Efficient Sequential Correspondence Selection by Cosegmentation

Jan Čech, *Member*, *IEEE*, Jiří Matas, *Member*, *IEEE*, and Michal Perdoch, *Member*, *IEEE*

Abstract—In many retrieval, object recognition, and wide-baseline stereo methods, correspondences of interest points (distinguished regions) are commonly established by matching compact descriptors such as SIFTs. We show that a subsequent cosegmentation process coupled with a quasi-optimal sequential decision process leads to a correspondence verification procedure that 1) has high precision (is highly discriminative), 2) has good recall, and 3) is fast. The sequential decision on the correctness of a correspondence is based on simple statistics of a modified dense stereo matching algorithm. The statistics are projected on a prominent discriminative direction by SVM. Wald's sequential probability ratio test is performed on the SVM projection computed on progressively larger cosegmented regions. We show experimentally that the proposed sequential correspondence verification (SCV) algorithm significantly outperforms the standard correspondence selection method based on SIFT distance ratios on challenging matching problems.

Index Terms—Correspondence, matching, verification, sequential decision, growing, cosegmentation, stereo, image retrieval, learning.

1 Introduction

Many successful image retrieval, object recognition, and wide-baseline stereo methods exploit correspondences of distinguished regions.[1] Most real-world visual recognition problems are large scale, where correspondences between regions from a query (test) image and many database (training) images of objects or scenes are sought. To achieve acceptable response times, large problems require the time complexity of the region matching process to be sublinear in the size of the database; the memory footprint of the database representation becomes a concern too. The standard solution is to describe regions with a compact descriptor such as SIFT [1] or some discretization of it (e.g., "visual words" [2]), and to store database image representations in a search tree (k-d [1], metric [3], and k-means [4], [5], [6]).

The matching process typically proceeds as follows [7], [8], [1]: Distinguished regions are detected in the image and local affine or similarity covariant coordinate frames are constructed for each region. *Measurement regions*, i.e., image patches of typically rectangular or elliptical shape, are specified in terms of the local coordinate frames. For each region, a descriptor (such as SIFT) is computed from the signal in the measurement region, after both photometric and geometric normalization. Additionally, the descriptor may be compressed by quantization.

This process, schematically visualized in Fig. 2, has the following main characteristics: 1) All steps are performed in individual images independently, 2) the shape and size of the measurement region are a fixed function of the shape and size of the distinguished region, and 3) the descriptor has the same form for all regions, e.g., it is a vector in R^d. These properties facilitate fast sublinear region matching, e.g., via search tree or hashing.

However, the fixed size and shape of the measurement region necessarily involve a compromise. In general, the larger the measurement region, the more discriminative the information inside is. On the other hand, large measurement regions may violate the local planarity assumption of wide-baseline matching methods, and are more likely to straddle object boundary or to be affected by occlusion. Moreover, they are more sensitive to localization errors of local frames. For "noncompact" objects, such as elongated and wiry ones, the compact fixed shape is problematic.

Consider, for instance, the two images depicted in Fig. 3a. Only a very small circular or rectangular region around the distinguished region on the branch will not include signal from the background, which is different for the two views. On the other hand, consider the images shown in Fig. 3b. The measurement region inside the circle is too small since any descriptor computed from the region will be close to identical for both images on the right—the correct match cannot be reliably established.

A better estimate of correspondence quality (a prediction of it being correct) can be obtained by looking at both test and training images simultaneously, e.g., by attempting to expand the correspondence domains, which is illustrated in Fig. 1. The value of correspondence growing methods has been demonstrated in [9] and [10], sometimes with impressive results, e.g., those achieved by the dual-bootstrap method [11], [12]. Most approaches to simultaneous cosegmentation and registration focus on the problem of

[1] Terms "transformation-covariant regions," "viewpoint invariant features," "interest points," "salient points," and "patches" also appear in the literature. We adopt the term "distinguished region" as a concise shortcut for the self-explanatory, but unwieldy "repeatably detectable transformation-covariant regions."

- *The authors are with the Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Technicka 2, 166 27 Praha 6, Czech Republic.*
E-mail: {cechj, matas, perdom1}@cmp.felk.cvut.cz.

Manuscript received 3 Nov. 2008; revised 6 May 2009; accepted 11 July 2009; published online 9 Oct. 2009.
Recommended for acceptance by B. Schiele.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-11-0760.
Digital Object Identifier no. 10.1109/TPAMI.2009.176.

0162-8828/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

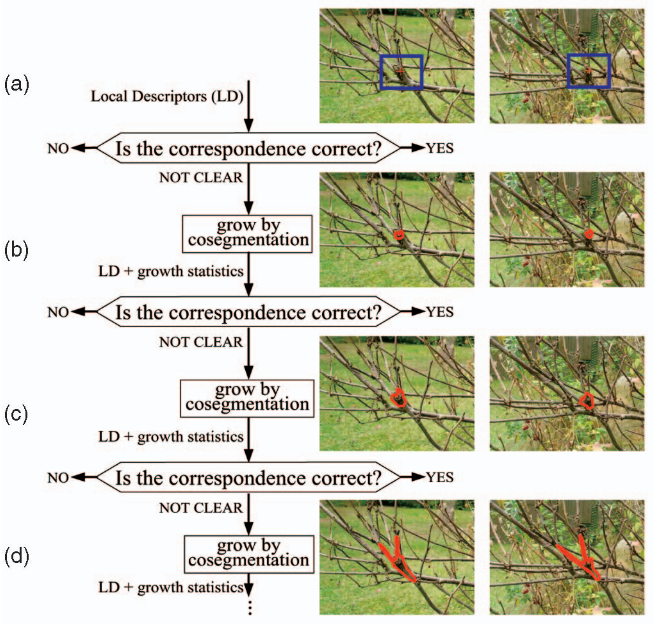


Fig. 1. The basic idea of the SCV algorithm. (a) First, a decision is attempted on the basis of local descriptors (LDs) computed from fixed shape measurement region (in blue). (b)-(d) If the correctness of the correspondence cannot be decided reliably, the correspondence is successively grown by cosegmentation (in red), collecting additional evidence for the decision.

finding the largest corresponding domain and codomain [11], [10], [13], [14].

Our objective is almost the 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 progressively growing size. We formulate the problem as sequential decision making, which is solved by performing Wald's sequential probability ratio test. The test is based on simple statistics of a modified dense stereo matching algorithm, which is projected on a single prominent discriminative direction by a linear support vector machine (SVM).

Of course, we do not want to lose the excellent large-scale matching properties of descriptors based on measurement regions of fixed size and scale. The cosegmentation process is therefore only applied to *tentative correspondences* obtained by a sublinear process, such as kD-tree search. In fact, if followed by correspondence verification, any such process for generating tentative correspondences can be set to be much more permissive, outputting a higher number of correspondences with lower inlier ratios, but containing a larger number of inliers. After filtering by simultaneous cosegmentation, inlier ratios are (more than) recovered and the larger number of inliers leads to higher recognition rates. We show on challenging problems 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, thus producing a higher number as well as higher percentage of correct correspondences.

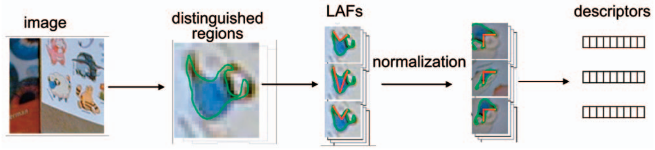


Fig. 2. Describing a distinguished region by a compact descriptor invariant to local geometric and photometric changes. “LAFs” stands for “local affine frames.”

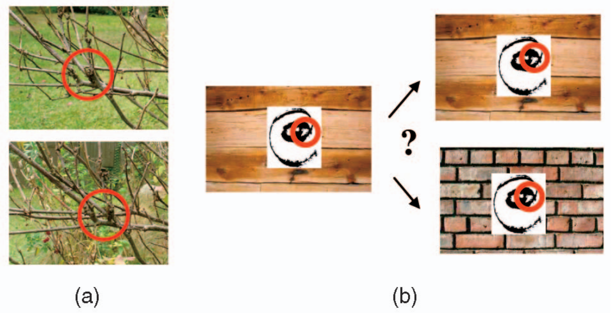


Fig. 3. Problems with a fixed shape and size of the measurement region. (a) Measurement region too large and/or with unsuitable shape. (b) Measurement region too small.

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.

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 expansion 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 [4], 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 Section 2, and the training data and learning procedure for the sequential classifier in Section 3. Experimental validation is presented in Section 4. Conclusions are summarized in Section 5.

This paper is a significantly extended and modified version of [15].

2 The Sequential Correspondence Verification Algorithm

The motivation of the approach is to distinguish, as fast as possible, correct and incorrect correspondences via dense matching, i.e., by a pixel-to-pixel correspondence 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 two error rates control in an intuitive way the trade-off between the number of correspondences (high when false negative rate is low), inlier ratio (high when false positive rate is low), and the speed of the method (low for low error rates). The importance of these three factors differs in applications.

The proposed sequential correspondence verification algorithm, i.e., Algorithm 1 (SCV) is overviewed in Fig. 4.

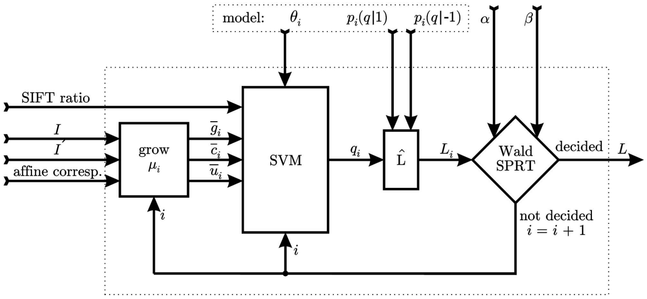


Fig. 4. The sequential correspondence verification algorithm.

The basic idea (see Fig. 1) is to perform Wald's sequential probability ratio test (SPRT), having learned the distributions of elementary statistics generated by the growing process.

The algorithm proceeds in decision stages indexed by i. In the first stage, a fast dense stereo matching algorithm, as given in Section 2.1, is initialized by a tentative correspondence of a pair of local affine frames. The verification proceeds by attempting to match discriminative, i.e., high variance, neighboring pixels. After a certain number of growing steps μ_i, the cosegmentation process returns three simple statistics $(\bar{g}_i, \bar{c}_i, \bar{u}_i)$ characterizing the quality of the correspondence: the growth rate $\bar{g}_i$—the size of the grown region divided by the maximum number of attempted growing steps μ_i, the average correlation $\bar{c}_i$ of the region, and the average number of pixels violating the uniqueness $\bar{u}_i$, i.e., nonbijectivity matching.

Algorithm 1. Sequential Correspondence Verification (`scv`)

```
Require: images $\mathbf{I}$, $\mathbf{I}'$,
    correspondence with affine frame $(x, y, \mathbf{A})$,
    SIFT ratio $s_r$,
    false positive and false negatives rates $(\alpha, \beta)$,
    model: learned SVM parameters $\theta_i$,
        likelihoods $\mathbf{p}_i(q|+1)$, $\mathbf{p}_i(q|-1)$.

1.1: init_grow$(x, y, \mathbf{A})$.
1.2: for $i = 1$ to max. number of decision stages do
1.3:     $\mu_i = \begin{cases} 0, & i=1, \\ \gamma^{i-2}, & i>1. \end{cases}$
1.4:     $(\bar{g}_i, \bar{c}_i, \bar{u}_i) := \mathrm{grow}(\mathbf{I}, \mathbf{I}', \mu_i)$.
1.5:     $q_i := \mathrm{SVM}(s_r, \bar{g}_i, \bar{c}_i, \bar{u}_i, \theta_i)$.
1.6:     $L_i := \frac{\mathbf{p}_i(q_i|+1)}{\mathbf{p}_i(q_i|-1)}$.
1.7:     if Wald SPRT$(L, \alpha, \beta)$ is conclusive then break.
1.8: end for
1.9: return likelihood ratio $L_i$ (of the last iteration).
```

The vector of statistics is projected by a linear SVM to a scalar quantity q_i, which avoids estimation of high-dimensional class-conditional probabilities (likelihoods). Instead, only the likelihoods of the projections $\mathbf{p}_i(q| + 1)$ and $\mathbf{p}_i(q| - 1)$ of correct and incorrect correspondence classes are computed.

The region statistics are augmented with the first to the second nearest SIFT descriptor distance ratio s_r, a standard measure for selection of tentative correspondences [1]. We call s_r the *SIFT ratio*. The Wald's SPRT is performed on the likelihood ratio $L_i = \mathbf{p}_i(q_i| + 1)/\mathbf{p}_i(q_i| - 1)$. If the SPRT test is conclusive, the algorithm terminates and the correspondence is assigned the likelihood ratio L_i of the decision. Otherwise, another decision stage i is performed, i.e., the cosegmentation is resumed with a new limit of attempted growing steps μ_i, Algorithm 1, Step 1.3, 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 100 and the largest pixel growth is $\mu_{100} = 1{,}000$ steps. We observed that the error does not decrease after a larger growth, see Fig. 10b. In principle, it is possible to perform Wald's SPRT test after each growing step, which would be the fastest strategy if the test execution time was zero. However, the statistics are average values; therefore, in order to have a constant influence of new measurements, we propose growing in steps of a geometric sequence instead, see Step 1.3. The number of decision stages was set empirically to minimize the decision time based on our implementation. These considerations determine γ in Step 1.3.

The choice of the three statistics of the growing process was driven mainly by computational requirements. The statistics were selected from a larger pool of easily computable characteristics. We experimented, e.g., with geometric deviation from the transformation implied by the local affine frame correspondence, mean intensity difference of the regions, and difference of Harris-like cornerness values. None of them discriminated well.

Besides benefiting computational speed, we attribute the good generalization of the sequential classifier to the simplicity of the characterization of the growing process. For instance, the sequential classifier performed almost equally well on correspondences established on distinguished regions other than those it was trained on, despite the fact that it is unlikely that image statistics in general are the same for different detection processes. Moreover, even after a very modest number of training examples, the classifier performed well on a very large test set. With satisfactory performance, we did not further investigate the feature selection problem (including feature number).

The linear SVM was our first choice of a projection method as it possessed the desirable properties of fast learning, fast execution, and trivial implementation. Since it leads to a good classification rate, following the structural risk minimization principle, we did not test more complex classifiers.

2.1 Growing Algorithm

The following algorithm explores regions around the input tentative correspondence. The growing mechanism is inspired by [16], [17], [13], [18].

Each correspondence defines a local affine mapping from the reference image $\mathbf{I}$ to the target image $\mathbf{I}'$. The mapping generates several pixel-to-pixel correspondences $\mathbf{s} = (x, y, \mathbf{A})$, where (x, y) is a point 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 $(x', y') = \mathbf{A}(x, y)$.

The procedure is presented in pseudocode as Algorithm 3. The inputs are the images $\mathbf{I}, \mathbf{I}'$, the set of initial correspondences, the seeds $\mathcal{S}$, and the maximum number of growing steps μ. The outputs consists of three statistics $\bar{g}$, $\bar{c}$, and $\bar{u}$, which characterize the (in)correctness of the input correspondence. The growing algorithm keeps its state: the set of correspondence seeds $\mathcal{S}$, matching maps $\mathbf{T}, \mathbf{T}'$, and the counters K, G, C, U that are initialized in Algorithm 1, Step 1.1. In the initialization (Algorithm 2), the image correlation $\mathrm{corr}(\mathbf{s})$ of all initial seeds[2] $\mathbf{s} \in \mathcal{S}$ is computed, Step 2.2, as Moravec's normalized cross-correlation [22] of a 5×5 pixel window $\mathbf{w}$ centered at pixel (x, y) in the reference image and window $\mathbf{w}'$ centered at $\mathbf{A}(x, y)$ in the target image, deformed according to the affinity[3] $\mathbf{A}$

$$\mathrm{corr}(\mathbf{s}) = \frac{2\mathrm{cov}(\mathbf{w}, \mathbf{w}')}{\mathrm{var}(\mathbf{w}) + \mathrm{var}(\mathbf{w}')}, \tag{2}$$

where $\mathrm{cov}(\mathbf{w}, \mathbf{w}')$ is a covariance and $\mathrm{var}(\mathbf{w})$ a variance.

```
Algorithm 2. Initialize growth state (init_grow)
Require: affine correspondence $(x, y, \mathbf{A})$.
2.1: Initialize
        seeds: $\mathcal{S} := (x, y, \mathbf{A})$,
        matching maps: $\mathbf{T}(:,:) := 0, \mathbf{T}'(:,:) := 0$,
        counters: $K := G := C := U := 0$.
2.2: Compute the image correlation for all seeds $\mathbf{s} \in \mathcal{S}$.
```

```
Algorithm 3. The Growing Algorithm (grow)
Require: images $\mathbf{I}$, $\mathbf{I}'$, maximum number of growing steps
    $\mu$, growth state.
3.1: while $K \leq \mu$ and $\mathcal{S}$ not empty do
3.2:     $K := K + 1$.
3.3:     Draw the seed $\mathbf{s} \in \mathcal{S}$ of the best similarity $\mathrm{corr}(\mathbf{s})$.
3.4:     for each of the best neighbors $\mathbf{t}_k^*$ in $\mathcal{N}_k(\mathbf{s})$:
           $\mathbf{t}_k^* = (x, y, \mathbf{A}) = \mathrm{argmax}_{\mathbf{t} \in \mathcal{N}_k(\mathbf{s})} \mathrm{corr}(\mathbf{t})$,
           $k \in \{1, 2, 3, 4\}$
         do
3.5:         $c := \mathrm{corr}(\mathbf{t}_k^*)$,
3.6:         if $c \geq \tau$ and $\mathbf{T}(x, y) = 0$ then
3.7:           $G := G + 1,\ C := C + c$.
3.8:           if $\mathbf{T}'(\mathbf{A}(x, y)) = 1$ then
3.9:             $U := U + 1$.
3.10:          end if
3.11           Update the matching maps
             $\mathbf{T}(x, y) := \mathbf{T}'(\mathbf{A}(x, y)) := 1$ and
3.12:          the seed queue $\mathcal{S} := \mathcal{S} \cup \{\mathbf{t}_k^*\}$.
3.13:        end if
3.14:    end for
3.15: end while
3.16: return statistics: growth rate $\bar{g} := \frac{G}{\mu}$, average
    correlation $\bar{c} := \frac{C}{G}$, average uniqueness violation
    $\bar{u} := \frac{U}{G}$.
```

Set $\mathcal{S}$ is organized as a priority queue according to the correlation. A seed is removed from the top of the queue, and for all its four neighbors (left, right, up, and down) in the reference image, the best correlating candidate in $\mathcal{N}_k(s)$ is found (out of nine possible positions in the target image), Algorithm 3, Step 3.4, such that

$$\begin{aligned} \mathcal{N}_1(s) &= \big\{(x-1, y, \mathbf{A}_{c-1,r}) \mid c, r \in \{-1, 0, 1\}\big\}, \\ \mathcal{N}_2(s) &= \big\{(x+1, y, \mathbf{A}_{c+1,r}) \mid c, r \in \{-1, 0, 1\}\big\}, \\ \mathcal{N}_3(s) &= \big\{(x, y-1, \mathbf{A}_{c,r-1}) \mid c, r \in \{-1, 0, 1\}\big\}, \\ \mathcal{N}_4(s) &= \big\{(x, y+1, \mathbf{A}_{c,r+1}) \mid c, r \in \{-1, 0, 1\}\big\}, \end{aligned} \tag{3}$$

where

$$\mathbf{A}_{c,r} = \begin{bmatrix} a_1 & a_2 & a_3 + a_1 c + a_2 r \\ a_4 & a_5 & a_6 + a_4 c + a_5 r \end{bmatrix}. \tag{4}$$

If the highest correlation exceeds threshold τ and the point is unmatched so far in the reference image, then a new match is found, Step 3.6. Next, the counter for the region size G is incremented and 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, Step 3.9. 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, Step 3.1.

We set correlation threshold $\tau = 0.5$, which was found experimentally, Fig. 9b. On our training set, we observed the classification error rate for several growing steps μ as a function of τ. The error is constant up to $\tau = 0.5$, then it rises. Lower thresholds make the decision process slower, due to worthless growth of incorrect correspondences.

Examples of region growth in the cosegmentation process are depicted in Fig. 5. Local affine frames are shown in blue as a pair of line segments. The three endpoints of the segments are the seeds of the growing process. Blue parallelograms delineate measurement regions, i.e., parts of the image where SIFT descriptors are computed. Yellow marks pixels inside the region at the time of the decision on the correctness of the correspondence made by the SCV algorithm. Red marks pixels that would be chosen if the process was left to grow the maximum number of $\mu = 1{,}000$ steps. Note that: 1) The SCV decision is often reached after growing over a very small number of pixels and 2) the shape of the region is data dependent, preferring areas with edges and high variance of the signal where correlation response is high. Unsurprisingly, pixel correspondences follow correctly the 3D surfaces (branches of the shrub, parts of the fence); in fact, a small local disparity map is computed.

The measurement regions include large parts of the background that is different in the two images. It might be surprising that the regions shown in Fig. 5 are correctly matched, given that the first test in the sequential classifier is based on the SIFT ratio. We believe there are (at least) three reasons for the favorable outcome. First, our test on the SIFT ratio is very permissive. Second, the centers of the regions are on corresponding 3D structures, and SIFT applies a Gaussian weighting function that reduces influence of the outer parts of the parallelogram, which are not corresponding. Finally, SIFT is an array of histograms of

[2] In our experiments, this is realized by local affine frames (LAF) constructed on maximally stable extremal region [19], [20] (MSER), and Hessian-affine points [21]. We take the three point-to-point correspondences of a pair of LAFs as the initial seeds of the growing process.

[3] The simplest and fastest interpolation, the nearest neighbor, was used to compute $\mathbf{w}'$. More advanced interpolation did not bring a significant improvement.

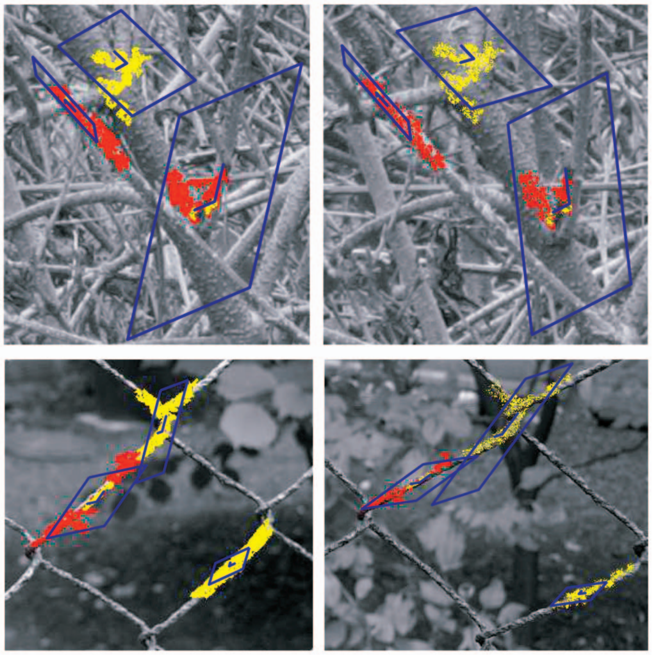


Fig. 5. Examples of region growth in the cosegmentation process. Selected pixels at the time of decision on correctness (in yellow) and after 1,000 growing steps (in red). The red pixels are shown for visualization purposes only, and their correspondence is not evaluated by the SCV algorithm. Corresponding local affine frames and measurement regions of SIFT descriptors are in blue.

gradients. The strong gradients are in correspondence in both pairs of images, and areas without strong edges are irrelevant for the SIFT representation.

2.2 Discussion

Unlike Vedaldi and Soatto's region growing algorithm [9], Algorithm 3 includes no explicit regularization either of the mapping or 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 correlation statistic[4] to stay above threshold τ, Step 3.6. Implicit surface smoothness is enforced. The disparity gradient change is constrained by (3), and similar constraint is applied in [13].

In wide-baseline dense stereo [17], [18], [23], local affine parameters (a_1, a_2, a_4, a_5) representing a window deformation due to surface slant are optimized after each growing step in order to facilitate maximum growth on curved or projectively distorted surfaces. 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 significantly faster than the iterative optimization. Experiments show that a small imprecision of the local affine parameters is not critical, possibly due to the fact that effects of transformation errors are subsumed in disparity (gradients).

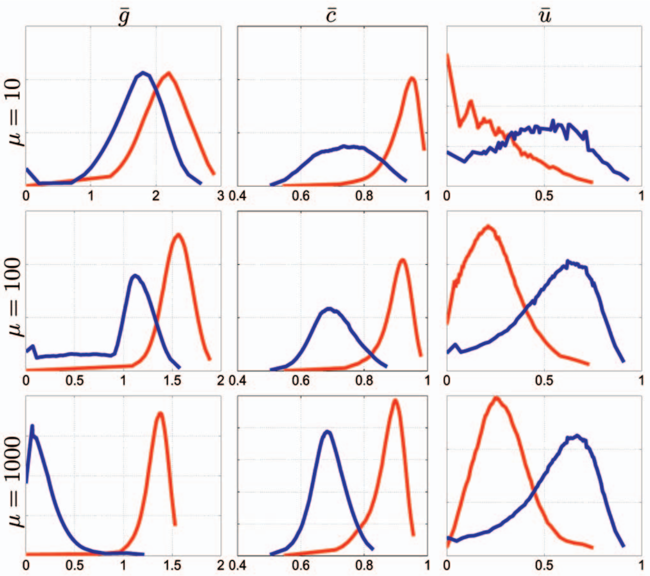


Fig. 6. Estimated class-conditional probability density functions of cosegmentation statistics. From left to right: growth rate $\bar{g}$, average correlation $\bar{c}$, and average uniqueness violation $\bar{u}$; from to top to bottom: for 10, 100, 1,000 growth cycles μ. Correct correspondences (in red) and incorrect correspondences (in blue). Note the gradual reduction in the overlap of the densities, especially in the two leftmost columns.

2.3 Statistical Correspondence Quality

Ideally, correspondence quality would be a function of the probability that a pair of grown patches is a projection of the same 3D surface, as calculated, e.g., via MRF on the image grid by global methods in dense stereo [24]. However, finding the MAP solution is computationally intensive even for simple fields. Therefore, we use the efficient growing algorithm as a suboptimal solution and model the correspondence quality on the basis of elementary statistics that were empirically shown to discriminate correct and incorrect correspondences.

The class-conditional probability densities of the adopted statistics are shown in Fig. 6. We observed that the growth rate $\bar{g}$ is typically larger for correct correspondences than for incorrect, as reported by Vedaldi and Soatto [9]; exceptions include, e.g., situations when a correct correspondence lies on a narrow surface or in cases of partial occlusion. The average correlation in the region $\bar{c}$ is also typically higher for correct correspondences, but incorrect correspondences may accidentally have high correlation on repetitive or locally similar structures, especially on small regions. The average uniqueness violation $\bar{u}$ (deviation from bijective matching) when growing the region is also quite discriminative, see Fig. 6 (right column). The statistics returned by the growing algorithm are combined with the ratio of the first to second closest distance of SIFT descriptors s_r [1], a standard method.

The problem of estimating a high-dimensional likelihood ratio is avoided by projecting the 4D feature vector onto a 1D scalar quantity $q_i = \mathrm{f}(s_r, \bar{g}_i, \bar{c}_i, \bar{u}_i)$, which expresses a confidence on correctness of the correspondence. This is done using an SVM trained on a set of positive and

[4] Note that Moravec's statistic is a zero-mean normalized correlation, see (2). 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.

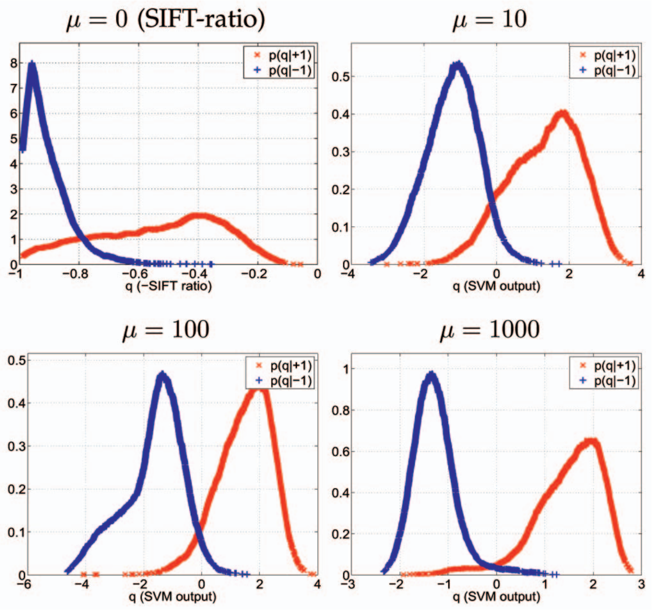


Fig. 7. Estimated class-conditional probability densities for the oriented distance to the SVM hyperplane, i.e., for projections on a normal to the maximum margin hyperplane, for correct (red) and incorrect (blue) correspondences, for four stages of the sequential decision process. Note the decrease in the overlap of the distributions with increasing μ.

negative correspondences, see Section 3. The SVM finds a discriminative direction maximizing a margin in combination with a hinge loss (the training data are not separable). Projecting on this direction is an effective feature extraction procedure, suggested already by Vapnik [25] and popularized by, e.g., Platt [26].

In consecutive decision stages i, the statistics are more discriminative, with the increase in the maximum number of growing steps μ_i, Step 1.3. Thus, a different SVM θ_i is trained for each decision stage i. The classification error due to the overlap of probability distributions is progressively decreasing, see plot shown in Fig. 10b.

The likelihoods $\mathbf{p}_i(q|+1)$ and $\mathbf{p}_i(q|-1)$ of positive and negative classes, respectively, were estimated by the Parzen window method with a moving average kernel. The likelihoods estimated from our training set are shown for four decision stages i in Fig. 7. In the first stage, there is no growth and the statistic is solely the SIFT ratio. Interestingly, the SIFT ratio threshold of 0.8 suggested for accepting a correspondence by Lowe [1] is confirmed, being close to the equal-error operating point. Note that in the sequential process, a significantly stricter test is applied in the first stage: Only correspondences having SIFT ratio smaller than about 0.4 are immediately accepted as correct, while the others are grown and decided in a later stage of a cascade, see Fig. 7 (top left).

The likelihood ratio L_i, given the SVM output q_i, is computed using linearly interpolated estimates of class-conditional probability.

2.4 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 $\{x_1,\dots,x_n\}$ on x be given. Here, measurements $x_i(=q_i)$ are scalar values, oriented distances from SVM decision boundaries after growing step i.

A sequential decision strategy is a set of decision functions $S = \{S_1,\dots,S_n\}$, where $S_i : \{x_1,\dots,x_i\} \rightarrow \{-1,+1,\sharp\}$. The strategy S makes one measurement at a time. The sign "$\sharp$" stands for "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 by 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

$$S^* = \arg\min_S \bar{T}_S, \tag{5}$$

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

for specified α and β.

Wald [27] proved that the solution of the optimization problem (5) is the *sequential probability ratio test*.

2.5 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 $x_1, x_2, \dots$. Let the joint conditional density $p(x_1,\dots,x_m|y=c)$ of the measurements $x_1,\dots,x_m$ be known for $c \in \{-1,+1\}$.

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

$$S^*_m = \begin{cases} +1, & L_m \geq A, \\ -1, & L_m \leq B, \\ \sharp, & B < L_m < A, \end{cases} \tag{6}$$

where L_m is the likelihood ratio

$$L_m = \frac{p(x_1,\dots,x_m|y=+1)}{p(x_1,\dots,x_m|y=-1)}. \tag{7}$$

The constants A and B are set according to the required error of the first kind α and error of the second kind β. Optimals 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

$$A = \frac{1-\beta}{\alpha}, \qquad B = \frac{\beta}{1-\alpha} \tag{8}$$

is close to optimal [27].

In the SCV algorithm, we assume that all information about a correspondence is contained in the statistics from the last growth step: $p(q_i|y) = p(q_1,\dots,q_i|y)$. Therefore, only 1D PDFs are needed to carry out the SPRT test. Estimation of scalar PDFs poses no technical problems, as discussed in Section 2.

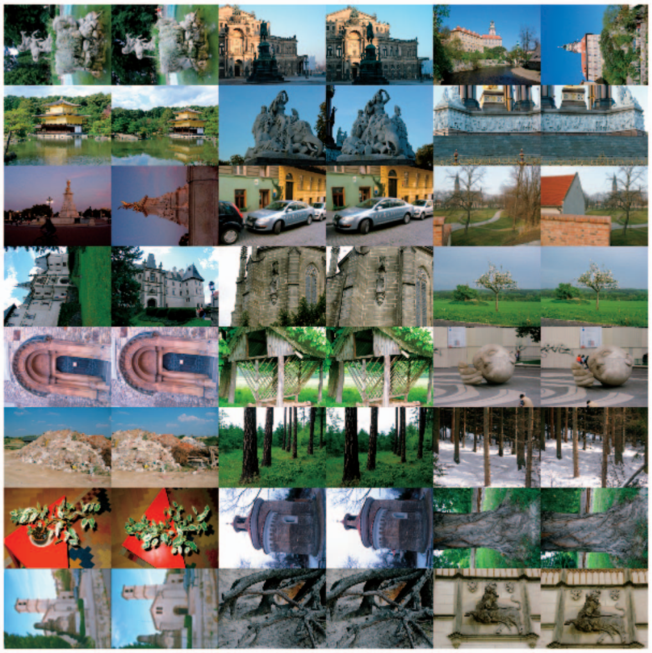


Fig. 8. The set of training images.

3 The Training Procedure

The set of 24 image pairs used in a training set of correspondences is shown in Fig. 8. For all image pairs, MSERs were detected, LAFs constructed [20], [19], and SIFT descriptors computed on normalized patches. Standard wide-baseline matching was performed using SIFTs and a set of tentative correspondences was obtained. Finally, RANSAC was run on each pair of the set to estimate the epipolar geometry. We have manually relabeled correspondences which were accidentally consistent with the epipolar geometry, but were, in fact, incorrect.[5] The remaining inlier correspondences formed the positive subset of the training set; all other correspondences were inserted as negative examples.

Approximately 6,200 positive and 9,800 negative correspondences were obtained, which means that tentative correspondences on the training set had, on average, approximately 40 percent of inliers.

The ground truth set was split randomly into two equal parts, half for training and half for testing. The learning stage included linear SVM training and probability density estimation via Parzen windowing. For SVM learning, we used a recently published efficient algorithm [28]. Regularization constant C of SVM with the hinge-loss criterion was estimated by cross-validation (which set $C = 1$).

A priori, we had no idea about the necessary size of a training set for the correspondence classification problem. We therefore carried out the following test: A progressively larger portion of the training set was used to estimate the SVMs and likelihoods. The resulting sequential classifier (SCV) was applied to the test set. The observed classification error is plotted as a function of the training set size shown in Fig. 9a. For reference, the classification error of the SIFT ratio is also plotted. The error does not improve significantly after about 300 samples, a surprisingly small number. We concluded that the size of our training set is sufficient.

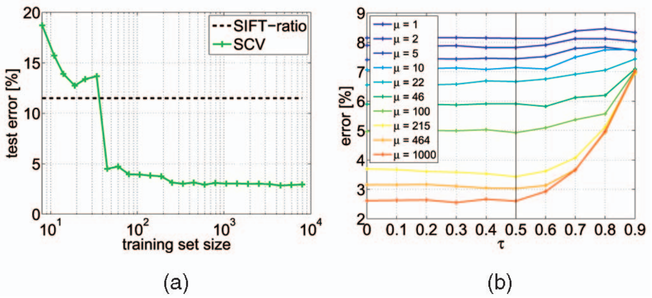


Fig. 9. (a) Test error as a function of a training set size. Note that only a very small training set with about 300 examples is required. (b) Cross-validated classification error as a function of the correlation threshold τ of the growing process plotted after several growing steps μ. The higher the value of τ, the faster the decision process. The selected value of $\tau = 0.5$ (highlighted) is the highest value with error close to minimal for all stages.

Note that the Wald SPRT is a non-Bayesian technique based on conditional probabilities, and its performance guarantees in terms of false positive and false negative rates hold for arbitrary prior probabilities. The insensitivity to the prior probability of (in)correct correspondence is an important property of the method since a wide range of inlier ratios is encountered in practical matching problems. In fact, the SCV procedure is extremely useful for matching problems where tentative correspondences have a very low inlier ratio and direct RANSAC application would require an astronomical number of samples. Such problems differ significantly in inlier percentage of tentative correspondences from our training set, but in a non-Bayesian setting, it does not matter. The training set must only be representative of the conditional probabilities of observations. Also note that all of the parameters and learned models (SVM weights and likelihoods) were kept fixed throughout all of the experiments.

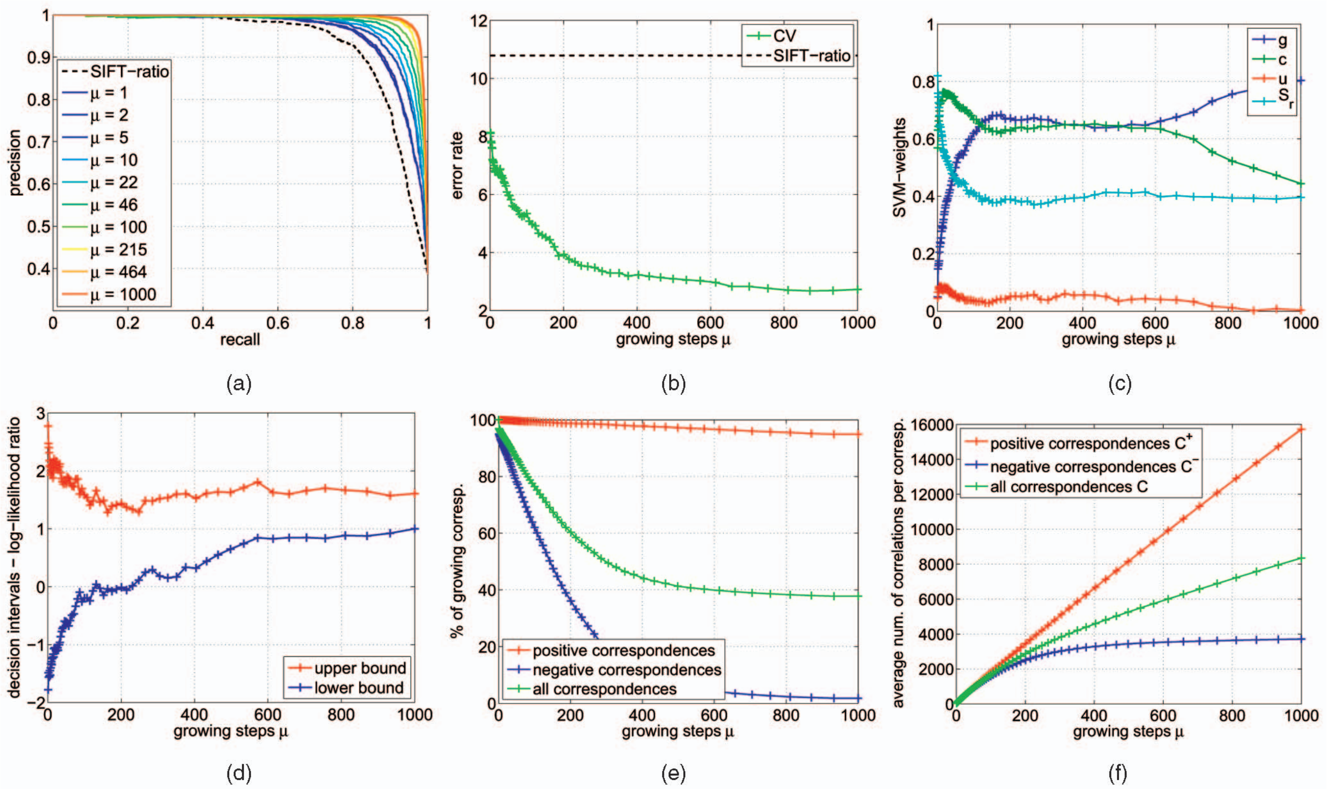
4 Experiments

4.1 Basic Properties of the Sequential Correspondence Verification Algorithm

We start the performance evaluation of the SCV algorithm by several experiments demonstrating some elementary properties of the algorithm. First, we measured discriminability of the SCV algorithm, i.e., its ability to distinguish correct and incorrect correspondences. The discriminability is characterized by a precision-recall curve, which is computed as follows: The SCV algorithm assigns likelihood ratio L to all N correspondences in the test set. The correspondences are sorted according to their likelihood ratio, $L_{(1)} \geq L_{(2)} \geq \cdots \geq L_{(N)}$. *Precision* is defined as Q_n^+/n, where Q_n^+ is the number of correct correspondences among $L_{(1)}, \ldots, L_{(n)}$. *Recall* is defined as Q_n^+/Q_N^+. The SCV algorithm is more discriminative than a standard ratio of SIFT descriptors, and the difference becomes more prominent with the number of growing steps μ, see Fig. 10a.

When a hard decision on the correctness of a correspondence is required, the likelihood ratio L is thresholded. The classification error rate for the threshold $L = 1$ is plotted in

[5] A mismatch in tentative correspondence lying on a corresponding pair of epipolar lines is not detected by RANSAC.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 10. Properties of the SCV algorithm as a function of the number of growing steps μ. (a) Precision-recall curves. (b) Classification error rate. (c) Weights of the SVM classifier. (d) Upper and lower bounds of decision intervals in log-likelihood scale. (e) Percentage of correspondences that continue growing. (f) The average number of computed correlations per correspondence.

Fig. 10b. In the case of the SIFT ratio, the threshold is 0.8, as discussed before.

Fig. 10c shows the evolution of the weights of the SVM classifier as a function of growing steps. The weights are the unit normal vector of the discriminative hyperplane, trained on the zero mean, unit variance normalized data. We show the absolute value of the weights, which can be interpreted as an importance of a particular statistic. A large value means the statistic is important, while value near zero refers about a small influence. At the beginning, the SIFT ratio s_r has a large impact, together with the average correlation $\bar{c}$. With more growing steps, the weight of the growth rate $\bar{g}$ quickly rises and becomes the most import statistic. The weight of the average uniqueness $\bar{u}$ violation is quite low and decreases with more growing steps. The weight of the average correlation $\bar{c}$ also decreases, which is probably due to an imprecise estimate of the LAFs which manifests itself by a lower correlation of correct correspondences far away from their seeds.

The next figure, Fig. 10d, focuses on decision thresholds for Wald's SPRT; the upper and lower bounds of the indecision intervals for Wald's SPRT are plotted for $\alpha = 0.05$ and $\beta = 0.001$ in log-likelihood ratio scale. The undecided interval is shrinking with increasing growing steps μ due to lower error.

Fig. 10e shows the percentage of correspondences which are still growing after maximum number of growing steps μ is performed. Note that almost all correct correspondences grow; 95 percent of correct correspondences can grow above the largest executed growth $\mu = 1{,}000$. Incorrect correspondences stop growing much earlier since there are no high correlating neighbors, and typically, the algorithm finishes by exhausting the seed queue $\mathcal{S}$ before the maximum number of growing steps is reached, see Algorithm 3, Step 3.1.

Finally, Fig. 10f shows the average number of computed window correlations per correspondence. This quantity is closely related to the computational complexity of the algorithm. For correct correspondences, the number of correlations C^+ grows almost linearly with growing steps μ, while for incorrect correspondences, the number of correlations C^- saturates at 4,000. It means that for the largest growth $\mu = 1{,}000$, negative correspondences are about four times faster to decide. This behavior is expected since the algorithm stops growing the incorrect correspondences earlier, see Fig. 10e.

4.2 The SCV Efficiently Increases Discriminability

We show that the SCV algorithm is more discriminative than the SIFT ratio and that the sequential decision-making process speeds the algorithm significantly at the expense of a very small discriminability loss. The comparison of the SCV algorithm was carried out with various settings of Wald's SPRT parameters (α, β), see Fig. 11. The SCV algorithm outperforms the SIFT ratio 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 ($\alpha = 0.001, \beta = 0.05$) allows more false positives, but they both are more efficient in terms of the number of window correlations they had to compute.

In Fig. 12, three (α, β) settings of SCV algorithm are compared with the nonsequential version (CV), which does

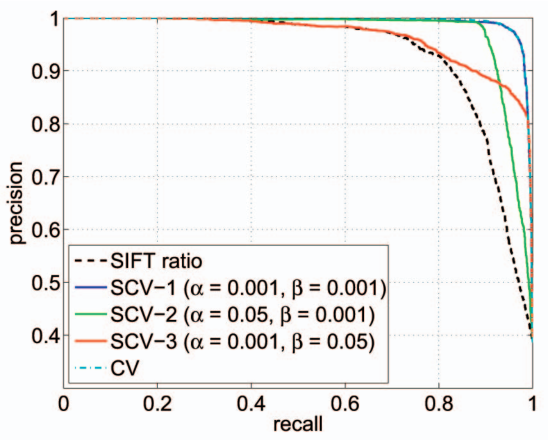


Fig. 11. Discriminability of the SCV algorithm. The precision-recall curves for SCV with various setting of false positive and false negative rates and for the SIFT ratio alone.

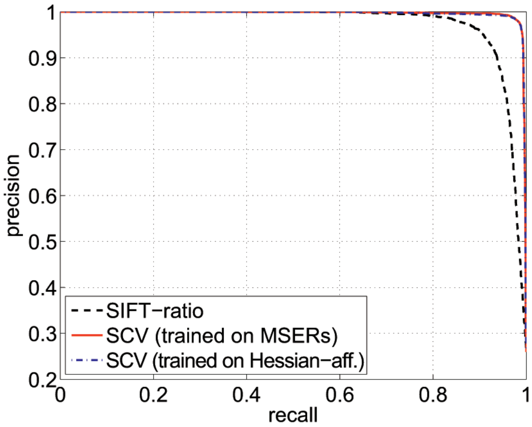
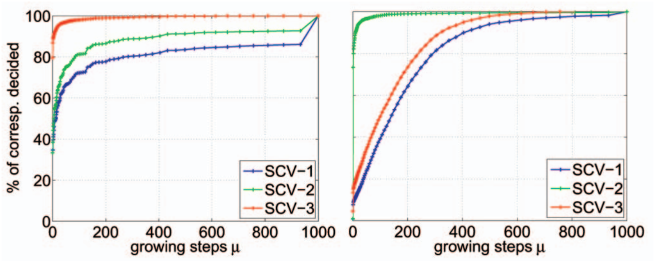


Fig. 13. Discriminability of the SCV algorithm for Hessian-affine correspondences. The SCV algorithm performs equally well when trained on MSER correspondences or specifically on correspondences of Hessian-affine points.

not decide until the last stage performing maximally $\mu = 1{,}000$ growing steps. We measured the average number of window correlations per correspondence C which had to be computed, and the percentage of correspondences decided (or stopped growing) in the ith stage of the algorithm after μ steps.

These values differ for correct and incorrect correspondences, so besides the mean values C (which depends on the percentage of correct correspondences in the test set), we show the values for correct correspondences C^+ and wrong correspondences C^- which differ as discussed in Section 4.1. The decision plots are shown for correct and incorrect correspondences as well.

The sequential decision speeds up the process by factor of more than 2 (SCV-1), more than 5 (SCV-3), or more than 9 in comparison to the nonsequential algorithm without losing much discriminability. The recall-precision curve shown in Fig. 11 of the nonsequential algorithm (CV) is almost identical to the SCV-1. Fig. 12 also shows that the SCV-2 with higher allowed false negative rate tends to decide negative correspondences in lower stages of the sequence speeding up the decision process by a factor of more than 18, while the SCV-3 is speeding up the decision process for positive correspondences by a factor of more than 100.



(a)

(b)

	$C/10^3$	$C^+/10^3$	$C^-/10^3$
CV	8.4	15.7	3.7
SCV-1	3.5	3.3	3.6
SCV-2	0.9	2.0	0.2
SCV-3	1.7	0.1	2.7

Fig. 12. Efficiency of the SCV algorithm. Plots show the percentage of correspondences decided after growing μ steps. The table shows an average number of window correlations per correspondence. (a) Correct correspondences. (b) Incorrect correspondences.

4.3 Computational Complexity

The dominant operation in the SCV algorithm is correlation computation, and other steps (SVM classification, Wald's SPRT) are negligible. The running time depends on the number of correspondences. Considering an example of 1,000 tentative correspondences, each requiring, on average, $C = 900$ correlations (see Fig. 12), we end up with approximately 10^6 correlations per image pair, which is computed on recent CPU in about 0.25 s and about 20-100 times faster on a modern GPU. It usually takes about 0.5 s on a standard C2 2.4 GHz with our implementation, also depending on the ratio of correct correspondences and on the setting of Wald's SPRT parameters.

4.4 SCV Performance on Hessian-Affine Points

Until now, all experiments have been carried out on correspondences of local affine frames on MSERs. We now show that the SCV algorithm performs equally well for verification of correspondences obtained from Hessian-affine points [21].

For all training image pairs shown in Fig. 8, a set of tentative correspondences was generated from Hessian-affine points and classified according to the ground-truth epipolar geometry, and split into a training and test set. This is the same procedure as described before for MSERs.

Fig. 13 shows that for Hessian-affine points, the SCV algorithm (SCV-1) improves the recall-precision curve obtained by SIFT ratio matching. Moreover, the performance is virtually equal for the two cases when the algorithm is trained specifically on Hessian-affine points or when the SCV algorithm trained on MSERs is used. This was not expected a priori as the image patches around the respective correspondences are quite different. But, the (simple) statistics of the growth process initialized from pixel correspondences are preserved.

Interestingly, the ratio of the SIFT descriptors has slightly better discriminability on Hessian-affine points than on

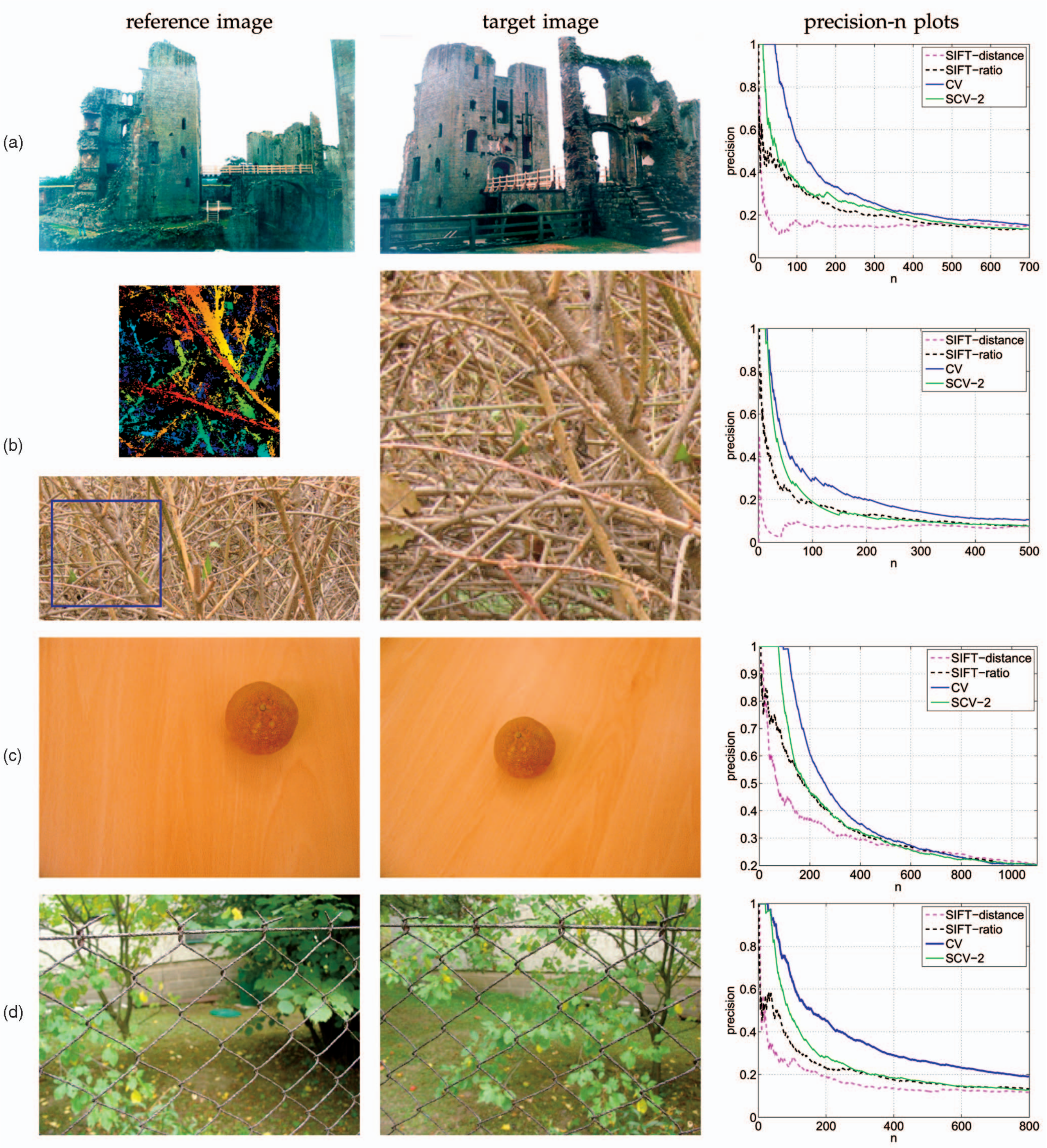


Fig. 14. Results on challenging wide-baseline scenes. For Forsythia, we show the color-coded depth map of a common part (inside the blue frame) to demonstrate the 3D structure of the image pair. Note that the Orange (c) is at a different place on the table and no correct correspondence exists on the background. Although not obvious, almost the same part of the fence appears in both images of the Fence scene. (a) Raglan. (b) Forsythia. (c) Orange. (d) Fence.

MSERs. The fact that MSERs are often detected on occlusion boundaries might play a role.

4.5 Challenging Wide-Baseline Stereo Scenes

Results of correspondence selection on difficult wide-baseline stereo scenes are shown in Fig. 14. These scenes are challenging due to small overlap, high degree of noise in the images (Raglan), and complex 3D structure with many occlusions (Forsythia, Fence). In the Orange pair, matching is difficult since the background is locally similar (same grain of wood), but not the same (different location on the same table). To find the epipolar geometry at all, the matching process generating the tentative correspondences had to be very permissive so that a sufficient number of correct correspondences was present among tentative correspondences. We allowed more than

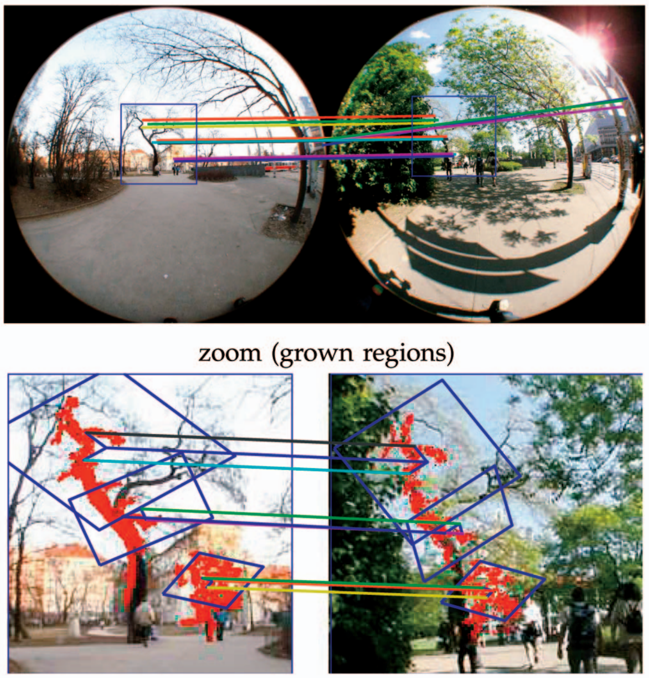


Fig. 15. Correspondences found by the SCV algorithm in challenging omnidirectional image pair with a "wide temporal baseline."

one-to-one mapping in tentative correspondences which led to a high number of outliers (about 90 percent).

Plots shown in the last column of Fig. 14 show the precision among the best n-retrieved correspondences. This is important for progressive RANSAC procedure [29], which samples tentative correspondences according to preferences defined by the matching processes (approximately speaking, in the order as sorted by the matcher of tentative correspondences). For correspondences sorted by the SCV algorithm, in all four scenes, the PROSAC procedure would terminate successfully after a single iteration since a sufficient number of top correspondences is correct. 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.

On the same images, we compared the sequential algorithm (SCV-2) and its nonsequential version (CV). For all the scenes, the results of SCV-2 are slightly worse than that of nonsequential CV, but much faster. The two algorithms evaluated the following numbers of window correlations (SCV-2 versus nonsequential CV): 0.5×10^3 versus 2.5×10^3 (Raglan), 0.6×10^3 versus 5.7×10^3 (Forsythia), 1.2×10^3 versus 6.3×10^3 (Orange), and 0.4×10^3 versus 4.3×10^3 (Fence). The reason why the decision is even faster here than on the test set in the previous experiment is the high number of wrong correspondences which are faster to decide.

4.6 Omnidirectional Images

The method was successfully tested on challenging image pairs like Fig. 15, obtained by a fish-eye camera. Besides a significant wide spatial baseline setup, the pair has a wide temporal baseline: The first image was captured in the winter when there were no leaves on trees, while the other was captured in the summer at a different time of day and in very different lighting conditions and shadows. Despite the difficult conditions, the SCV algorithm was able to find several correct correspondences, as can be checked visually; the ground truth does not exist in this case. The algorithm selected four LAF correspondences (three correct, one incorrect) out of more than 2,000 tentative correspondences. Note that the correspondences shown are selected solely by the SCV algorithm, i.e., before robust model fitting.

The method works because the initial affine transformation obtained from LAF correspondence locally approximates the nonlinear neighborhood deformation in omnidirectional images. This is true for the central part of the images, while the problems occur at the boundary of spheres, where the distortion is not negligible. This is probably the reason for the incorrect correspondence which occurred close to boundary of the sphere.

4.7 Test on the Oxford Data Set

We used a subset of the Oxford data sets[6] which has been used for performance evaluation of affine region detectors [30] and local descriptors [31]. The data sets consist of images which are distorted by various degradation: projective distortion (due to change of the camera position), image blur (from defocusing), JPEG compression artifacts, and illumination changes. The ground truth correspondences are known since the database contains a homography mapping between the reference and distorted target images.

The input is a set of several hundred tentative correspondences per each pair. The results for correspondence selection based on standard SIFT ratio and on the SCV algorithm are shown in Fig. 16 as precision-recall curves. We can see that the SCV algorithm is better in all cases. The most difficult distortion seems to be the blur, but it is destroying for SIFT as well. The projective distortion is well captured by local affine transformation, and the illumination change also does not create serious problems since Moravec's correlation used in the growing algorithm is insensitive (however, not fully invariant like the NCC statistic) to affine illumination changes. Surprisingly, the SCV does not deteriorate that much with the JPEG compression. Although the images look seriously (unnaturally) corrupted, the frequencies preserved by the JPEG compression resulted in enough correlation.

4.8 Image Retrieval

The benefits of the SCV algorithm are demonstrated on a large-scale image retrieval setup, using the data set from Nister and Stewenius benchmark [4]. It consists of 10,200 images in groups of four that show the same object. In the benchmark experiment, each image becomes a query. For each query, the top N images are returned, and a score is computed that counts how many of the correct answers are in top K images. In the benchmark, K is set to 4, giving the highest score 4 if the algorithm manages to retrieve as the top four images the four instances of the object in the data set. Since the query image is also present in the data set, the worst score of the algorithm returning only the query in the top K is 1. The overall performance of the algorithm is computed as the average score of all 10,200 queries from the data set.

[6] http://www.robots.ox.ac.uk/~vgg/research/affine/.

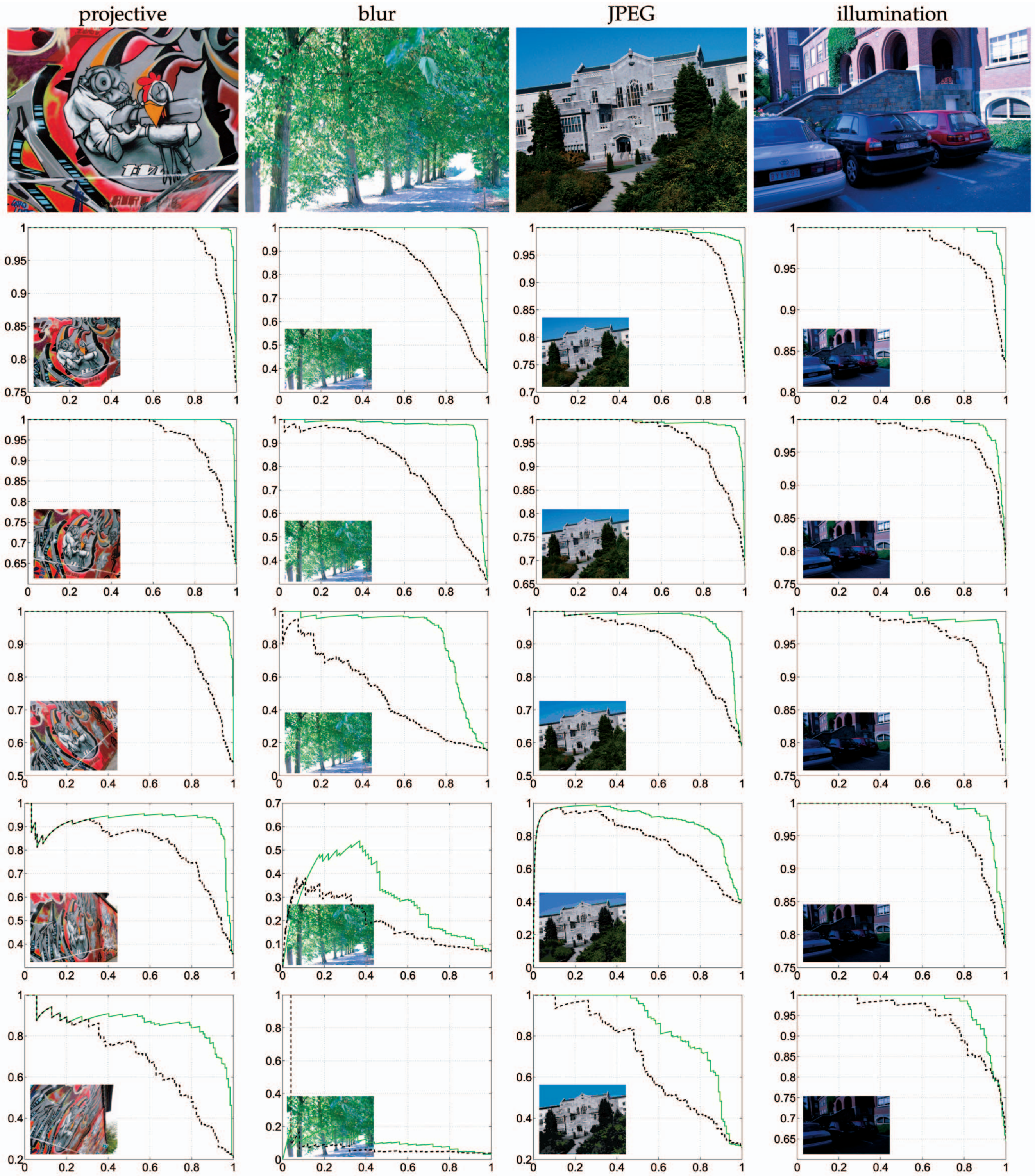


Fig. 16. Results on the Oxford data sets. Precision-recall curves of the SCV algorithm (green) and the SIFT ratio (black dashed), for images with an increasing degree of distortion. Reference images in the first row, target images are shown in graphs. See the electronic version of the article to better view of the distorted images.

We reimplemented a part of Nister's approach. MSERs [20] and LAFs [19] were computed on each of the images. Each of approximately 7.4 millions LAFs was described using SIFT [1] computed on an affine normalized patch. Then, similarly to the visual words approach proposed by Sivic and Zisserman [2], we built a visual word vocabulary consisting of 1 million k-means in the SIFT descriptor space and assigned all of the descriptors in the images to the nearest visual word. Each visual word in a given document is weighted using term frequency-inverse document frequency (TFIDF) measure from text retrieval. The similarity of two documents is then the L_1 distance between their vectors of the visual words' weights. The top K most

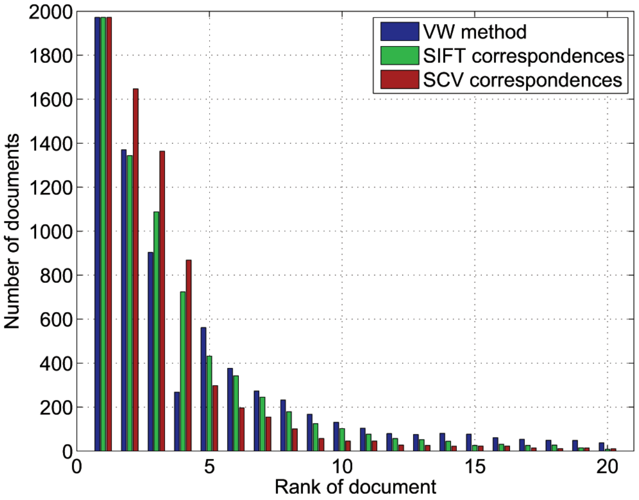


Fig. 17. Ranking of documents based on the visual word method, the number of SIFT correspondences with distance $\mathrm{ratio} < 0.8$, and the number of SCV correspondences.

similar documents are retrieved and an average score of 3.41 images retrieved per query is achieved.

To evaluate the performance of the SCV algorithm, we reranked retrieved images according to the number of SCV-validated correspondences. Since, for efficiency reasons, the reranking could include only a small number of images (e.g., 20), we only considered queries that had at least one image of the retrieved object with rank 5-20. There are 1,972 such query images. The overall score is 2.29 for these queries (note that these are difficult ones; the average on the full data sets is 3.41). The top 20 score, i.e., the average number of correct images among the top 20, is 3.51 on this subset. This is the upper bound of the performance for a retrieval algorithm that resorts the top 20 retrieved images. The correspondences were verified by the SCV $(\alpha = 0.01, \beta = 0.001)$ algorithm. Finally, new ranking was established according to the number of SCV correspondences found.

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. 17). Clearly, SCV significantly improves the ranking of the correct images bringing most of them to top 4. Its overall top 4 score on the selected query images is 5,865, resulting in average 2.97, and the average top 5 score is 3.12.

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 overall top 4 score for SIFT correspondences is 5,004, resulting in average 2.60, and the average top 5 score is 2.82.

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

TABLE 1
Comparison of the Scores of the VW Method and Rankings Based on SIFT and SCV Correspondences

score	higher	same	lower
SIFT	786	934	252
SCV	1258	631	83

5 Conclusions

We have presented the SCV algorithm, which is able to efficiently distinguish correct and incorrect correspondences, via collecting statistics while cosegmenting gradually larger regions. We have shown that this significantly benefits the matching process in challenging wide-baseline scenes and improves results in a large-scale image retrieval. The process is computationally efficient and very fast in practice, e.g., the method was successfully applied in a paper by Chum et al. [32] on large-scale image retrieval algorithm with impressive results.

The SCV software is available for download at http://cmp.felk.cvut.cz/software/SCV.

Acknowledgments

The research was supported by the Czech Academy of Sciences project 1ET101210406, by EC projects FP6-IST-027113 eTRIMS and ICT-215078 DIPLECS, and by the Czech Government research program MSM6840770038. The Valbonne images were provided by INRIA Sophia and the Raglan images by Julian Morris.

References

[1] D. Lowe, “Distinctive Image Features from Scale Invariant Keypoints,” *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[2] J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1470-1477, 2003.

[3] B. Leibe, K. Mikolajczyk, and B. Schiele, “Efficient Clustering and Matching for Object Class Recognition,” *Proc. British Machine Vision Conf.,* 2006.

[4] D. Nister and H. Stewenius, “Scalable Recognition with a Vocabulary Tree,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2161-2168, 2006.

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object Retrieval with Large Vocabularies and Fast Spatial Matching,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, “Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval,” *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[7] T. Tuytelaars and L.V. Gool, “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions,” *Proc. British Machine Vision Conf.,* pp. 412-425, 2000.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions,” *Image and Vision Computing,* vol. 22, no. 10, pp. 761-767, 2004.

[9] A. Vedaldi and S. Soatto, “Local Features, All Grown Up,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1753-1760, 2006.

[10] V. Ferrari, T. Tuytelaars, and L. van Gool, “Simultaneous Object Recognition and Segmentation from Single or Multiple Image Views,” *Int'l J. Computer Vision,* vol. 67, no. 2, pp. 159-188, 2006.

[11] G. Yang, C.V. Stewart, M. Sofka, and C.-L. Tsai, “Registration of Challenging Image Pairs: Initialization, Estimation, and Decision,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 11, pp. 1973-1989, Nov. 2007.

[12] C.V. Stewart, C.-L. Tsai, and B. Roysam, "The Dual-Bootstrap Iterative Closest Point Algorithm with Application to Retinal Image Registration," *Medical Imaging,* vol. 22, no. 11, pp. 1379-1394, 2003.

[13] M. Lhuillier and L. Quan, "Match Propagation for Image-Based Modeling and Rendering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1140-1146, Aug. 2002.

[14] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of Image Pairs by Histogram Matching—Incorporating a Global Constraint into MRFs," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[15] J. Cech, J. Matas, and M. Perdoch, "Efficient Sequential Correspondence Selection by Cosegmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[16] J. Cech and R. Sara, "Efficient Sampling of Disparity Space for Fast and Accurate Matching," *Proc. BenCOS Workshop at IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[17] G.P. Otto and T.K.W. Chau, "'Region-Growing' Algorithm for Matching of Terrain Images," *Image and Vision Computing,* vol. 7, no. 2, pp. 83-94, 1989.

[18] Z. Megyesi, G. Kos, and D. Chetverikov, "Dense 3D Reconstruction from Images by Normal Aided Matching," *Machine Graphics and Vision,* vol. 15, pp. 3-28, 2006.

[19] S. Obdrzalek and J. Matas, "Sub-Linear Indexing for Large Scale Object Recognition," *Proc. British Machine Vision Conf.,* 2005.

[20] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proc. British Machine Vision Conf.,* pp. 384-393, 2002.

[21] F. Schaffalitzky and A. Zisserman, "Multi-View Matching for Unordered Image Sets, or How Do I Organize My Holiday Snaps?" *Proc. European Conf. Computer Vision,* pp. 414-431, 2002.

[22] H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. Int'l Joint Conf. Artificial Intelligence,* p. 584, 1977.

[23] J. Kannala, E. Rahtu, S.S. Brandt, and J. Heikkila, "Object Recognition and Segmentation by Non-Rigid Quasi-Dense Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[24] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions Using Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 508-515, 2001.

[25] V.N. Vapnik, *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[26] J.C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers,* pp. 61-74, MIT Press, 1999.

[27] A. Wald, *Sequential Analysis.* Dover, 1947.

[28] V. Franc and S. Sonnenburg, "OCAS Optimized Cutting Plane Algorithm for Support Vector Machines," *Proc. Int'l Machine Learning Conf.,* 2008.

[29] O. Chum and J. Matas, "Matching with PROSAC—Progressive Sample Consensus," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 220-226, 2005.

[30] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision,* vol. 65, nos. 1/2, pp. 43-72, 2005.

[31] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[32] O. Chum, M. Perdoch, and J. Matas, "Geometric Min-Hashing: Finding a (Thick) Needle in a Haystack," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

Jan Čech received the MSc degree (with honors) in cybernetics from the Faculty of Electrical Engineering, and the PhD degree in artificial intelligence and biocybernetics from the Czech Technical University, Prague, in 2002 and 2009, respectively. His research interests include stereoscopic vision, 3D reconstruction, image understanding, and interpretation. He serves as a reviewer for several conferences in the field. He is a member of the IEEE.

Jiří Matas received the MSc degree (with honors) in cybernetics from the Czech Technical University, Prague, in 1987, and the PhD degree from the University of Surrey, Guildford, United Kingdom, in 1995. From 1991 to 1997, he was a research fellow in the Center for Vision, Speech, and Signal Processing at the University of Surrey. In 1997, he joined the Center for Machine Perception, Czech Technical University, where he has been the holder of various positions since 1997. He has published more than 100 papers in refereed journals and conference proceedings. His publications have more than 800 citations in the Science Citation Index. He was a recipient of the Best Paper Prize at the British Machine Vision Conferences in 2002 and 2005 and at the Asian Conference on Computer Vision in 2007. He has served in various roles at major international conferences (e.g., the IEEE International Conference on Computer Vision, the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Pattern Recognition, and the Neural Information Processing Symposium), cochairing the European Conference on Computer Vision 2004 and CVPR 2007. He is an associate editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and a member of the editorial board of the *International Journal of Computer Vision*. He is a member of the IEEE and the IEEE Computer Society.

Michal Perdoch received the BSc degree in software engineering from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovak Republic, in 2001, and the MSc degree in computer science in 2004 from the Faculty of Electrical Engineering, Czech Technical University, Prague, where he is currently working toward the PhD degree at the Center for Machine Perception. His current research interests include low-level image processing, feature detection and description, object recognition, and large-scale object retrieval. He is a member of the IEEE.