Efficient Sequential Correspondence Selection by Cosegmentation

Jan Čech, Jiří Matas, Michal Perďoch
Center for Machine Perception, Department of Cybernetics,
Faculty of Electrical Engineering, Czech Technical University, Prague
{cechj,matas,perdom1}@cmp.felk.cvut.cz

Abstract

In many retrieval, object recognition and wide baseline stereo methods, correspondences of interest points are established possibly sublinearly by matching a compact descriptor such as SIFT. We show that a subsequent cosegmentation process coupled with a quasi-optimal sequential decision process leads to a correspondence verification procedure that has (i) high precision (is highly discriminative) (ii) good recall and (iii) is fast. The sequential decision on the correctness of a correspondence is based on trivial attributes of a modified dense stereo matching algorithm. The attributes are projected on a prominent discriminative direction by SVM. Wald's sequential probability ratio test is performed for SVM projection computed on progressively larger co-segmented regions. Experimentally we show that the process significantly outperforms the standard correspondence selection process based on SIFT distance ratios on challenging matching problems.

1. Introduction

Many successful image retrieval, object recognition and wide baseline stereo methods exploit correspondences of local transformation-covariant regions. Most real-world visual recognition problems are large scale where correspondences between regions from a query (test) image and many database (training) images (multiple views of a many objects or scenes) are sought. To achieve acceptable response times, large problems require the time complexity of the region matching process be sublinear in the size of the database; memory footprint of the database representation becomes a concern too. The standard solution is to describe regions with a compact descriptor such as SIFT [9] or some discretization of it (*e.g.* "visual words" [17]) and to store database image representations in a search tree (k-d [9], metric [7], k-means [13, 16, 3]). [1]

[1]Terms "viewpoint invariant features", "interest points", "patches", "distinguished regions" also appear in the literature.

A better estimate of correspondence quality (a prediction of it being correct) can be obtained by looking at both test and training image simultaneously, *e.g.* by attempting to expand the correspondence domains or to improve the precision of registration. The value of correspondence growing methods has been demonstrated in [19, 4], sometimes with impressive results, *e.g.* those achieved by the dual bootstrap method [21, 18]. Most approaches to simultaneous cosegmentation and registration focus on the problem of finding the largest corresponding domain and codomain [21, 4, 8].

Our objective is almost opposite: given acceptable false positive and false negative rates, design the fastest possible test for correctness of a correspondence, based on cosegmentation of regions of growing size. We formulate the problem as sequential decision making performing Wald's sequential probability ratio test. The test is based on simple statistics of a modified dense stereo matching algorithm which are projected on a single prominent discriminative direction by a linear SVM.

On challenging matching problems, *we show that the selection of correspondences based on sequential cosegmentation is very efficient, runs near to real-time and significantly outperforms the standard correspondence process based on SIFT distance ratios*, producing a higher number as well as higher percentage of correct correspondences. Consequently, combinatorial procedures for estimation of a geometrically consistent subset of correspondences with time complexity sensitive to inlier ratios (polynomial dependence), *e.g.* RANSAC, should always adopt sequentially terminated cosegmentation as a preprocessing step. In fact, the process of generating tentative correspondences can be set to be much more permissive, outputting higher number of correspondences with lower inlier ratios but containing larger number of inliers. After filtering by simultaneous cosegmentation, inlier ratios are recovered and the larger number of inliers leads to higher recognition rates.

The method scales well: the number of potential correspondences for a query image region can be controlled. If it is constant, the total time complexity of the region ex-

Algorithm 1 SCV: Sequential Correspondence Verification

Require: images $\mathbf{I}$, $\mathbf{I}'$,
correspondence with affine frame $(x, y, \mathbf{A})$,
SIFT ratio s_r,
false positive and false negatives rates (α, β),
model: learned SVM parameters θ_i,
likelihoods $p_i(q|+1),\ p_i(q|-1)$.

1.1: **for** $i = 1$: maximum number of decision stages **do**
1.2: $\mu_i = \begin{cases} 0, & i=1, \\ 10^{i-1}, & i>1. \end{cases}$
1.3: $(\bar{g}, \bar{c}, \bar{u}) = \text{grow}(\mathbf{I}, \mathbf{I}', (x, y, \mathbf{A}), \mu_i)$.
1.4: $q = \text{SVM}(s_r, \bar{g}, \bar{c}, \bar{u}, \theta_i)$.
1.5: $L = \frac{p_i(q|+1)}{p_i(q|-1)}$.
1.6: **if** Wald SPRT(L, α, β) is conclusive **then break**.
1.7: **end for**
1.8: **return** likelihood ratio L.

pansion process is independent of the size of the database and linear in the size of the input (number of regions in the query image). On a large scale retrieval experiment [13], we observed that the time needed to carry out the sequential procedure is not significant in comparison with the time needed for the initial indexing process for establishing tentative correspondences.

The rest of the paper is organized as follows. The method is described in Sec. 2, the experiments are found in Sec. 3, and the conclusion is given in Sec. 4.

2. The sequential correspondence verification algorithm

The basic idea of the approach is to distinguish, as fast as possible, correct and incorrect correspondences via a dense matching (pixel-to-pixel) growing algorithm. The requirements of high speed and quality of the decision process are contradictory. We therefore propose a quasi-optimal sequential decision algorithm that minimizes time to decision, given user-specified probabilities of false positive and false negative rates.

The Sequential Correspondence Verification algorithm (SCV) is summarized in Fig. 1. It proceeds in decision stages i. In the first decision stage, a fast dense stereo matching growing algorithm, Sec. 2.1, is initialized by a tentative correspondence with a Local Affine Frame. The verification proceeds by attempting to match discriminative neighboring pixels. After a maximum number of growing steps μ_i, this cosegmentation produces three simple statistics $(\bar{g}, \bar{c}, \bar{u})$ characterizing the quality of the correspondence: the growth rate $\bar{g}$ is the size of the grown region divided by the maximum number of growing steps μ_i, the average correlation $\bar{c}$ of the region, and the average number of pixels violating the uniqueness $\bar{u}$, *i.e.* non-bijectivity matching.

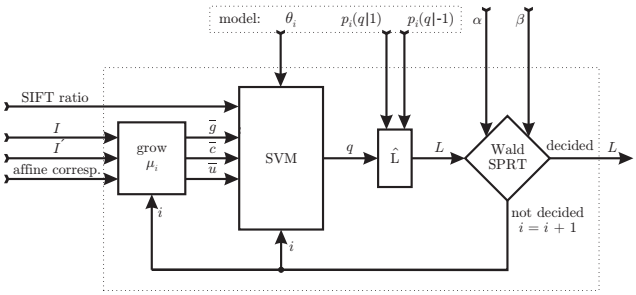

Figure 1. The Sequential Correspondence Verification algorithm.

The vector of statistics is projected by SVM to a scalar quality q which simplifies the estimation of likelihoods $p_i(q|+1), p_i(q|-1)$ of correct and incorrect correspondence classes respectively. The region statistics are augmented with the first to second nearest SIFT descriptor distance ratio s_r, a standard tentative correspondence selection technique [9].

The Wald's Sequential Probability Ratio Test (SPRT) is performed on the likelihood ratio L_i. If the SPRT test is conclusive, the algorithm terminates and the correspondence is assigned the decision and the likelihood ratio L_i of the decision. Otherwise, another decision stage i is performed, *i.e.* the cosegmentation is continued with exponentially larger maximum number of growing steps μ_i, Alg. 1, step 2 (Alg. 1.2), potentially producing more discriminative statistics, since it is based on more measurements. Note, that $\mu_1 = 0$ which means the decision in the first stage is based solely on the SIFT ratio without growing.

The process continues until the maximum number of decision stages i is reached. In our experiments, we set the maximum number of decision stages to 4, so the largest growth is by $\mu_4 = 1000$ steps. Details of the algorithm are described below.

2.1. Growing algorithm

The following simple algorithm explores regions around the input tentative correspondence. The growing mechanism is inspired by [1, 15, 8, 11].

Each affine correspondence defines a local mapping from the reference image $\mathbf{I}$ to the target image $\mathbf{I}'$. The mapping generates several pixel to pixel correspondences, the seeds.[2] Seed $\mathbf{s} = (x, y, \mathbf{A})$ is a point (x, y) in $\mathbf{I}$ with associated affine transformation $\mathbf{A}$ which maps the local neighborhood to the other image $\mathbf{I}'$:

$$\begin{aligned} x' &= a_1 x + a_2 y + a_3, \\ y' &= a_4 x + a_5 y + a_6, \end{aligned} \tag{1}$$

or simply $(x', y') = \mathbf{A}(x, y)$.

[2] In our experiments, this is realized by a Local Affine Frame (LAF) constructed on Maximally Stable Extremal Region [14, 10] (MSER). We take all three points in LAF as the initial seeds of the growing process.

Algorithm 2 The Growing Algorithm

Require: images $\mathbf{I}, \mathbf{I}'$,
initial correspondence seeds $\mathcal{S}$
maximum number of growing steps μ.

2.1: Initialize matching tables $\mathbf{T}(:,:) = 0,\ \mathbf{T}'(:,:) = 0$, variables $K := G := C := U := 0$.
2.2: Compute the image correlation for all seeds $\mathbf{s} \in \mathcal{S}$.
2.3: **while** $K \leq \mu$ **and** $\mathcal{S}$ not empty **do**
2.4: $K := K + 1$.
2.5: Draw the seed $\mathbf{s} \in \mathcal{S}$ of the best similarity $\mathrm{corr}(\mathbf{s})$.
2.6: **for** each of the four best neighbors

$$\mathbf{t}_i^* = (x, y, \mathbf{A}) = \underset{\mathbf{t} \in \mathcal{N}_i(\mathbf{s})}{\mathrm{argmax}}\, \mathrm{corr}(\mathbf{t}),\ i \in \{1, 2, 3, 4\}$$

do
2.7: $c := \mathrm{corr}(\mathbf{t}_i^*),\ c_2 := \max_{\mathbf{t} \in \{\mathcal{N}_i(\mathbf{s}) \setminus \mathbf{t}_i^*\}} \mathrm{corr}(\mathbf{t})$.
2.8: **if** $c \geq \tau$ **and** $c - c_2 \geq \epsilon$ **and** $\mathbf{T}(x, y) = 0$ **then**
2.9: $G := G + 1,\ C := C + c$.
2.10: **if** $\mathbf{T}'(\mathbf{A}(x, y)) = 1$ **then**
2.11: $U := U + 1$.
2.12: **end if**
2.13: Update the matching maps $\mathbf{T}(x, y) := \mathbf{T}'(\mathbf{A}(x, y)) := 1$ and
2.14: the seed queue $\mathcal{S} := \mathcal{S} \cup \{\mathbf{t}_i^*\}$.
2.15: **end if**
2.16: **end for**
2.17: **end while**
2.18: **return** growth rate $\bar{g} := \frac{G}{\mu}$, average correlation $\bar{c} := \frac{C}{G}$, average uniqueness violation $\bar{u} := \frac{U}{G}$.

The procedure is presented in pseudo-code as Alg. 2. The input are the images $\mathbf{I}, \mathbf{I}'$, the set of initial seeds $\mathcal{S}$ and the maximum number of growing steps μ. The output are three statistics $\bar{g}, \bar{c}, \bar{u}$ which characterize the (in)correctness of the input correspondence.

The algorithm computes the image correlation $\mathrm{corr}(\mathbf{s})$ of all initial seeds $\mathbf{s} \in \mathcal{S}$, Alg. 2.2, as Moravec's normalized cross-correlation [12] (MNCC) of 5×5 pixel windows centered at pixels (x, y) in the reference image and $\mathbf{A}(x, y)$ in the target image, deformed with accordance to the affinity $\mathbf{A}$. Set $\mathcal{S}$ is organized as a correlation-priority queue. A seed is removed from the top of the queue, and for all its 4-neighbors (left, right, up, down) in the reference image, the best correlating candidate in $\mathcal{N}_i$ is found (out of 9 possible positions in the target image), Alg. 2.6, such that

$$\begin{aligned}
\mathcal{N}_1(s) &= \left\{(x-1, y, \mathbf{A}_{i-1,j}) \mid i, j \in \{-1, 0, 1\}\right\}, \\
\mathcal{N}_2(s) &= \left\{(x+1, y, \mathbf{A}_{i+1,j}) \mid i, j \in \{-1, 0, 1\}\right\}, \\
\mathcal{N}_3(s) &= \left\{(x, y-1, \mathbf{A}_{i,j-1}) \mid i, j \in \{-1, 0, 1\}\right\}, \\
\mathcal{N}_4(s) &= \left\{(x, y+1, \mathbf{A}_{i,j+1}) \mid i, j \in \{-1, 0, 1\}\right\},
\end{aligned} \tag{2}$$

where

$$\mathbf{A}_{i,j} = \begin{bmatrix} a_1 & a_2 & a_3 + a_1 i + a_2 j \\ a_4 & a_5 & a_6 + a_4 i + a_5 j \end{bmatrix}. \tag{3}$$

If the highest correlation exceeds threshold $\tau = 0.5$ and the difference of the first and second highest correlations is above $\epsilon = 0.01$ and the point is not matched in the reference image, a new match is found, Alg. 2.8. Next, the counter for the region size G is incremented, correlation value c is added to sum C. If the pixel in the target image $\mathbf{I}'$ is already matched, the counter for uniqueness violation U is incremented, Alg. 2.11. The binary matching maps $\mathbf{T}$ and $\mathbf{T}'$ are updated and the found match becomes a new seed. Up to four seeds are created in each growing step.

The process continues until there are no seeds in the queue or the algorithm is stopped when reaching the maximum number of growing steps μ, Alg. 2.3.

Discussion. Unlike Vedaldi and Soatto's region growing algorithm [19], Algorithm 2 includes no explicit regularization either of the mapping or of the shape of the cosegmented regions. The reason is that the algorithm grows only in informative areas with distinguishing signal (texture), so regularization is not needed. Areas without texture are ambiguous and do not help to distinguish correct and incorrect correspondences. Growth is restricted to unambiguous areas by requiring MNCC statistic[3] to stay above a threshold τ, and by requiring the distance of the first and second highest correlation to be above ϵ, Alg. 2.8. Parameters τ and ϵ were set empirically, as a tradeoff between reliable growth of correct correspondences and preventing the growth into ambiguous regions. There is only an implicit surface smoothness via the disparity gradient cannot change too much in (2), similarly as [8].

Usually in wide baseline dense stereo [15, 11], local affine parameters (a_1, a_2, a_4, a_5) representing a matching window deformation due to surface slant are optimized after each growing step, in order to enable the growth on curved or projectively distorted surfaces as far as possible. However, our goal is different; for correspondence verification the surface need not be grown too far. Therefore, in our algorithm, the parameters inherited from the initial seed are kept constant, which is faster than the iterative optimization. A small imprecision of the local affine parameters is not critical.

2.2. Statistical correspondence quality

Ideally, the correspondence quality would be a function of the probability that the pair of grown patches from the correspondence is a projection of a 3D surface, calculated *e.g.* via MRF on the image grid as by global methods in dense stereo [6]. However, finding the MAP solution is computationally intensive even for simple fields. Therefore, we use the efficient growing algorithm as a subopti-

[3] Note, the MNCC is a zero mean normalized correlation. For areas without texture, after subtracting the mean values of signals in windows, the rest is an uncorrelated noise which results in a low value of the statistic.

mal solution and model the correspondence quality based on elementary statistics which discriminate the correct and incorrect correspondences.

We observed, the growth rate $\bar{g}$ is typically larger for correct correspondences than for incorrect as reported by [19], but not always since the correct correspondence may lie on a small surface or be partially occluded. The average correlation in the region $\bar{c}$ is also typically higher for correct correspondences, but incorrect correspondences may accidently have high correlation due to locally similar structures especially for small regions. The average uniqueness violation $\bar{u}$ (deviation from bijective matching) when growing the region is also quite discriminative. It is often higher for wrong correspondences, while for the correct ones the mapping is more coherent. To forbid the uniqueness violation as in [8] is not suitable in our wide-baseline setup due to possibly high surface slant or scale changes.

The statistics returned by the growing algorithm are combined with a ratio of the first to second closest distance of SIFT descriptors s_r [9]. The problem of estimation of high dimensional likelihood ratio is avoided by projecting the four dimensional statistic into a 1D scalar quality $q_i = \mathrm{f}(s_r, \bar{g}_i, \bar{c}_i, \bar{u}_i)$ which expresses a confidence on correctness of the correspondence. This is done using the Support Vector Machine (SVM) trained on a set of exemplar positive and negative correspondences, see Sec. 3.

In consecutive decision stages i, the statistics are more discriminative, as the growth has an increasing maximum number of steps μ_i, Alg. 1.2. Thus a different SVM θ_i is trained for each decision stage i.

The likelihoods $p_i(q|+1)$ and $p_i(q|-1)$ of positive and negative class respectively were estimated by Parzen window method with a moving average kernel. The likelihood ratio L_i given the SVM output q_i is computed from linearly interpolated likelihood estimates. When the q_i is out of estimated bounds a Gaussian extrapolation is applied.

2.3. Wald's sequential decision

Let x be an object belonging to one of two classes $\{-1, +1\}$. In our case, the classified objects are correspondences and the classes are "correct" (1) and "incorrect" (-1). Next, let an ordering on the set of measurements $\{x_1, \ldots, x_n\}$ on x be given. Here measurements x_i are scalar values, oriented distances from SVM decision boundaries after growing step i.

A sequential decision strategy is a set of decision functions $S = \{S_1, \ldots, S_n\}$, where $S_i : \{x_1, \ldots, x_i\} \rightarrow \{-1, +1, \sharp\}$. The strategy S takes one measurements at a time. The '$\sharp$' sign stands for a "continue" (do not decide yet). If a decision is '$\sharp$', x_{i+1} is obtained and S_{i+1} is evaluated. Otherwise, the output of S is the class returned by S_i.

In two-class classification problems, errors of two kinds can be made by strategy S. Let us denote α_S the probability of rejecting a correct correspondence (x belongs to $+1$ but is classified as -1) and β_S the probability of accepting an incorrect correspondence (x belongs to -1 but is classified as $+1$). A sequential strategy S is characterized by its error rates α_S and β_S and its average evaluation time $\bar{T}_S = E(T_S(x))$ where the expectation is over $p(x)$, and $\bar{T}_S$ is the expected evaluation time (or time-to-decision) for strategy. An optimal strategy for the sequential decision making problem is then defined as

$$S^* = \arg\min_S \bar{T}_S \tag{4}$$

$$\begin{aligned} \text{s.t.} \quad \beta_S &\leq \beta, \\ \alpha_S &\leq \alpha \end{aligned}$$

for specified α and β.

Wald [20] proved that the solution of the optimization problem (4) is the *sequential probability ratio test*.

Sequential Probability Ratio Test. Let x be an object characterized by its hidden state (class) $y \in \{-1, +1\}$. The decision about the hidden state is based on successive measurements $x_1, x_2, \ldots$. Let the joint conditional density $p(x_1, \ldots, x_m | y = c)$ of the measurements $x_1, ..., x_m$ be known for $c \in \{-1, +1\}$.

SPRT is a sequential strategy S^*, which is defined as

$$S^*_m = \begin{cases} +1, & L_m \geq A \\ -1, & L_m \leq B \\ \sharp, & B < L_m < A \end{cases} \tag{5}$$

where L_m is the likelihood ratio

$$L_m = \frac{p(x_1, ..., x_m | y = -1)}{p(x_1, ..., x_m | y = +1)}. \tag{6}$$

The constants A and B are set according to the required error of the first kind α and error of the second kind β. Optimal A and B are difficult to compute in practice, but tight bounds are easily derived. It can be shown that setting the thresholds A and B to

$$A = \frac{1-\beta}{\alpha}, \qquad B = \frac{\beta}{1-\alpha} \tag{7}$$

is close to optimal.

In the SCV algorithm, we assume that all information about a correspondence is contained in the statistics from the last growth step: $p(q_i|y) = p(q_1, \ldots, q_i|y)$. Therefore only 1D PDFs are needed to carry out the SPRT test. Estimation of scalar PDFs poses no technical problems as discussed in previous section.

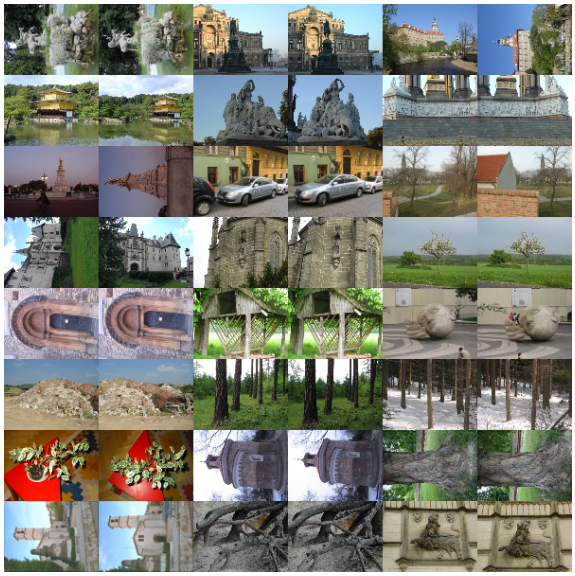

Figure 2. The set of training images.

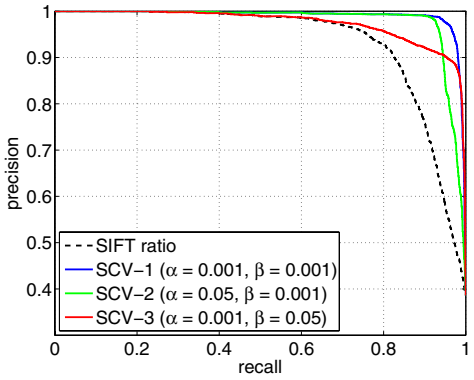

Figure 3. Discriminability of the SCV algorithm. The precision-recall curves for SCV with various setting of false positive and false negative rates and for the SIFT ratio alone.

3. Experiments

The complete set of 24 image pairs used in a training set of correspondences is shown in Fig. 2. For all image pairs, MSERs were detected, LAFs were constructed [10, 14] and SIFT descriptors were computed on normalized patches. Preliminary matching was performed which produced a set of tentative correspondences. Finally, RANSAC was run on each pair of this set to estimate the epipolar geometry. We took as the positive correspondence examples inliers of the epipolar geometry, while the outliers were taken as the negative examples. We have manually relabeled the correspondences which were accidently consistent with the epipolar geometry but were in fact incorrect. We obtained approximately 6200 positive and 9800 negative correspondence examples. This is our ground-truth set.

The ground-truth set was used to adapt SVM models and to estimate the likelihoods via Parzen windowing. We used a publicly available Statistical Pattern Recognition Toolbox [5] to train the linear SVM.

3.1. The SCV efficiently increases discriminability

Discriminability is the algorithm's ability to distinguish correct and incorrect correspondences while the efficiency is related to speed of the decision. We show the SCV algorithm is more discriminative than a standard SIFT ratio and the sequential decision making process speeds the algorithm up of the expense of a small discriminability loss.

All the measurements are on an independent test set, which is a randomly taken half of the ground-truth set, while the other half was used to the model learning: SVM training and likelihood estimates.

The discriminability of SCV was measured using a precision-recall curve. The SCV algorithm assigns likelihood ratio L to all N correspondences in test set. The correspondences are sorted according to their likelihood ratio, such that $L_{(1)} \geq L_{(2)} \geq ... \geq L_{(N)}$. The *precision* is defined as Q_n^+/n, where Q_n^+ is a number of correct correspondences when retrieving first n samples according to the likelihood ratio ordering. The *recall* is defined as Q_n^+/Q_N^+.

We compared the discriminability of the SCV algorithm with various settings of Wald's test parameters (α, β) and the SIFT ratio, Fig. 3. The SIFT ratio curve is computed in the same way as explained above, the sorting is performed on the negative ratio of SIFT distances. The SCV algorithm outperforms the SIFT ratio alone for all three settings. The SCV-1 ($\alpha = 0.001, \beta = 0.001$) is the most strict setting which has the highest discriminability. The SCV-2 ($\alpha = 0.05, \beta = 0.001$) allows more false negatives, while the SCV-3 ($\alpha = 0.001, \beta = 0.05$) more false positives, but they both are more efficient in terms of number of window correlations they had to compute.

In Fig. 4, we compared the three (α, β) settings of SCV algorithm with the non-sequential version (CV), which which does not decide until the last stage performing maximally $\mu_4 = 1000$ growing steps. We measured the average number of window correlations per correspondence C which had to be computed, and the percentage d_i of correspondences decided in i-th stage of the algorithm.

These values differ for correct and incorrect correspondences, so besides the mean values d_i, C (which depends on the percentage of correct correspondences in the test set), we show the tables for correct correspondences d_i^+, C^+ and wrong correspondences d_i^-, C^-.

We can see that in the non-sequential (CV) algorithm, wrong correspondences take more than four times fewer correlations than correct correspondences. This behavior is expected, since the algorithm stops growing when there are no high correlating neighbors and typically finishes by exhausting the seed queue $\mathcal{S}$ before the maximum number of

all correspondences

	d_1	d_2	d_3	d_4	$C \times 10^3$
CV	0	0	0	100	8.6
SCV-1	9.7	12.7	9.5	68.1	4.1
SCV-2	9.7	63.1	16.4	10.8	2.1
SCV-3	33.6	9.5	4.2	52.8	2.0

correct correspondences only

	d_1^+	d_2^+	d_3^+	d_4^+	$C^+ \times 10^3$
CV	0	0	0	100	16.2
SCV-1	23.8	23.5	22.9	29.8	4.8
SCV-2	23.8	27.9	24.8	23.4	4.2
SCV-3	79.9	11.3	5.8	3.1	0.4

incorrect correspondences only

	d_1^-	d_2^-	d_3^-	d_4^-	$C^- \times 10^3$
CV	0	0	0	100	3.7
SCV-1	0.8	6.0	1.1	92.2	3.6
SCV-2	0.8	85.1	11.1	2.9	0.7
SCV-3	4.5	8.3	3.1	84.0	3.1

Figure 4. Efficiency of the algorithm. The d_i is a percentage of correspondences decided in ith stage of the decision process. The C is an average number of window correlation per correspondence.

growing steps is reached, see Alg. 2.3. This is convenient, as the matching of tentative correspondences can be much more permissive without losing much efficiency which is shown in next experiment.

Further we can see, the sequential decision can speed up the process by factor of two (SCV-1) or more than four (SCV-2, SCV-3) compared to the non-sequential algorithm without losing much discriminability. The curve in Fig. 3 of the non-sequential algorithm (CV) is almost identical to the SCV-1, therefore it is not shown. In the tables we can also verify that the SCV-2 having higher allowed false negative rate tends to decide negative correspondences in lower stages of the sequence speeding up the decision process by factor of more than 5, while the SCV-3 vice-versa, speeding up the decision process of positive correspondences by factor of 12.

Computational complexity. The dominant operation in SCV algorithm is computation of correlations, the other overheads (SVM classification, Wald's SPRT) are negligible. Considering the average number of tentative correspondences 1000, each requiring on average $C = 2100$ correlations (*c.f*. Fig. 4) we end up with approximately 2×10^6 correlations per image pair. This can be computed on recent CPU in about 0.5 seconds and about 20–100 times faster in parallel computation on a modern GPU.

3.2. Challenging wide baseline stereo scenes

The results on correspondence selection on difficult wide baseline stereo scenes are shown in Fig. 5. These scenes are challenging due to a small overlap of a common part, a high degree of noise in the images (Raglan), a complex 3D structure with many occlusions (Forsythia), and due to locally similar background which is not the same (Orange). To find the epipolar geometry at all, the matching process generating the tentative correspondences preceding RANSAC had to be much more permissive, otherwise there were not enough correct among tentative correspondences. We allowed more than one-to-one mapping in tentative correspondences which lead to a higher number of inliers but also a higher number of outliers (about 90 percent).

Plots in the last column in Fig. 5 show the curves of precision in the best n retrieved correspondences. This is important for progressive RANSAC procedure [2] which samples tentative correspondences in a given order to fit the model. So, we can see that for our algorithm, for all three scenes, this procedure would terminate successfully after 1 iteration, since there is most of the correct correspondences evaluated with high quality placed in first ranks. This is neither the case when the ordering of tentative correspondences is given by the negative ratio of SIFT distances, nor the SIFT distances alone which is even worse.

The sequential algorithm (SCV-2) and its non-sequential version (CV) are compared. For all the scenes, the results of SCV-2 are slightly worse than for CV, but it is much faster. For the Raglan scene it took 0.5×10^3 and 2.5×10^3, for the forsythia 0.6×10^3 and 5.7×10^3, and for the orange scene it was 1.2×10^3 and 6.3×10^3 of average number of computed window correlations per correspondence for the sequential and full algorithm respectively. The reason why the decision is even faster here than on the test set in previous experiment is that there are many more wrong correspondences which are faster to decide and these outliers are quickly decidable in early stages of the sequence.

3.3. Image retrieval

We show the benefits of SCV algorithm in a large scale image retrieval setup, using the data set from Nistér and Stewénius benchmark [13]. It consists of 10200 images in groups of four that show the same object. Each image is retrieved from the whole data set. For each query the top N images are returned, and the score counting how many of the correct answers are in top K is computed. In the benchmark K is set to 4, giving the highest score 4, if the algorithm manages to retrieve top four images that matches four instances of the object in the data set. Since the query image is also present in the data set, the worst score of algorithm returning only the query in top K is 1. The overall performance of the algorithm is computed as the average score of all 10200 queries from the data set.

A part of the solution proposed by Nistér was reimplemented. The MSERs [10] and LAFs [14] were computed on each of the images. Each of approximately 7 millions

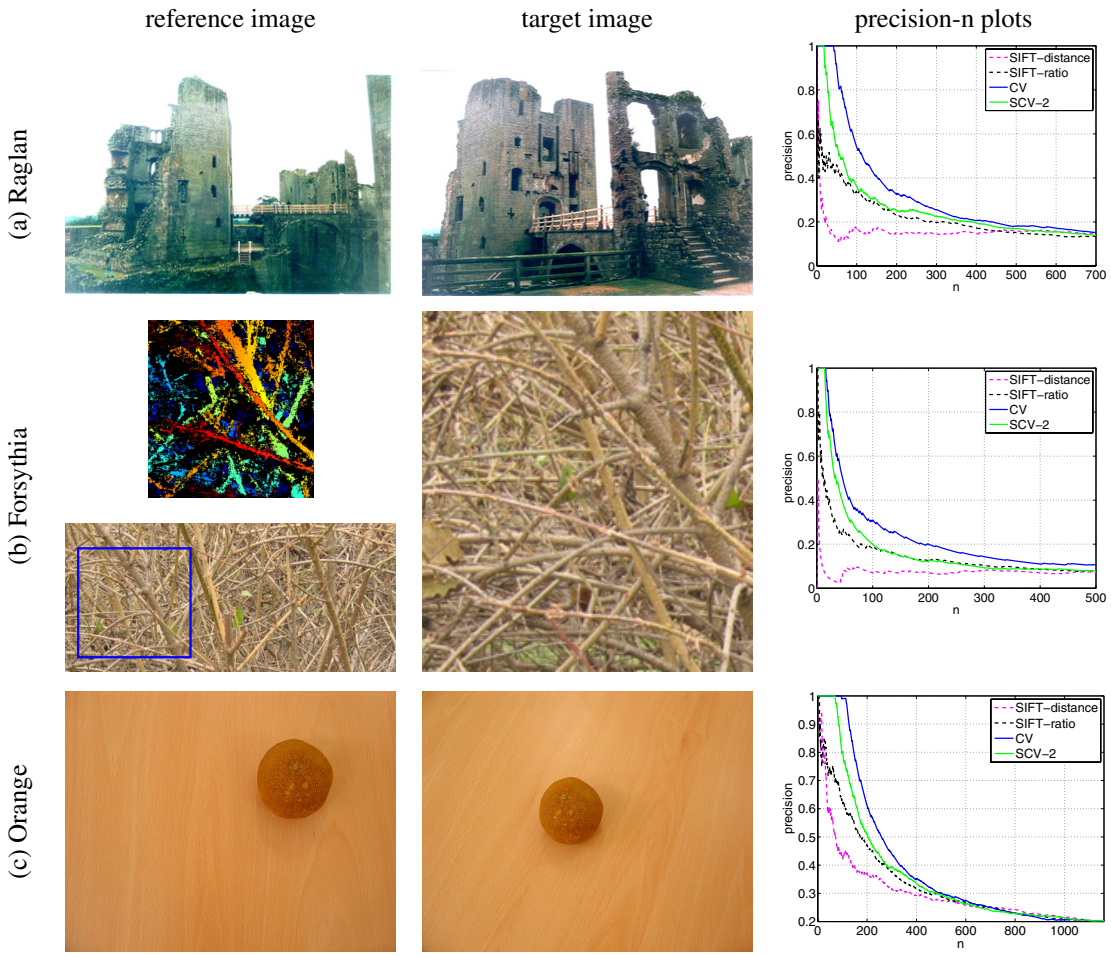

Figure 5. Results on challenging wide-baseline scenes. For Forsythia, we show the color coded depth map of a common part (marked approximately by blue frame) to prove the 3D structure of the image pair. Notice, the orange is placed in a different place in the table with no true correspondence on it.

LAFs was described using SIFT descriptor [9] computed on an affine normalized patch. Then, similarly to visual words approach proposed by Sivic and Zisserman [17], we build visual words vocabulary consisting of 1 million k-means in the SIFT descriptor space and assign all the descriptors in the images to the nearest visual word. Each visual word in a given document is weighted using TFIDF (Term Frequency – Inverse Document Frequency) measure from text retrieval. The similarity of two documents is then the L1 distance between their vectors of the visual words weights. The top K most similar documents are retrieved. Described approach is similar to the flat scoring of Nistér and Stewénius. It achieves the average score of 3.40 images retrieved per query on the whole dataset.

To evaluate the performance of the SCV algorithm we took all queries that can be improved by verifying reasonable number of images retrieved by VW method, *i.e*. queries where there is at least one image of the retrieved object with rank 5 to 20. There are 1904 such query images. The overall score, the average number of correct images among top 4, achieved by TFIDF visual word ranking is 2.32 for these queries. Top 20 score, *i.e*. the average number of correct images among top 20, is 3.54. This is the upper bound of the performance for a retrieval algorithm that resorts the top 20 retrieved images.

For comparison, tentative correspondences were computed as nearest SIFT descriptions for each of 1904 query images and its top 20 retrieved images giving altogether 38080 pairs. Tentative correspondences of each pair were then verified using the SCV ($\alpha = 0.02, \beta = 0.001$) algorithm. Finally, new ranking was established according to the number of SCV correspondences found in each pair of the images. We also compared our method to the ranking based on SIFT correspondences (rank is based on the number of correspondences with SIFT distance ratio < 0.8). The performance of the SCV algorithm is compared in a histogram of ranks of the four correct images in answer to each query (see Fig. 6). Clearly, SCV significantly improves the rank-

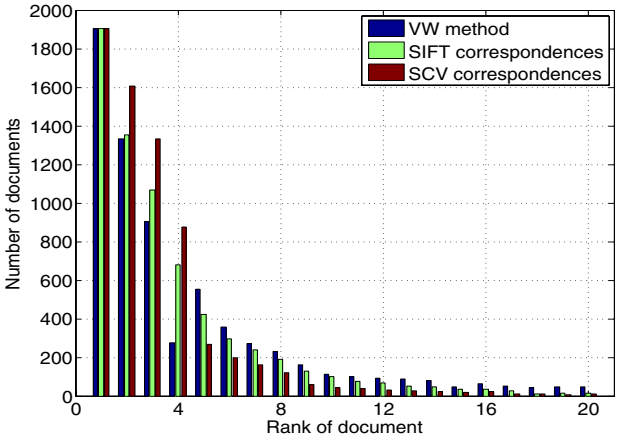

Figure 6. Ranking of documents based on the visual word method, the number of SIFT correspondences with distance ratio < 0.8 and the number of SCV correspondences.

score	higher	same	lower
SIFT	750	926	228
SCV	1224	597	83

Table 1. Comparison of the scores of the VW method and rankings based on SIFT and SCV correspondences.

ing of the correct images bringing most of them to top 4. Its overall top 4 score on the 1904 query images is 5717 resulting in average 3.00, the average top 5 score is 3.14. The overall top 4 score for SIFT correspondences is 5004 resulting in average 2.63 and the average top 5 score is 2.85.

At last, we compared the achieved top 4 scores of both methods to the visual words method in Tab. 1. It shows the ranking is improved or unchanged with SCV in 95% of cases. Wrong ranking occurs typically for images of different objects with little texture (usually slightly blurred) on the same structured background. In this case, the most of, in fact correct, correspondences are found in the background which does not help retrieving a correct image.

4. Conclusions

We have presented a method which is able to efficiently distinguish correct and incorrect correspondences, via collecting statistics while cosegmenting gradually larger regions. We have shown it benefits the matching process in challenging wide baseline scenes and improves results in a large scale image retrieval. Note that, the statistical model – parameters of SVM and likelihoods in SPRT, was learned on a small and probably non-representative database. We expect, the results would further improve if the set was enlarged or adapted to a specific domain.

Acknowledgement. The research was supported by Czech Academy of Sciences project 1ET101210406, by EC projects FP6-IST-027113 eTRIMS, ICT-215078 DIPLECS, and by Ministry of Education as a part of the specific research at the CTU in Prague.

References

[1] J. Čech and R. Šára. Efficient sampling of disparity space for fast and accurate matching. In *BenCOS Workshop, CVPR*, 2007.

[2] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *CVPR*, pages 220–226, 2005.

[3] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[4] V. Ferrari, T. Tuytelaars, and L. van Gool. Simultaneous object recognition and segmentation from single or multiple image views. *IJCV*, 67(2):159–188, 2006.

[5] V. Franc and V. Hlavac. Statistical Pattern Recognition Toolbox for Matlab, 2007.

[6] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, pages 508–515, 2001.

[7] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *BMVC*, September 2006.

[8] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *PAMI*, 24(8):1140–1146, 2002.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, 2002.

[11] Z. Megyesi, G. Kós, and D. Chetverikov. Dense 3D reconstruction from images by normal aided matching. *Machine Graphics and Vision*, 15:3–28, 2006.

[12] H. P. Moravec. Towards automatic visual obstacle avoidance. In *IJCAI*, page 584, 1977.

[13] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[14] S. Obdrzalek and J. Matas. Sub-linear indexing for large scale object recognition. In *BMVC*, 2005.

[15] G. P. Otto and T. K. W. Chau. 'Region-growing' algorithm for matching of terrain images. *IVC*, 7(2):83–94, 1989.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[17] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[18] C. V. Stewart, C.-L. Tsai, and B. Roysam. The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *Medical Imaging*, 22(11):1379–1394, 2003.

[19] A. Vedaldi and S. Soatto. Local features, all grown up. In *CVPR*, pages 1753–1760, 2006.

[20] A. Wald. *Sequential analysis*. Dover, New York, 1947.

[21] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Alignment of challenging image pairs: Refinement and region growing starting from a single keypoint correspondence. *PAMI*, 23(11):1973–1989, Nov. 2007.