

Fast Target Recognition on Mobile Devices: Revisiting Gaussian Elimination for the the Estimation of Planar Homographies

Olexa Bilaniuk, Hamid Bazargani, Robert Laganière

*School of EECS - University of Ottawa
Ottawa, ON, Canada, K1N 6N5
laganier@eecs.uottawa.ca*

Abstract—This work analyzes the problem of homography estimation for robust target matching in the context of real-time mobile vision. We present a device-friendly implementation of the Gaussian Elimination algorithm and show that our optimized approach can significantly improve the homography estimation step in a hypothesize-and-verify scheme. Experiments are performed on image sequences in which both speed and accuracy are evaluated and compared with conventional homography estimation schemes.

I. INTRODUCTION

Object recognition is an important task of machine vision. The past few years have seen important progress in this direction. Of particular interest are the recent methods for fast feature point matching among which are BRIEF [1], FREAK [2], BRISK [3] and ORB [4]. These allow reliable matching of feature points at very low computational costs. Their secret sauce resides in their use of binary descriptors to represent a patch surrounding a keypoint. Matching then involves only simple binary operators which can be computed very efficiently on modern CPU. However, the recognition of multiple objects at frame rate and at full resolution on devices with limited resources such as mobile phones remains an important algorithmic challenge.

Planar targets are particularly interesting for recognition because, in this case, the different views are related by a 2D homographic transformation. Real-time recognition in video is generally achieved using a matching framework in which keypoints detected in each frame are matched with the ones associated with a reference view of the planar target. The resulting set of putative correspondences is then validated through a robust estimation scheme that aims at identifying a plausible homography mapping the current target view to its reference image.

In this work, we analyze the problem of model parameter estimation in robust target matching. This process represents one of the key steps in planar target recognition and we demonstrate through our experimentation that, in this context, reliable homography estimates can be obtained using a device-friendly implementation of the Gaussian Elimination algorithm. More specifically, we show, in this paper, that our optimized approach can improve by a factor of 20

the homography estimation step in a hypothesize-and-verify scheme.

The rest of this paper is organized as follows. Section 2 presents a review of planar target recognition based on robust homography estimation. Section 3 describes the proposed homography estimation algorithm. Section 4 presents the target recognition framework used to test our homography estimation scheme. Experimental results are presented in Section 5. Section 6 is a conclusion.

II. FAST RECOGNITION OF PLANAR TARGETS

Efficient recognition of planar targets is achieved by building, during an off-line phase, a rich model of a planar target that will then be used to reliably detect this target in live video under a wide range of viewpoints. In [5], the model is built by considering random pair of pixel locations inside a defined neighborhood around the keypoint. A simple binary test is performed to compare the intensity values of those pixels leading to binary features that are grouped together to create small binary space partitions called Ferns. By generating thousand of viewpoints, the training phase estimates the class conditional probability of each Fern for each keypoint. The likelihood of a new patch to correspond to a patch in the model can then be computed by assuming independence of these estimated probability distribution. The BRIEF [1] feature point descriptor also uses the concept of intensity comparison to generate a binary string describing a keypoint patch. ORB [4] is a variant of BRIEF to make it more robust to noise and rotationally invariant. In both cases, Hamming distance is then simply used for evaluating the similarity of two putative matches. In their Histogrammed Intensity Patch method, Taylor et al. [6] introduced the idea of grouping the generated random views into viewpoint bins. The model is built by first computing coarse histograms of the intensities of selected pixels around a keypoint for each viewpoint bin. Once computed these histograms are binarized by identifying bins that are rarely hit with the idea that corresponding patches in the live view should have a small number of pixel values falling into these bits.

During the matching phase, these fast algorithms produce a large set of putative matches. Depending on the quality of

the target descriptors, this match set will be contaminated by a more or less large number of false matches (outliers). This is where robust model estimation comes into play. In the case of a planar targets, the model to be estimated is the two-view homography between the current frame and the reference target. The most commonly used techniques to robustly estimate a model from data contaminated by outliers is RANSAC [7]. It proceeds by randomly selecting a small subset from the data from which a model is hypothesized. This model is then verified against all data points in the set in order to compute its support that is the number of data points that are in agreement with the hypothesized geometric model. By repeating several times this process with different random sampling of the data set, a solution with a high support should be found.

Several variants of the basic RANSAC scheme have been proposed in the past that aim at improving the efficiency of the algorithm. Among the most notable ones is the PROSAC algorithm [8] that observed that by sampling the data from the best quality matches, the probability of hitting a valid model is increased. The ARRISAC [9] framework has been designed to provide accurate estimates in spite of the fixed time budget that imposes a real-time application. It operates by generating a first set of candidate hypotheses that are evaluated on a subset of the data. Based on the current estimate of the inlier ratio, additional hypotheses are then adaptively generated from the highest quality matches. More recently, [10] introduced the USAC framework which proposes a unified package that incorporates the most recent development in RANSAC-based robust estimation. It includes a number of key elements for building a computationally efficient solution and thus offers an ideal tool for benchmarking new approaches.

To achieve fast recognition performance on a low-power mobile device, high efficiency and low computational cost of each of these tasks is crucial. The contribution of this paper resides in the hypothesis generation step. We show that the robust estimation of a homography can be significantly improved by using a computationally efficient implementation of the well-known Gaussian Elimination method to solve the underlying set of equations. Additionally, we show that in the context of target recognition, the solutions found provides reliable estimates of the homography with an accuracy comparable to solutions based on the more commonly used Singular Value Decomposition.

We tested our homography estimation implementation in a target recognition framework that uses FAST9 keypoints described by the BRIEF descriptor. Our robust homography estimation approach follows a PROSAC scheme. It is however important to note that any other approaches in which a target is recognized by estimating a homography from a contaminated set of tentative correspondences could have been considered and would benefit from our efficient implementation.

III. HOMOGRAPHY ESTIMATION BY GAUSSIAN ELIMINATION

A homography is a plane-to-plane relation in a projective space. It is algebraically defined by a 3×3 matrix H mapping two views of a planar object. If we define X and x to be the projective coordinates of the same point respectively on the reference planar target and on an image of it, the 2D homography transformation is then defined as:

$$X = Hx \quad (1)$$

This equality being up to a scale factor, the homography has 8 degrees of freedom and can consequently be computed from four point correspondences. The corresponding homogeneous system of equations is solved through the Direct Linear Transform algorithm by posing:

$$X_i \times Hx_i = 0 \quad (2)$$

in which H is computed using the source points (x_i, y_i) and target points (X_i, Y_i) . This equation can then be rewritten as:

$$\begin{pmatrix} \mathbf{0}^T & -\mathbf{x}_i^T & Y_i \mathbf{x}_i^T \\ \mathbf{x}_i^T & \mathbf{0}^T & -X_i \mathbf{x}_i^T \\ -Y_i \mathbf{x}_i^T & X_i \mathbf{x}_i^T & \mathbf{0}^T \end{pmatrix} \mathbf{h} = \mathbf{0} \quad (3)$$

Which results in equations of the form $\mathbf{A}_i \mathbf{h} = \mathbf{0}$, with \mathbf{A}_i being the lines of the left matrix and \mathbf{h} being a 9×1 vector made of the entries of the homography matrix. The typical approach to solve this system of equation is to use Singular Value Decomposition. This technique is particularly useful when more point correspondences are available, in which case SVD will identify the optimal least-square algebraic solution. Other more computationally expensive (and iterative) approaches could also be used to obtain a geometrically optimal solution [11].

Even if the SVD estimation from four point correspondences in a RANSAC-based framework can be performed with a relative efficiency, its repetitive computation can still impose a significant computational load in the context of real-time estimation using low-power devices. This observation leads us to consider simpler approaches to resolve the 4-point homography estimation problem. In particular, we selected the well-know Gaussian Elimination scheme that can be used to solve the non-homogeneous 4-point set of equations. Even if this approach is known to be less numerically stable, we show here that in the context of target recognition, stable and accurate solutions can still be obtained.

Our implementation of the reduction to reduced-row-echelon form of the matrix is summarized here. It assumes that the minimum configuration is used to estimate the homography, that is 4 matches. If we take the matrix in

(3) to be decomposed as (after appropriate row shuffling):

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0X_0 & -y_0X_0 & X_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 & X_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 & X_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -y_3X_3 & X_3 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0Y_0 & -y_0Y_0 & Y_0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 & Y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 & Y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3Y_3 & -y_3Y_3 & Y_3 \end{pmatrix} \quad (4)$$

We notice here that the matrix is somewhat sparse, and what's more, the top left 4×3 matrix minor is identical to the bottom middle 4×3 minor. This is of great help, since it means that initially, the same operations will be applied to the top 4 rows and bottom 4 rows of the matrix. Even better, when 4-lane or 8-lane vector processing engines (such as SSE, AVX, AltiVec or NEON) are available, the loads of x_i , X_i , y_i and Y_i , the multiplies xX , xY , yX and yY and the row operations can be done in parallel.

We now subtract rows 2 and 6 from the rows 0, 1, 3 and 4, 5, 7 respectively, thus eliminating almost all 1's in column 2 and 5. Since we choose not to scale the rows containing said 1's, they will remain unaffected throughout the remainder of the computation and therefore no storage needs to be reserved for them.

$$\sim \begin{pmatrix} x_0 - x_2 & y_0 - y_2 & 0 & 0 & 0 & 0 & x_2X_2 - x_0X_0 & y_2X_2 - y_0X_0 & X_0 - X_2 \\ x_1 - x_2 & y_1 - y_2 & 0 & 0 & 0 & 0 & x_2X_2 - x_1X_1 & y_2X_2 - y_1X_1 & X_1 - X_2 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 & X_2 \\ x_3 - x_2 & y_3 - y_2 & 0 & 0 & 0 & 0 & x_2X_2 - x_3X_3 & y_2X_2 - y_3X_3 & X_3 - X_2 \\ 0 & 0 & 0 & x_0 - x_2 & y_0 - y_2 & 0 & x_2Y_2 - x_0Y_0 & y_2Y_2 - y_0Y_0 & Y_0 - Y_2 \\ 0 & 0 & 0 & x_1 - x_2 & y_1 - y_2 & 0 & x_2Y_2 - x_1Y_1 & y_2Y_2 - y_1Y_1 & Y_1 - Y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 & Y_2 \\ 0 & 0 & 0 & x_3 - x_2 & y_3 - y_2 & 0 & x_2Y_2 - x_3Y_3 & y_2Y_2 - y_3Y_3 & Y_3 - Y_2 \end{pmatrix} \quad (5)$$

We note here that at this stage, of the 72 potential floating-point values in the matrix, only 32 (excluding the two remaining 1's) are distinct and non-zero. This neatly fits in half of a vector register file with 16 4-lane registers, a common configuration in most modern architectures.

For brevity, after this point only the row operations are given. They were designed to delay the use of reciprocals as long as possible. and the first part is duplicated on both top and bottom half.

First we eliminate column 0 of rows 1 and 3:

$$\begin{aligned} \vec{R}_1 &= r_{1,x} * \vec{R}_1 - r_{1,x} * \vec{R}_0, & \text{idem on } \vec{R}_5 \\ \vec{R}_3 &= r_{0,x} * \vec{R}_3 - r_{3,x} * \vec{R}_0, & \text{idem on } \vec{R}_7 \end{aligned}$$

We eliminate column 1 of rows 0 and 3.

$$\begin{aligned} \vec{R}_0 &= r_{1,y} * \vec{R}_0 - r_{0,y} * \vec{R}_1, & \text{idem on } \vec{R}_4 \\ \vec{R}_3 &= r_{1,y} * \vec{R}_3 - r_{3,y} * \vec{R}_1, & \text{idem on } \vec{R}_7 \end{aligned}$$

We eliminate columns 0 and 1 of row 2.

$$\begin{aligned} \vec{R}_0 &= \frac{1}{r_{0,x}} * \vec{R}_0, & \text{idem on } \vec{R}_4 \\ \vec{R}_1 &= \frac{1}{r_{1,y}} * \vec{R}_1, & \text{idem on } \vec{R}_5 \\ \vec{R}_2 &= \vec{R}_2 - (r_{2,x} * \vec{R}_0 + r_{2,y} * \vec{R}_1), & \text{idem on } \vec{R}_6 \end{aligned}$$

Columns 0-5 of rows 3 and 7 are zero, and the matrix now resembles this:

$$\begin{pmatrix} 1 & & & & & & 0 & a_{06} & a_{07} & a_{08} \\ & \ddots & & & & & & a_{16} & a_{17} & a_{18} \\ & & 1 & 0 & & & & a_{26} & a_{27} & a_{28} \\ & & 0 & 0 & & & & a_{36} & a_{37} & a_{38} \\ & & 0 & 1 & & & & a_{46} & a_{47} & a_{48} \\ & & & & \ddots & & & a_{56} & a_{57} & a_{58} \\ & & & & & 1 & & a_{66} & a_{67} & a_{68} \\ 0 & & & & & & 0 & a_{76} & a_{77} & a_{78} \end{pmatrix} \quad (6)$$

Let's now cease treating the matrix as two independent 4×9 halves and now consider the rightmost three columns as one 8×3 matrix. We use the barren rows 3 and 7 to eliminate columns 6 and 7, thus:

First, we normalize row 7.

$$\vec{R}_7 = \frac{1}{r_{76}} * \vec{R}_7$$

We eliminate column 6 of rows 0 through 6.

$$\begin{aligned} \vec{R}_0 &= \vec{R}_0 - r_{06} * \vec{R}_7 & \vec{R}_1 &= \vec{R}_1 - r_{16} * \vec{R}_7 \\ \vec{R}_2 &= \vec{R}_2 - r_{26} * \vec{R}_7 & \vec{R}_3 &= \vec{R}_3 - r_{36} * \vec{R}_7 \\ \vec{R}_4 &= \vec{R}_4 - r_{46} * \vec{R}_7 & \vec{R}_5 &= \vec{R}_5 - r_{56} * \vec{R}_7 \\ \vec{R}_6 &= \vec{R}_6 - r_{66} * \vec{R}_7 \end{aligned}$$

We normalize row 3.

$$\vec{R}_3 = \frac{1}{r_{37}} * \vec{R}_3$$

We eliminate column 7 of rows 0 through 2 and 4 through 6.

$$\begin{aligned} \vec{R}_0 &= \vec{R}_0 - r_{07} * \vec{R}_3 & \vec{R}_1 &= \vec{R}_1 - r_{17} * \vec{R}_3 \\ \vec{R}_2 &= \vec{R}_2 - r_{27} * \vec{R}_3 & \vec{R}_4 &= \vec{R}_4 - r_{47} * \vec{R}_3 \\ \vec{R}_5 &= \vec{R}_5 - r_{57} * \vec{R}_3 & \vec{R}_6 &= \vec{R}_6 - r_{67} * \vec{R}_3 \end{aligned}$$

The last column of the matrix now contains the homography, normalized by setting $h_{22} = 1$:

$$\sim \begin{pmatrix} 1 & & & & & & h_{00} \\ & \ddots & & & & & h_{01} \\ & & 1 & 0 & & & h_{02} \\ & & 0 & 0 & & 1 & h_{21} \\ & & 0 & 1 & & & h_{10} \\ & & & & \ddots & & h_{11} \\ & & & & & 1 & h_{12} \\ 0 & & & & & 0 & 1 & h_{20} \end{pmatrix} \rightarrow \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{pmatrix}$$

With this non-homogeneous solution, poor estimation would be obtained if the element h_{22} should actually have a value close to zero. Gaussian elimination is however numerically stable for diagonally dominant or positive-definite matrices. For general matrices, Gaussian elimination is usually considered to be stable, when using partial pivoting [12]. In practice, we observed reliable stability when the Z -component of the translation is significant with respect to the $X - Y$ ones; this is the common situation when a hand-held device is used for target recognition.

IV. ROBUST TARGET MATCHING FRAMEWORK

The proposed homography estimation algorithm has been designed for real-time planar target matching using hand-held devices. The first step is to extract features from a video frame. These are then compared with the features of a reference target model resulting in a set of putative matches. In our implementation, we used the FAST9 feature detector [13] and the BRIEF binary descriptor [1] for fast performance. In order to improve the robustness of the matching, we synthesize multiple views of the reference target by warping it using different random perspective transforms. Following the scheme proposed by Taylor et al. [6], we group these generated views into viewpoint bins from which stable keypoints are identified. A majority vote is then applied on each the individual descriptors of corresponding keypoints to produce a top-level descriptor used during the matching phase as in [14].

A planar target is detected by using a RANSAC scheme based on a model hypothesize-and-verify loop in which each iteration implies a homography estimation step and its parameters are evaluated using our Gaussian Elimination algorithm. We chose to use here the PROSAC variant [8] in which samples are selected from the ordered set of correspondences based on their Hamming distance. Compared to the typical RANSAC method, PROSAC greatly reduces the number of attempts for the best support hypothesis, in the sense that it is more likely to choose outlier-free samples from the subset of highest score correspondences. Algorithm (1) summarizes our PROSAC implementation for

fast homography estimation. For the stopping criterion, we selected the SPRT test described in [15] and also used in [10].

Algorithm 1 PROSAC for H estimation

Require: Set of putative matches with corresponding matching score and a level of confidence η_0 .
Initialize $I_{best} := 0$ and k_{MAX} with the maximum allowed value.
Sort the set with respect to the similarity score.
for $i = 1$ to k_{MAX} **do**
 Select minimal samples (4 pairs) from the highest scored correspondences.
 $H \leftarrow$ Generate a hypothesis using the selected pairs.
 $I_k \leftarrow$ Evaluate the current hypothesis's support using SPRT method [15].
 if $I_k < I_{best}$ **then**
 $I_{best} \leftarrow I_k$
 Update H with the hypothesis with the strongest support.
 $k_{MAX} \leftarrow$ Update iteration bound to achieve the level of confidence η_0 .
 end if
end for

V. EXPERIMENTATION

This section presents experimental results showing the performance of our homography estimation method. We assessed the reliability and accuracy of the homography estimation itself as well as the resulting recognition rate and efficiency when used in the context of planar target recognition.

A. Homography Estimation Accuracy

In order to validate our homography estimation algorithm, we used the test set proposed in [10] to benchmark the USAC framework. We produced the same performance table as in [10] in which a homography is estimated using different image pairs (see Table II). The first column shows the performance we obtained using the standard USAC 1.0 framework. In the second column, we simply replaced the USAC SVD estimation by our Gaussian Elimination (GE) implementation. Very similar performances are obtained which demonstrate that GE estimation is also able to provide accurate estimates. The computational timings are also similar and this is explained by the fact that under the full USAC framework, the homography estimation stage does not represent a significant portion of the total computation. We therefore ran a new set of experiments in which we removed the more costly local optimization and symmetrical re-projection error steps as these are not required in a target recognition context. In such a case, the benefit of using GE in the estimation of the homography becomes apparent

(compared to the third column SVD results, fourth column GE results are 2 to 5 times faster). Finally, the last column shows the performance of our PROSAC implementation based on GE estimation of the homography.

B. Target recognition performance

In order to assess the performance of our GE implementation in the context of planar target recognition, we captured four sets of image sequences for four different types of targets, with each sequence comprising of around 250 to 300 frames¹. The image sequences were captured using an LG Optimus 2X smartphone camera with a resolution of 480×480 . The camera was rotated by approximately 45° in all directions (i.e. 45° in- and out-of-plane rotation). The scale of the target varies from full resolution (where the target fully occupies the frame) to about one third the image size. A majority of the images suffer from perspective distortions and severe motion blur in some cases. The ground truth target locations were manually obtained by identifying the four corners of the target in each image of each sequence.

The matching scheme based on BRIEF described in Section 2 was used to match the target features with the ones detected in each frame of the test sequences. Each matching set thus obtained is then feed to our PROSAC estimator in order to obtain a putative homography. The same experiments was repeated for the different homography estimation methods, all of them using the same initial match sets.

Table I shows the number of matches in the initial set and the number of matches in the final set with best support as found by PROSAC. We report these results for the SVD solution (as implemented in OpenCV) and for our Gaussian Elimination implementation.

The recognition rate is determined by analyzing the maximum error between the estimated target corner locations to the corresponding ground truth corner location. This error, given in pixels, is obtained as follows:

$$\mathcal{E}_i(\tilde{C}) = \max_j \|H_i \hat{p}_j - \tilde{p}_{ij}\|, 1 \leq j \leq 4, \quad (7)$$

where H_i is the estimated homography at frame i , \hat{p}_j is the coordinate of target corner j in the reference frame and \tilde{p}_{ij} is the manually obtained location of corner j in frame i . If we consider that a target is successfully detected if $\mathcal{E}(\tilde{C}) \leq 10$ pixels, we then obtain a recognition rate of 72.56% for SVD and 73.44% for Gaussian Elimination (last column of Table I). To illustrate the behavior of the two tested homography estimation methods, we show in Figure 2 the evolution of the maximal positional error (reported every 5 frames) for one of the test sequences. As it can be seen, except for one large error made by Gaussian Elimination, both estimation scheme exhibits very similar behavior.

¹available at www.eecs.uottawa.ca/~laganier/projects/mobilevision

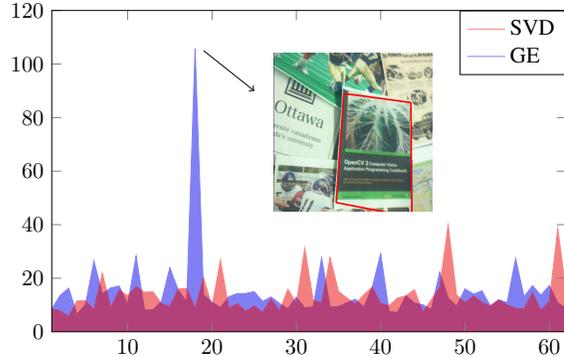


Figure 2: Maximum positional error for the *book* sequence.

C. Computational efficiency

We report in this section the global computational efficiency of different homography estimation methods in the context of robust target detection. Speed is here measured in cycle count. For completeness, we evaluate the performance of different methods under different contexts; the results are shown in Figure 3. First, we measured the speed of the OpenCV `cv::findHomography` function (version 2.4) under the RANSAC mode. We also built our own implementation of the RANSAC scheme inside which we used the OpenCV 2.4 SVD function. We then integrated the same OpenCV 2.4 function under our PROSAC implementation. We also tested the DEGSVD function from the LAPACK package. We also tested a publicly available but non-optimized Gaussian Elimination implementation [16]. Finally, the last results shown in Figure 3 is the one obtained our proposed optimized Gaussian Elimination scheme. For a device equipped with a $2.26GHz$ CPU, a $30fps$ detection rate corresponds to a maximum number of about 75 millions of cycles.

VI. CONCLUSION

We have presented in this paper a device-friendly implementation of the well-known Gaussian Elimination algorithm for homography estimation. We have shown that this simplified approach significantly speeds-up the homography estimation process. Since this estimation step is repeated many times in target recognition frameworks that are based on the hypothesize-and-verify scheme, this improvement considerably reduces the computational load in real-time implementation on low-powered devices. Additionally, we showed from experimentations that the homographies obtained using our optimized *GE* implementation have an accuracy comparable to the ones obtained using the more conventional *SVD* solution.

The homography estimation algorithm proposed in this paper was integrated into a fast target recognition framework based on trained BRIEF features. Any other recognition

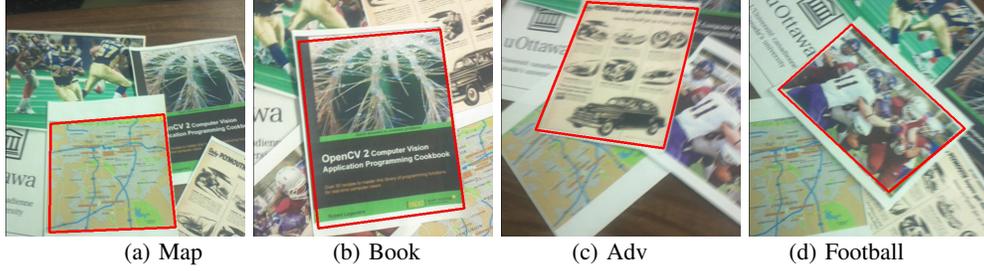


Figure 1: Target recognition for one frame of each of our test videos.

| Target | Total matches | Total Inliers | | Iterations | | Recognition rate(%) | |
|----------|---------------|---------------|-------------|---------------|---------------|---------------------|-------|
| | | GE | SVD | GE | SVD | GE | SVD |
| Book | 169.0 ± 33.1 | 67.1 ± 41.3 | 61.0 ± 34.7 | 756.0 ± 710.6 | 772.0 ± 722.5 | 48.82 | 45.40 |
| Map | 75.3 ± 18.7 | 38.1 ± 17.2 | 38.6 ± 17.3 | 317.4 ± 546.8 | 299.4 ± 526.3 | 72.24 | 74.75 |
| Football | 232.7 ± 41.9 | 82.2 ± 44.8 | 79.8 ± 40.8 | 747.9 ± 723.7 | 742.2 ± 717.4 | 84.58 | 79.06 |
| Adv | 200.8 ± 52.8 | 74.8 ± 41.0 | 83.0 ± 39.6 | 693.1 ± 726.8 | 604.1 ± 661.0 | 88.09 | 90.49 |
| Average | 175.6 ± 65.6 | 67.5 ± 41.6 | 67.3 ± 38.9 | 658.8 ± 709.7 | 635.4 ± 694.0 | 73.44 | 72.56 |

Table I: Average number of total matches, inliers ,required iterations and recognition rate are shown for the four targets with both GE and SVD method

| | | USAC 1.0 | USAC GE | USAC SVD (No LO) | USAC GE (No LO) | our PROSAC GE |
|--|----------|---------------|----------------|---------------------|--------------------|------------------|
| A: $\epsilon = 0.46$, $N = 2540$  | I | 1147.6 ± 0.1 | 1147.7 ± 0.1 | 1074.4 ± 9.1 | 1017.2 ± 10.1 | 969.6 ± 10.2 |
| | K | 4.8 ± 0 | 5.9 ± 0.1 | 7.8 ± 0.1 | 9.1 ± 0.2 | 8.4 ± 0.2 |
| | K_rej | 0.0 ± 0.0 | 0.0 ± 0.0 | 0.0 ± 0.0 | 0.0 ± 0.0 | 0.0 ± 0.0 |
| | models | 4.8 ± 0 | 5.9 ± 0.1 | 7.8 ± 0.1 | 9.1 ± 0.2 | 8.4 ± 0.1 |
| | VPM | 755.6 ± 15.6 | 667 ± 16.4 | 1021.6 ± 16.6 | 869.3 ± 16.1 | 1193.5 ± 14.8 |
| | error | 1.27 | 1.27 | 1.18 | 2.22 | 2.27 |
| | time(ms) | 24.78 | 24.4 | 0.4494 | 0.3477 | 0.0810 |
| B: $\epsilon = 0.15$, $N = 514$  | I | 68.1 ± 0.0 | 68.0 ± 0.0 | 67.7 ± 0.5 | 61.5 ± 0.9 | 64.3 ± 0.4 |
| | K | 925 ± 316 | 14557 ± 3676 | 57.0 ± 11.9 | 165.7 ± 23.0 | 13.6 ± 0.4 |
| | K_rej | 711.2 ± 263.8 | 12446.8 ± 3226 | 35.2 ± 10.2 | 128.0 ± 19.7 | 3.0 ± 0.1 |
| | models | 214.1 ± 53.7 | 2104.8 ± 451.3 | 21.8 ± 1.8 | 36.4 ± 3.3 | 10.6 ± 0.3 |
| | VPM | 49 ± 1.4 | 42.4 ± 2.3 | 29.6 ± 2.1 | 100.4 ± 3.6 | 294.3 ± 3.6 |
| | error | 0.87 | 0.87 | 2.08 | 2.35 | 2.38 |
| | time(ms) | 4.93 | 3.78 | 0.2873 | 0.07323 | 0.02511 |
| C: $\epsilon = 0.23$, $N = 1317$  | I | 301.0 ± 0.0 | 300.56 ± 0.3 | 211.4 ± 1.2 | 210.9 ± 1.3 | 202.9 ± 1.4 |
| | K | 4.8 ± 0.1 | 7.5 ± 0.3 | 4.7 ± 0.1 | 6.3 ± 0.2 | 5.0 ± 0.1 |
| | K_rej | 0.3 ± 0.0 | 0.5 ± 0.0 | 0.3 ± 0.0 | 0.4 ± 0.1 | 2.3 ± 0.0 |
| | models | 4.5 ± 0.1 | 4.9 ± 0.3 | 4.4 ± 0.1 | 3.9 ± 0.2 | 2.7 ± 0.1 |
| | VPM | 372.6 ± 4.5 | 593.5 ± 17.5 | 435.1 ± 6.6 | 694.9 ± 16.8 | 1215.7 ± 7.8 |
| | error | 0.80 | 0.8 | 0.98 | 1.42 | 1.35 |
| | time(ms) | 6.33 | 6.3 | 0.1127 | 0.07363 | 0.03406 |
| D: $\epsilon = 0.34$, $N = 495$  | I | 146.2 ± 0.1 | 146.3 ± 0.1 | 137.0 ± 1.0 | 139.6 ± 1.1 | 136.7 ± 1.2 |
| | K | 14.0 ± 0.4 | 16 ± 0.5 | 5.1 ± 0.1 | 5.8 ± 0.1 | 5.5 ± 0.1 |
| | K_rej | 3.7 ± 0.1 | 4.2 ± 0.2 | 1.9 ± 0.0 | 2.0 ± 0.0 | 2.7 ± 0.0 |
| | models | 10.3 ± 0.3 | 10.8 ± 0.4 | 3.2 ± 0.1 | 3.0 ± 0.1 | 2.8 ± 0.1 |
| | VPM | 103.4 ± 2.3 | 111.4 ± 3.4 | 307.3 ± 5.4 | 342.1 ± 5.9 | 482.4 ± 1.5 |
| | error | 1.16 | 1.16 | 5.72 | 5.70 | 5.87 |
| | time(ms) | 2.73 | 2.68 | 0.07764 | 0.04241 | 0.016903 |

Table II: Performance result of Homography estimation as in [10]. (I) is the number of inliers found. (K) and (K_rej) are the number of samples drawn and the number of samples rejected by the degeneracy test. (models) is the number of total hypotheses, (VPM) the number of verification per model. The symmetrical reprojection (error) is measured w.r.t. the ground truth. (time) indicates the execution time per frame in *ms*. Note that all reported results are averaged over a total of 500 runs.

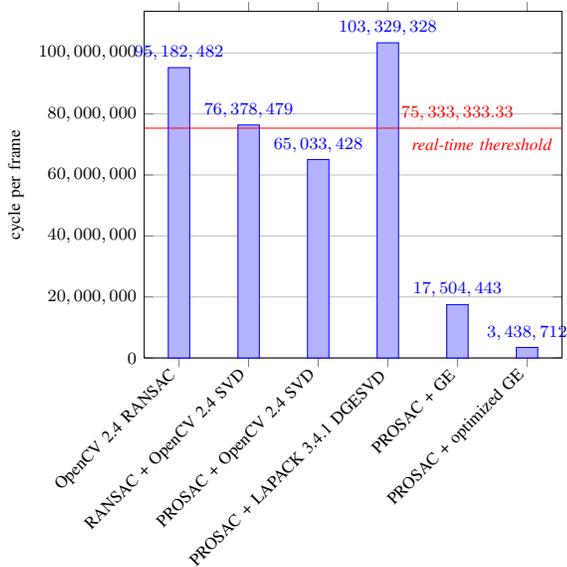


Figure 3: Per-frame average cycle count for each H-estimator

framework would similarly benefit from our optimized implementation.

REFERENCES

- [1] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *ECCV 2010*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6314, pp. 778–792.
- [2] A. Alahi, R. Ortiz, and P. Vanderghenst, “Freak: Fast retina keypoint,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 510–517.
- [3] S. Leutenegger, M. Chli, and R. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2548–2555.
- [4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.
- [5] M. Ozuysal, P. Fua, and V. Lepetit, “Fast keypoint recognition in ten lines of code,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, June 2007, pp. 1–8.
- [6] S. Taylor, E. Rosten, and T. Drummond, “Robust feature matching in 2.3 μ s,” in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, June 2009, pp. 15–22.
- [7] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

- [8] O. Chum and J. Matas, “Matching with prosac - progressive sample consensus,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 220–226 vol. 1.
- [9] R. Raguram, J.-M. Frahm, and M. Pollefeys, “A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus,” in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV ’08, 2008, pp. 500–513.
- [10] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm, “Usac: a universal framework for random sample consensus,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013.
- [11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.
- [12] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
- [13] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *ECCV 2006*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 3951, pp. 430–443.
- [14] S. S. Akhouri and R. Laganière, “Training binary descriptors for improved robustness and efficiency in real-time matching,” in *ICIAP 2013*, ser. Lecture Notes in Computer Science, A. Petrosino, Ed., 2013, vol. 8157, pp. 288–298.
- [15] J. Matas and O. Chum, “Randomized ransac with sequential probability ratio test,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct 2005, pp. 1727–1732 Vol. 2.
- [16] <https://github.com/camilosw/ofxVideoMapping>, 2012.