Motion estimation through efficient matching of a reduced number of reliable singular points

Carlos R. del-Blanco[a], Fernando Jaureguizar[b], Luis Salgado[c] and Narciso García[d]

Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, 28040, Madrid, Spain

ABSTRACT

Motion estimation in video sequences is a classical intensive computational task that is required for a wide range of applications. Many different methods have been proposed to reduce the computational complexity, but the achieved reduction is not enough to allow real time operation in a non-specialized hardware. In this paper an efficient selection of singular points for fast matching between consecutive images is presented, which allows to achieve real time operation. The selection of singular points lies in finding the image points that are robust to the noise and the aperture problem. This is accomplished by imposing restrictions related to the gradient magnitude and the cornerness. The neighborhood of each singular point is characterized by a complex descriptor vector, which presents a high robustness to illumination changes and small variations in the 3D camera viewpoint. The matching between singular points of consecutive images is performed by maximizing a similarity measure based on the previous descriptor vector. The set of correspondences yields a sparse motion vector field that accurately outlines the image motion. In order to demonstrate the efficiency of this approach, a video stabilization application has been developed, which uses the sparse motion vector field as input. Excellent results have been obtained in synthetic and real sequences, demonstrating the efficiency of the proposed motion estimation technique.

Keywords: Real Time, Motion Estimation, Singular Point, Noise Adaptive, Point descriptor, Point Matching

1. INTRODUCTION

A number of different techniques has been proposed for the estimation of motion in video sequences. The techniques[1–3] that yield a dense motion vector field demand a high computational power that can only be reached by means of specialized hardware. But the use of specialized hardware is expensive and needs a specific implementation. On the other hand, block-matching techniques generate a non-dense motion vector field, in which each motion vector represents the motion of a rectangular region in the image. A large reduction in the computational burden is achieved combining these block techniques with gradient minimization methods,[4,5] used to find the best correspondence of each block. However, the quality of the estimation can be seriously affected by the aperture problem.[6] Feature oriented techniques[7–9] overcome this problem computing only the motion at points with a distinguishing feature such as, for example, the cornerness. Nevertheless, illumination changes and little variations of 3D viewpoint can still produce a lot of erroneous motion vectors, which can not be easily detected.

In this paper a real-time motion estimation technique is proposed, which is robust to noise, the aperture problem, illumination changes and small variations of 3D viewpoint. Real-time processing is achieved through the combination of two strategies: restricting the motion estimation to a reduced set of singular points and using properly look-up tables to avoid the computation of complex mathematical operations. The selection of singular points is carried out by imposing three different restrictions: the first one selects the points with a gradient magnitude response greater than the image noise level, obtaining SP_{gm}. The second one rejects, among those in SP_{gm}, the points with low cornerness, i.e. points located in straight edges. The resulting set, SP_{cor}, is composed by those points that globally stand out by their gradient magnitude and cornerness. The final selection, SP_{fin}, is obtained by applying a non-maximal suppression algorithm in the cornerness space, what removes points that are not very significative in comparison with their neighborhood. Points in SP_{fin} are the most reliable cues to estimate the image motion, since image points that are specially affected by the noise and the aperture

[a]cda@gti.ssr.upm.es, [b]fjn@gti.ssr.upm.es, [c]L.Salgado@gti.ssr.upm.es, [d]narciso@gti.ssr.upm.es; www.gti.ssr.upm.es

Real-Time Image Processing 2008, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6811, 68110N, © 2008 SPIE-IS&T · 0277-786X/08/$18

problem have been discarded. Each singular point in SP_{fin} is characterized by a sophisticated descriptor which is particularly robust to illuminations changes and small variations in the 3D viewpoint. In order to compute the descriptor, an array of gradient phase histograms in the neighborhood of each singular point is calculated. The concatenation of the bins of all histograms forms the descriptor DV_{hist}, which is normalized to the unit length to minimized the influence of illumination changes, resulting in DV_{fin}. Singular points of consecutive images are matched using the Euclidean distance between the corresponding descriptor as similarity measure. Erroneous correspondences are discarded comparing its similarity measure with the one related to the second best correspondence. The set of matchings yields an accurate and sparse motion vector field, $SMVF$, that represents the motion in the image. In order to demonstrate the efficiency of the proposed motion estimation technique, a video stabilization application based on the computed $SMVF$ is developed.

The organization of the paper is as follows: in Sec. 2 the strategy to select reliable singular points for motion estimation is presented. The descriptor used to characterize each singular point is explained in Sec. 3. Section 4 describes the matching between singular points. The look-up tables used to reduce the processing time are explained in Sec. 5, and Sec. 6 presents a video stabilization application based on the proposed motion estimation technique. Results about processing times and the quality of the motion estimation are shown in Sec. 7. Finally, the conclusions are presented in Sec. 8.

2. SELECTION OF SINGULAR POINTS

A chain of restrictions is imposed to the image points to select the best ones in order to compute the image motion. The first restriction selects the points whose gradient magnitude value is above a noise-adaptive threshold (Th_{gm}), obtaining SP_{gm}. The image gradient magnitude, $\|\nabla I(x,y)\| = \sqrt{I_x^2 + I_y^2}$, is calculated using a look-up table (described in Sec. 5) to reduce the processing time. The gradient magnitude image is thresholded by Th_{gm}, which is computed as a function of the image noise distribution. Assuming image noise as Gaussian, the corresponding image gradient magnitude has a Rayleigh distribution[10] (see Eq. (1)).

$$R(\|\nabla I(x,y)\|) = \frac{\|\nabla I(x,y)\|}{\sigma^2} e^{-\frac{\|\nabla I(x,y)\|^2}{2\sigma^2}} \tag{1}$$

Then, Th_{gm} is computed as a function of the Rayleigh parameter, σ, by means of Eq. (2),

$$Th_{gm} = \sigma\sqrt{-2\ln(P_f)} \tag{2}$$

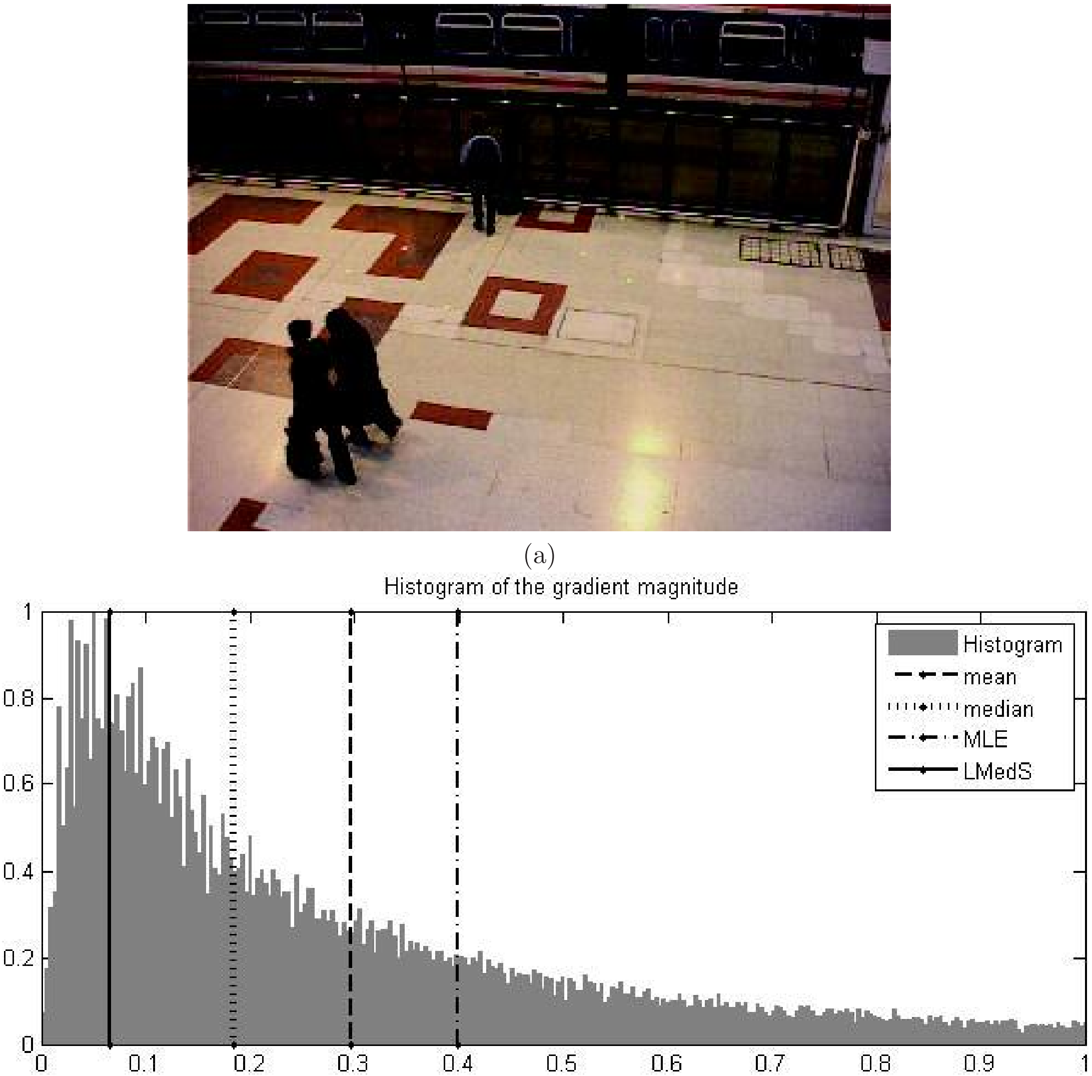
where P_f is the acceptable proportion of low reliable singular points due to noise peaks. In real images the estimation of σ is a hard problem, since the gradient magnitude distribution is a combination of different sources of noise and edges that contaminate the expected Rayleigh distribution. The Rosin's approach[11] based on the Least Median Squares algorithm (LMedS) computes a good approximation of σ by means of the Eq. (3),

$$\hat{\sigma} = 0.968 \min_j(\text{median}_{\forall i}(H(i) - j)^2) \tag{3}$$

where $H(i)$ is the histogram count of gradient magnitude i, and 0.968 is a correction factor that relates the LMedS estimation of an uncontaminated Rayleigh distribution with σ. Figure 1(a) shows an intensity image and Fig. 1(b) its normalized histogram of the gradient magnitude along with several estimations obtained by different algorithms. As can be observed, the LMedS algorithm produces the best estimation in comparison with the mean (Eq. (4)), the median (Eq. (5)) and the maximum likelihood estimations (Eq. (6)),

$$\sigma = \sqrt{\frac{2}{\pi}} \cdot \overline{R(\|\nabla I(x,y)\|)} \tag{4}$$

$$\sigma = \frac{\text{median}(\|\nabla I(x,y)\|)}{\sqrt{\ln(4)}} \tag{5}$$



(a)

(b)

Figure 1. An original intensity image is shown in (a) and its normalized histogram of gradient magnitude in (b) along with the estimations obtained by the LMedS, mean, median and Maximum Likelihood Estimation algorithms (MLE).

$$\sigma = \sqrt{\frac{1}{2N} \cdot \sum^{N} \|\nabla I(x,y)\|^2} \tag{6}$$

where N is the total number of pixels in the image.

Figure 2 shows the points in SP_{gm}, resulting of thresholding the gradient magnitude image by Th_{gm}.

The points of SP_{gm} located along straight edges are not very reliable to estimate their motion since they are very sensitive to small amounts of noise. These points are characterized by a low cornerness response, what means that they have a large principal curvature along the edge but a small one in the perpendicular direction. The principal curvatures of a point are proportional to the eigenvalues of the Hessian matrix, $\mathbf{H}$, calculated in the location of the point. In order to reduce the computational cost the approach of Harris and Stephen[12] is adopted, which computes the ratio of the principal curvatures without explicitly calculating the eigenvalues. The

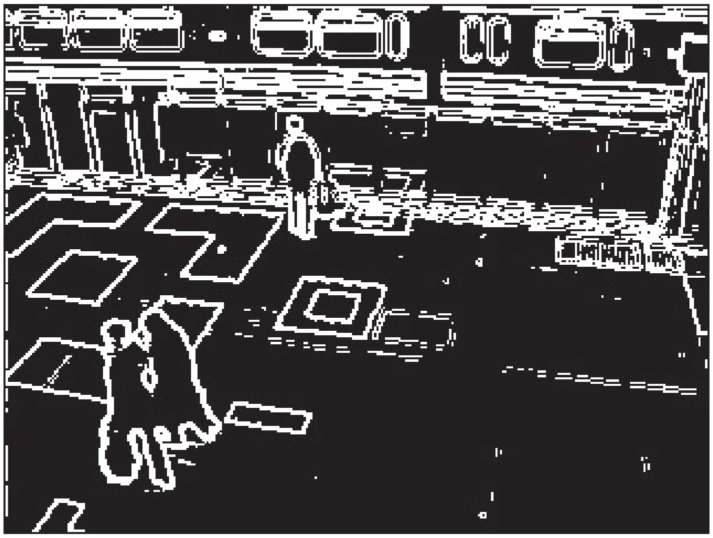


Figure 2. Singular points belonging to SP_{gm} (in white), for the original image presented in Fig 1(a).

trace (Tr) and the determinant (Det) of $\mathbf{H}$ are computed as in Eq. (7) and Eq. (8) respectively,

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} \tag{7}$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 \tag{8}$$

where D_{xx}, D_{yy} and D_{xy} are the elements of $\mathbf{H}$ (see Eq. (9)), which are calculated using a Sobel filter as derivative operator.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{9}$$

The ratio between the largest and smallest eigenvalue magnitudes, R_{eig}, is related to Tr and Det as shown in Eq. (10).

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(R_{eig}+1)^2}{R_{eig}} \tag{10}$$

The value $\frac{(R_{eig}+1)^2}{R_{eig}}$ is minimum when the two eigenvalues are equal (maximum cornerness) and it increases with R_{eig}. Therefore, Eq. (11) can be used to check that the cornerness of a point is larger than a threshold Th_{cor}.

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(R_{eig}+1)^2}{R_{eig}} = Th_{cor} \tag{11}$$

As result of applying the cornerness restriction, SP_{cor} is obtained, which is composed by the points that globally have the most significant values of gradient magnitude and cornerness. Nevertheless, several points lying in the same neighborhood will result little distinctive. Therefore, a non-maximal suppression in the cornerness space is applied to obtained the final selection of singular points, SP_{fin}, which locally selects the most significant points. Figure 3(a) and (b) respectively show the points in SP_{cor} and in SP_{fin} from the singular points presented in Figure 2.

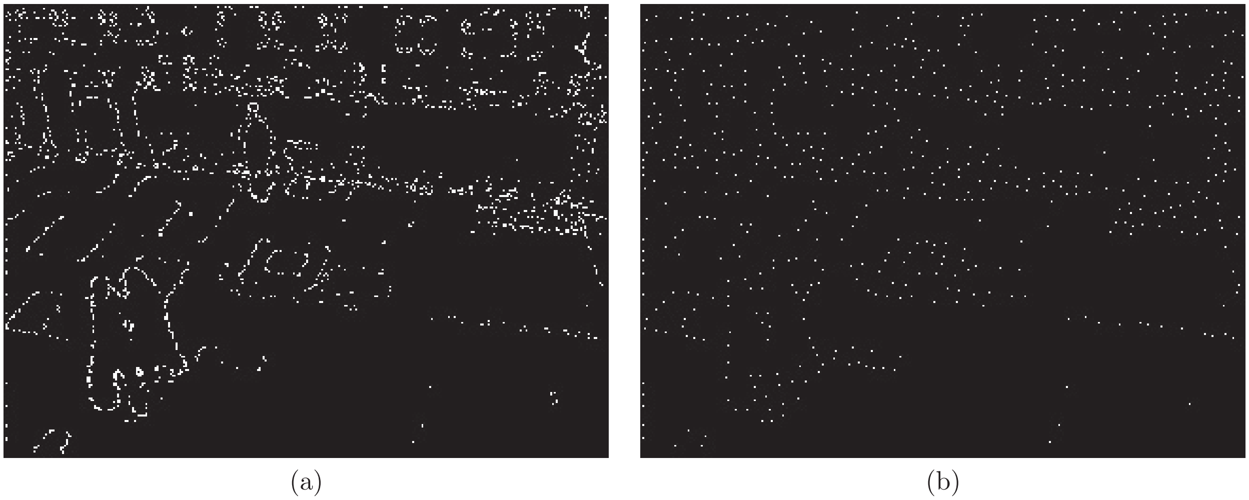


Figure 3. (a) shows the singular points belonging to SP_{cor}, and (b) the singular points in SP_{fin}.

3. DESCRIPTION OF SINGULAR POINTS

The descriptor used to characterize the neighborhood of each singular point must be very distinctive in order to match correctly the singular points between two consecutive images. In addition, it must be as robust as possible to variations produced by noise, rotations, changes in illumination, 3D viewpoint and non-rigid deformations. Classical descriptors based on a patch of intensity or gradient values are very sensitive to the aforementioned variations. On the other hand, the descriptors that compute the intensity or gradient histogram are more robust but their distinctiveness is significantly lower. Lowe[13–15] proposed a combination of both strategies that imitates the biological vision of human beings. Lowe's approach consists in allowing small shiftings in the localization of gradient values for computing a similarity measure, rather than computing it in precise localizations. In order to compute the Lowe's descriptor, the gradient magnitude and phase are calculated in the neighborhood of the singular point defined by a squared window named description window. This is carried out by means of look-up tables, as it is explained in Sec. 5, to reduce the computational cost.

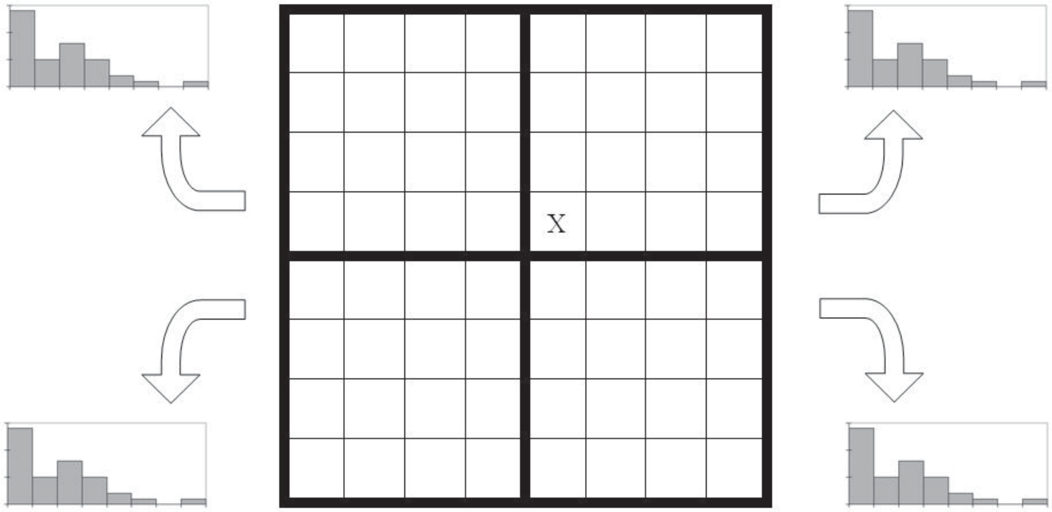


Figure 4. Computation of an array of orientation histograms inside the descriptor window, used to generate the descriptor of a singular point market as X, for $N_h = 2$, $N_p = 4$ and $N_d = 8$.

The gradient magnitude is smoothed by a Gaussian function centered in the singular point and with a standard deviation equal to one half the width of the descriptor window. This Gaussian weighting gives less emphasis to gradient values that are far from the center of the singular point, avoiding sudden changes in the descriptor due to small changes in the position of the descriptor window. The descriptor window is divided into $N_h \times N_h$ squared regions composed by $N_p \times N_p$ pixels each one. Then, an orientation histogram is computed in each region. An orientation histogram is defined as a gradient phase histogram composed by N_d bins, where the bin contributions are the gradient magnitude values. Figure 4 shows an example of an array of orientation histograms computed in the descriptor window. The strategy of splitting the neighborhood into several histograms allows for small positional shifts, since the gradient magnitude of a pixel contributes to the same histogram as long as its location keeps inside of the same squared region. However, the descriptor can still abruptly change if the positional shift of the gradient magnitude modifies the histogram contribution, or if a smooth variation of the gradient phase changes the phase bin contribution. A trilinear interpolation solves this problem by distributing each gradient magnitude value into adjacent orientation histograms. This means that each orientation histogram entry is multiplied by three different weights: w_h, w_v and w_p, which are computed as in Eqs. (12), (13) and (14),

$$w_h = 1 - (\frac{x}{N_h} - \lfloor \frac{x}{N_h} \rfloor) \tag{12}$$

$$w_v = 1 - (\frac{y}{N_h} - \lfloor \frac{y}{N_h} \rfloor) \tag{13}$$

$$w_p = 1 - (\frac{\theta}{N_d} - \lfloor \frac{\theta}{N_d} \rfloor) \tag{14}$$

where x and y are the coordinates of the gradient magnitude location; θ is the gradient phase in degrees; and $\lfloor \rfloor$ means the nearest lowest integer.

The descriptor, DV_{hist}, is formed by concatenating the phase bins of all orientation histograms by raws, as in Eq. (15). The resulting descriptor has a length of: $L_{DV} = N_h \times N_h \times N_d$,

$$DV_{hist} = [Hist_{(1,1)}(1), Hist_{(1,1)}(2), ...Hist_{(r,s)}(i), .., Hist_{(N_h,N_h)}(N_d)];\ 0 < i < N_d,\ 0 < j < N_h \tag{15}$$

where $Hist_{r,s}(i)$ is the i^{th} bin of the orientation histogram of coordinates (r, s).

DV_{hist} is invariant to brightness changes, which result from a constant added to each image pixel, as it uses gradient values rather than intensity values. However, contrast changes, in which each pixel value is multiplied by the same constant, affects the value of the descriptor vector. This is overcome by normalizing DV_{hist} to unit length, which makes the descriptor invariant to affine illumination changes, but it is not invariant to non-linear illumination changes, such as camera saturation and puntual illumination sources. These non-linear illumination changes can cause large changes in the gradient magnitude, but they do not affect so much to the gradient phase. Based on this fact, each component of the descriptor is limited to a maximum value, Th_{illu}, thus reducing the influence of the gradient magnitude while giving more emphasis to the gradient phase. Normalizing again the previous result, the final version of the descriptor, DV_{fin}, is obtained.

4. MATCHING OF SINGULAR POINTS

Singular points of consecutive images, SP^n_{fin} and SP^{n+1}_{fin}, are matched by means of the Euclidean distance computed in the descriptor domain, which is used as similarity measure. Camera frame rate is assumed to be high enough to consider a slow motion between frames. Therefore, the matching can be restricted to singular points whose distance to each other is less than a threshold, Th_{dist}. This restriction not only reduces the computational cost but also improves the matching accuracy, since the descriptor vector of the singular point only must be enough distinctive in its neighborhood, rather than in the whole image. Matching between a singular point $\vec{p_i}$ belonging to SP^n_{fin} and the set of singular points of SP^{n+1}_{fin} is computed as in Eq. (16),

$$\text{Match}(\vec{p_i} \in SP^n_{fin}) = \min_{\vec{p_j} \in SP^{n+1}_{p_j}} \| DV_{fin}(\vec{p_i}) - DV_{fin}(\vec{p_j}) \| \tag{16}$$

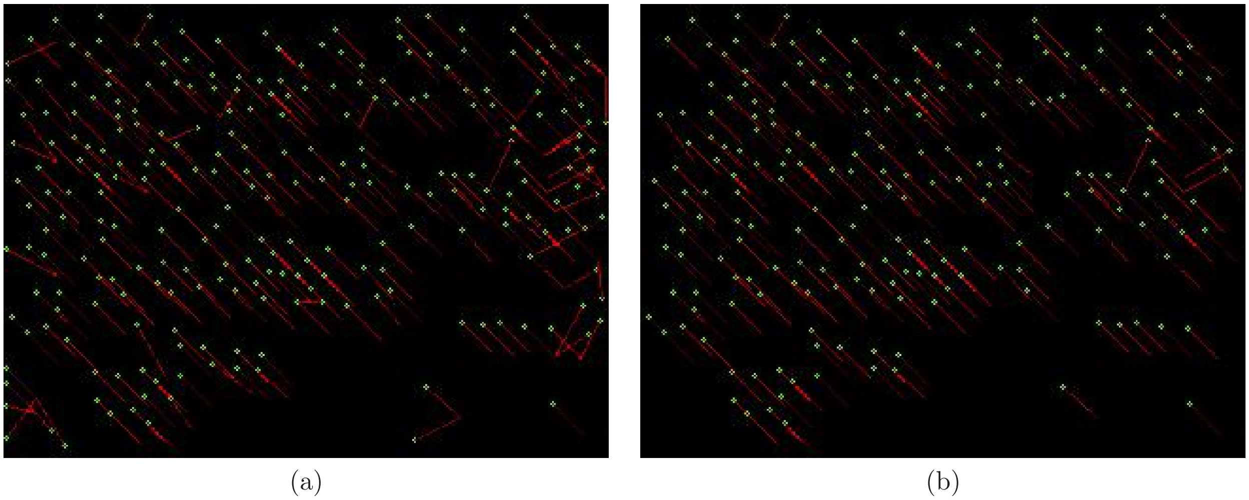


Figure 5. (a) and (b) respectively shows the $SMVF$ between two images before and after verifying the second best matching condition.

where $\| \ \|$ is the Euclidean distance and $SP_{p_j}^{n+1} = \{p_j \in SP_{fin}^{n+1} \mid \|\vec{p_i} - \vec{p_j}\| < Th_{dist}\}$, i.e. the set of singular points of SP_{fin}^{n+1} whose distante to $\vec{p_i}$ is less than Th_{dist}.

Distances in the descriptor domain between the first and second best matching of a point are compared to determine the reliability of each correspondence. A correct correspondence must satisfied the condition $d_{FM} > 0.51 d_{SM}$, where d_{FM} and d_{SM} are respectively the Euclidean distances between the corresponding descriptor vectors of the first and second best matching. Correspondences that fulfill the previous condition represents a motion vector of the sparse motion vector field, $SMVF$. Figures 5(a) and (b) respectively show the $SMVF$ between two images before and after applying the condition of correct correspondence. In this example, the image in instant $n + 1$ is a noisy and translated version of the first one in instant n, therefore the outliers are easily identified. Figure 5(a) contains several outliers near to the margins, which have been discarded in Fig. 5(b).

5. LOOK-UP TABLES

The proposed motion estimation strategy uses three different look-up tables in order to reduce the computational cost. A look-up table is used to calculate the gradient magnitude. The table stores the precalculated values of the gradient magnitude, according to the ranges of the horizontal and vertical image gradient. Then, to compute a specific gradient magnitude value, the horizontal and vertical gradient values are used as indexes to the table. The computation of the gradient phase is performed in a similar way. In this case, the table contains gradient phase values.

The last look-up table is used to compute the trilinear interpolation related to the descriptor of a singular point. The table stores weighting factors and bin contributions for the phase histograms. The table indexes are the pixel coordinates belonging to the singular point neighborhood, and the gradient phase value of the pixel.

Processing times for computing the gradient magnitude, the gradient phase and the descriptor have been measured using and without using look-up tables. The time measures have been performed with a 2.00 GHz T2500 Intel Core Duo Mobile PC, using a 320×240 pixel image in which 211 singular points have been detected. Table 1 shows the processing times in milliseconds, in which the measure corresponding to the descriptor represents the total time for computing all 211 singular points. A reduction in the processing time can be observed in all cases, specially in the gradient phase computation.

Table 1. Processing times in milliseconds of the gradient magnitude, the gradient phase and the descriptor vector using and without using a look-up table.

	With Look-up Table (ms)	Without Look-up Table (ms)
Gradient Magnitude	1.2	3.5
Gradient Phase	1.2	10.7
Descriptor	13.2	19.4

6. VIDEO STABILIZATION APPLICATION

The proposed real-time motion estimation technique can be used in a wide range of applications that do not require a dense motion vector field. In this section its application for video stabilization is described to show its performance and accuracy. This application removes the camera ego-motion, i.e. the motion induced in the image by the own camera motion, that typically arises using a hand-held camera or due to the instability of the platform where it is placed. This is achieved by using $SMVF$ to estimate the best affine transformation, AT_B, that describes the camera motion. The affine transformation is described by Eq. (17),

$$\begin{bmatrix} x^n \\ y^n \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^{n+1} \\ y^{n+1} \\ 1 \end{bmatrix} \tag{17}$$

where a, b, d and e are the parameters of a linear transformation; c and f are respectively horizontal and vertical translation; and, x^n, y^n, x^{n+1}, y^{n+1} are the coordinates of a pixel in the previous image, I^n, and in the current image I^{n+1}.

The parameters of AT are estimated using RANSAC, a robust iterative parameter estimation technique that rejects outlier motion vectors (due to erroneous motion vectors or moving objects). RANSAC[16,17] randomly selects N_{mv} motion vectors from $SMVF$. Then, $\widehat{AT}$ is estimated from them through the Least Mean Squares algorithm (LMS). The number of inliers in $SMVF$ with regard to $\widehat{AT}$ is calculated by thresholding the corresponding residuals, as it is shown in Eq (18),

$$N_{in} = \{\vec{mv}_i \in SMVF \mid \|\vec{p}_{ori} - AT_B \cdot \vec{p}_{ext}\| < Th_{in}\} \tag{18}$$

where $\vec{p}_{ori}$ and $\vec{p}_{ext}$ are respectively the origin and extremum of the motion vector $\vec{mv}_i$; and Th_{in} is the maximum residual distance allowed, whose value has been heuristically fixed to 4.

If the number of inliers is equal or greater than 50% of the total number of motion vectors in $SMVF$ (which represents the breakdown point of the RANSAC algorithm), the $\widehat{AT}$ is chosen as AT_B, i.e. the best affine transformation. Otherwise, another set of N_{mv} motion vectors is selected, and the entire process is repeated up to a maximum of N_{it} times. This maximum number of iterations is calculated as in Eq. (19),

$$N_{it} = \frac{\log(1 - P_s)}{\log[1 - (1 - \varepsilon^{N_{mv}})]} \tag{19}$$

which ensures that al least a set of N_{mv} motion vectors is free of outliers with a probability P_s, given a maximum fraction of outliers ε.

If after N_{it} iterations, no $\widehat{AT}$ has passed the condition of minimum number of inliers, the image can not be stabilized, what occurs in cases of rarely large displacement of the camera. Under normal conditions AT_B is obtained, which is used to compute the compensated location of each pixel. The stabilized image is obtained by applying a bilinear interpolation over the non-integers coordinates resulting of applying the affine transformation. Figure 6 shows the Peak Signal to Noise Ratio (PSNR) of a stabilized and an unstabilized sequence, respectively represented by a solid line and a dashed line. The PSNR measures the quality of a compensated image compared to the original image, and its value is increased according to the quality of the compensation.

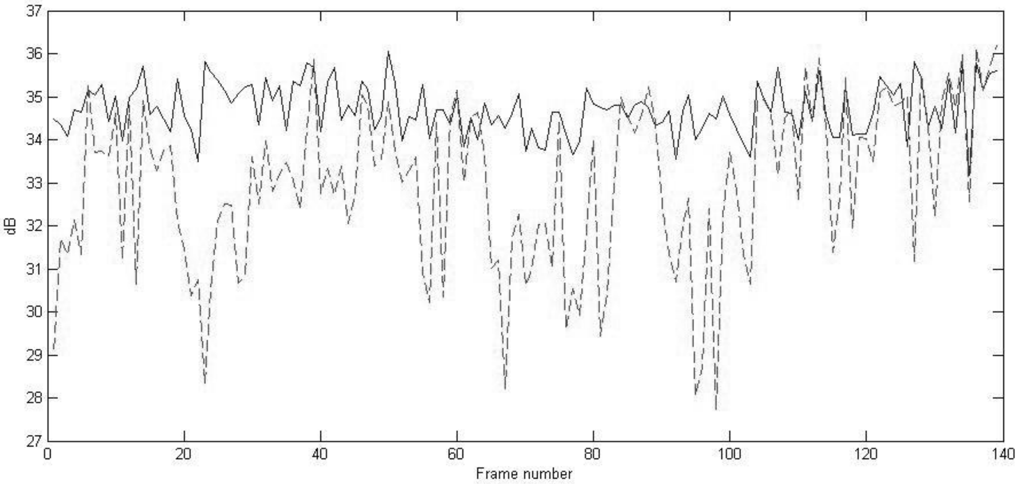


Figure 6. PSNR measures of the unstabilized (dashed line) and stabilized (solid line) video sequence.

As can be observed, the PSNR measure of the stabilized sequence is higher and more stable along the time than the unstabilized sequence. This demonstrates the high accuracy and performance of the video stabilization application in combination with the proposed real-time motion estimation technique.

7. RESULTS

Different types of experiments has been carried out to test the performance and efficiency of the proposed motion estimation technique and to find the descriptor parameters that maximizes the ratio of correct correspondences (defined as the quotient of the correct correspondences and the total number of correspondences) while minimize the computation time. Using these parameters, the robustness of the motion estimation technique to noise and rotation has been tested. Finally, the dependency between the number of detected singular points in an image and the processing time is shown. For the experiments have been used a 2.00 GHz T2500 Intel Core Duo Mobile PC with a set of 320 × 240 pixel images.

Table 2 shows the ratio of correct correspondences, R_{cc}, and the processing times for several combinations of the descriptor parameters N_h, N_p and N_d. These results have been obtained using an image and the same image translated 16 pixels in the horizontal and vertical axes. A Gaussian noise of $\mu_N = 0$ and $\sigma_N = 2$ has been added to both images. As it is expected, the processing time increases with the value of the descriptors parameters. This increment is more significative for N_h and N_p, since they control the size of the neighborhood of each singular point that is used to computed the descriptor. The ratio of correct correspondences also increases with the value of the descriptors parameters. Again, N_h and N_p are the most influential parameters, since the distinctiveness of the descriptor increases with the area of the neighborhood of the singular point. A good compromise between performance and computational cost is obtained by selecting $N_h = 4$, $N_p = 4$ and $N_d = 8$, which will be used in the rest of experiments.

The robustness to the noise has been measured by means of R_{cc}. The images used are the same as for Table 2, but adding different levels of Gaussian noise of mean $\mu_N = 0$ and standard deviation σ_N. Table 3 shows the obtained results. As it is expected, R_{cc} decreases as σ_N value increases. However, its value keeps high even for elevated values of noise.

Table 4 shows the robustness of the motion estimation technique to rotation changes, measured through R_{cc} and the total number of correspondences, N_{TC}. An image and the same image rotated θ degrees have been used, to which a Gaussian noise of $\mu_N = 0$ and $\sigma_N = 2$ has been added. As can be observed, R_{cc} is very stable with the rotation angle, while N_{TC} decreases. The conclusion is that the motion estimation technique is robust to rotation changes, but not invariant.

Table 2. Influence of the descriptor parameters N_h, N_p and N_d in the ratio of correct correspondences (R_{cc}) and in the processing time in milliseconds. The results have been obtained using an image and the same image translated 16 pixels in the horizontal and vertical axes.

N_h	N_p	N_d	R_{cc}	Processing time (ms)
2	4	8	0.84	12.9
4	4	8	0.89	25.8
6	4	8	0.92	48.9
8	4	8	0.93	84.9
4	2	8	0.85	15.6
4	6	8	0.91	46.6
4	8	8	0.92	78
4	4	4	0.87	24.1
4	4	6	0.88	24.7
4	4	10	0.89	25.8
4	4	12	0.91	26.6

Table 3. Robustness of the motion estimation technique to noise, measured by means of the ratio of correct correspondences (R_{cc}). A Gaussian noise of mean $\mu_N = 0$ and standard deviation σ_N has been added to the images.

σ_N	0	1	2	3	4	5	6	7	8	9	10
R_{cc}	0.95	0.93	0.89	0.86	0.86	0.81	0.8	0.77	0.74	0.72	0.71

Table 5 shows the dependency of the processing time with the image size and the number of singular points. A set of images of size 640 × 480 and 320 × 240 have been used. As it is expected, the processing time increases with both parameters. The obtained values demonstrate the applicability of the proposed motion estimation technique for real time requirements.

8. CONCLUSIONS

The presented motion estimation technique is able to compute an accurate sparse motion vector field in real time without the need of specialized hardware. Real time is accomplished by restricting the motion computation to a subset of image points, together with the incorporations of three look-up tables to effiently handle complex mathematical operations. The combination of both strategies allows obtaining processing rates of up to 40 frames per second (fps) for 320 × 240 pixel images and up to 20 fps for 640 × 480 pixel images, as shown the Sec.7.

The robustness of the motion estimation technique to noise, aperture problem, illumination changes and small variations in the 3D viewpoint yields sparse motion vector fields of high quality. In addition, the matching process evaluates the reliability of each correspondence, being able to discard erroneous motion vectors.

Table 4. Robustness of the motion estimation technique to image rotation (indicated by θ), measured through R_{cc} and N_{TC}.

θ	1	2	3	4	5	6	7	8	9	10
R_{cc}	0.71	0.72	0.73	0.75	0.73	0.73	0.73	0.72	0.72	0.71
N_{TC}	149	150	144	128	142	134	137	125	122	114

Table 5. Processing time in milliseconds as a function of the image size and the number of singular points.

Image size	Number of detected points	Processing time (ms)
640 × 480	891	99.4
640 × 480	733	86.5
640 × 480	610	75.9
640 × 480	522	68.2
640 × 480	438	61.6
640 × 480	377	57.1
640 × 480	323	55.2
640 × 480	285	51
640 × 480	250	48
640 × 480	218	46
640 × 480	196	44.6
320 × 240	572	74.4
320 × 240	416	50
320 × 240	323	37.2
320 × 240	259	29.6
320 × 240	221	25.2
320 × 240	171	21.9

The sparse motion vector field has been used satisfactory in an application of video stabilization, where the sparse motion vector field is used to estimate and compensate the camera ego-motion. Other areas of applicability are the detection of moving objects and mosaicking.

ACKNOWLEDGMENTS

This work has been partially supported by the Ministerio de Ciencia y Tecnología of the Spanish Government and FEDER/ERDF funds through project TIN2004-07860 (Medusa) and by the Comunidad de Madrid under project P-TIC-0223-0505 (Pro-Multidis).

REFERENCES

1. S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Comput. Surv.* **27(3)**, pp. 433–466, 1995.
2. J. L. Barron, S. S. Beauchemin, and D. J. Fleet, "On optical flow," *Proc. AIICSR*, pp. 3–14, 1994.
3. H. Spies and H. Scharr, "Accurate optical flow in noisy image sequences," *Proc. ICCV*, pp. 587–592, 2001.
4. M. Al-Mualla, C. Canagarajah, and D. Bull, "Reduced complexity motion estimation techniques: review and comparative study," *Proc. ICECS* **2**, pp. 607–610, 2003.
5. S. Mattoccia, F. Tombari, L. Stefano, and M. Pignoloni, "Efficient and optimal block matching for motion estimation," *Proc. ICIAP*, pp. 705–710, 2007.
6. S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambrigde, London, 1979 (seventh edition).
7. B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. IJCAI*, pp. 674–679, 1981.
8. J. Shi and C. Tomasi, "Good features to track," *Proc. CVPR*, pp. 593–600, 1994.

9. T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features track better," *Proc. CVPR*, pp. 178–183, 1998.
10. H. Voorhees and T. Poggio, "Detecting textons and texture boundaries in natural images," *Proc. ICCV*, pp. 250–258, 1987.
11. P. Rosin, "Edges: Saliency measures and automatic thresholding," *Machine Vision and Applications* **9(4)**, pp. 139–159, 1999.
12. C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. Fourth Alvey Vision Conference*, pp. 147–151, 1988.
13. D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Comp. Vision* **60**, pp. 91–110, 2004.
14. D. Lowe, "Local feature view clustering for 3d object recognition," *Proc. CVPR*, pp. 682–688, 2001.
15. D. Lowe, "Object recognition from local scale-invariant features," *Proc. ICCV*, pp. 1150–1157, 1999.
16. C. Stewart, "Robust parameter estimation in computer vision," *SIAM Reviews* **41(3)**, pp. 513–537, 1999.
17. P. Meer, C. Stewart, and D. Tyler, "Robust computer vision: an interdisciplinary challenge," *Computer Vision and Image Understanding* **78(1)**, pp. 1–7, 2000.