

# An Improved Methodology for Image Feature Matching

Vilar F. da Camara Neto

Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI) — Manaus, AM — Brazil

neto@dcc.ufmg.br

Mario Fernando M. Campos

Universidade Federal de Minas Gerais (UFMG) — Belo Horizonte, MG — Brazil

mario@dcc.ufmg.br

## Abstract

*This paper presents a novel methodology to perform matching between image points described by their respective features. Traditionally, such correspondences are determined by computing the similarity between descriptor vectors associated to each point which are obtained by invariant descriptors. Our methodology first obtains a coarse global registration among images, which constrains the correspondence space. Then, it analyzes the similarity among descriptors, thus reducing both the number and the severity of mismatches. The approach is sufficiently generic to be used with many feature descriptor methods. We present several experimental results that show significant increase in accuracy, number of successful matches, and execution time.*

## 1 Introduction

*Matching or correspondence* is the problem of establishing a biunivocal mapping between points of two images that correspond to the same point in the scene, and is a fundamental step to solve several other problems in Computer Vision, such as depth/disparity map estimation, multiple view geometry, scene mapping, egomotion determination, and object tracking. A typical approach to solve the image matching problem is to first select a set of points in each image that exhibit peculiar characteristics in its vicinity (edges, corners, derivative peaks, etc.); then pair the elements of these sets together, possibly discarding points for which no good matching was found.

It is highly desirable to ensure a low probability of generating false correspondences, as outliers can severely compromise other process down the line that rely on accurate matching results. For this reason, several methods were developed to assign a distinctive signature, or *descriptor*, to

each image feature during the selection phase. After the selection is made, a correspondence step is carried out by selecting pairs of features whose descriptors are most similar. However, mismatches still occur, mostly due to the criteria used for selecting pairs of features that are based on descriptor similarity alone. One unpleasant effect of this criteria is that mismatches frequently generates correspondences between seemingly random points of the two images, which can be disastrous if the results are the input of non-robust processes.

Ideally, a matching process should be able to generate the largest set of correct correspondences and their precise location on each image at the lowest computational cost possible, while avoiding the occurrence of incorrect correspondences (mismatches). This paper presents a methodology that fulfills the aforementioned requirements. The key idea is to analyze the consistency of several geometrical clues before blindly accepting a candidate match. This approach not only greatly reduces the occurrence of outliers, but also increases the number of matching pairs and takes less time to execute when compared to classical descriptor similarity criteria. It is also worth noting that our methodology is generally applicable to several types of feature descriptors. Experiments show that our approach consistently outperforms current matching criteria.

The rest of this paper is organized as follows: Section 2 gives an overview of the main issues behind feature descriptors and the matching process; Section 3 presents the proposed methodology, which is evaluated by several experiments whose results are shown in Section 4. Finally, Section 5 concludes with a discussion of the results and possible further directions.

## 2 Feature descriptors and matching

In Computer Vision, it is desirable that image matching be reliably carried out even under substantial geometric

and/or photometric deformations in image characteristics. In the last decade, several feature descriptors have been proposed that exhibit important qualities: invariance to affine transformations (translation, scaling, rotation), illumination changes, and robustness to perspective transformations, such as changes in the point-of-view. The SIFT algorithm [6, 7] is considered an important breakthrough since it provides a simple descriptor — an 128-element vector — that can be directly compared using the Euclidean distance (a relatively low-cost metric). In the following years other methods have been proposed, notably PCA-SIFT [5], GLOH [8] and SURF [1]. These methods provide descriptor vectors that are either faster to evaluate, more distinctive and/or have less elements (which reduces the comparison cost) when compared to SIFT.

The typical matching step for these algorithms can be described as follows:

- Given the set of the descriptors of all distinct features found in the two images,  $\mathcal{D}_q = \{\mathbf{d}_{q,1}, \dots, \mathbf{d}_{q,N_q}\}$  for  $q \in \{1, 2\}$ , where  $\mathbf{d}_{*,*}$  are vectors with  $K$  elements and  $N_q$  is the number of features detected in image  $q$ ;
- given a *distance function*,  $\text{dist} : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R}^+$  (possibly, but not necessarily, the Euclidean distance), where  $\mathbb{R}^+$  is the set of non-negative real numbers;
- for each descriptor  $\mathbf{d}_{1,i}$ , search for a  $\mathbf{d}_{2,j}$  that minimizes the distance  $\text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j})$ .

The output is a set of pairs  $\langle i, j \rangle$  of corresponding features. Formally, the set  $\mathcal{M}$  of all pairs can be described by:

$$\mathcal{M} = \left\{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j}) \rangle \right\} \quad \forall 1 \leq i \leq N_1. \quad (1)$$

This approach is obviously error-prone since *all* features from the first image will be matched to some feature in the second image, which will inevitably generate false correspondences. To overcome this undesirable effect, a typical solution is to accept a match only if the distance from  $\mathbf{d}_{1,i}$  to the second best match in  $\mathcal{D}_2$  is significantly larger. Formally, a pair  $\langle i, j \rangle$  is rejected if there exists a pair  $\langle i, k \rangle$ ,  $1 \leq k \leq N_2$  and  $k \neq j$ , for which  $\text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,k}) < \tau \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j})$  for a given “distinctiveness” threshold  $\tau \geq 1$ , or:

$$\mathcal{M} = \left\{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j}) \rangle \mid \forall k \neq j [\text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,k}) \geq \tau \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j})] \right\}. \quad (2)$$

(Eq. (1) represents the unconstrained case where  $\tau = 1$ ). However, this solution prevents matchings with repeated features, such as recurrent patterns or homogeneous textures. Ideally, matchings should be searched for all rich features, no matter how many times they appear duplicated in the images.

A better approach would be to rely on other constraints while keeping  $\tau = 1$ , effectively allowing matchings of repeated features. This can be achieved by searching a general geometrical consensus between features of the images before actually evaluating individual matches. For that matter, most feature detector algorithms conveniently return not only the descriptor vector, but also a set of associated geometric data. In general, a feature  $F$  can be described as a tuple with at least the following data:

$$F = \langle x, y, s, \phi, \mathbf{d} \rangle, \quad (3)$$

where  $x$  and  $y$  are the coordinates of the feature centroid, in pixels;  $s$  is a scale factor;  $\phi$  is the orientation of some particular characteristic of the feature; and  $\mathbf{d}$  is the descriptor vector.

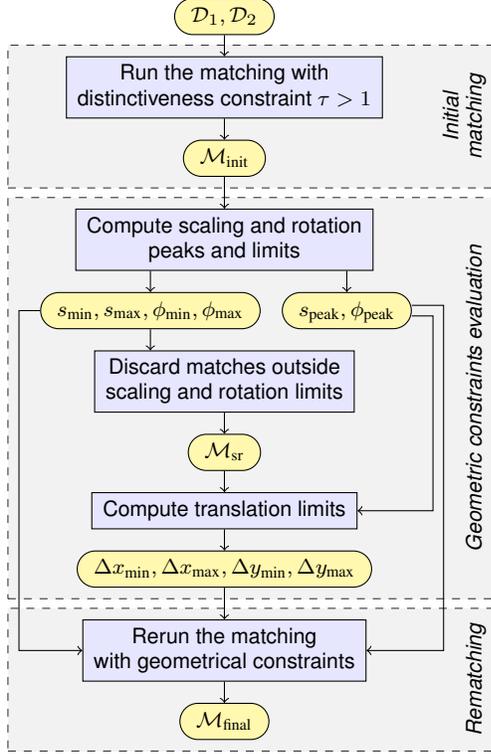
In general, the scale factor  $s$  and the orientation angle  $\phi$  may have no meaning in the image space, serving only to compare the geometry between local patches (around the neighborhood of  $(x, y)$ ) in the two images. For example, given a set of  $\{\langle i, j \rangle\}$  “good” matches between two images,  $I_1$  and  $I_2$ , where  $I_2$  is a zoomed view of a particular region of  $I_1$ , then all  $s_j/s_i$  should maintain a general consistence about that scaling transformation, i.e.,  $s_j/s_i \sim \text{constant}$ . The same should be true for the difference in orientations  $(\phi_j - \phi_i)$  if the second image is a rotated view (along camera axis) of the first image.

In the general case, we cannot expect the same scaling factors  $s_j/s_i$  nor a constant orientation change  $\phi_j - \phi_i$  for all matched pairs  $\langle i, j \rangle$  (which limits the applicability of robust methods that search for extreme values, such as iterative optimization methods or RANSAC [4]). However, if the scene is static and the camera movement between shots does not cause significant perspective transformation from one image to the other, then we can expect both scaling and rotation to be bounded. The identification of these limits is the inspiration of our methodology, which is presented next.

### 3 Methodology

Our methodology is depicted in Figure 1, and it works as follows. Given  $\mathcal{M}_{\text{init}} = \{\langle i, j \rangle\}$ , the set of matchings generated by any matching method (e.g., Eq. (2)), we first search for a global consensus for scaling, rotation, and translation of the image features and define “acceptable” ranges for these transformations. We then build a new set of matchings based only on geometric constraints (as if the distinctiveness threshold  $\tau$  was set to 1).

While Eq. (2) can be used to evaluate the initial set of matched features  $\mathcal{M}_{\text{init}}$ , this would also cause our algorithm to be slower than current methods (since the total time required to run our methodology includes the time spent to compute  $\mathcal{M}_{\text{init}}$ ). A more time-efficient method to evalu-



**Figure 1. Overview of our methodology. From an initial matching set, we first analyze scaling, rotation and translation to evaluate the geometric constraints, then the matching is rerun under these constraints.**

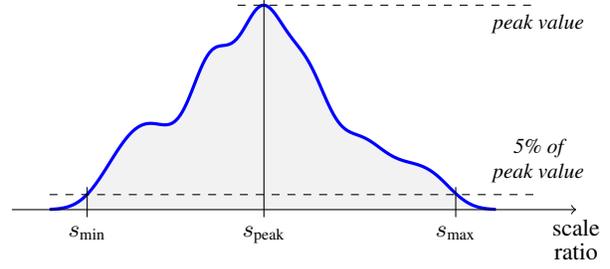
ate the initial set is discussed next. Further details on the following steps will be described in Subsections 3.2 and 3.3.

### 3.1 Initial matching

Evaluation of a matching set according to Eq. (2) has asymptotic time complexity of  $O(KN_1N_2)$ , where  $K$  is the descriptor vector length and  $N_1$  and  $N_2$  are the number of features found in each input image respectively. While reducing the asymptotic cost profile is not a trivial task, we can actually reduce the computational cost by taking only a subset of descriptors of  $\mathcal{D}_1$  to evaluate  $\mathcal{M}_{\text{init}}$ . Although the resulting matching set is probably smaller than that evaluated in Eq. (2), this does not compromise the rest of the methodology, since  $\mathcal{M}_{\text{init}}$  is used only to compute the limits for the rematching step.

We define  $\mathcal{Z}$  as a random subset of  $\mathcal{D}_1$ , where the number of elements of  $\mathcal{Z}$  is defined as

$$|\mathcal{Z}| = \left\lfloor \frac{|\mathcal{D}_1|}{z} \right\rfloor \quad (4)$$



**Figure 2. Evaluation of the acceptable range for the scale ratio. The same procedure applies to the evaluation of the rotation acceptable range.**

for a given *subsampling factor*  $z \geq 1$ . Then we redefine  $\mathcal{M}_{\text{init}}$  by adapting the definition presented in Eq. (2):

$$\mathcal{M}_{\text{init}} = \left\{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j}) \rangle \mid \forall k \neq j [ \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,k}) \geq \tau \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j}) ] \right\} \quad \forall \mathbf{d}_{1,i} \in \mathcal{Z}. \quad (5)$$

While the asymptotic time complexity is still the same (since  $O(K \frac{N_1}{z} N_2) = O(KN_1N_2)$ ), the actual time required to compute  $\mathcal{M}_{\text{init}}$  as in Eq. (5) will be significantly reduced for a large  $z$  when compared to the original proposition shown in Eq. (2). The optimal determination of  $z$  will not be addressed here. We will empirically determine the subsampling ratio for our experiments in Section 4.

## 3.2 Geometric constraints evaluation

The objective of this step is to define acceptable ranges for all geometric transformations considered in this work: scaling, rotation, and translation. The first two transformations — scaling and rotation — can be directly estimated from the scaling factors  $s$  and orientations  $\phi$  computed by the feature detector (Eq. (3)). However, feature translations in image coordinates can only be consistently estimated after determining the other transformations. Thus, the evaluation of geometric constraints is performed in two steps: (i) scale and rotation and (ii) translation.

### 3.2.1 Scale and rotation constraints evaluation

For scale and rotation constraints evaluation, we first analyze the scaling ratios and rotation angles that occur between matched features. We then define an acceptable range for these transformations and discard all pairs that fall outside these constraints.

For scaling ratios we proceed as follows: Given the set of all scaling ratios, represented by  $\mathcal{S}$  and defined as:

$$\mathcal{S} \triangleq \{s_{2,j}/s_{1,i} \mid \forall \langle i, j \rangle \in \mathcal{M}_{\text{init}}, \quad (6)$$

we first estimate the Probability Density Function (PDF) of  $\mathcal{S}$  using Parzen windows [9] with a Gaussian kernel. For the bandwidth value we use the automated method discussed in [2]. From the estimated PDF we pick the following values (see Figure 2):

- the mode of scale ratio (the value at the PDF's peak),  $s_{\text{peak}}$ ;
- the largest scale ratio below  $s_{\text{peak}}$  for which the likelihood equals 5% of the mode,  $s_{\text{min}}$ ; and
- the smallest scale ratio above  $s_{\text{peak}}$  for which the likelihood equals 5% of the mode,  $s_{\text{max}}$ .

The same procedure is applied to find the rotation angle's (difference of orientations) peak and limits. Given the set of all rotation angles,  $\mathcal{R}$ , defined as:

$$\mathcal{R} \triangleq \{\phi_{2,j} - \phi_{1,i} \mid \forall \langle i, j \rangle \in \mathcal{M}_{\text{init}}, \quad (7)$$

we estimate the PDF of  $\mathcal{R}$ , again using Parzen windows. However, since angles form a cyclic space, the estimated PDF must be confined to a  $(2\pi \text{ rad})$ -wide window centered on the likelihood peak  $\phi_{\text{peak}}$ , i.e., the PDF is defined over the domain  $(\phi_{\text{peak}} - \pi \text{ rad}, \phi_{\text{peak}} + \pi \text{ rad}]$ . The acceptable range limits  $\phi_{\text{min}}$  and  $\phi_{\text{max}}$  are defined from this PDF exactly as  $s_{\text{min}}$  and  $s_{\text{max}}$ .

Finally, we build the set  $\mathcal{M}_{\text{sr}} \subseteq \mathcal{M}_{\text{init}}$  containing only matching pairs whose scaling and rotation transformations fall within the limits previously defined:

$$\mathcal{M}_{\text{sr}} \triangleq \{ \langle i, j \rangle \in \mathcal{M}_{\text{init}} \mid (s_{\text{min}} \leq s_{2,j}/s_{1,i} \leq s_{\text{max}}) \wedge (\phi_{\text{min}} \leq \phi_{2,j} - \phi_{1,i} \leq \phi_{\text{max}}) \}, \quad (8)$$

where  $\phi_{2,j} - \phi_{1,i}$  is normalized to the interval  $(\phi_{\text{peak}} - \pi \text{ rad}, \phi_{\text{peak}} + \pi \text{ rad}]$ .

### 3.2.2 Translation constraints evaluation

As explained before, translation is analyzed on the transformed (scaled and rotated) feature coordinates. For each pair  $\langle i, j \rangle \in \mathcal{M}_{\text{sr}}$ , we define the *transformed displacement vector*  $\vec{v}_{i,j}$  as:

$$\vec{v}_{i,j} \triangleq \begin{bmatrix} \Delta x_{i,j} \\ \Delta y_{i,j} \end{bmatrix} \triangleq \begin{bmatrix} x_{2,j} \\ y_{2,j} \end{bmatrix} - s_{\text{peak}} \mathbf{R} \begin{bmatrix} x_{1,i} \\ y_{1,i} \end{bmatrix}, \quad (9)$$

where  $\mathbf{R}$  is the peak angle  $\phi_{\text{peak}}$  rotation matrix:

$$\mathbf{R} \triangleq \begin{bmatrix} \cos \phi_{\text{peak}} & -\sin \phi_{\text{peak}} \\ \sin \phi_{\text{peak}} & \cos \phi_{\text{peak}} \end{bmatrix}. \quad (10)$$

Translation limits are defined as follows: Given a bidimensional histogram over  $x$  and  $y$  axes of vectors  $\vec{v}_{i,j}$ , the rectangle that covers the connected island of all histogram bins around the histogram peak defines the acceptable ranges for both axes:  $\Delta x_{\text{min}}$ ,  $\Delta x_{\text{max}}$ ,  $\Delta y_{\text{min}}$ , and  $\Delta y_{\text{max}}$ .

### 3.3 Rematching

From the constraints evaluated during the previous step, we now build the final set of matching pairs  $\mathcal{M}_{\text{final}}$ :

$$\mathcal{M}_{\text{final}} \triangleq \{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_{1,i}, \mathbf{d}_{2,j}) \rangle \mid (s_{\text{min}} \leq s_{2,j}/s_{1,i} \leq s_{\text{max}}) \wedge (\phi_{\text{min}} \leq \phi_{2,j} - \phi_{1,i} \leq \phi_{\text{max}}) \wedge (\Delta x_{\text{min}} \leq \Delta x_{i,j} \leq \Delta x_{\text{max}}) \wedge (\Delta y_{\text{min}} \leq \Delta y_{i,j} \leq \Delta y_{\text{max}}) \}, \quad (11)$$

again with  $\phi_{2,j} - \phi_{1,i}$  normalized to the interval  $(\phi_{\text{peak}} - \pi \text{ rad}, \phi_{\text{peak}} + \pi \text{ rad}]$ . Notice that, as opposed to Eqs. (2) and (5), no distinctiveness constraint is used.

## 4 Experiments and results

Both the classical and the proposed methods were implemented in Matlab/C++. The test platform consists of an Intel® 2GHz Core™2 Duo processor with 4GB of RAM, running a 64-bit GNU/Linux Ubuntu box (kernel v2.6.28). Images were taken using a Canon PowerShot SX10 IS digital camera with 10.0 megapixels and subsampled to 1/4 of the original dimensions, resulting in images of  $912 \times 684$  pixels.

Some images of a planar, feature-rich scene, shown in Figure 3, were used as the base for the experiments. We evaluate the proposed methodology by analyzing the correspondences from the base image (Figure 3(a)) to the others, thus covering three important transformations: change of point-of-view (Figure 3(b)), camera approximation (Figure 3(c)), and camera rotation (Figure 3(d)). Feature sets  $\mathcal{D}_1$  and  $\mathcal{D}_2$  are evaluated using the SIFT algorithm [6, 7].

Although the methodology is not limited to planar scene patches, this setup allows for a direct qualitative evaluation of the results based on image homography. Here we refer to the homography as a function  $H : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  which maps bidimensional coordinates  $(x, y)$  from the first image to the coordinate space of the second image:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = H \left( \begin{bmatrix} x \\ y \end{bmatrix} \right). \quad (12)$$

The homography between two images is recovered using Bouguet's *Camera Calibration Toolbox for Matlab* [3] based on the checkerboard calibration patterns present in all images.



**Figure 3. Set of images used in the experiments: (a) base image, (b) image after change of point-of-view, (c) image after camera approximation, and (d) image after camera rotation.**

Under ideal conditions, given any matching pair  $\langle i, j \rangle$ , the feature coordinates in the second image,  $(x_{2,j}, y_{2,j})$ , should match the transformed coordinates of the corresponding feature in the first image,  $(x_{1,i}, y_{1,i})$ . In practice, noise from several sources affect this equality: mismatched pairings, calibration uncertainties, determination of feature coordinates, and so on. Thus, we define the *error vector*  $\vec{\epsilon}_{i,j}$  as the difference between observed (from feature detector and matcher) and predicted (from homography) coordinates:

$$\vec{\epsilon}_{i,j} \triangleq \begin{bmatrix} x_{2,j} \\ y_{2,j} \end{bmatrix} - H \left( \begin{bmatrix} x_{1,i} \\ y_{1,i} \end{bmatrix} \right). \quad (13)$$

From the set of the error vectors evaluated from a matching set  $\mathcal{M}$ , we analyze two global error measurements: the *Root Mean Square Error*,  $\text{RMSE}(\mathcal{M})$ , defined as

$$\text{RMSE}(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sqrt{\sum_{\langle i,j \rangle \in \mathcal{M}} \|\vec{\epsilon}_{i,j}\|^2}, \quad (14)$$

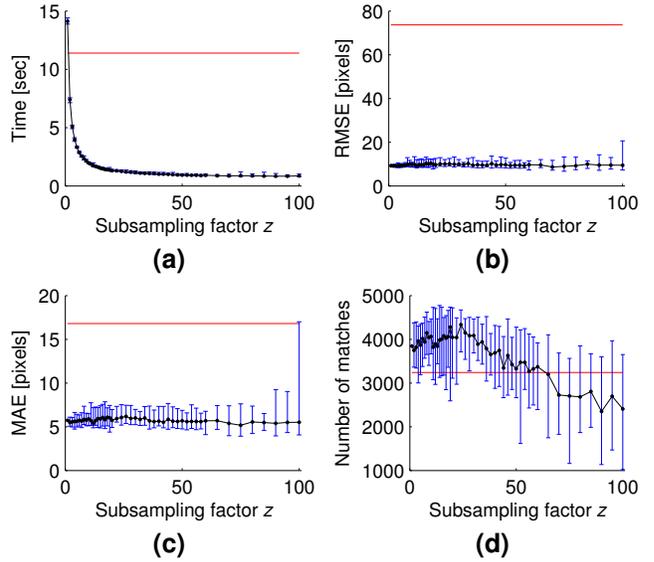
and the *Mean Absolute Error*,  $\text{MAE}(\mathcal{M})$ , defined as

$$\text{MAE}(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{\langle i,j \rangle \in \mathcal{M}} \|\vec{\epsilon}_{i,j}\|. \quad (15)$$

Smaller values obtained for  $\text{RMSE}(\mathcal{M})$  and  $\text{MAE}(\mathcal{M})$  indicate better results. RMSE is particularly sensitive to outliers (i.e., a single big value of  $\|\vec{\epsilon}_{i,j}\|$ ) than MAE.

To compare the results obtained from current methods (Eq. (2)), referenced here as  $\mathcal{M}_{\text{curr}}$ , with those obtained from our method (Eq. (11)), our experiments estimate the following performance factors:

- the number of pairs of each matching set,  $|\mathcal{M}_{\text{curr}}|$  vs.  $|\mathcal{M}_{\text{final}}|$ ;
- the error measurements, RMSE and MAE, for both  $\mathcal{M}_{\text{curr}}$  and  $\mathcal{M}_{\text{final}}$ ; and
- the time required to run each matching method.



**Figure 4. Results obtained with various values for the subsampling factor  $z$  for camera rotation, showing results after 30 runs: median (black line), minimum and maximum (error bars). Red (horizontal) line is the result obtained with classical method. (a) execution time, (b) RMSE, (c) MAE, and (d) number of matches ( $|\mathcal{M}_{\text{final}}|$ ). Observe the decrease of execution time as  $z$  increases, the instability of RMSE and MAE for large values of  $z$ , and the gradual decrease of the number of matches approximately from  $z \geq 30$ .**

The experiments are divided in two steps. In Subsection 4.1 we perform some tests to select a reasonable value for the subsampling factor,  $z$ ; and in Subsection 4.2 we perform a deep analysis of the results obtained with the selected subsampling factor.

Test case Method	POV-change			Camera approximation			Camera rotation		
	Pairs	RMSE [px]	MAE [px]	Pairs	RMSE [px]	MAE [px]	Pairs	RMSE [px]	MAE [px]
Classical method	2,887	90.50	22.70	1,571	129.54	41.60	3,242	73.65	16.81
Proposed method	3,255	14.10	8.40	1,968	9.33	6.18	4,245	11.24	7.16
Comparison (%)	113%	15.6%	37.0%	125%	7.2%	14.9%	131%	15.3%	42.6%

**Table 1. Matching count and error measures comparison between classical (with  $\tau = 1.5$ ) and proposed methods. Notice the consistent increase of number of matches and significant decrease of both RMSE and MAE error measurements.**

#### 4.1 Selection of the subsampling factor

A typical profile obtained for various values of  $z$  is shown in Figure 4. For each value of  $z$  we ran the algorithm 30 times, each one with a different selection of the random subset  $\mathcal{Z} \subseteq \mathcal{D}_1$  (Eq. (4)). In Figure 4(a), it can be seen that the time required to execute the algorithm steadily decreases when  $z$  increases. The only case where the proposed algorithm takes more time than the classical method is when  $z = 1$ , i.e., when there is no subsampling at all. Median of error measures (RMSE and MAE, shown in Figures 4(b) and 4(c), respectively) remain approximately the same for all tested values of  $z$ ; however, for large subsamplings ( $z \geq 90$ ) a poor selection of  $\mathcal{Z}$  can compromise the quality of results, which is clearly seen by the increase of the maximum values of MAE. As for the number of matches (Figure 4(d)), in general it is possible to outperform the classical method (red line) for values of  $z \leq 60$ ; however, quality starts to decrease approximately from  $z \geq 30$ .

From these results, we have chosen  $z = 20$  for the remaining experiments. This subsampling factor is a good compromise between time cost reduction and number of matches maximization.

#### 4.2 Quality and performance analysis of the proposed method

For the adopted subsampling factor  $z = 20$ , results for both the number of correspondences ( $|\mathcal{M}_{\text{curr}}|$  vs.  $|\mathcal{M}_{\text{final}}|$ ) and the error measurements (RMSE and MAE) are shown in Table 1. All results show a consistent increase in the number of correspondences (about 13%–31% more matches) and an expressive decrease in both RMSE and MAE error measurements (to less than 1/6 for RMSE and to less than 1/2 for MAE).

Table 2 shows the time required to compute the matches. Results demonstrate that our method is about 10 times faster than the classical method. The most expensive step is the initial matching, accounting for approximately 37%–51% of the total time of our test cases. This step would be much more expensive if the optimization proposed in Eq. (5) was

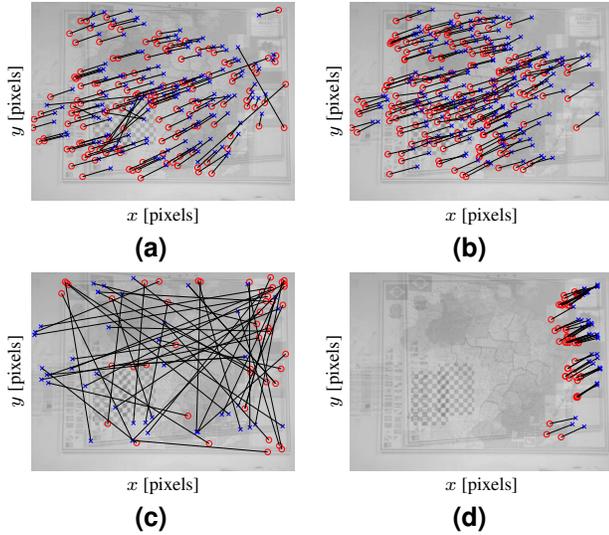
Test case Method / step	Time [s]		
	POV- -change	Camera approximation	Camera rotation
Classical method	7.86	13.49	11.20
Proposed method	0.86	1.06	0.98
Pre-matching	0.32	0.54	0.46
Filtering	0.28	0.23	0.21
Rematching	0.27	0.29	0.31
Comparison (%)	11.0%	7.8%	8.7%

**Table 2. Time comparison between classical (with  $\tau = 1.5$ ) and proposed methods. The times spent for each step are also shown. Notice that the proposed method runs approximately 10 times faster than the classical method.**

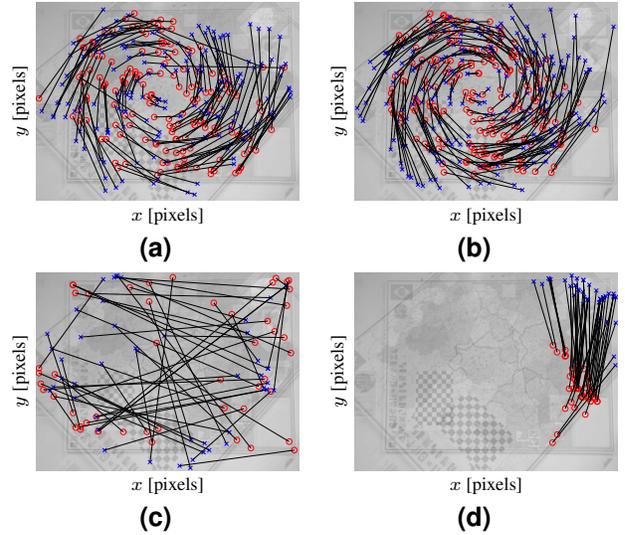
not used.

The improvement in the quality of the results can be also visually assessed, as shown in Figures 5, 6 and 7. For each of these figures, we plot 5% of the matches (first row of images) and the 50 worst matches, i.e., those corresponding to the 50 highest values for  $\|\vec{\epsilon}_{i,j}\|$  (second row). We used only 5% of the matches for the first row in order to avoid visual cluttering caused by the plot of thousands of matches. While some severe outliers can be seen in Figures 5(a), 6(a) and 7(a), no significant outliers can be seen in Figures 5(b), 6(b) and 7(b). Also, the 50 worst matches are clearly mismatches in Figures 5(c), 6(c) and 7(c), while the respective worst matches of Figures 5(d), 6(d) and 7(d) still follow the general geometric matching pattern. Even so, it is worth noting that these worst matches appear consistently far from the calibration patterns, which may indicate a poor homography model in these regions due to camera lens distortion.

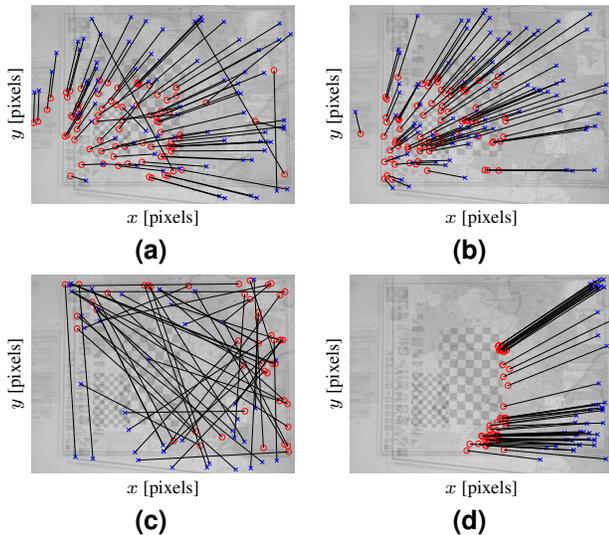
Figure 8 shows the distribution of the error vectors  $\vec{\epsilon}_{i,j}$  for all test cases. All experiments show that our method confines error vectors to much narrower bounds, about one order of magnitude when compared to the values obtained from the classical method (notice the scale change for all plots).



**Figure 5. Matches for point-of-view change (from Figure 3(a) to Figure 3(b)): 5% of the matches for (a) classical method with  $\tau = 1.5$ , (b) proposed method; 50 worst matches for (c) classical method, (d) proposed method.**



**Figure 7. Matches for camera rotation (from Figure 3(a) to Figure 3(d)): 5% of the matches for (a) current method with  $\tau = 1.5$ , (b) proposed method; 50 worst matches for (c) current method, (d) proposed method.**

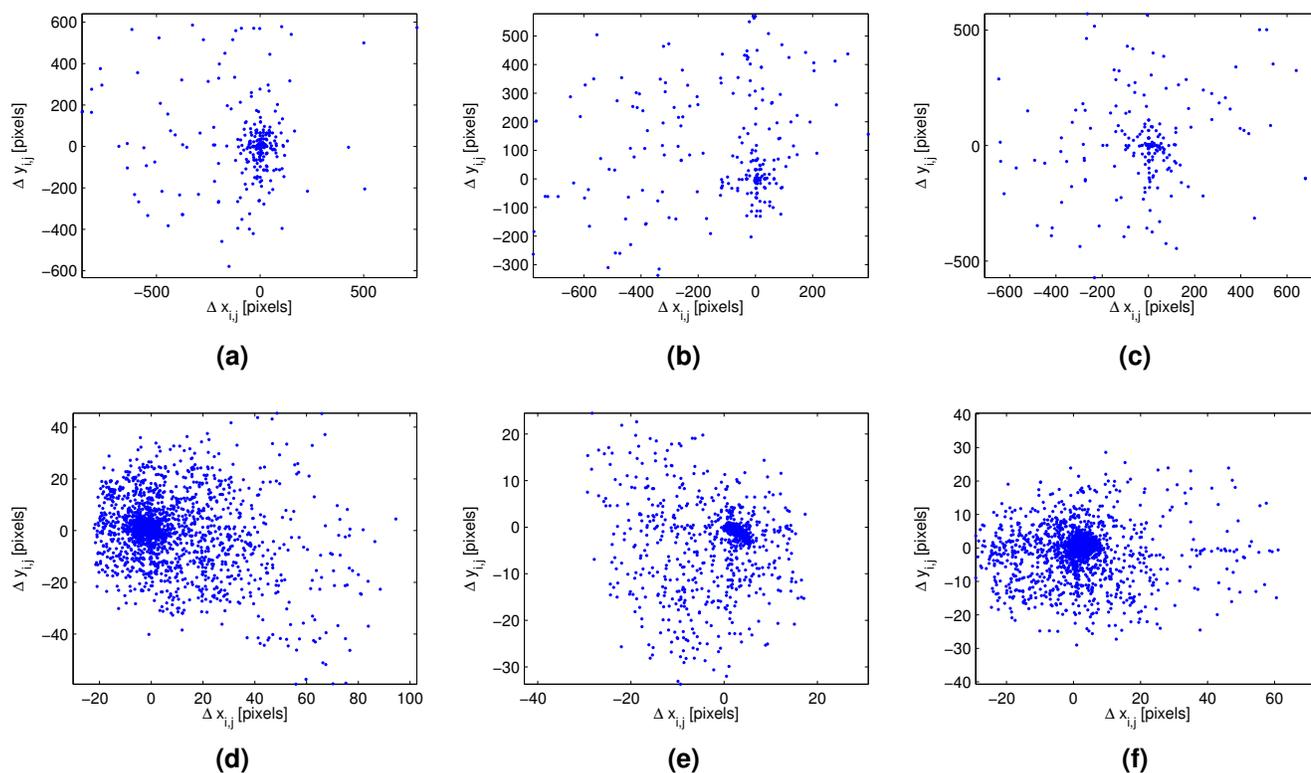


**Figure 6. Matches for camera approximation (from Figure 3(a) to Figure 3(c)): 5% of the matches for (a) current method with  $\tau = 1.5$ , (b) proposed method; 50 worst matches for (c) current method, (d) proposed method.**

## 5 Conclusions and future work

We have presented a novel methodology for feature matching in a pair of images that provides more accurate results at considerably smaller computational cost. Compared to current methods based on similarity between descriptor vectors alone, our approach is able to significantly reduce the occurrence of mismatches, as evidenced by the considerable decrease in both RMSE and MAE error measurements. Since RMSE is very sensitive to outliers, the expressive decrease in its magnitude, when compared to typical approaches in the literature, is a clear indication that occurrence of feature mismatch is much less frequent in the proposed method. Also, as we have shown, the increase in the number of successful matches suggests that classical techniques are likely to discard correct matches due to the use of a distinctiveness threshold. This undesirable effect is not present in the proposed method, since no distinctiveness threshold is used. Finally, we have also shown that our method presents a tenfold increase in performance, while correctly determining all viable matches.

Our method expects that all geometric transformations of features (rotation, scale, and translation) are bounded to a single interval, an assumption that is not true in the general case where the scene is not static or if the images exhibit large point-of-view changes of the camera between shots. In these cases, it is possible to improve the proposed method by accepting more than one set of geometric transformation



**Figure 8. Homography errors ( $\vec{\epsilon}_{i,j}$ ) for point-of-view change (first column), for camera approximation (second column), and for camera rotation (third column). First row: current method with  $\tau = 1.5$ ; second row: proposed method. Notice the difference of one order of magnitude in the scales of the axis of plots on the first (current method) and second (proposed method) rows.**

intervals, effectively segmenting the images in regions for which geometric transformations are locally consistent. This generalization is being currently addressed by the authors, as part of ongoing researches in scene geometry reconstruction and SLAM.

## Acknowledgments

This work was partially supported by *Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (Analysis, Research and Technological Innovation Center) — FUCAPI*, Manaus, AM, Brazil.

## References

- [1] H. Bay, T. Tuytelaars, and L. J. V. Gool. SURF: Speeded Up Robust Features. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proc. ECCV*, pages 404–417, 2006.
- [2] Z. I. Botev. A novel nonparametric density estimator. Technical report, The University of Queensland, 2006.
- [3] J.-Y. Bouguet. Camera calibration toolbox for Matlab. Available at [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc), 2008.
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [5] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR*, volume 2, pages 506–513, 2004.
- [6] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, volume 2, pages 1150–1157, 1999.
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [8] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.
- [9] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.