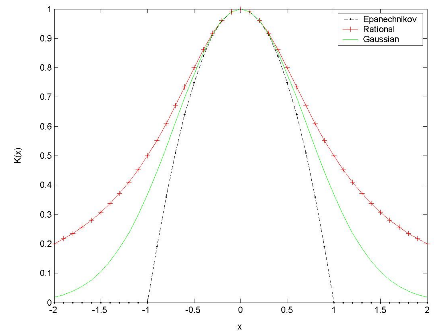


Fig. 1. 1D representation of several kernels

more or less data. The most mixing one is the rational kernel, the least mixing one is the Epanechnikov kernel [1], and the compromise is the gaussian kernel [11]. It is interesting to mix data if we want several colors to have similar weights, even if this means adding some noise. If one wants a clear cut color selection, then one should choose the Epanechnikov kernel. If one wants farther colors to be assimilated to the reference one, then one should choose the rationnal kernel.

III. Application through particle filtering

This section presents the tracking procedure chosen to test our new texture-color feature. This procedure is a particle filter. Particle filters have many advantages for tracking problems. They produce an optimal solution to the problem in terms of bayesian inference. In addition because of their random aspect, they can manage object occlusion problems. Indeed many tracking procedure use criterion minimization techniques, or convergence techniques, which are not enough reactive in the presence of occlusions. Particle filters scan more largely the image and can detect the object back again without difficulty when the occlusion ends. Furthermore particle filters can run in real time, in fact it does just depend on the feature computation time itself.

Given an unknown state X_t (for example the object location and bounding box dimensions), and observations Y_t (for example our new feature), particle filters return an estimation of the filtering density: $p\left(X_t|Y_{1:t}\right)$, where $Y_{1:t}$ corresponds to all available observations at time t. This estimation is obtained by drawing samples $\left(X_t^{(i)}\right)_{i:1..N}$, also called particles, from a law $q(.)$. Then according to probability theory, the following estimation is unbiased:

$$\hat{p}\left(X_t|Y_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)} \delta_{X_t^{(i)}}\left(X_t\right) \quad (4)$$

with the weights $w_t^{(i)} = \frac{p(X_t|Y_{1:t})}{q(X_t|Y_{1:t})}$, which annihilate the influence of density q. In addition, under markovian hypotheses, it can be shown that a recursive estimation of weights is possible using the formula:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(Y_t | X_t^{(i)}\right) p\left(X_t^{(i)} | X_{t-1}^{(i)}\right)}{q\left(X_t^{(i)} | X_{t-1}^{(i)}, Y_t\right)} \tag{5}$$

So by choosing $q = p(X_t|X_{t-1})$, which is called the prior density, the weights calculation is simplified: $w_t^{(i)} \propto w_{t-1}^{(i)} p\left(Y_t|X_t^{(i)}\right)$. As a transition density $p(X_t|X_{t-1})$, a centered gaussian noise is added to the estimate to propagate the particles. Note that this density characterizes the state dynamics, which guide the bounding box evolution. Making such a choice enables us to fairly judge the efficiency of our feature, because the filter success will only rely on the likelihood $p\left(Y_t|X_t^{(i)}\right)$, which is obtained thanks to the tested feature.

At this step only the likelihood is required to be able to sequentially estimate the filtering density. In this article the following likelihood expression is proposed:

$$p(Y_t|X_t) \quad \propto \quad \exp\left[-\lambda d_{min}(Y_t, Y_{Ref})^2\right] \tag{6}$$

Y_{Ref} is the set of reference matrices learnt on a subwindow containing the car. Y_t is the set of matrices computed on the local subwindow. d_{min} is the minimal Bhattacharyya distance between matrices of the two sets. To achieve faster processing, only one matrix is computed on the subwindow and compared to the whole reference set. If the reference set was computed, on the four directions of vector $\vec{d}$, as explained in section II, then the feature is invariant to rotation. Parameter λ helps to strengthen the difference between subwindows containing background and subwindows containing the object. A similar filter was proposed in [2], but it used only color information feature, we propose to use our color-texture feature, which contains more information.

Note that in terms of hardware implementation, calculation can be parallelized for each particle, which can strongly fasten the process. For more details about particle filters, the following reference articles are recommended [9], [10].

IV. Results and discussions

The tests performed in this section used the same settings. For the particle filter, over 100 particles were used, and an image could be processed in less than 1 second with a 2,4 GHz Athlon AMD processor implemented on a laptop. The computation of a matrix from an m by n image is in $O(mnN_k)$. With $N_k = 3$ or 4, an object is sufficiently well characterized.

A. Color-Texture combining results

One of the goals of weighted color cooccurrence matrices is to nicely aggregate color and texture information. The filter was firstly compared to rather color approach, in which features are color densities. This comparison is represented by the figure 2.

The object to track is the red cap of the felt pen. Using color approach the algorithm is lured by the presence of another red object and the track is lost. Since the red cap possesses some horizontal ridges, the textural information overcomes the difficulty. Our color-texture was also compared to gray-level cooccurrences, and also proved to better perform. This comparison is represented by the figure 3. The targeted object is the hand of the character. This video was strongly compressed, so the textural information was damaged, and turns out to be insufficient to carry out the tracking successfully. As the hand and the head of the character have similar skin color, we can think that a color approach would be unsatisfying too. Consequently, the fact that the track is maintained when using the color-texture cue shows that the feature does really take advantage from the two sources of information.

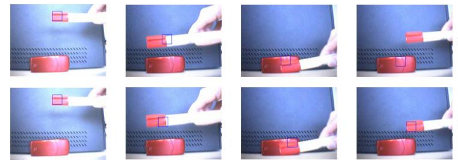


Fig. 2. (a) the red cap is tracked using color approach. (b) the cap is tracked using our color-texture approach, the filters bears the same settings.

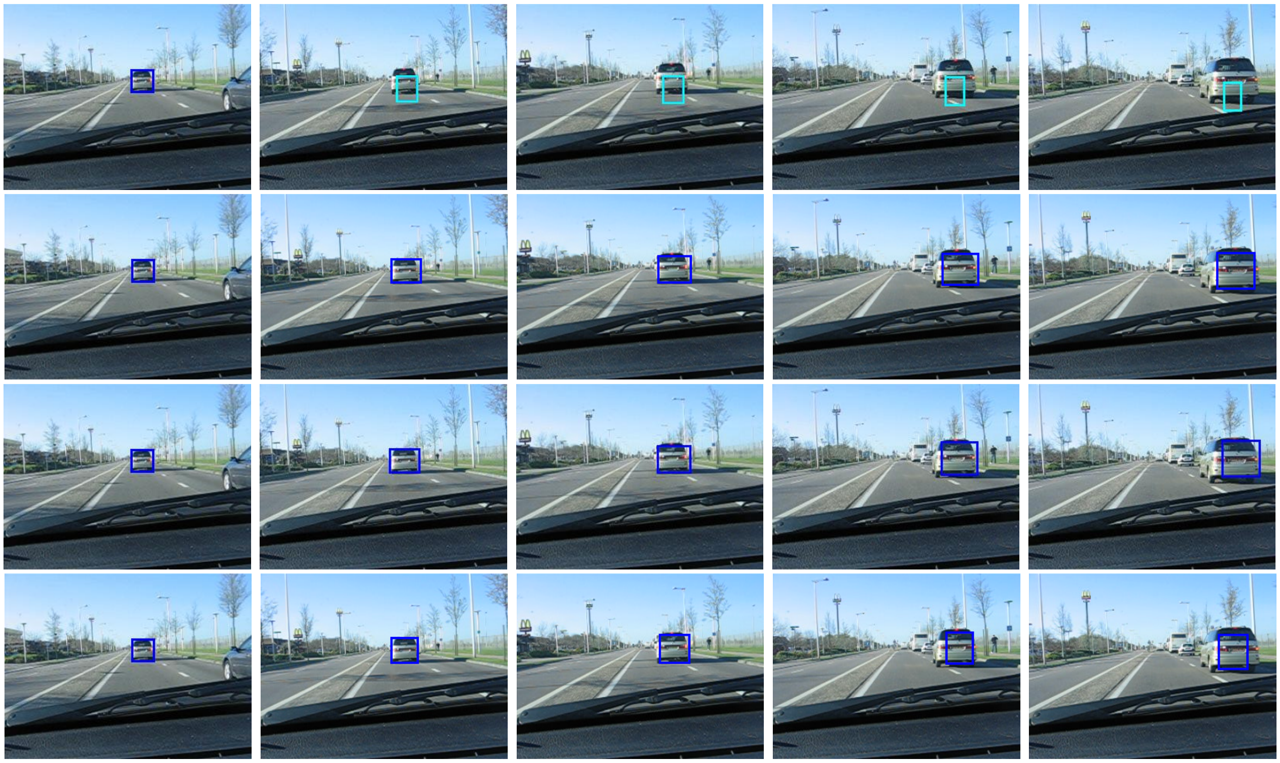
Fig. 3. (a) tracking the hand with grey-level cooccurrences, (b) tracking the hand with our color-texture feature, the filters bear the same settings

B. Tracking precision and robustness

It is always difficult to objectively measure the precision of a tracking procedure. The best way to build some opinion about it is still the human eye. On figure 4 the particle filter was used with classical cooccurrence matrices, and weighted cooccurrence matrices (WCM) with three different kernels.

According to this figure, color cooccurrence method does perform some tracking, which proves that cooccurrences contain reliable information. However this tracking is very imprecise. As you may see, only part of the car rear view and its shadow is included in the bounding box. This kind of tracking quality does not match computer vision goals. When looking at tracking results from WCM methods, it is clear that the tracking efficiency is enhanced, because the car is still globally contained by the bounding box. Concerning the different kernels, it appears that Gaussian and Epanechnikov ones both give very satisfying results, and that rational kernel is a little less accurate at the end of the sequence.

In terms of robustness, note that the filter has successfully handled slight illuminations changes, scale changes and view changes. In another part of the this sequence, the filter was able to deal with an occluding shadow coming from another car (see fig 5), which points out its capacity to deal with slight illuminations changes.



(a)

(b)

(c)

(d)

Fig. 4. (a): tracking with classical cooccurrence, (b): tracking with epanechnikov WCM, (c): tracking with Gaussian WCM, (d): tracking with rational WCM

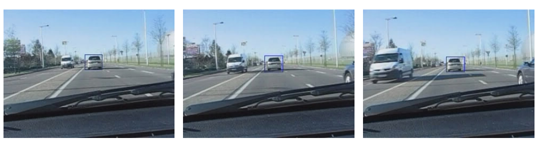


Fig. 5. maintaining track over illumination change due to shadow

V. Conclusion

In this article a new color-texture feature tuned for tracking was introduced. It is built upon the color cooccurrence matrices. If an object has to be modeled by this method, it is possible to choose a few representative colors of the object. Thus, we propose to weight the vote of colors in the occurrence increment, depending on whether or not a color is close to some of the representative chosen ones, hence the name of weighted color cooccurrences for this new feature.

The weighting is performed thanks to a kernel. Three kernels have been examined: rational, Gaussian and Epanechnikov. The experiment proved that weighted cooccurrences outperform classical cooccurrences in a tracking context, and lose no information compared to color-only or texture-only approaches.

Thanks to weighted cooccurrence matrices, a reliable color-texture source of information is available for tracking purposes. Yet color and texture are not the only cues that can be processed within an image. In future works, we intend to propose data fusion between our new feature and other features extracted from shape or movement analysis techniques. Thanks to this fusion, we hope to be able to handle more disrupting events for tracking.

References

[1] D. Comaniciu, V. Ramesh, "Real time tracking of non rigid objects using mean shift", *Comp. Vis. And Patten recognition conf.*, Hilton Head Island, USA, 2000, vol II, pp. 142-149

[2] P. Perez, D. Vermaak, C. Hue, M. Gangnet "Color-based Probabilistic Tracking", *ECCV.*, Copenhagen, Denmark, 2002, pp. 661-675

[3] C.R. Jung, Unsupervised multiscale segmentation of color images, *Pattern Recognition Letters, art. in press*, vol. X, 2006, pp XX-XX.

[4] C. Palm, Color Texture Classification by integrative Co-occurrence Matrices *Pattern Recognition*, vol. 37, 2003, pp 965-976.

[5] P. Chang, J. Krumm, "Object Recognition with Color Cooccurence Histograms", *in IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, USA, 1999.

[6] L. Cinque, S. Levialdi, A. Pellicano, Color-based image retrieval using spatial chromatic histograms *IEEE Multimadia Systems*, vol. 2, 1999, pp 969-973.

[7] G. Paschos, I. Radev, N. Prabakar, Image content-based retrieval using chromacity moments *IEEE trans. on Knowl. Data Eng.*, vol. 15, 2003, pp 1069-1072.

[8] R. M. Haralick, K. Shanmugan, I. Dinstein, Textural Features for Image Classification *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, n6, 1973, pp 610-621.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A Tutorial on Particles Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking *IEEE Trans. on Signal Processing*, vol. 50, n2, 1993, pp 174-188.

[10] A. Doucet, On sequential Simulation-based Methods for Bayesian Filtering *Technical report, Univ. of Cambridge*, vol. 310, 1998.

[11] F. Suard, V. Guigue, A. Rakotomamonjy, A. Bensrhair, Pedestrian detection using stereo-vision and graph kernels *Intelligent Vehicule Symposium*, Las Vegas, USA, 2005, pp. 267-272.