Feature Points Tracking Adaptive to Saturation

Andrés Romero [#1], Michele Gouiffes [#2], Lionel Lacassagne [*3]

[#] *IEF, UMR 8622, Paris XI*
Bat. 220, 91405 Orsay cedex, France
[1] andres.romero@ief.u-psud.fr
[2] michele.gouiffes@ief.u-psud.fr

[*] *CEA, LIST, Embedded Computation Lab*
91191 Gif sur Yvette, France
[3] lionel.lacassagne@cea.fr

***Abstract*—This paper proposes a color tracking strategy designed to improve the robustness against luminance and saturation changes due to illumination variations. On the one hand, color is helpful in terms of photometric invariance and separability power. On the other hand, it is more costly in time and resources and most color invariants are ill-defined at low saturation. To answer these issues, the proposed method weights the color and luminance information adaptively with respect to the current color saturation. A particular attention is also given to chose a color representation with low computational cost. The method is compared to classical color tracking in terms of accuracy, robustness and executing times. The experimental results achieved on several image sequences confirm the good performances of the method.**

I. Introduction

It is proved that color provides valuable information to improve many computer vision applications, such as segmentation, indexing or tracking [1]. Color feature points can be detected by combining the gradient of Dizenzo defined for vectorial images and the Harris operator in the same way as for gray-scale images [2]. The color feature points are usually complementary to the luminance points and more discriminative, since two different colors can have the same luminance. In applications like motion analysis for which a large amount of points have to be tracked, these aspects can be particularly useful.

Another advantage of color is the opportunity to use color invariants and color constancy techniques which are robust to illumination intensity and geometry changes [3], [4].

Despite the good performance of such features, they are unfortunately dedicated to the well-saturated colors. Indeed, since they do not depend on the luminance variations, their use can lead to the lost of some useful textural information. Another fact against their indiscriminate use is that they tend to be noisy when luminance and/or saturation is low. In these conditions, it is recommended to favor luminance information: it can be more reliable, less noisy, requires less computing power and converges faster.

Within an image sequence, it is possible to experiment an important decrease of lighting intensity. Even worst, the illumination change is generally not uniform in the whole image. Therefore, it is not straightforward to identify the best feature for tracking, either color or luminance information. The aim of this article is to go further with the approach proposed in [5] where the author addresses the problem of color relevance by defining a coefficient depending on the saturation and intensity.

We propose a KLT-like tracking method [6], [7] which adapts to the context depending on the color relevance. It is known that large displacements are difficult to follow with KLT, therefore the implementation of a multiscale tracking pyramid formed by a stack of different resolution images provides a partial solution to this problem. The algorithm starts at the lower resolution level and refines its results at next level until it gets the final locations of the points at the original resolution level [8].

The remainder of the paper is structured as follows. Section II gives a short explanation of the chosen color features and explains the color relevance coefficient. The tracking method is detailed in Section III and additional technical details are provided by Section IV. The experiments are shown in Section V. Several sequences tracking results are compared in terms of robustness, accuracy and time costs.

II. Color Spaces and Attributes

Considering matte surfaces, each point belonging to the surface of an object reflects the incoming light equally in all directions, forming an hemisphere of light centred around this point. The color of the light captured by a sensor, depends mainly on the light spectrum arriving to the surface's patch, on the reflectance coefficient which characterizes that surface point, on the geometry conditions and of course on the sensor's sensitivity to the captured light spectrum.

Considering the dichromatic model [9], the color vector $\mathbf{C} = (R, G, B)$ for a point p in the image, which is the projection of a physical point P is modeled as

$$\mathbf{C}(p) = m_b(P)\,\mathbf{C}_b(p) \tag{1}$$

where $\mathbf{C}_b = \left(C_b^R, C_b^G, C_b^B\right)$ is the color related to the body (or diffuse) reflection. The term m_b models the dependence on the scene geometry (lighting angle, orientation of surface). As an example, for the red channel, C_b^R:

$$C_b^R(p) = \int_\lambda S_R(\lambda)\,\mathcal{I}(\lambda, P)\,\mathcal{R}_b(\lambda, P)\,d\lambda \tag{2}$$

where $S_R(\lambda)$ is the camera sensitivity which is a function of the wavelength λ, $\mathcal{I}$ is the illuminant spectrum and $\mathcal{R}_b$ the reflectance of the material. If we assume that $S_R(\lambda)$ is band limited around the red wavelength λ_R then we can approximate it with a constant S_R for the band of interest. By considering the same for all the sensors, we get the following simplification:

$$\mathbf{C}(p) = \mathbf{a}(p) \,.\, \mathbf{C}_I(p)\, m_b(p)) \tag{3}$$

where . represents the Hadamard product. In (4), $\mathbf{C}_I$ represents the color of the illuminant and it depends on the gains of the camera (S_R, S_G, S_B). Finally, $\mathbf{a}(p)$=(a_R, a_G, a_B) depends on the reflectance and is supposed to be constant during the time.

A. Color features

The most classical invariant colorspace is (H, S, V) where the hue H gives an interpretation of color which is invariant to shadows and specular reflections [3], V is the luminance and S the saturation. While Hue has interesting invariance properties, its value is not so reliable when color saturation is low. In addition, its calculation is more elaborated than luminance and less easy to accelerate since it is based on trigonometric functions.

A very direct way to separate chrominance and luminance from the original (R, G, B) components is to normalize them, obtaining the components (r, g, b) which depend only on the albedo and therefore have no luminance information. Considering the red component and the model of (3), the equation reductions lead finally to:

$$r = \frac{R}{R+G+B} = \frac{\mathbf{a}_R}{\mathbf{a}_R + \mathbf{a}_G + \mathbf{a}_B} \tag{4}$$

Starting from rgb, the scalar value $L1 = max(r, g, b)$ represents the more saturated channel while being easier to manipulate compared to a three-dimensional data. As many color invariants, $L1$ reduces the separability between colors. Indeed, as r, g or b, all gray colors become indistinct $\left(r = g = b = \frac{1}{3}\right)$. In addition, $L1$ can reduce the distinction between two colors with same maximum value. In the context of the paper, where some small windows of interest are considered, it is assumed that the probability that two neighbour pixels have same maximum but different colors is low in most real sequences.

Fig.1 shows a few images from the sequence *Cardgame* from the ALOI database[1]. The first row displays the classical RGB images. Obviously, such features suffer from the illumination changes. The invariant features $L1$ and H are shown on the second and third rows respectively. As expected, the photometric variations are no more visible in these colorspaces.

However, this robustness is reached at the price of a lower separability between colors, especially when their saturation is low. This is noticeable for instance on the eyes of the character drawn on the box of Fig.1, which become uniformly gray

[1]http://staff.science.uva.nl/~aloi/

Fig. 1. a)Original RGB images. b) L1 invariant images. c) Hue invariant images.

using $L1$. This is also true with Hue since the title of the game can not be read. It is also obvious that the hue produces more noise than $L1$, especially when saturation is low[2]. In a tracking context, this problem can lead to the detection of outliers points and to matching instabilities. The following subsection introduces the color relevance function, which is used to determine when the color invariant feature can be used or not.

B. Relevance of color versus luminance.

Carron has proposed in [10] a method for color contour detection in the HSV space by fusing the information from Hue, Saturation and Value channels while keeping coherency with the introduction of relevance measure which depends on the saturation channel. Hue can be considered as a complement to Value and Saturation channels and be exploited only when it is considered relevant. We can also privilege it over Value and Saturation and consider the latter only when Hue is not relevant. This approach can be extended to other invariants with similar properties.

We use for each point p a relevance coefficient $\beta(p)$ which results from a sigmoid function applied to the saturation S at point p:

$$\beta(S) = \frac{1}{\pi}\left[\frac{\pi}{2} + atan\left(u_\beta\left(S - S_0\right)\right)\right] \tag{5}$$

where S_0 is the inflection point and u_β the slope parameter. Under S_0, the luminance is privileged, otherwise the color invariant feature is considered to be sufficiently well-defined and can be used properly.

For an RGB image, the saturation is calculated as

$$S = \frac{max(R,G,B) - min(R,G,B)}{max(R,G,B)} \tag{6}$$

for each point p.

[2]Hue is artificially represented in range [0-255] although it is an angular value. Therefore some of the noise is due to that representation, and some of the noise is due to ill definition of Hue.

Coefficient β is defined at each pixel involved in the tracking, in order to use color when it is meaningful and to rely on luminance otherwise.

Finally, several colorspaces will be compared for tracking purpose:

- The luminance I.
- A color invariant: H or $L1$.
- The colorspaces RGB, HSV, (r, g, b).
- The mixture of a color invariant with I (with photometric normalisation [11]), hereafter represented respectively as H,$L1$,HP and $L1P$.

Following section we explain the adaptive tracking procedure which combines the two components.

III. A TRACKING PROCEDURE ADAPTIVE TO THE SATURATION

The tracking is done considering (D_k, I_k) and $(D_{k'}, I_{k'})$, where D is the color invariant for the colorspace in consideration (H or $L1$) and I their corresponding luminance images at their respective times k and k'. A physical point P is located at the image in p and p' for frames k and k'.

Each point p and p' has its corresponding coordinates (x, y). For each point p to be tracked, let be a small window of interest $\mathcal{W}$ centred around it, and q a point located in $\mathcal{W}$. The motion undergone by $\mathcal{W}$ is modeled by a function $\delta(p, \mathbf{A})$ where the vector $\mathbf{A}$ describes the deformation of the window from one frame to another. In that manner, the point P at time k' is located at $p' = \delta(p, \mathbf{A})$.

The tracking procedure consists in computing the parameters $\mathbf{A}$ that minimize the following error function

$$\epsilon(\mathbf{A}) = \sum_{q \in \mathcal{W}} \left(\gamma(q, q')\, \epsilon_D(q, \mathbf{A}) + (1 - \gamma(q, q'))\, \epsilon_I(q, \mathbf{A})\right)^2 \tag{7}$$

with:

$$\epsilon_D(q, \mathbf{A}) = \|D_k(q) - D_{k'}(\delta(q, \mathbf{A}))\| \tag{8}$$

$$\epsilon_I(q, \mathbf{A}) = \|I_k(q) - I_{k'}(\delta(q, \mathbf{A}))\| \tag{9}$$

In addition, $\gamma(p, p')$ is the geometric mean of the relevance coefficients for each of the points compared:

$$\gamma(p, p') = \sqrt{\beta_k(p)\, \beta_{k'}(p')} \tag{10}$$

After a Taylor expansion for $D_{k'}(\delta(q, \mathbf{A}))$ and $I_{k'}(\delta(q, \mathbf{A}))$ and keeping only the first order coefficients, it yields the following approximation:

$$D_{k'}(\delta(q, \mathbf{A})) = D_{k'}\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) + \mathbf{G}_D\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) J_\delta^{\hat{\mathbf{A}}} \Delta\mathbf{A} \tag{11}$$

$$I_{k'}(\delta(q, \mathbf{A})) = I_{k'}\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) + \mathbf{G}_I\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) J_\delta^{\hat{\mathbf{A}}} \Delta\mathbf{A} \tag{12}$$

where $\mathbf{G}_D$ and $\mathbf{G}_I$ are the Jacobian matrices of $D_{k'}$ and $I_{k'}$ calculated for both directions x and y. Working with equations (11),(12) and (7), it finally results in the following linearized system

$$\left(\sum_{q \in \mathcal{W}} \mathbf{V}_C \mathbf{V}_C^T\right) \Delta A = \sum_{q \in \mathcal{W}} \gamma \Delta_D^k \mathbf{V}_D + (1 - \gamma)\, \Delta_I^k \mathbf{V}_I$$

with:

$$\Delta_D^k = D_k(q) - D_{k'}\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) \tag{13}$$

$$\Delta_I^k = I_k(q) - I_{k'}\left(\delta\left(q, \hat{\mathbf{A}}\right)\right) \tag{14}$$

The vectors $\mathbf{V}_D$ and $\mathbf{V}_I$ are defined for an affine motion model as

$$\mathbf{V}_D = \begin{bmatrix} g_x^D & g_y^D & xg_x^D & xg_y^D & yg_x^D & yg_y^D \end{bmatrix}^T \tag{15}$$

$$\mathbf{V}_I = \begin{bmatrix} g_x^I & g_y^I & xg_x^I & xg_y^I & yg_x^I & yg_y^I \end{bmatrix}^T \tag{16}$$

where g_x^D, g_y^D and g_x^I, g_y^I are respectively the invariant chrominance and the luminance image gradients. Vector $\mathbf{V}_C$ used for the calculation of the Hessian matrix is defined as

$$\mathbf{V}_C = \begin{bmatrix} g_x & g_y & xg_x & xg_y & yg_x & yg_y \end{bmatrix}^T$$

where the gradients g_x and g_y are defined as a combination of the chrominance and luminance gradients:

$$g_x = \left[\gamma g_x^D + (1 - \gamma)\, g_x^I\right] \tag{17}$$

$$g_y = \left[\gamma g_y^D + (1 - \gamma)\, g_y^I\right] \tag{18}$$

The forward-backward error [12] is employed for validating the tracking. This method relies on the assumption that correct tracking should be independent of the direction of time-flow. When a feature is occluded or when the tracker fails, it outputs a random position that almost certainly does not correspond with the output position obtained while tracking on the reverse sense. By measuring these discrepancies it is possible to immediately reject unreliable features and eliminate feature drift.

A point is lost or rejected when some of these situations occurs: 1) the point is located outside the image; 2) its convergence residue is greater than a threshold (we considered a mean difference of $\epsilon_D < 15$ and $\epsilon_I < 15$ in (11) and (12), therefore $\epsilon(\mathbf{A}) < 15 \times n$ in (19) where n is the number of points in $\mathcal{W}$); 3) its forward-backward error is greater than a threshold (we considered a maximum distance of 1.5px); 4) its Hessian matrix is not invertible and 5) it requires to much iterations to converge.

The tracking adaptive to saturation will be compared to the classical color tracking using the three components c_1, c_2 and c_3 (RGB, rgb or HSV), *i.e* by solving the following cost function:

$$\epsilon(\mathbf{A}) = \sum_{q \in \mathcal{W}} \sum_{i \in c_1, c_2, c_3} \left(D_k^i(q) - D_k^i(\delta(q, \mathbf{A}))\right)^2 \tag{19}$$

The solving is based on the similar stages as in the previous method.

IV. IMPLEMENTATION DETAILS

The implementation has made use of the *Inverse Compositional* algorithm as detailed in [13], calculating the Hessian matrix only for the first iteration. As stated at the beginning of the paper, the procedure follows a multiple scale approach, starting with a coarse resolution, and refining the results

ascending in the pyramid as in [8]. Three levels are considered in the pyramid. The implementation takes advantage of the SIMD (Single Instruction Multiple Data) instructions either by processing multiple pixels on a single color channel or by processing multiple color components of a single pixel simultaneously. For this purpose it is better to have data aligned in memory: for 32 bit wide floating point data, we can process 4 elements at the same time in 128 bits wide registers. Therefore, the initial RGB— images have been converted to $RGBX$ so each pixel occupies 128 bits in memory instead of 96 bits. Except for the simple tracking method based on the minimization of 19 the extra memory is not a waste since it can be used to store and additional channel relevance coefficient required for the computation of γ in equation 10.

After color conversion when necessary, the feature points are detected by the Harris operator dedicated to color images. Then the color channels are loaded in the tracking pyramid.

Note that when processing multiple components at the same time, the colorspace HSV cannot directly take advantage of the SIMD instructions with the original order of allocation of its components, because the Hue component is an angular quantity and requires modular arithmetic, obviously this has a negative impact on the performance of the algorithm, which will require more complex processing.

V. Experiments

A. Experimental Setup

In order to determine which colorspace can track more points feature lists were obtained for each of the tested colorspaces and grouped on a single list. Each of these feature lists is built obtaining their gradients in vertical and horizontal directions (applying a Sobel Filter), measuring for each point the Harris operator response, applying a threshold that depends on the global maxima of this function. Finally, local maxima are selected. DiZenzo gradient [2] is employed in RGB colorspace as well as Carron2 [10] gradient for HSV.

Sequences are played forward and then in reverse order to verify if points come back to their initial location. Feature positions on the forward trajectory $T_f^k = (\mathbf{x}_t, \mathbf{x}_{t+1}, \cdots, \mathbf{x}_{t+k})$, where k is the number of frames, are compared with their corresponding positions on the backward trajectory $T_b^k = (\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_{t+1}, \cdots, \hat{\mathbf{x}}_{t+k})$. If the calculated distances are less than a defined distance $d_{fb} = 1.5px$ the features are considered as tracked correctly.

For the pyramidal tracking we use the exactly the same parameters (window sizes, number of levels,etc) as the ones described in [8].

B. Experiment Results

Five image sequences were considered during experiments, the first two sequences were captured indoors in a controlled illumination environment, 1) *Basketball* [14] has stable illumination conditions, it exhibits colourful and textured matte objects that are somehow easy to follow. 2)*Cardgame* [4] shows a colourful and textured box with a little bit of specular reflections in its borders, it suffers a strong illumination direction change during time. 3) *Pedestrian 2* [15] is a low-resolution outdoors sequence experimenting a relatively strong camera movement, it displays highly and low-saturated objects. 4) *Road*[3] is also an outdoor sequence recorded at very adverse illumination conditions, camera moves in a random fashion harming the stability of tracking, and finally, 5) *Dtneu_schnee*[4] is a traffic scene recorded during a snow fall, coloured objects appear low-saturated, this sequence gives the opportunity to test the tolerance to noise.

Fig. 2. Optical flow detected with $L1P$ at *Basketball* sequence second frame. Displacement vectors in yellow are displayed magnified.

The optical flow detected from the first to the second frame of *Basketball* sequence is represented in Fig.2 with magnified yellow vectors, tracking was made with the $L1P$ model. The flow appears quite coherent specially in textured and coloured regions. The opposite happens on the upper-left corner, where there is no texture nor coloured regions. Those features detected by the $L1$ invariant will be removed by the Forward-Backward error because of their random behaviour.

Table I shows the number of features successfully tracked in each sequence and colorspace, the time required to track each feature is also shown. Indeed, the executing times are different from one method to the other because of the color conversion but also on the number of iterations required to converge.

In well-saturated sequences ($Basketball$ and $Pedestrian2$), the luminance I fails to track many of the features that other techniques are able to track employing color information. For example, in the *Cardgame* sequence which suffers an important change of illumination, Luminance is able to track many of the features applying the photometric normalisation, while RGB and HSV drastically fail to handle this, while they behave pretty well in many of the sequences.

The relatively high number of features successfully tracked with $L1$ in many sequences prove its tracking capability. As expected, this invariant fails in sequences where color saturation is weak. Tracking with $L1P$ (mixture of $max(r, g, b)$

[3]available on http://vasc.ri.cmu.edu/idb/html/jisct/index.html

[4]available on http://i21www.ira.uka.de/image_sequences/#taxi

Seq	Features	I	RGB	(r,g,b)	HSV	H	HP	L1	L1P
Basketball 07-14	2843	1147	1946	1598	1364	667	470	1814	1778
101_l6c 1-3	3179	1642	210	1266	54	696	130	1765	1400
Pedestrian 2 1-10	1061	251	301	386	358	100	336	320	324
Road 16-54	2825	0	37	0	59	0	0	0	14
dtneu_schnee 1-50	2772	2192	2033	63	1521	0	1624	31	2026
Tracking time/feature $[s]$		0.000571	0.001331	0.001557	0.001353	0.000517	0.001342	0.000511	0.001307

TABLE I
TRACKING RESULTS

and I) we are able to handle this situation thanks to the adaptive capability of this technique. $L1P$ maintains the tracking performance of the $L1$ invariant in high-saturated sequences and significantly improves the results for the low-saturated ones. Tracking with (r,g,b) does not improve at all the results obtained with $L1$ while incrementing significantly the calculations.

$L1$ and $L1P$ behave better than their equivalents for the Hue invariant, being able to track more points for all the tested sequences (except the peculiar case of the *Pedestrian 2* sequence), that is probably due to the noisy character of the Hue component, mentioned in II-A.

A qualitative analysis of the experimental results can be done with the help of colormaps as those built for the *Basketball* sequence and displayed in Fig.4. Points are assigned a color from a scale that goes from red to green denoting the frame where features were lost. Red for features that were lost on the first frames and green for features correctly tracked during the whole sequence. $L1P$ keeps the good response of $L1$ including areas that neither I nor $L1$ were able to track (like the ear of the man on the right).

Fig.5 display the colormaps built for the *Pedestrian 2* sequence, Luminance I succeeds to track the features over the car on the left corner, mostly because this is the most textured region of the sequence. On the other side, it fails to track the people walking at the center of the image that can be regarded as color blobs. Almost any colorspace (except Hue) succeeds to track correctly both the car (luminance region) and the people walking at the center.

It is also noticeable in Fig.5 that $L1P$ tracked less color features than $L1$ for this sequence. The influence of the Luminance component, the same that improves the robustness to track low-saturated regions, reduces at some proportion the ability to track color features. It is possible to reduce this phenomenon by finding more appropriate sigmoid function parameters: S_0 and u_β.

Execution times scale-up proportionally to the number of components exploited during tracking. Single component tracking techniques like I, Hue and $L1$ require less time than multiple component techniques. Despite its simplicity in terms of calculations, Luminance I requires almost the same execution time as Hue and $L1$, because of the photometric normalisation. Multiple component techniques require more execution time, a price to pay for the robustness and capability to track complementary features.

Fig.3 displays the mean forward-backward euclidean distances $d_E(x_t, \hat{x}_t)$ computed from all the correctly tracked

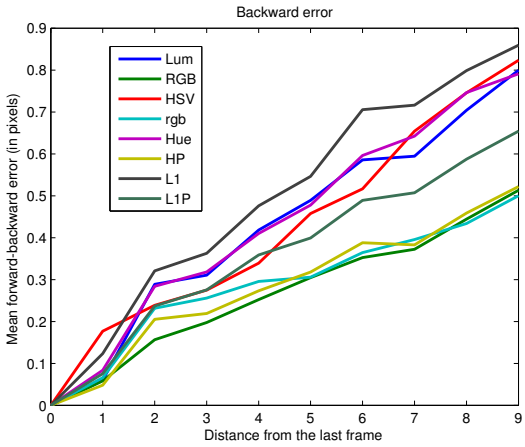

Fig. 3. Mean forward-backward euclidean distances $d_E(x_t, \hat{x}_t)$

points in each frame of the *Pedestrian 2* sequence. The error increases with the number of frames between x_t and $\hat{x}_t$ in the forward and backward trajectories, but remains lower than one pixel for each method. Note that the use of a single color component ($L1$, H or $L1$) instead of the whole colorspace (RGB, HSV or rgb) usually leads to a lower accuracy. Indeed the separability between colors can be lower due to the reduction of components. However, as written in Table I the computational costs are reduced with one component and the number of points correctly tracked higher in some difficult sequences, when illuminations changes occur. Mixed approaches $L1P$ and HP provide a better accuracy compared to $L1$ and H respectively while maintaining the invariance properties needed for challenging tracking situations.

VI. CONCLUSION

This article has exposed a feature points tracking procedure which combines luminance and color invariants for matte surfaces with the help of a relevance coefficient. The use of the color invariant $L1 = max(r,g,b)$ has been proposed. It requires less calculations, it is easier to parallelize and it provides better tracking results in terms of robustness. Experimental results have shown that this combination helps to distinguish more features and makes also possible to track them better even if feature patches change in orientation, expanding the regions and conditions suitable for tracking.

The proposed method is somehow dependent on the sigmoid parameters, this is a subject to be explored on future

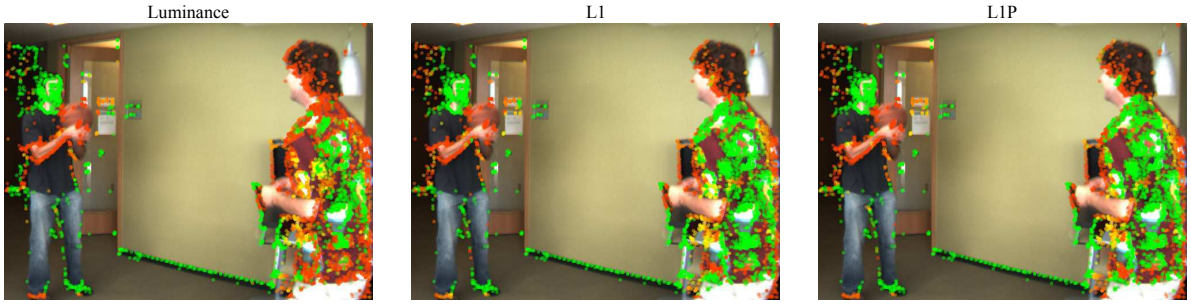

Fig. 4. Tracking colormaps with Luminance I, $L1$ ($max(r, g, b)$) and $L1P$ (mixed I and $L1$) for the *Basketball* sequence. Points are painted in a color-scale that goes from red to green representing the time the feature was lost

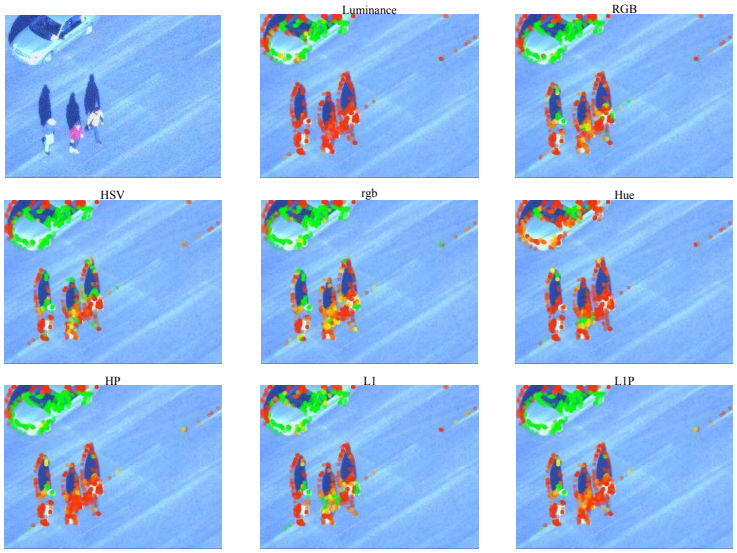

Fig. 5. Tracking colormaps for the Pedestrian 2 sequence for all the tested colorspaces.

works, an on-line estimation of these parameters could benefit significantly the speed of the algorithm.

VII. Acknowledgements

This research is supported by the European Project ITEA2 SPY[5].

References

[1] J. Stoettinger, A. Hanbury, N. Sebe, and T. Gevers, "Do colour interest points improve image retrieval?" in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, 16 2007-oct. 19 2007, pp. I–169–I–172.

[2] V. G. Philippe and R. Deriche, "Evaluation de détecteurs de points d'intêret pour la couleur evaluation of point of interest detectors for color images," 2000.

[3] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 3, pp. 453–464, 1999. [Online]. Available: http://www.science.uva.nl/research/publications/1999/GeversIJPRAI1999

[4] J. Geusebroek, G. Burghouts, and A. Smeulders, "The Amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[5] M. Gouiffes, "Tracking by combining photometric normalization and color invariants according to their relevance," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 6, 16 2007-oct. 19 2007, pp. VI–145–VI–148.

[6] T. Kanade and B. Lucas, "An Iterative Image Registration Technique with an Application to Stereo Vision." 1981.

[7] C. Tomasi and T. Kanade, *Detection and tracking of point features*. Citeseer, 1991.

[8] J. Bouguet *et al.*, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, vol. 3, 1999.

[9] S. A. Shafer, "Color," G. E. Healey, S. A. Shafer, and L. B. Wolff, Eds. , USA: Jones and Bartlett Publishers, Inc., 1992, ch. Using color to separate reflection components, pp. 43–51. [Online]. Available: http://portal.acm.org/citation.cfm?id=136809.136817

[10] T. Carron, "Segmentation d'images couleur dans la base teinte luminance saturation : approche numérique et symbolique," Thesis, Université de la Savoie, France, December 1995.

[11] T. Tommasini, A. Fusiello, V. Roberto, and E. Trucco, "Robust feature tracking in underwater video sequences," in *OCEANS'98 Conference Proceedings*, vol. 1. IEEE, 1998, pp. 46–50.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *2010 International Conference on Pattern Recognition*. IEEE, 2010, pp. 2756–2759.

[13] S. Baker, R. Patil, G. Cheung, and I. Matthews, "Lucas-kanade 20 years on: Part 1," Tech. Rep., 2004.

[14] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[15] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," *Conference on Computer Vision and Pattern Recognition*, 2010.

[5] Surveillance imProved sYstem http://www.ppsl.asso.fr/spy.php