Deterministic Sample Consensus with Multiple Match Hypotheses

Paul McIlroy
http://mi.eng.cam.ac.uk/~pmm33
Ed Rosten
http://mi.eng.cam.ac.uk/~er258
Simon Taylor
http://mi.eng.cam.ac.uk/~sjt59
Tom Drummond
http://mi.eng.cam.ac.uk/~twd20

Machine Intelligence Laboratory
Department of Engineering
University of Cambridge
Cambridge, UK

Abstract

RANSAC (Random Sample Consensus) is a popular and effective technique for estimating model parameters in the presence of outliers. Efficient algorithms are necessary for both frame-rate vision tasks and offline tasks with difficult data. We present a deterministic scheme for selecting samples to generate hypotheses, applied to data from feature matching. This method combines matching scores, ambiguity and past performance of hypotheses generated by the matches to estimate the probability that a match is correct. At every stage the best matches are chosen to generate a hypothesis. This method will therefore only spend time on bad matches when the best ones have proven themselves to be unsuitable. The result is a system that is able to operate very efficiently on ambiguous data and is suitable for implementation on devices with limited computing resources.

1 Introduction

In computer vision, a common task is to estimate model parameters from a set of feature matches. This has applications in motion estimation, tracking, SLAM, image stitching and object detection. Feature matching schemes generate data with outliers and so models are often estimated using some form of RANSAC (Random Sample Concensus) [3], in which small samples of data are used to hypothesize models. Efficient RANSAC schemes are important in both frame-rate vision and off-line vision tasks, where inefficient schemes can prove intractable. We make the following contributions:

- A new method for determining the prior probability of a match being an inlier based on gathered data and ambiguity in matching.
- A way to fuse the prior probability with knowledge that a point was used to successfully or unsuccessfully generate a hypothesis.
- A *deterministic* sampling scheme which always chooses the best known points to generate a hypotheses.

The advantage of this scheme is that it always generates models from the most promising set of points, and it is able to address the challenge of multiple matches caused by ambiguities in matching.

Since the inception of RANSAC, a number of schemes have been proposed which aim to improve its performance. The basic framework involves selecting points to generate a hypothesis and then testing the correspondence of all data against the hypothesis. Speed improvements can be achieved by optimizing either or both of these stages. The second stage can be improved by testing hypotheses against a random subset of points [5] or by detecting bad hypotheses early [7]. MLESAC (Maximum Likelihood Estimation Sample Consensus) [13] instead makes the measurement of correspondence more reliable and improves the estimate of the hypotheses.

This paper belongs to the class of algorithms that optimize the first stage by improving the way in which points are selected. Some improvement in sampling can be made if there are good assumptions to be made about the distribution of points [6]. However, most feature matching schemes compare features for similarity, in order to decide if features match. The implicit assumption is that better scoring matches are more likely to be correct. If the probabilities are available (for instance by observing how often points with a given score are inliers) then 'Guided Sampling' can be used which samples points according to their probability of correctness [12]. On the other hand, if probabilities are unavailable it is still reasonable to assume a monotonic relationship between score and probability. The PROSAC (Progressive Sample Consensus) algorithm [2] makes use of this, spending more time considering points with a high score.

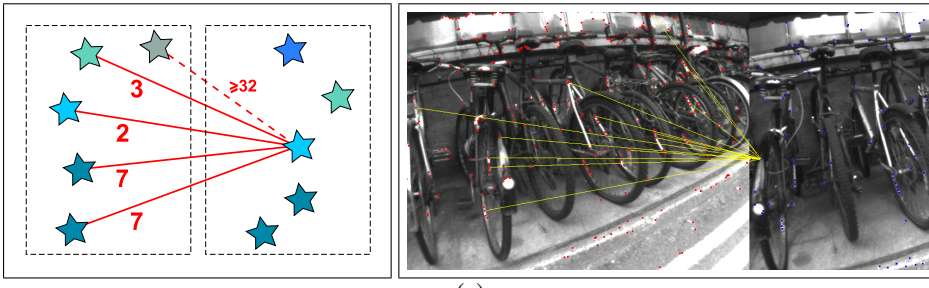
This paper sits somewhere between the PROSAC ordered-sampling approach and the probabilistic approach of Guided Sampling. Intuitively, the first iteration of PROSAC does the right thing: as its first act it creates a hypothesis from the best points, whereas in guided sampling, the best points can get swamped by a large number of lower score points. However, if the first iteration fails to yield a correct solution we use a very different scheme for choosing the next solution. The key is that when a point participates in a failed hypotheses some information has been generated that the point is an outlier. Therefore we use this to update the probabilities of all the points participating in such a hypothesis at the end of an iteration. On the next iteration we again choose the most likely points.

The result is a deterministic scheme which performs sampling guided by the probabilities, but always picks the best known points at each iteration.

In order to correctly deduce match probabilities, one needs a rigorous method for dealing with ambiguities in matching. The problem with ambiguities is well known. Even with distinctive features, better matching performance is achieved by considering the ratio of scores between the best and second best matches [9]. This effectively scores matches on their lack of ambiguity indirectly.

We take a more direct approach and make use of both the score and the ambiguity. Consider the case where two competing matches sharing a common feature have the same good match score. Naturally, only one can be correct so the probability of each must be ≤ 0.5, but the sum of the two probabilities might still be ≈ 1. Without accounting for the ambiguity, the probability of selecting either one of these two matches will therefore be the same as selecting as single good match with probability ≈ 1. We actually consider all matches above a low threshold, and weight the probability of matches according to both their score and the level of ambiguity.

In order to demonstrate this technique we apply it to very simple features, with discrete matching scores, used in frame-to-frame matching tasks. These simple features can be im-



(a) (b)

Figure 1: Multiple matches hypothesis: (a) a discrete error score is calculate for each correspondence; (b) multiple matches from a single feature on the right.

plemented efficiently on low power mobile computing devices. The lack of a database of features prevents the use of efficient, discriminatively trained features [8, 11]. We demonstrate that our system is capable of overcoming the considerable ambiguity in matching, due to the simplicity of the features, to yield a very effective model estimation scheme. In particular we demonstrate the method estimating essential matrices and homographies for a variety of scenes.

2 Multiple match hypotheses

Multiple matches belonging to a single feature are mutually exclusive events and the probabilities assigned to each alternative match must sum to ≤ 1 with the remainder being the probability that none of the putative matches are correct. One consquence of this approach is that if two such matches have the same quality score they must each have a probability ≤ 0.5 and are therefore significantly less likely to be valid than a unique match with the same score. Thus, similar match scores present a challenge to strategies that select a single best match. A common heuristic, employed in SIFT, throws away all matches associated with a feature if the ratio between the best and next best quality scores falls below a threshold. In this section we exploit the additional information provided by the alternative match scores in order to compute a probability for each putative correspondence.

2.1 Heavily quantised patch descriptors

The feature descriptor used in this paper is based on the quantised patches used to build the Histogrammed Intensity Patch of Taylor and Drummond [11]. Interest points are first detected using the efficient FAST-9 algorithm [9]. An 8×8 pixel patch is extracted by subsampling from the 15×15 pixel region surrounding each interest point. The normalised intensity of each pixel is then quantised into one of five bins. The 8×8 bits representing each level of quantisation are stored separately as a 64-bit word with five words in total for each descriptor. This permits an efficient implementation of the match score computation between two descriptors using bitwise operations. The match score is an integer in the range $[0,64]$ representing the total number of pixels that differ between two descriptors. The descriptor optimises for speed at the expense of performance and is not invariant to scale or rotation, yet it successfully matches features over reasonably wide baseline frame to frame motions if false positives can be tolerated.

2.2 Match probability

The frame to frame matching process is illustrated by Figure 1. Each feature in the current frame is compared to all n features in the previous frame. The putative correspondences are assigned a discrete match score using the method described in the previous section. Each feature in the current frame will generate at most one correct match. If one match is correct, the remaining $(n-1)$ match scores are generated by incorrect correspondences. Alternatively, all n match scores may be generated by incorrect correspondences if, for instance, the correct feature is occluded or moves outside the field of view.

Each false match generates a match score drawn from a probability distribution, with high error scores more likely than low ones. The match scores are generated by each putative correspondence as independent trials from a catagorical distribution. Since the match scores are discrete the probability that N false matches will generate a particular set of match scores is therefore given by the multinomial distribution,

$$\mathsf{Mult}(m_1, m_2, \ldots, m_K | \mu, N) = \frac{N!}{m_1! m_2! \cdots m_K!} \prod_{k=1}^{K} \mu_k^{m_k}, \tag{1}$$

where m_k is the number of false matches that generated a match with score k following the formulation in Bishop [1]. The parameters $\mu = (\mu_1, \ldots, \mu_K)^\top$ give the probability of a single false match generating match score k, and are subject to the constraints $0 \leq \mu_k \leq 1$ and $\sum_k \mu_k = 1$.

Let $P(c_k)$ be the probability that a feature has a correct correspondence with match score k with $0 \leq k \leq K$. There is also a probability $P(c_\emptyset)$ that a feature has no correct correspondence in the previous image.

The match scores generated by a set of n putative correspondences are observed. Several competing explanations are compatible with the observed match scores. We first consider the event E_j that there is a correct match with score j and the other $(n-1)$ possible matches are false. The probability of the data given this event is then:

$$P(E_j) = P(c_j)\mathsf{Mult}(m_1, m_2, \ldots, m_j - 1, \ldots, m_K | \mu, n-1). \tag{2}$$

Now we consider the event $P(E_\emptyset)$ that the feature generated the no correct match event and the match scores $m_1 \ldots m_K$ were generated by the n false matches. The probability the data given this event is then:

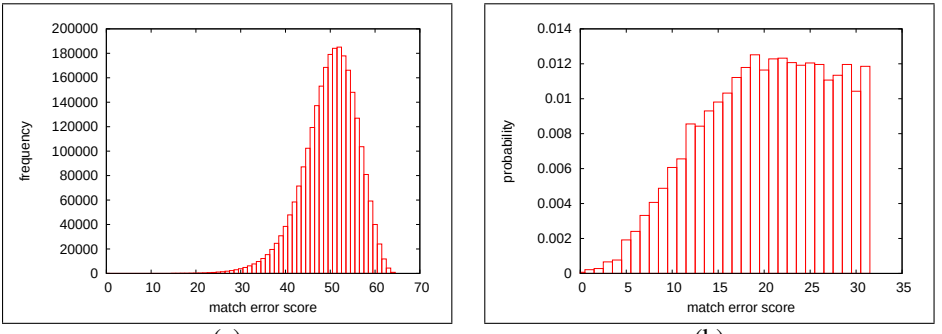
$$P(E_\emptyset) = P(c_\emptyset)\mathsf{Mult}(m_1, m_2, \ldots, m_K | \mu, n). \tag{3}$$

The posterior probability $P(c_i | m_1, \ldots, m_K)$ that a correct match exists with error i given the observed set of match scores is therefore

$$P(c_i | m_1, \ldots, m_K) = \frac{P(E_i)}{\sum_{j=0}^{K} P(E_j) + P(E_\emptyset)}. \tag{4}$$

Two or more correspondences may share the same match score, so the probability that a single correspondence s with match score i is correct is given by

$$P(s_i | m_1, \ldots, m_K) = \frac{P(c_i | m_1, \ldots, m_K)}{m_i}. \tag{5}$$



(a) (b)

Figure 2: Probability distributions for error scores generated by (a) false matches; (b) correct matches.

This probability is calculated for each putative correspondence. It provides the subsequent sampling strategy with a quality measure that takes into account not only the individual match score between two descriptors, but also the ambiguity in matching this feature. In order to calculate these probabilities the parameters μ must first be determined as described in the next section.

2.3 Match score distributions

The distribution for match scores generated by a false match is shown in Figure 2. The probability distribution for false matches is obtained by considering the *intra*frame matches. The match score between two distinct features that coexist in the same image is guaranteed to have been generated from the false match distribution. Large numbers of false matches over long sequences of video are very easily harvested to produce an accurate distribution.

If the patches were generated randomly from white noise the probability distribution would follow a binomial distribution for 64 trials with probability 0.8 due to the five quantisation bins. In practice the distribution is skewed towards lower match scores by scene structure and repeated texture in real world images.

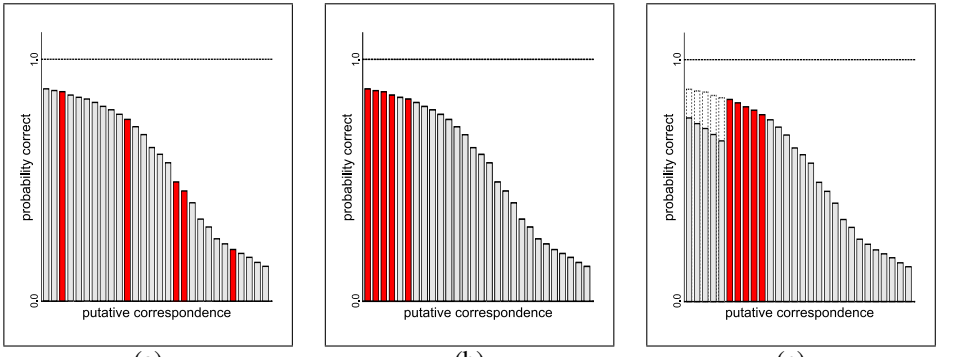
Match scores above a fixed threshold may be considered as a single event as most of the relevant information is given by the lower match scores. This modification greatly reduces computation required. Let K now be the upper match score threshold such that all match scores $k \geq K$ are assigned to event bin K. The threshold is set sufficiently low that most false matches generate match scores above the threshold. A further simplification is then possible noting that for $n \gg 1$ and $\mu_K \approx 1$,

$$\mathsf{Mult}(m_1,\ldots,m_{K-1},m_K-1|\mu,n-1) \approx \mathsf{Mult}(m_1,\ldots,m_{K-1},m_K|\mu,n). \tag{6}$$

This allows the 'no correct match' event and correct matches scoring $\geq K$ to be combined together in event bin K.

$$P(E_K)+P(E_\emptyset) \approx P(c_K \cap c_\emptyset)\mathsf{Mult}(m_1,\ldots,m_{K-1},m_K-1|\mu,n-1). \tag{7}$$

The probability distribution for features generating correct matches is harvested from sequences with ground truth for the correct matches. Figure 2 shows both distributions for match scores 0 to $K-1$, with the remainder of the probability occupying event bin K in each case.



(a) (b) (c)

Figure 3: The second minimal set sampled by: (a) RANSAC; (b) PROSAC (c) DESAC

This section described how the multiple match hypotheses generated by a feature descriptor may be used to determine the probability that a putative correspondence is correct given its match score and the match scores of the alternative hypotheses. The next section describes how the probability of each correspondence can be used to guide the selection of a minimal set of correspondences in a sample and test strategy to determine the camera motion between two images.

3 DESAC: Deterministic sampling

This section introduces a hypothesize-and-verify method that exploits the prior probability for each putative correspondence from Section 2. We seek to determine the model that best explains the feature matches observed between two frames. The RANSAC algorithm widely used for this purpose proceeds by selecting a minimal set of correspondences by random sampling then testing the motion calculated from this minimal set against the remaining matches. PROSAC uses the match score to order the correspondences and begins by selecting the minimal set with the best match scores. The pool of samples from which PROSAC draws its minimal set is gradually expanded at a rate designed to provide a balance between reliance on the initial sorting and the RANSAC approach which treats all correspondences as equally likely.

If the probabilities for each match are available, rather than just a match score, more sophisticated sampling is possible. If the incoming matches have similar probability to those already in the pool, then the pool should expand more rapidly than if they have significantly lower probability. Furthermore, if a hypothesis generated from a set of samples fails to generate significant consensus, then it is likely that at least one of the samples used is a false match and this can be used to update (by reducing) the probability that each match being considered is correct. These matches can then be removed from and reinserted into the list so that it remains sorted by match probability. At this point we can always choose the most likely matches at the top of the list.

Figure 3 illustrates this aproach by comparing the samples chosen at the second iteration of RANSAC, PROSAC and DESAC (our algorithm). The matches are shown in order according to the prior probabilities before the first test. RANSAC ignores the probabilities and selects each minimal set by random sampling from the set of all matches. PROSAC expands the sample pool by one and chooses a minimal set from this limited pool compris-

ing the recently introduced match and four of the others chosen at random. Our algorithm reduces the probabilities of the matches tested after the first iteration and then chooses the most likely matches from the updated probabilities. Given the nearly uniform distribution at the top of the match list our algorithm favours fresh matches over the ones known to contain one or more false matches. If the distribution decreased in probability more rapidly some of matches from the first iteration may be chosen at the second iteration.

Each failed consensus test implies that the minimal set contains one or more false matches. Consider a minimal set consisting of four matches A, B, C and D. The prior probability of the minimal set containing only correct matches is given by $P(A \cap B \cap C \cap D)$. The outcome of a failed test removes this event from the joint distribution and the posterior probability that match A is correct becomes

$$P(A|\bar{A} \cup \bar{B} \cup \bar{C} \cup \bar{D}) = \frac{P(A) - P(A \cap B \cap C \cap D)}{1 - P(A \cap B \cap C \cap D)}. \tag{8}$$

The joint distribution presents a challenge as it is not feasible to represent the full joint probability for large numbers of matches. We treat the events as independent and approximate the joint distribution using $P(A)P(B)P(C)P(D)$. Given the improved mapping between match scores and probabilities described in 2 we proceed with this line of research and investigate the performance of our algorithm, testing against both synthetic and real world data in the next section.

Algorithm 1: DESAC

1. Store correspondences in a sorted data structure, S, sorted according to probability;
2. Remove a minimal set, K, of correspondences from the top of S and fit a model;
3. Test consensus of the model (inlier count above a minimum threshold or sufficient inlier ratio). Stop if the model has high consensus.;
4. Modify the probabilities of the correspondences in K according to 8, reinsert them into S and goto 2;

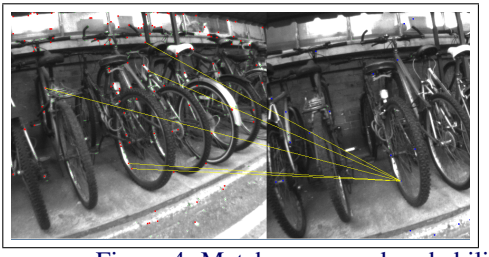
4 Results

An example of the multiple match hypotheses associated with a single feature is shown in Figure 4. The correct match in this case has a match score of 21 and four alternative hypotheses are generated from the false matches with higher error scores. The first match is assigned a probability of 0.3870 taking into account the alternative hypotheses and the event X that all matches are false.

4.1 Simulation

Our algorithm was first tested against synthetic data to investigate the best-case performance of the method. Two sets of n points were generated to represent the features in two images. A Monte Carlo sampling approach was used to select either a correct match score or a not matched event for each feature in turn. The probability distribution used for the correct matches was generated from real images. If a correct match was generated for a given feature, $(n-1)$ samples were drawn from the distribution for false matches. If the not matched

h



Match	Score	Probability
1	21	0.3870
2	29	0.0866
3	29	0.0866
4	30	0.0596
5	31	0.0483
X	X	0.3320

Figure 4: Match scores and probabilities for alternative hypotheses.

b

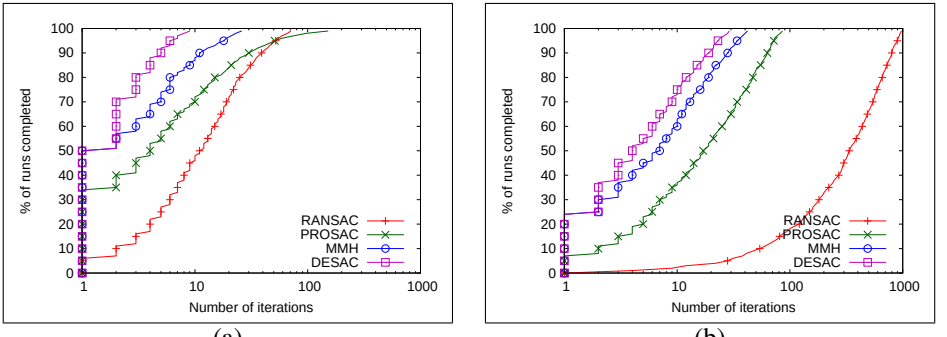


Figure 5: Time-to-solution for synthetic data: (a) 200 features; (b) 1000 features. Multiple Match Hypothesis (MMH) combines the scoring method proposed in Section 2 with PROSAC. DESAC combines the scoring of Section 2 with the sampling method proposed in Section 3.

event was drawn from the correct match probability distribution, n false match scores were drawn from the distribution for false matches.

The multiple match hypotheses for each feature in the right image were used to assign probabilities to each correspondence according to the method described in Section 2. This set of synthetic data with known ground truth was then used to compare the time to solution in iterations for a number of robust matching algorithms. Each test run terminated when a minimal set was selected consisting only of correspondences generated by the correct match distribution. The DESAC algorithm was compared with both PROSAC and RANSAC. Two versions of PROSAC were implemented, one using the match score directly to order the correspondences and the other making use of the probabilities with multiple match hypotheses from Section 2 named PROSAC-MMH. Figure 5 compares the time to solution for each of the methods measured in iterations. This is a reasonable measure of speed as the same consensus test is performed on the minimal set selected at each iteration and this step dominates.

Tests were performed with $n = 200$ and $n = 1000$. The DESAC algorithm requires the lowest number of iterations to find a correct minimal set. The PROSAC version ordered by probabilities clearly outperforms the version ordered by match scores, but this is unsurprising as the synthetic data is drawn from the distributions used to calculate the probabilities. RANSAC requires many more iterations for $n = 1000$ as each feature now has five times as many false matches which are equally likely to be sampled as the correct ones. The ordering in PROSAC and DESAC makes these algorithms more robust to increased noise as the correct matches have higher probability of being selected.

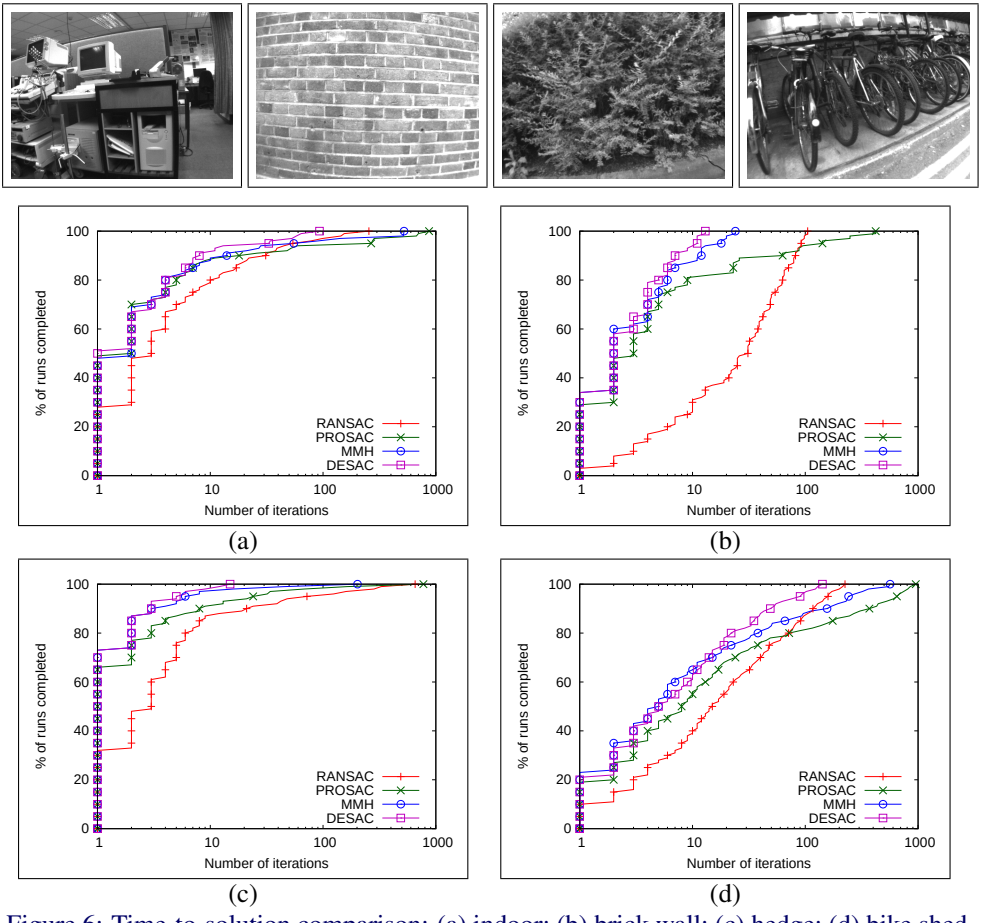


Figure 6: Time-to-solution comparison: (a) indoor; (b) brick wall; (c) hedge; (d) bike shed.

4.2 Experiments on video sequences

Figure 6 presents the results of experiments on four video sequences. The lab sequence represents a standard indoor 3D scene with distinct corner features and some repeated texture. The brick wall is a more challenging planar scene with fewer distinct features. The hedge sequence represents a difficult natural scene. The bike sequence frustrates matching by generating corner features between overlapping structures that only exist temporarily in a single frame.

An adaptive FAST threshold was used to extract 100 features from the first image drawing 25 from each quadrant. The brick wall scene has few distinct corner features and many similar interest points generated by the brick texture. The adaptive threshold selects twice as many interest points in the second frame to address the redetection challenge caused by the top 100 interest points varying from frame to frame. For the 3D sequences the epipolar geometry was recovered from the minimal set of five matches using [10]. In the 2D brick wall scene the planar homgraphy was calculated from a minimal set of four matches.

The time-to-solution tests on real video sequences reveal the same overall trend as the synthetic experiments. The version of PROSAC ordered using the improved match probabil-

ities again shows a marked improvement on standard PROSAC using the individual match scores. Our algorithm requires the fewest iterations overall.

5 Conclusion

We introduced a deterministic sampling scheme that combines evidence from matching scores, ambiguity and past performance of matches in generating hypotheses. At every stage the best set of matches is chosen to generate a new hypothesis. The performance of this method was examined using both synthetic tests and real world video sequences. The results show that we require fewer iterations than existing sample consensus schemes.

Acknowledgements

This work was supported by the EPSRC and through the EU funded project HYDROSYS (EU-FP7-224416).

References

[1] C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

[2] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[3] 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*, 1981.

[4] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[5] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Proc. IEEE International Conference on Computer Vision*, 2005.

[6] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. NAPSAC: high noise, high dimensional robust estimation - it's in the bag. In *Proc. British Machine Vision Conference*, 2002.

[7] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 2005.

[8] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[9] E. Rosten and T.W. Drummond. Machine learning for high-speed corner detection. In *Proc. European Conference on Computer Vision*, 2006.

[10] E. Rosten, G. Reitmayr, and T.W. Drummond. *Improved RANSAC performance using simple, iterative minimal-set solvers.* University of Cambridge Technical Report, 2010. URL http://xxx.lanl.gov/abs/1007.1432.

[11] S.J. Taylor and T.W. Drummond. Multiple target localisation at over 100 FPS. In *Proc. British Machine Vision Conference*, 2009.

[12] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conference on Computer Vision*, 2002.

[13] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000.