**Journal of Real-time Image Processing manuscript No.**
(will be inserted by the editor)

**Pablo Márquez-Neila · Javier López-Alberca · José M. Buenaposada · Luis Baumela**

# Speeding-up homography estimation in mobile devices

Received: date / Accepted: date

**Abstract** A critical problem faced by computer vision on mobile devices is reducing the computational cost of algorithms and avoiding visual stalls. In this paper we introduce a procedure for reducing the number of samples required for fitting a homography to a set of noisy correspondences using a random sampling method. This is achieved by means of a geometric constraint that detects invalid minimal sets. In the experiments conducted we show that this constraint not only reduces the number of random samples at a negligible computational cost, but also balances the processor workload over time preventing visual stalls. In extreme situations of very large outlier proportion and noise level, it reduces in about one order of magnitude the number of required random samples.

**Keywords** mobile devices · random sampling · homography estimation

**Mathematics Subject Classification (2000)** MSC 65D19

## 1 Introduction

Consumer devices can easily create and manage multimedia information in digital format and yet the connection between the digital and the physical world, *e.g.* identifying a character in a film, recognizing a piece of music or reading the title of a book, is still a difficult task. These perceptual problems have traditionally been addressed assuming the availability of a high-performance computer where a computationally intensive algorithm performs the recognition. Now there is, nevertheless, an ongoing shift in the computing paradigm from powerful decentralized computers connected though a network towards an entirely pervasive computing system where a myriad of resource-limited inter-communicated computing devices are spread worldwide, "like pigment in the wall paint" Castells (1996). Providing perceptual abilities to these small devices is an open challenge for computer science.

Mobile devices such as cameras, media players and tablets are an ubiquitous part of our daily life, being the mobile phone the most pervasive digital apparatus. Nowadays they are equipped with high-quality color displays, high resolution digital cameras, hardware accelerated 3D graphics as well as GPS and broadband data connections, enabling it to be applied in a wide spectrum of applications. Initially they were used as sensors for two-dimensional visual codes attached to physical objects, acting as a key to access object-related information and functionality Rohs and Gfeller (2004). For certain types of objects, such as sights, buildings or living beings, the tag-based approach is not adequate. Markerless solutions are based on extracting a set of object discriminant features Lowe (2004) and performing the recognition on a remote server Pielot et al (2008) or on the mobile device itself Bruns and Bimber (2009); Henze et al (2009). A combination of marker-less recognition and pervasive tracking has been used for building a large-scale museum guide Bruns et al (2007). Augmented reality (AU) is another practical application on mobile phones. It augments indoor or outdoor scenes with graphical elements providing new ways of interaction and giving information about objects and locations in the scene. The relative orientation between camera and scene is estimated computing the positions of a

Pablo Márquez-Neila, Javier López-Alberca, Luis Baumela
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus Montegancedo s/n
28660 Boadilla del Monte, Spain
Tel.: +34 91 336 7440
Fax: +34 91 352 4819
E-mail: p.mneila@upm.es, lbaumela@fi.upm.es

José M. Buenaposada
Departamento de Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos
C/ Tulipán, s/n
28933 Móstoles (Spain)
Tel.: +34 91 488 8129
E-mail: josemiguel.buenaposada@urjc.es

set of image features Takacs et al (2008) or by learning and recognizing the perspective orientation of a texture patch Lee et al (2010).

In this paper we introduce a procedure to accelerate the estimation of homographies and evaluate the impact of the approach on a mobile device. Homographies are a simple and precise way to obtain the relative orientation between the camera and a planar patch in the scene. They have been extensively used for robot navigation López-Nicolás et al (2010a,b); Guerrero et al (2005) as well as tracking and mapping Klein and Murray (2009) and constructing image panoramas Xiong and Pulli (2010) on mobile phones or creating augmented reality books Kim et al (2010). Although there are procedures to estimate multiple homographies between a pair of images all constrained to have the same relative orientation Kirchhof (2008), single Homographies are usually estimated finding the correspondences of a set of discriminant features using a Random Sampling Method (RSM) Hartley and Zisserman (2004); Klein and Murray (2009); Kim et al (2010). The main drawback for implementing this approach in a resource-limited device, such as a mobile phone, is the computational cost involved both in identifying the discriminant image features and in the RSM procedure itself, which involves the repetitive fitting of a model to a large enough number of sets of correspondences. The problem of efficiently extracting and classifying image features has been extensively studied Roblee et al (2011); Wagner et al (2008); Taylor and Drummond (2009); Taylor et al (2009). In this paper we deal with the second problem, i.e. alleviating the computational cost of the RSM.

RSMs are a family of techniques used to robustly fit a model to data in presence of outliers. The most prominent examples of these procedures are Random Sample And Consensus (RANSAC) Fishler and Bolles (1981) and Least Median of Squares (LMEDS) Rousseeuw (1984), both of which have been extensively used in computer vision. RANSAC often replaces LMEDS in vision algorithms since it has more tuning parameters and, consequently, it can be better adapted to complex data analysis situations Meer et al (2000). In this paper we will consider the RANSAC algorithm, although our results are immediately applicable to LMEDS. MLESAC Torr and Zisserman (2000); Tordoff and Murray (2005) is an improvement of RANSAC that replaces the inlier count with a weighted voting based on an M-estimator. Randomized RANSAC Chum and Matas (2002) improves efficiency by randomizing the RANSAC hypothesis evaluation step. PROSAC Chum and Matas (2005) prioritizes correspondences which are more likely to be inliers in the sampling step, thereby increasing the probability of finding a good hypothesis earlier. MAPSAC Torr and Davidson (2000) uses a Bayesian approach to rank the hypothesis, estimating the posterior of the hypothesis given the data. In the same work, Torr *et al.* introduce IMPSAC, which works with an image pyramid and proceeds in a coarse-to-fine procedure. They apply MAPSAC in the coarsest level, and propagate the believes in the space of parameters of finer levels using sampling-importance-resampling and Markov Chain Monte Carlo. The consensus sampling technique of Cheng and Lai (2009) provides several improvements to RANSAC: it ranks the quality of the data samples and gives priority to those ranked higher than a threshold (similar to PROSAC); it uses a preemptive scheme to evaluates a large set of hypothesis with only a few data samples; it robustly estimates the standard deviation of the inliers noise to determine the error scale. These works introduce improvements which are applicable in fairly general cases. However, none of those improvements takes advantage of some specific conditions that hold in homography estimation. Our contribution is complementary to all these works.

If we assume that our camera follows a pinhole model, then only points in front of the camera are visible. This condition is modeled using the oriented projective geometry framework Laveau and Faugeras (1996). In this paper we extend the use of the orientation restriction in the estimation of homographies. Using the oriented projective geometry we derive a circular order-preserving constraint. This is not the first time that such a constraint has been used. A similar constraint was previously used in Tell and Carlsson (2002). However, it was not used for homography estimation. Chum Chum et al (2004) also applies the oriented projective geometry to establish a constraint on epipolar geometry estimation via RANSAC. In the experiments conducted we show that the constraint reduces the computational cost of RSMs at almost no overhead. Moreover, the geometrically-constrained algorithm not only achieves the lowest computational cost, but also the lowest variance and consequently the best balanced processor workload over time.

The organization of the paper is as follows. In Section 2 we recall the general scheme of RSMs for homography estimation. Section 3 describes the technique used for improving the performance of RSMs and introduces the geometric constraint. In Sections 4 and 5 we present the implementation of the constraint for OpenCV and the results of an extensive set of synthetic and real experiments. Finally, in Section 6 we draw conclusions.

## 2 Random sampling homography estimation

For the estimation of plane homographies, RSMs need a set of point correspondences $\mathcal{C}$ between two images, in which there may be outliers. In Algorithm 1 we give the general scheme of RANSAC. It iteratively takes random subsets from the original set $\mathcal{C}$ and fits a model (in this case, a homography) to them. The number of elements in each subset is the minimum for fitting the model (specifically, four for estimating a homography). This is what we call a minimal subset, $S_h$. In the case of RANSAC, the quality of a model is given by the number of inliers.

**Algorithm 1** RANSAC scheme for homography estimation.

1: **for** $h = 1$ to $h = it_{\max}$ **do**
2: Take randomly a subset $S_h$ of hypothetical correspondences from the set $\mathcal{C}$.
3: Fit a homography $M_h$ to $S_h$.
4: Compute the quality $u_h$ of model $M_h$.
5: **if** $u_h > u^*$ **then**
6: Store current homography $M_h$ as the best one $M_h^*$.
7: $u^* \leftarrow u_h$.
8: **end if**
9: **end for**
10: Return the best model $M_h^*$.

Here, an inlier is a sample whose distance to the model is below a given threshold. After $it$ iterations, the model with largest number of inliers is selected.

Steps 3 and 4 in Algorithm 1 are the most expensive in term of computational requirements, since they involve a repetitive model fitting process which is costly computationally. The goal of the geometrical restriction introduced in Section 3 is avoiding unnecessary executions of these steps.

Given a proportion $p$ of inliers in the data set, the probability $P$ of finding at least one correct hypothesis after $it$ iterations is given by Fishler and Bolles (1981):

$$P = 1 - (1 - p^m)^{it},$$

where $m$ is the size of the minimal subset ($m = 4$ for estimating homographies). Therefore, given a desired confidence level $P$, the theoretical number of necessary iterations is

$$it_{\max} = \frac{\log(1-P)}{\log(1-p^m)}. \tag{1}$$

This is an optimistic estimation of the number of iterations, since it does not consider the noise contaminating inliers' location. Note that a minimal subset of correct inliers may not lead to a valid hypothesis (see Fig. 1). So, the actual number $it^*$ of random sampling iterations is always larger than the theoretical value $it_{\max}$. In Section 5.1 we perform synthetic experiments that validate these hypothesis.

## 3 Geometric constraint in homography estimation

A set of points in a plane are visible only when two necessary (but not sufficient) conditions hold:

- Points must be in front of the camera as required by the oriented projective geometry Laveau and Faugeras (1996).
- The plane containing the points must face towards the camera.

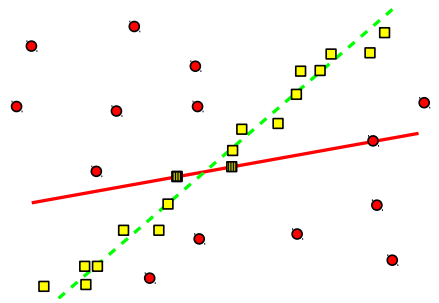


**Fig. 1** Example of noisy inliers which lead to a wrong hypothesis. □'s are noisy inliers, and ∘'s are outliers. Inliers with a line pattern lead to a wrong model (continuous line). The correct model is given by the dashed line.

RSMs spend a large portion of computational resources processing sets $S_h$ of correspondences that do not verify these properties. Here we introduce a geometric constraint to check whether a minimal subset $S_h$ is valid, before actually fitting a model $M_h$ (step 3 in Algorithm 1) and evaluating it (step 4).

Let $I_0$ and $I_1$ be two camera images in which a plane $\boldsymbol{\pi}$ is visible. We want to estimate the plane $\boldsymbol{\pi}$ induced homography $H$ between these two images. Let $P_0 \subset \mathbb{P}^2$ be the set of two-dimensional coordinates of visible key-points in $I_0$, and $P_1 \subset \mathbb{P}^2$ be the set of two-dimensional coordinates of visible key-points in $I_1$. $\mathcal{C}$ is the set of tentative correspondences between points in $I_0$ and $I_1$. A correspondence $v_i \in \mathcal{C}$ is a tuple $(\mathbf{p}_0, \mathbf{p}_1)$ which maps an element of $P_0$ to an element of $P_1$.

Let $\mathbf{p}_i \in P_i$ be a visible point in image $I_i$. Since it is visible, we assume it is in front of the camera, *i.e.*, there exists a positive scale $\lambda_{\mathbf{p}_i}$ such that

$$\mathbf{p}_i = \lambda_{\mathbf{p}_i} \mathbf{P}_i, \quad \lambda_{\mathbf{p}_i} > 0 \tag{2}$$

where $\mathbf{P}_i \in \mathbb{R}^3$ is the location of the projected point $\mathbf{p}_i$ in the reference frame of the camera $i$. We denote the similarity up to a positive scale with $\overset{+}{\sim}$:

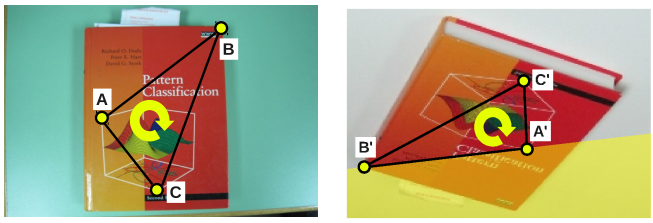
$$\mathbf{p}_i \overset{+}{\sim} \mathbf{P}_i.$$

In each iteration, we are given a minimal subset $S_h = \{v_a, v_b, v_c, v_d\}$ of four correspondences:

$$v_a = (\mathbf{a}_0, \mathbf{a}_1),\ \mathbf{a}_0 \in P_0,\ \mathbf{a}_1 \in P_1,$$
$$v_b = (\mathbf{b}_0, \mathbf{b}_1),\ \mathbf{b}_0 \in P_0,\ \mathbf{b}_1 \in P_1,$$
$$v_c = (\mathbf{c}_0, \mathbf{c}_1),\ \mathbf{c}_0 \in P_0,\ \mathbf{c}_1 \in P_1,$$
$$v_d = (\mathbf{d}_0, \mathbf{d}_1),\ \mathbf{d}_0 \in P_0,\ \mathbf{d}_1 \in P_1.$$

We want to check whether $S_h$ is valid before estimating its corresponding homography. The plane induced by $S_h$ must be visible in both images $I_0$ and $I_1$ to be valid, *i.e.*, both camera centers must be located at the same side of the plane:

$$\mathbf{n}_0^T \mathbf{C}_0 + d_0 \overset{+}{\sim} \mathbf{n}_1^T \mathbf{C}_1 + d_1,$$




(a) (b)

**Fig. 2** Geometric constraint that must be satisfied by $S_h$ in each random sample. (*a*) Points $A, B, C$ in image $I_1$. (*b*) Corresponding points $A', B', C'$ in image $I_0$. Point $C'$ must not be located in the marked region, since $A', B', C'$ must have the same relative order than $A$, $B$, $C$. If this constraint does not hold, this set of correspondences should be discarded.

where $\mathbf{n}_i$, $d_i$ and $\mathbf{C}_i$ are, respectively, the plane normal, the plane distance to the origin and the camera center coordinates in the camera $i$ reference frame. Since $\mathbf{C}_i = [0,0,0]^T$, this expression reduces to

$$d_0 \overset{+}{\sim} d_1. \tag{3}$$

By definition, we know that $d_i = -\mathbf{n}_i^T \mathbf{A}_i$, with $\lambda_{\mathbf{a}_i}\mathbf{A}_i = \mathbf{a}_i$. Determining the plane normal $\mathbf{n}_i$ requires three points (out of the four points provided by $S_h$). Without loss of generality, we take the points of the first three correspondences of $S_h$. Hence,

$$\begin{aligned} -d_i &= \mathbf{n}_i^T \mathbf{A}_i, \\ &= ((\mathbf{B}_i - \mathbf{A}_i) \times (\mathbf{C}_i - \mathbf{A}_i))^T \mathbf{A}_i, \\ &= (\mathbf{B}_i \times \mathbf{C}_i)^T \mathbf{A}_i. \end{aligned}$$

Using (2),

$$-d_i = \lambda_{\mathbf{a}_i}\lambda_{\mathbf{b}_i}\lambda_{\mathbf{c}_i}(\mathbf{b}_i \times \mathbf{c}_i)^T \mathbf{a}_i. \tag{4}$$

Since $\lambda_{\mathbf{a}_i}\lambda_{\mathbf{b}_i}\lambda_{\mathbf{c}_i} > 0$,

$$-d_i \overset{+}{\sim} (\mathbf{b}_i \times \mathbf{c}_i)^T \mathbf{a}_i.$$

Substituting this expression in (3) we get the geometric constraint:

$$(\mathbf{b}_0 \times \mathbf{c}_0)^T \mathbf{a}_0 \overset{+}{\sim} (\mathbf{b}_1 \times \mathbf{c}_1)^T \mathbf{a}_1. \tag{5}$$

Expression (5) holds if and only if the relative order of points $\mathbf{a}_0$, $\mathbf{b}_0$, $\mathbf{c}_0$ and that of points $\mathbf{a}_1$, $\mathbf{b}_1$, $\mathbf{c}_1$ is the same. Figure 2 depicts it graphically. Every subset of three correspondences in $S_h$ must verify eq. (5). Otherwise, $S_h$ should be discarded, since it leads to an invalid homography.

When estimating homographies, in which four correspondences are given, one might apply (5) over the

1. first three correspondences ($v_a$, $v_b$ and $v_c$),
2. all four possible sets of three correspondences, or
3. over some of these four possible sets.

This gives three alternatives for assessing the minimal sets with our constraint. The first two alternatives will be experimentally evaluated in Section 5. From now on we will call *weak* constraint to the first alternative and *strong* constraint to the second one. In Algorithm 2 we improve the standard RANSAC procedure including this test.

**Algorithm 2** Improved RANSAC for homography estimation.

```
for h = 1 to h = it_max do
    Take a subset S_h of hypothetical inliers
    if S_h passes geometric test in (5) then
        Fit a homography M_h to the hypothetical inliers S_h.
        Compute the homography quality u_h
        if u_h > u* then
            Store current homography M_h as the best one.
            u* ← u_h.
        end if
    end if
end for
```

The geometric constraint presented above is linear in the size of $\mathcal{C}$. It requires only 8 multiplications and 11 additions (note that $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ have their third component set to 1). Hence, if a set $S_h$ verifies (5), the computational cost of the improved method (Algorithm 2) is almost equal to the standard one (Algorithm 1). However, when the minimal set does not pass the test, neither the homography fitting nor the validation step have to be performed. Since these are the most expensive steps in the algorithm, we get a drastic reduction of the computational cost in that iteration. In this way, it is possible to speed up the random sampling procedure virtually for free. In the next section we validate this experimentally.

## 4 OpenCV implementation

In this section describe the implementation of the geometric constraints described in the previous section in OpenCV 2.4[1].

Robust model estimation algorithms in OpenCV share a C++ class interface. The implementation of such algorithms belongs to the 3D camera calibration module. The base class defining the interface is `CvModelEstimator2` (see classes hierarchy in Fig. 3).

The class `CvModelEstimator2` implements two robust model estimation techniques: Least Median of Squares (LMedS) (in `CvModelEstimator2::runLMeDS`) and RANSAC (in `CvModelEstimator2::runRANSAC`). Any specialization of the class `CvModelEstimator2` must implement the core routine for model estimation, `runKernel`, and the method `computeReprojError` used in `CvModelEstimator2::findInliers`. The section of code in the

[1] `http://opencv.willowgarage.com`

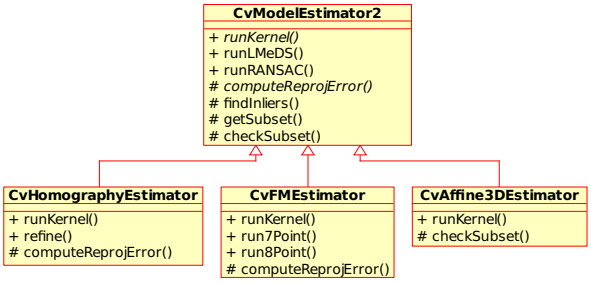



**Fig. 3** OpenCV's UML class hierarchy for robust estimators.

`CvModelEstimator2::runRANSAC` method is shown in Listing 1. The implementation follows that in Algorithm 1 except for the number of remaining RANSAC iterations that is updated taking into account the number of inliers of the last accepted model.

**Listing 1** Code in `CvModelEstimator2::runRANSAC`

```
1  for( iter = 0; iter < niters; iter++ )
   {
     int i, goodCount, nmodels;
     if( count > modelPoints )
     {
6      //Get minimal subsect for model estimation:
       //e.g. 4 correspondences for an homography
       bool found = getSubset(m1,m2,ms1,ms2,300);
       if( !found )
       {
11       if( iter == 0 )
           return false;
         break;
       }
     }
16
     // Core model estimation algorithm: e.g. 4
     // correspondences homography estimation.
     nmodels = runKernel( ms1, ms2, models );
     if( nmodels <= 0 )
21     continue;
     for( i = 0; i < nmodels; i++ )
     {
       // Check for number of inliers
       CvMat model_i;
26     cvGetRows(models,&model_i,i*modelSize.height,
                 (i+1)*modelSize.height);
       goodCount = findInliers(m1, m2,&model_i,err,
                               tmask,reprojThreshold);

31     //If the model has more inliers than the
       // current best model, declare it as the new
       // best one.
       if(goodCount>MAX(maxGoodCount,modelPoints-1))
       {
36       std::swap(tmask, mask);
         cvCopy( &model_i, model );
         maxGoodCount = goodCount;
         //Update the number of remaining iterations
         //as a function of the number of
41       //inliers found so far.
         niters = cvRANSACUpdateNumIters( confidence,
                  (double)(count - goodCount)/count,
                  modelPoints, niters );
       }
46   }
   }
```

At this point we can add our geometrical constraint to OpenCV's RANSAC homography estimation code. We have to check if a selected subset satisfies the constraint. We do so by adding the following code on line 16 in Listing 1:

```
if ( !isMinimalSetConsistent( ms1, ms2 ) )
  continue;
```

This code checks the geometrical constraint and skips the homograhy estimation and inliers checking altogether going to the next RANSAC iteration, if the the set does not satisfy the constraint.

## 5 Experiments

We have performed six synthetic and three real experiments to asses the influence in the performance of RANSAC of the geometric restrictions presented in Section 3. The synthetic experiments let us systematically study the number of fitted homographies depending on the proportion of outliers and noise level. In the real experiments we analyze the actual gains in processing time per image achieved on a mobile device implementation.

### 5.1 Synthetic experiments

In these experiments we analyze the computational cost of the algorithms, measured in terms of the number of fitted homographies, *i.e.* number of RANSAC iterations that satisfy the geometric constraints.

We create the synthetic data as follows. We choose an inlier proportion $p$ and noise level $\sigma$. We then generate an arbitrary homography. This homography is used to synthesize a set of inlier correspondences. We also generate a set of random outliers correspondences. The size of both sets is adjusted according to proportion $p$. The union of the inlier and outlier sets generates the correspondence set $\mathcal{C}$ of the experiment. Finally, we perturb the points in $\mathcal{C}$ by adding a random Gaussian value with mean zero and standard deviation $\sigma$ pixels. Table 1 summarizes the configurations $(p, \sigma)$ of the first four experiments performed.

For the experiment we run RANSAC 5000 times using the correspondence set $\mathcal{C}$. Each of these executions is halted at the iteration in which a good homography (*i.e.*, a homography that fits to at least 85 percent of inliers in $\mathcal{C}$) is found. From each execution we record the required number of fitted homographies using standard RANSAC (Algorithm 1) and the *weak* and *strong*-constrained RANSAC (Algorithm 2). We also compute the number of theoretical iterations with equation (1). In Fig. 4 we show the cumulative proportion of runs that complete at or before a given number of fitted homographies. Looking it another way, the figure represents the probability of finding a good homography for a given number of fitted homographies.

In absence of noise (experiment 1), the number of required fitted homographies in the standard RANSAC matches the theoretical estimate. Since the geometrical constraints filter out many invalid minimal sets, the actual number of fitted homographies is smaller than the theoretical value when the constraints are used.

When noise is present (experiments 2, 3 and 4), the theoretical estimate provided by (1) is optimistic compared to the standard RANSAC algorithm. The higher

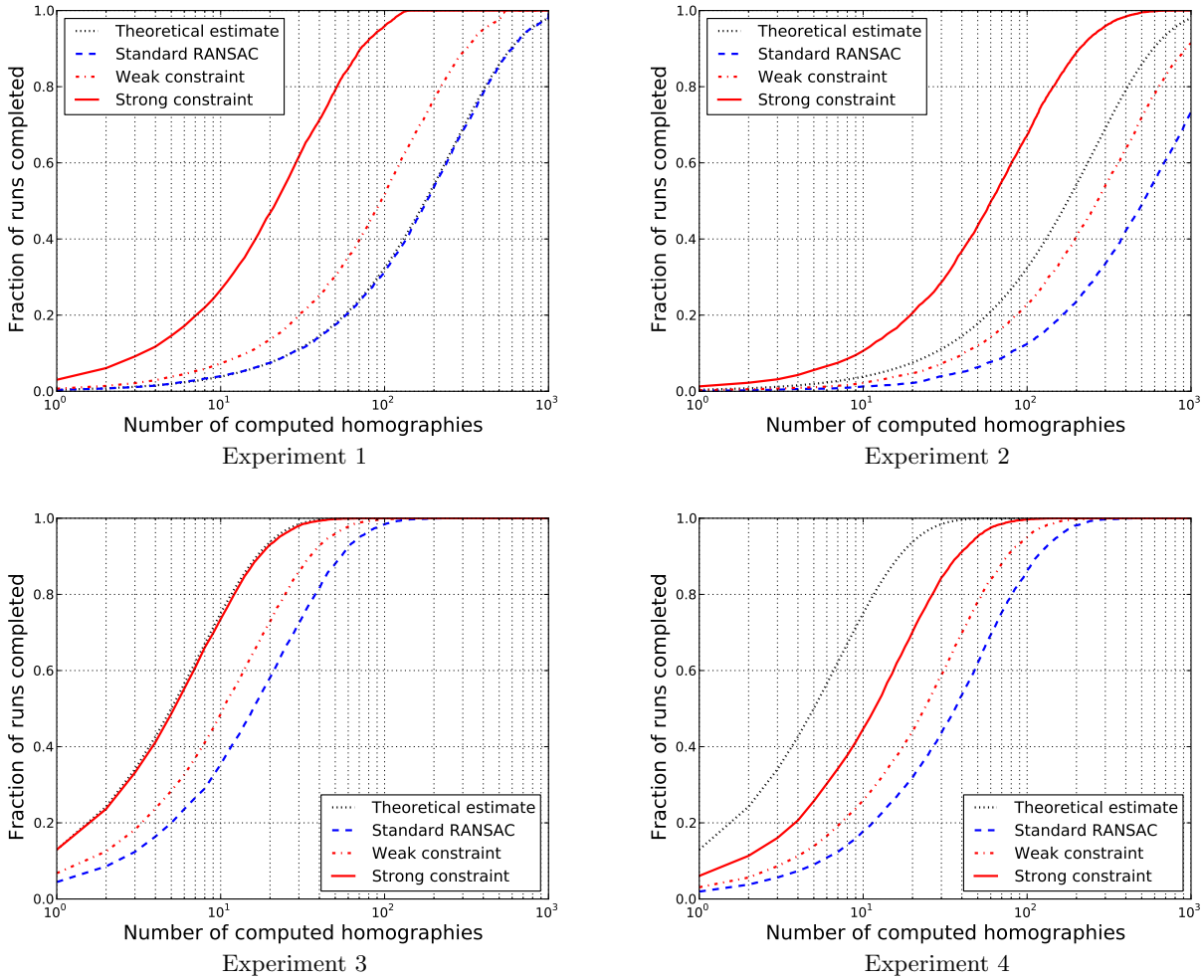



**Fig. 4** Probability of finding a good homography for a given number of fitted homographies. Curves show the differences between the theoretical estimate provided by (1), the standard RANSAC, the *weak* constrained and *strong* constrained RANSAC. The configuration for the experiments is given in Table 1.

**Table 1** Configuration of synthetic experiments. Each experiment has a different value of inlier proportion $p$ and Gaussian noise $\sigma$.

| | **Inlier prop.** | **Noise level** |
|---|---|---|
| Exp. 1 | $p = 0.25$ | $\sigma = 0$ |
| Exp. 2 | $p = 0.25$ | $\sigma = 1$ |
| Exp. 3 | $p = 0.6$ | $\sigma = 1$ |
| Exp. 4 | $p = 0.6$ | $\sigma = 2$ |

**Table 2** Percentage of reduction in the number of computed homographies provided by the *weak* and *strong* constraints.

| | Exp. 1 | Exp. 2 | Exp. 3 | Exp. 4 |
|---|---|---|---|---|
| Weak constraint | 47.48% | 47.74% | 33.64% | 34.27% |
| Strong constraint | 87.66% | 88.13% | 66.26% | 66.80% |

the noise level, the higher the number of iterations required by the standard RANSAC. In this case the geometrical constraints reduce the number of computed homographies to bring it closer to the theoretical prediction. This happens to be the case in experiment 3. The higher the proportion of outliers, the better the results achieved by the geometrical restrictions. In experiments 1 and 2, since the proportion of outliers is highest, the *strong* restriction achieves the best results. On the other hand, in experiment 4, since the noise level is the highest, the theoretical prediction is the most optimistic and the furthest from the reality of the standard RANSAC.

In all cases, the geometrical constraints lead to a reduction in the number of the fitted homographies required by the standard RANSAC. This reduction is about 40% for the weak constraint and about 75% for the *strong*

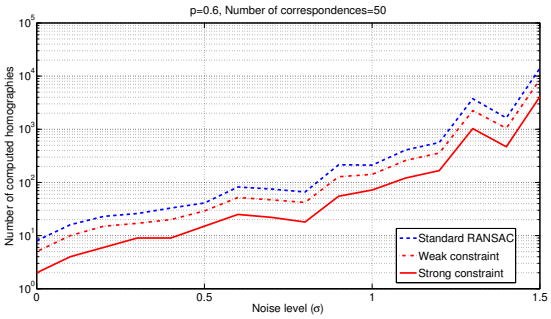



**Fig. 5** Experiment 5. Average number of fitted homographies required to obtain a good fit at different noise levels.

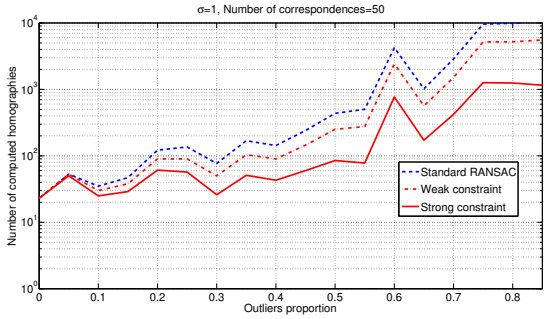



**Fig. 6** Experiment 6. Average number of fitted homographies required obtain a good fit at different percentage of outliers.

constraint. Table 2 gives details of the percentage of improvement achieved in each experiment. For example, in experiment 4, the standard RANSAC requires 153 homography estimations to reach a confidence level of 0.95, while only 50 estimations are needed with the *strong* constraint. This fact is more noticeable when the inlier proportion decreases. Experiment 2, with less noise but a higher outlier proportion, requires 2342 homography fits without the constraints (out of the bounds in Fig. 4) and only 280 with the *strong* constraint, about one order of magnitude less fits.

The number of homographies that have to be fitted to find a good model grows exponentially with noise and outliers. In synthetic experiments 5 and 6 we analyze this number for increasing noise level and outlier proportion respectively. We have run RANSAC 5000 times per noise and outlier value. We stop when a homography with 85 percent of inliers is found and plot the average number of required fitted homographies for the 5000 runs. As we can see in Fig. 5, the number of computed homographies for both the *weak* and *strong*-constrained RANSAC is always smaller than for the standard algorithm when noise increases, although the rate of growth is the same for all algorithms. Results in Fig. 6 confirm again that the geometrically constrained algorithms always require the estimation of less homographies than standard RANSAC. Remarkably, in this case, as the proportion of outliers increases, the rate of growth in the number of required fits is smaller for the constrained algorithms.

In summary, the actual number of required RANSAC iterations $it^*$ is always larger than theoretical number $it_{\max}$ if the point locations are contaminated with noise. The geometric constraints introduced in Section 3 reduce the actual number of homography fits. This reduction is more prominent the larger the proportion of outliers and noise level. Lastly, and most importantly, as the proportion of outliers grows, the rate of growth in this number is notably lower for the geometrically constrained algorithms

## 5.2 Real Experiments

In this Section we perform static experiments with real images (experiments 7 and 8) and dynamic experiments with a large image sequence (experiment 9). Our goal here is to analyze the actual gain in processing time when using the geometric constraints in a resource-limited device, such as a mobile phone.

In these experiments we use the SURF feature detector and descriptor Bay et al (2008) implemented in OpenCV. To compute correspondences between detected features we use the euclidean distance between their 64 element descriptors. We declare a putative correspondence between descriptor $\mathbf{d}_i$ from image $I_1$ and descriptor $\mathbf{d}_j$ from image $I_2$ if the euclidean distance between $\mathbf{d}_i$ and $\mathbf{d}_j$, $d(\mathbf{d}_i, \mathbf{d}_j)$, is less than $0.6 \times d(\mathbf{d}_i, \mathbf{d}_k)$, where $\mathbf{d}_k$ is the second nearest descriptor to $\mathbf{d}_i$ from image $I_2$. Note that finding the nearest neighbors to a given descriptor can be performed very quickly with a kd-tree data structure.

In the first real experiments, numbers 7 and 8, we confirm the synthetic results obtained in Section 5.1. First, we estimate the ground truth homography by using a large number of standard RANSAC iterations. Then, we run RANSAC 5000 times halting each of them at the iteration in which a homography has at least 85 percent of inliers. Since in these experiments we cannot change the percentage of inliers nor the noise level, we have used two images with different characteristics. In experiment 7 we use an image with no clutter in the background (*i.e.* large percentage of inliers, see Fig. 7). For experiment 8 we use an image with clutter in the background (*i.e.* low percentage of inliers, see Fig. 8). In Figs. 7 and 8 we show the probability of finding a good homography after fitting a given number of putative homographies.

In experiment 7, with no clutter in the background, we have a very low amount of outliers (see Fig. 7). The proportion of true inliers is 0.82, 64 out of 74 putative correspondences, which is higher than that in the synthetic experiments (see Table 1). In Fig. 7 (right) we can see that if we want to find a good homography with probability 0.9, standard RANSAC needs to fit 87 homogra-

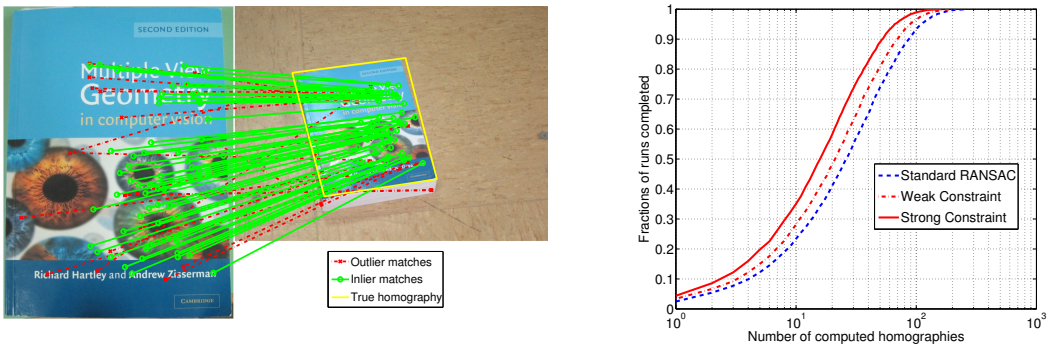



**Fig. 7** Experiment 7. Clean background. On the left we show overlayed in yellow the true homography. The 64 inlier (resp 14 outlier) correspondences are displayed in green (resp in red). On the right, we show the percentage of RANSAC runs (from 5000) that achieved a good homography (85% of the true inliers).

phies, the *weak*-constrained RANSAC needs to estimate 70 whereas the *strong*-constrained only requires 52. This means that the *weak* constraint estimates 20% less homographies than standard RANSAC and the *strong* one 41% less. This is an expected result when the number of inliers is high, since the geometrical constraints are satisfied by most of the $S_h$ sets. However, we still have an improvement in the computational cost with respect to standard RANSAC.

In experiment 8 the proportion of inliers is 0.32 (20 out of 61 putative matches). In Fig. 8 (right) we can see that if we want to find a good homography with probability 0.9, standard RANSAC needs to fit 2363 homographies, the *weak*-constrained RANSAC needs to estimate 1242 whereas the *strong*-constrained only requires 431. In this case, the *weak* constraint estimates 48% less homographies than standard RANSAC and the *strong* one 82% less. These real experiments confirm the results obtained with the synthetic ones and prove that the geometrical constraints notably reduce the number of fitted homographies.

A reduction in the number of estimated homographies will be useful only if the overall computation time is reduced, *i.e.* the constraint computation is cheap. Our last experiment is based on a real image sequence from a museum guide application running on a Samsung Galaxy. The time figures shown only account for the homography fitting part of the application. We have processed a large image sequence with different paintings. We compare the standard `FindHomography` RANSAC routine in OpenCV with the modified versions of the same routine, described in Section 4, using the *weak* and *strong* constraints. The experiment has been executed on a 2005 desktop processor, an AMD Opteron 254 at 2.8GHz, and on Samsung Galaxy's ARM Cortex V8 processor at 1GHz. We provide the results for the desktop processor to put in context the mobile processor's performance. We present results for three sub-sequences: experiment 9.1, with a proportion of inliers of about 0.5 (see Fig. 9), experiment 9.2 varying between low (0.25) and high (0.75) inlier proportion (see Fig. 10) and experiment 9.3 with a low proportion of inliers, around 0.25, in all images.

Experiment 9.1 is an example of an execution when the proportion of outliers is low (0.5). In Table 3 we can see that OpenCV's implementation, which is a standard RANSAC code, doubles the time of the *strong* constraint and, what is more important, the constrained solution is more regular than OpenCV's, since we get half its standard deviation (14.95 ms versus 37.85 ms).

Experiment 9.2 (see Fig. 10) involves a wider range of situations. Initially a poster is located on one side of the image, which includes a large portion of cluttered background (frames 2330 to 2388). Then the camera moves fast and some images are blurred (frames 2388 to 2406). After that, the camera moves slowly and close to the poster (frames 2407 to 2473). Finally, the camera moves away , showing more background (frames 2474 to 2540). Each of the sub-sequences has a different proportion of inliers and shows different behavior of the tested algorithms:

- Frames 2330-2388: Since the background is visible and the poster is small w.r.t. the image, the proportion of inliers is low, around 0.25. In this case the *strong*-constrained algorithm is 4 times faster than OpenCV's RANSAC (see Table 4). Here the standard deviation in the computation time of the constrained algorithm is 5 times lower than OpenCV's implementation, which gives an smoother execution. We thus confirm that, as observed in experiment 7, a reduction in the number of fitted homographies implies an equivalent reduction in the computation time.
- Frames 2389-2406: The camera moves towards the poster and the clutter almost disappears from the background, giving a proportion of inliers about 0.4. Some images are blurred making feature detection difficult and consequently decreasing the inlier proportion to 0.25 (see image 2406 and 2413). When the proportion of inliers is 0.4 the *strong*-constrained algorithm is three times faster than OpenCV's RANSAC

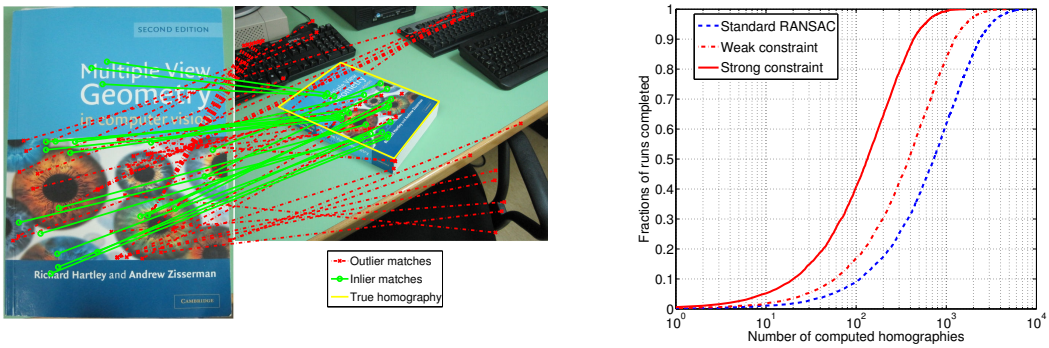



**Fig. 8** Experiment 8. Cluttered background. On the left we show overlayed in yellow the true homography, the 20 inlier (resp 42 outlier) correspondences are displayed in green (resp in red). On the right, we show the percentage of RANSAC runs (from 5000) that achieved a good homography (85% of the true inliers).

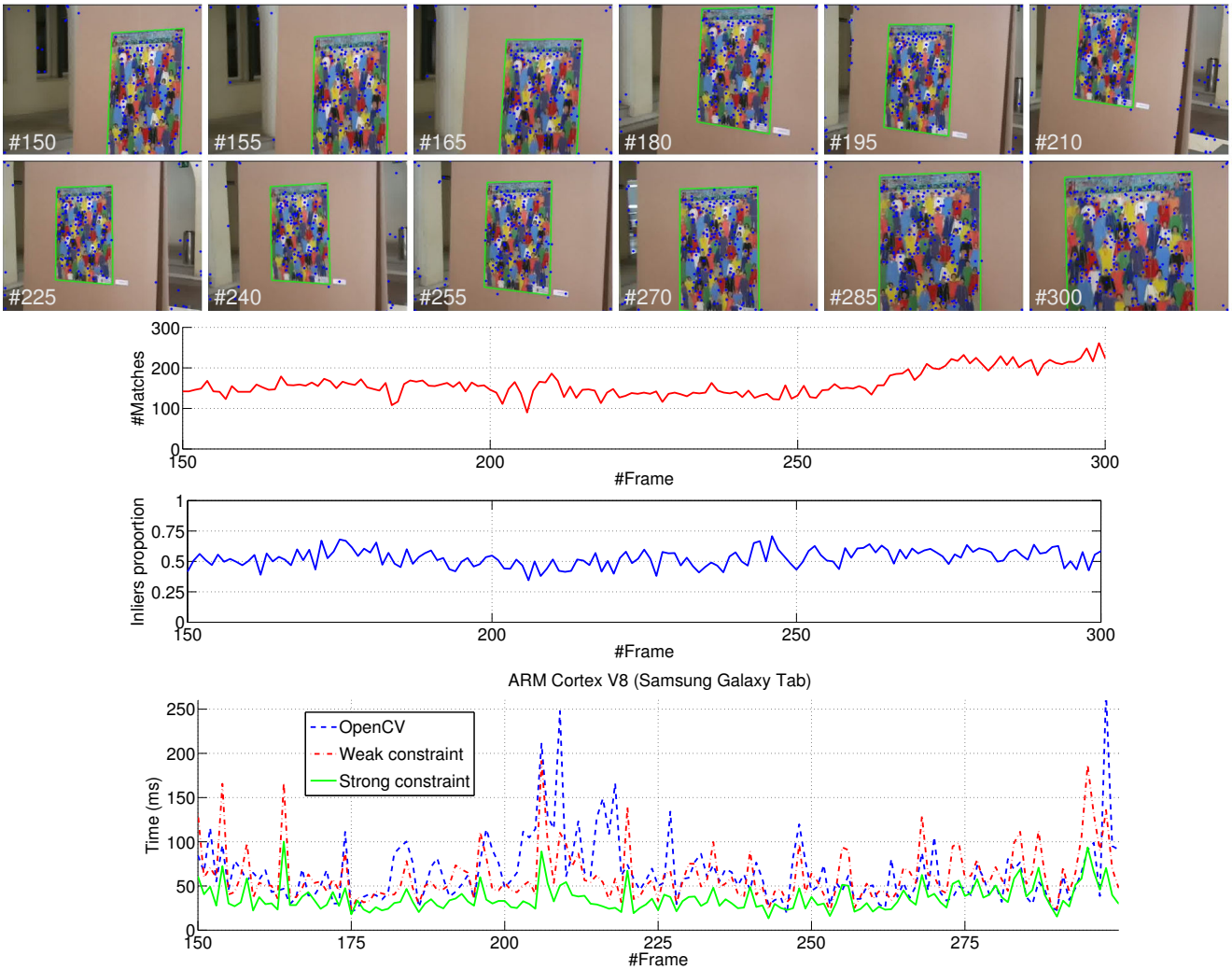



**Fig. 9** Experiment 9.1. Real sequence experiment. In the first two rows, we show the estimated homography overlayed on the input images (in green). Features belonging to putative matches are shown in blue. In the third row we show the number of putative correspondences (matches) per frame. In the fourth row we display the percentage of inliers. Finally, in the last row we show the execution times for RANSAC homography estimation with OpenCV's `FindHomography` and the two alternatives for the geometrical constraint.

**Table 3** Experiment 9.1. Time to fit a homography on an ARM Cortex V8, in milliseconds per frame.

| frames | algorithm | mean (ms) | median (ms) | std (ms) |
|---|---|---|---|---|
| 150-300 | OpenCV | 65.94 | 57.89 | 37.85 |
| | Strong constraint | 36.41 | 31.93 | 14.95 |

(see Table 4). The standard deviation is 10 times lower with the constrained algorithm.

- Frames 2407-2473: The poster is so close to the camera that only a part of it is visible. It has good resolution. In this situation the feature detector finds more feature points and, hence, the number of putative matches is increased (more than 200). This also boost the proportion of inliers to 0.75. The execution times show that the *strong*-constrained algorithm is still 1.61 times faster than OpenCV's RANSAC implementation. Here again the standard deviation is 2.5 times lower (see Table 4).
- Frames 2474-2540: In the last part of the sequence the poster is far from the camera and the background clutter is visible. This scene configuration makes the proportion of inliers fall below 0.25. The geometric-constrained homography estimation is 4.5 times faster than OpenCV's. The standard deviation of the constrained method is 5.65 times lower (see Table 4).

In the last sub-sequence, experiment 9.3, we have a moving camera that performs fast in-plane rotations (see Fig. 11). The inlier proportion ranges from 0.14 to 0.48 with a mean value of 0.32. In this case the speed-up for the *strong*-constrained RANSAC over OpenCV's `FindHomography` ranges from 2.11 times (frames 3591 to 3606) to 3.7 times (frames 3521 to 3590) (see Table 5). This is a difficult but typical example of planar localization in which the background is cluttered and the camera is rotating and moving fast. This kind of camera motion is difficult for the feature detectors and therefore the proportion of inliers decreases. In such cases the proposed constraints are a must to cope with the computational burden introduced by the large amount of outliers.

In this section we have shown that the constrained RANSAC algorithm introduced in this paper consistently outperforms the standard RANSAC implementation in OpenCV's `FindHomography`. We have shown experimentally that the reduction in the number of fitted homographies has an equivalent reduction in computation time, proving that the constraints are cheap to compute. Moreover, an important requirement to achieve a constant frame rate on a resource-limited device is a balanced processor workload over time. A varying number of matches and, most importantly, of outliers, may slow down the performance of a standard homography estimation algorithm to values above 1 sec per frame on an ARM Cortex V8 processor. The geometrically-constrained RANSAC algorithm introduced in this paper not only achieves the lowest fitting time, but also the lowest variance and consequently the best balanced processor workload over time.

With the geometrical restriction we give more time to other vision application tasks. In an ARM Cortex V8 the average values for our RANSAC homography estimation times vary between 36 and 166 ms per frame with a proportion of inliers between 0.6 and 0.2. On the other hand, in the same conditions, OpenCV's `FindHomography` routine takes between 65 and 629 ms, on average. In certain specially adverse situations, the performance of the geometrically constrained RANSAC is about one order of magnitude faster than OpenCV's.

## 6 Conclusions

The main contribution of this work is a geometric constraint for faster homography estimation with random sampling. It filters out minimal sets of correspondences that will not lead to valid homographies. In extreme situations with large noise and outlier proportion, the constraint is a necessary requirement for achieving real-time performance on a mobile device.

In the synthetic experiments conducted we have proved that the usual expression used for estimating the number of iterations in RANSAC (1) is optimistic when noise contaminates feature positions. In this case, the geometric constraint brings the actual number of iterations closer to the theoretical prediction. If noise is sufficiently low, the geometrically-constrained RANSAC may outperform the theoretical prediction. As noise and outliers grow, the geometrically constrained algorithms always require less iterations. Most importantly, when the proportion of outliers increases, the rate of growth in the number of iterations of the geometrically-constrained RANSAC is clearly below that of the standard algorithm.

The real experiments on an ARM Cortex V8 1Ghz prove that the geometric constraint introduces very little computational overhead in the algorithm. Consequently, fitting a homography with a geometrically constrained RANSAC always requires less time than with the standard algorithm. The variance is also smaller, so it better balances the processor workload over time. This is a key feature for avoiding stalls in augmented reality applications on resource-limited devices.

The code with the improved OpenCV's `FindHomography` may be downloaded from the aditional material in the journal and from the web page `http://www.dia.fi.upm.es/~pcr/fast_homography.html`.

**Acknowledgements** The authors gratefully acknowledge funding from Cenit project "mIO!:Tecnologías para prestar servicios en movilidad en el futuro universo inteligente" and the

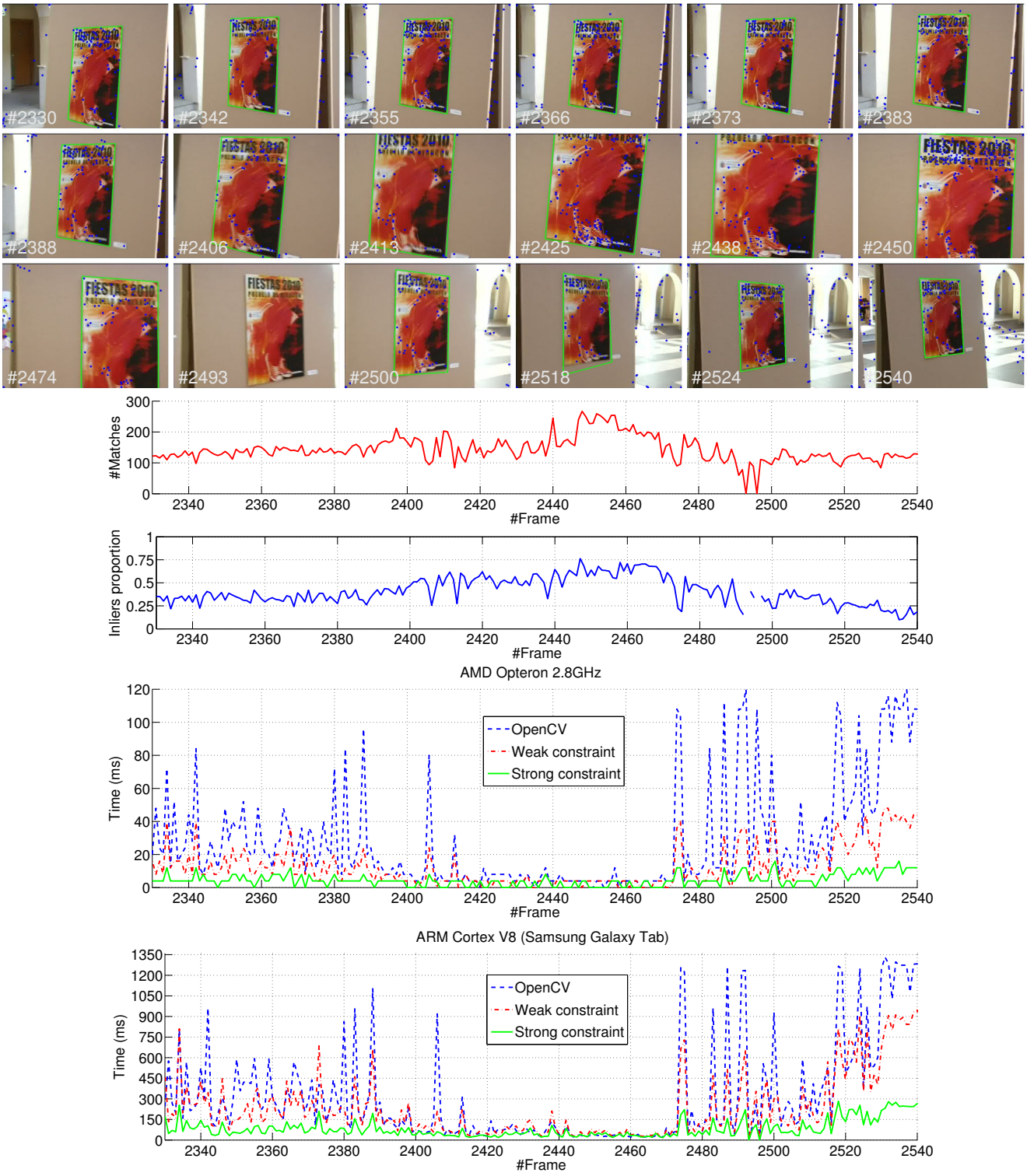



**Fig. 10** Experiment 9.2. Real sequence experiment. In the first three rows, we show the estimated homography overlayed on the input images (in green). The features belonging to putative matches are shown in blue. In the fourth row we show the number of putative correspondences (matches) per frame. In the fifth row we display the percentage of inliers. Finally, in the last two rows we show the execution times for RANSAC homography estimation with OpenCV's `FindHomography` and the two alternatives for the geometrical constraint.

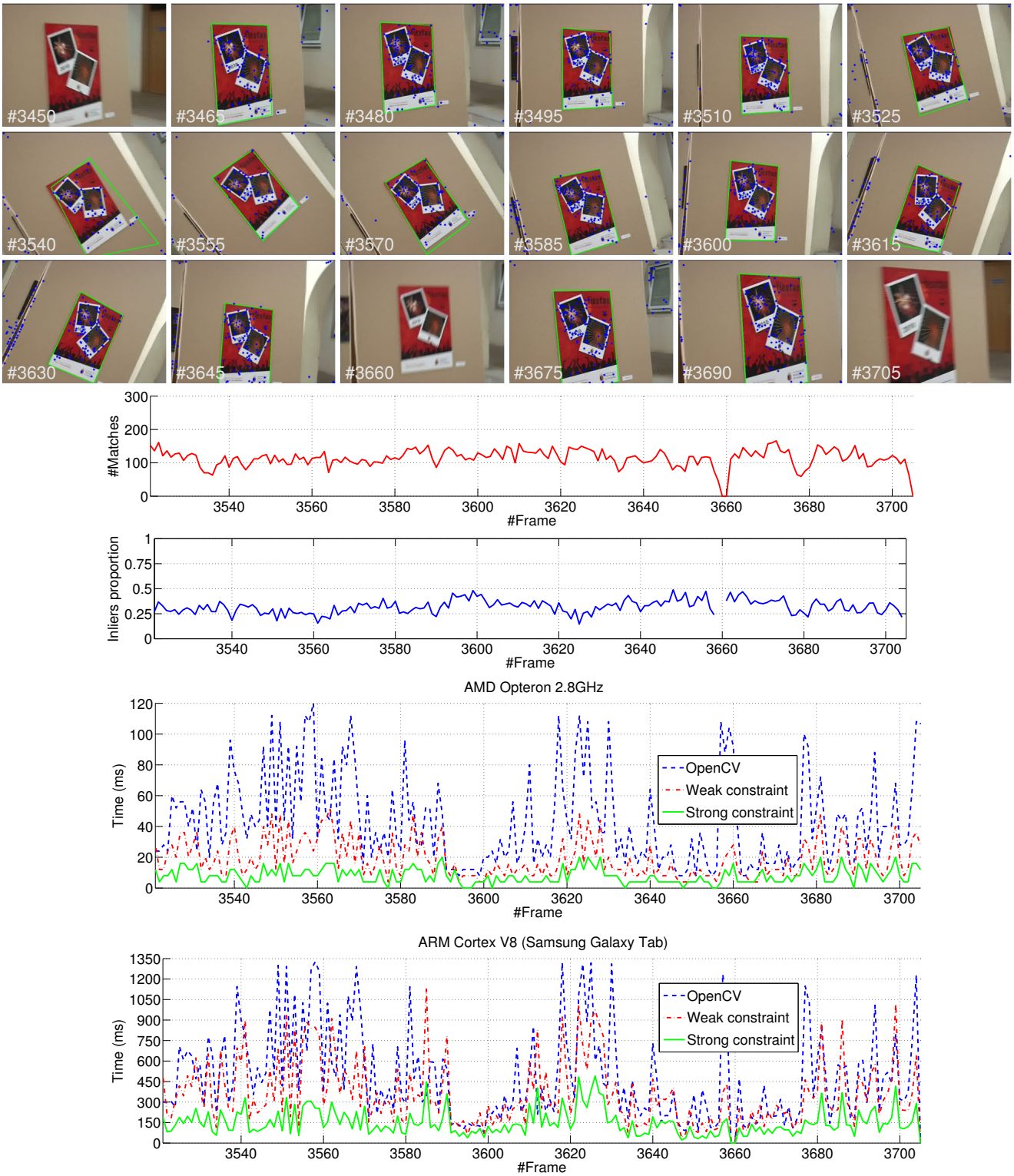



**Fig. 11** Experiment 9.3. Real sequence experiment. In the first three rows, we show the estimated homography overlayed over the input images (in green). The features belonging to putative matches are shown in blue. In the fourth row we show the number of putative correspondences (matches) per frame. In the fifth row we display the percentage of inliers. Finally, in the last two rows we show the execution times for RANSAC homography estimation with stock OpenCV's `FindHomography` and the two alternatives for the geometrical constraint.

**Table 4** Experiment 9.2. Time to fit a homography on an ARM Cortex V8, in milliseconds per frame.

| frames | algorithm | mean (ms) | median (ms) | std (ms) |
|---|---|---|---|---|
| 2330-2388 | OpenCV | 356.16 | 290.01 | 226.42 |
| | Strong constraint | 86.00 | 79.84 | 44.68 |
| 2389-2406 | OpenCV | 169.11 | 110.82 | 197.86 |
| | Strong constraint | 57.06 | 54.43 | 19.35 |
| 2407-2473 | OpenCV | 56.45 | 41.70 | 44.81 |
| | Strong constraint | 38.10 | 32.74 | 17.72 |
| 2474-2540 | OpenCV | 585.14 | 439.79 | 469.69 |
| | Strong constraint | 129.73 | 111.61 | 83.33 |

**Table 5** Experiment 9.3. Time to fit a homography on an ARM Cortex V8, in milliseconds per frame.

| frames | algorithm | mean (ms) | median (ms) | std (ms) |
|---|---|---|---|---|
| 3521-3590 | OpenCV | 629.41 | 576.42 | 321.34 |
| | Strong constraint | 166.51 | 140.43 | 85.70 |
| 3591-3606 | OpenCV | 179.75 | 149.73 | 76.10 |
| | Strong constraint | 84.83 | 77.48 | 29.52 |
| 3607-3705 | OpenCV | 441.91 | 324.06 | 335.11 |
| | Strong constraint | 148.19 | 127.55 | 105.69 |

Spanish *Ministerio de Ciencia e Innovación* under contract `TIN2010-19654` and the *Consolider Ingenio Program* under contract `CSD2007-00018`. Pablo Márquez-Neila was funded by the *Programa Personal Investigador de Apoyo* from the *Comunidad de Madrid*.

## References

Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Computer Vision and Image Understanding 110(3):346 – 359

Bruns E, Bimber O (2009) Adaptive training of video sets for image recognition on mobile phones. Personal Ubiquitous Comput 13:165–178

Bruns E, Brombach B, Zeidler T, Bimber O (2007) Enabling mobile phones to support large-scale museum guidance. IEEE MultiMedia 14:16–25

Castells M (1996) The Rise of the Network Society. Blackwell Publishers

Cheng CM, Lai SH (2009) A consensus sampling technique for fast and robust model fitting. Pattern Recognition 42:1318–1329

Chum O, Matas J (2002) Randomized RANSAC with $T_{d,d}$ test. In: Proc. of British Machine Vision Conference, vol 2, pp 448–457

Chum O, Matas J (2005) Matching with PROSAC - progressive sample consensus. In: Proc. of Int. Conf. on Computer Vision and Pattern Recognition, vol 1, pp 220 – 226

Chum O, Werner T, Matas J (2004) Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In: Proc. of the Int. Conf. on Pattern Recognition, pp 112–115

Fishler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM (6):381–395

Guerrero J, Martínez-Cantin R, Sagüés C (2005) Visual map-less navigation based on homographies. Journal of Robotic Systems 22(10):569–581

Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press

Henze N, Schinke T, Boll S (2009) What is that? object recognition from natural features on a mobile phone. In: Mobile Interaction with the Real World

Kim K, Lepetit V, Woo W (2010) Scalable real-time planar targets tracking for digilog books. Computer Graphics International

Kirchhof M (2008) Linear constraints in two-view multiple homography estimation of uncalibrated scenes. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol XXXVII, p B3a

Klein G, Murray D (2009) Parallel tracking and mapping on a camera phone. In: Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality, Orlando

Laveau S, Faugeras OD (1996) Oriented projective geometry for computer vision. In: Proc. of the European Conf. on Computer Vision, pp 147–156

Lee W, Park Y, Lepetit V, Woo W (2010) Point-and-shoot for ubiquitous tagging on mobile phones. In: Proc. of the International Symposium on Mixed and Augmented Reality

López-Nicolás G, Gans N, Bhattacharya S, Sagüés C, Guerrero J, Hutchinson S (2010a) Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40(4):1115 –1127

López-Nicolás G, Guerrero J, Sagüés C (2010b) Multiple homographies with omnidirectional vision for robot homing. Robotics and Autonomous Systems 58(6):773 – 783

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int Journal of Computer Vision 2(60):91–110

Meer P, Stewart CV, Tyler DE (2000) Robust computer vision: An interdisciplinary challenge. Computer Vision and Image Understanding 78:1–7

Pielot M, Henze N, Nickel C, Menke C, Samadi S, Boll S (2008) Evaluation of camera phone based interaction to access information related to posters. In: Proc. of Mobile Interaction with the Real World

Roblee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. In: Proc. International Conference on Computer Vision, pp 2564 – 2571

Rohs M, Gfeller B (2004) Using camera-equipped mobile phones for interacting with real-world objects. In: Ad-

vances in Pervasive Computing, pp 265–271
Rousseeuw PJ (1984) Least median of squares regression. Journal of the American Statistical Association 79:871–880
Takacs G, Chandrasekhar V, Gelfand N, Xiong Y, Chen WC, Bismpigiannis T, Grzeszczuk R, Pulli K, Girod B (2008) Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In: Proc. ACM Int. Conf. on Multimedia Information Retrieval, MIR'08, pp 427–434
Taylor S, Drummond T (2009) Multiple target localisation at over 100 fps. In: Proc. British Machine Vision Conference
Taylor S, Rosten E, Drummond T (2009) Robust feature matching in 2.3ms. In: CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond
Tell D, Carlsson S (2002) Combining appearance and topology for wide baseline matching. In: Proc. of the European Conference on Computer Vision, pp 68–81
Tordoff BJ, Murray DW (2005) Guided-MLESAC: Faster image transform estimation by using matching priors. IEEE Transactions Pattern Analysis Machine Intelligence 27(10):1523–1535
Torr P, Zisserman A (2000) MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78:138–156
Torr PHS, Davidson C (2000) IMPSAC: Synthesis of importance sampling and random sample consensus. In: Proc. of European Conf. on Computer Vision, pp 819–833
Wagner D, Reitmayr G, Mulloni R, Drummond T, Schmalstieg D (2008) Pose tracking from natural features on mobile phones. In: Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality
Xiong Y, Pulli K (2010) Fast image stitching and editing for panorama painting on mobile phones. In: IEEE International Workshop on Mobile Vision