



Published in Image Processing On Line on 2017–12–23.  
 Submitted on 2015–10–26, accepted on 2017–12–17.  
 ISSN 2105–1232 © 2017 IPOL & the authors CC–BY–NC–SA  
 This article is available online with supplementary materials,  
 software, datasets and online demo at  
<https://doi.org/10.5201/ipol.2017.154>

# Local Region Expansion: a Method for Analyzing and Refining Image Matches

Erez Farhan, Elad Meir, Rami Hagege

EE, BGU, Israel

{farhan, meirela}@post.bgu.ac.il, hagege@ee.bgu.ac.il



This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:  
 Erez Farhan and Rami Hagege. "Geometric Expansion for Local Feature Analysis and Matching" *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2771-2813, 2015. <http://epubs.siam.org/doi/abs/10.1137/140997671>

## Abstract

We present a novel method for locating large amounts of local matches between images, with highly accurate localization. Point matching is one of the most fundamental tasks in computer vision, extensively used in applications such as object detection, object tracking and structure from motion. The major challenge in point matching is to preserve large numbers of accurate matches between corresponding scene locations under different geometric and radiometric conditions, while keeping the number of false positives low. Recent publications have shown that applying an affine transformation model on local regions is a particularly suitable approach for point matching. Yet, affine invariant methods are not used extensively for two reasons: first, because these methods are computationally demanding; and second because the derived affine estimations have limited accuracy. In this work, we propose a novel method of region expansion that enhances region matches detected by any state-of-the-art method. The method is based on accurate estimation of affine transformations, which are used to predict matching locations beyond initially detected matches. We use the improved estimations of affine transformations to locally verify tentative matches in an efficient way. We systematically reject false matches, while improving the localization of correct matches that are usually rejected by state-of-the-art methods.

## Source Code

The source code and documentation are available from [the web page of this article](#)<sup>1</sup>. The code is mainly a Matlab code that requires some Matlab toolboxes detailed in the ReadMe.txt file attached to the source code. Specific instructions on how to run the code, including some other dependencies, are also found in the ReadMe.txt file.

**Editorial Warning:** This paper appears as a useful supplementary material and online illustration of the SIIMS paper [5]. The referees have not confirmed that the published text is an accurate or complete account of the published code.

**Keywords:** local matching; affine transformation; outlier rejection; registration

<sup>1</sup><https://doi.org/10.5201/ipol.2017.154>

# 1 Introduction

Point matching, the process of finding corresponding scene points in two images, constitutes a fundamental challenge in many computer vision problems and applications. The problem inherent in point matching is that the image of a patch around a particular point of a scene can vary dramatically as a result of variations in the pose of the camera, the lighting conditions, and the camera specifications. A key factor leading to variability in the appearance of a patch is its geometric transformation across different views. For smooth surfaces, this variability can be approximated locally by an affine transformation. Thus, assuming we have small patches around corresponding points that lie on a smooth surface of the scene, the geometric transformation between them can be well approximated by an affine transformation. This approach has been explored in many previous works for finding point correspondences [12, 11], but most of the methods used today for point matching do not exploit affine transformations. As shown, for example in [9, 2], for many applications it is sufficient to consider the effect of two dimensional translation, scale (zoom) and rotation around the optical axis (which are a subset of the *affine* model). Methods neglecting affine transformations (and other sources of variability) will usually produce many false matches, which then have to be filtered out in later stages of the algorithms [7] and will also result in the loss of many other potential matches. In [11], a method to adaptively normalize the affine shape of every point match is suggested. In [10] a method to normalize the affine transformation of detected regions is suggested. In [12], a method to simulate the space of affine transformations is suggested. These methods offer a considerable improvement in terms of accuracy in position and performance vis-à-vis methods that neglect these geometric variations [9, 2]. The fact that affine-based methods are not frequently used is either due to *numerical reasons*, since corresponding regions do not contain enough information to accurately estimate the affine transformations (like in [11, 10]); or for *computational reasons*, when trying to simulate such transformations [12]. A possible solution to the numerical challenge is simply extracting larger features from each image. But, while allowing better affine transformation estimation in some cases, extracting larger features might prohibit the affine model (which is usually a local approximation) in many other cases. In [12], it is shown how simulating a finite set of affine transformations on a given image, outperforms the affine normalization (e.g. MSER) based methods in many different examples, especially when the affine transformation includes a hard tilt. In practice, this approach offers a promising way of by-passing the numerical challenges of affine estimation from small patches. In this work, we suggest an algorithm to estimate affine transformations and create a matching scheme which takes affine transformation fully into account. The algorithm can be used on top of any state-of-the-art method (e.g. SIFT, MSER), with little extra computational demand.

## 1.1 Point Matching

The process of finding point matches usually comprises four stages:

1. Feature Extraction. Finding potential candidate points (a.k.a *features*), for example corners [13, 8], blobs [9], or level-line based regions [10].
2. Feature Description. Attaching to each potential candidate a descriptor [9, 2, 1, 3] that considers a small image patch around it (the descriptor must be invariant to any variations that the patch might undergo).
3. Descriptor Matching. Applying some metric to find potential matches, assuming that points that correspond to the same physical patch in the scene should provide similar descriptors, and those that correspond to different points should provide different descriptors.

4. False Match Rejection (Optional). This is usually achieved assuming and verifying global transformations using methods such as RANSAC [7].

Since false matches are a seemingly unavoidable artifact of common point matching solutions, stage 4 is essential. Specifically, tuning point matching solutions for finding more matches will usually result in finding more false matches. While these false matches may prove harmful for many tasks, systematically eliminating them has still remained a challenge. The most common approach for eliminating false matches [7, 4], is based on an assumption that detected matches agree on some global geometric model (e.g. co-planar). This enables the detection of a sub-set of the matches that strongly agree on a consensus model. Matches that disagree on the consensus model are considered outliers and thus rejected. This paradigm encapsulates two key caveats:

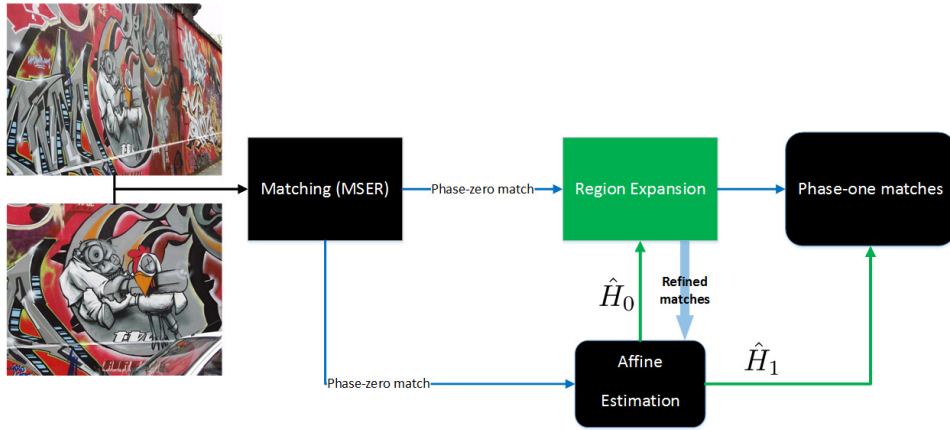
1. Efficiency. Most methods address stage 4 only after completing stages 1-3. In practice, a considerable amount of matches detected in stages 1-3 are discarded in stage 4. The need to discard many candidate matches raises both functional and computational issues. Specifically, there are two main concerns: 1) Increasing the rate of false matches (*outliers*) could significantly increase the number of iterations required to detect a sub-set of correct matches; 2) Significant computational resources are dedicated to detecting and describing matches that are eventually discarded.
2. Global Model Existence. False match elimination also relies on the existence of a global geometric model. While this is easily the case for rigid planar surfaces (modeled by perspective transformations) or any other rigid smooth surfaces, global geometric models are much less applicable in the general case where the scene is only piecewise-smooth, or perhaps composed of separate dynamic parts (imagine two different cars on a road).

In this work, we implement stage 4 using a local analysis and integrate it along with stages 1-3 in order to avoid falsely detected matches in early stages, such that most detected matches do not demand any further iterative global analysis. We show that these local considerations can be applied on any locally smooth surface of the scene with no regard to global geometric assumptions.

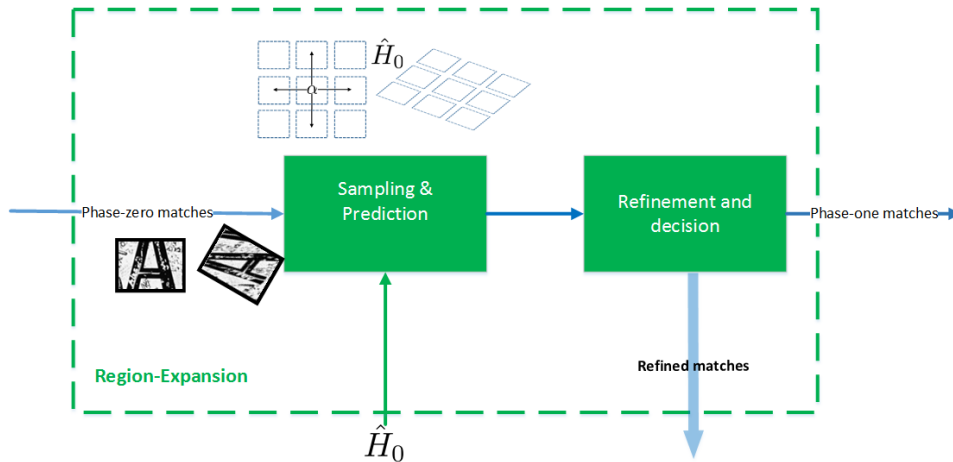
## 2 Region Expansion

In this section, we present a method that both dramatically improves the results of local affine transformation estimations and locates a massive number of accurate point matches between images. The method is also very useful in detecting false or inaccurate matches for rejection. We start by giving an overview of the method, and then give the details in the following sub-sections. As we illustrate in Figure 1a, the inputs to the region expansion mechanism are region matches between two images (*source* and *target*) produced by any standard region matching algorithm (specifically, we chose MSER for this explanation). Each of these matches, which we designate *phase-zero point matches*, also contains the information about the feature shape in each image (in MSER, this is given by the affine normalization method), which we utilize to extract the local affine transformations between the corresponding matched features. The basic idea is to enhance each *phase-zero* match separately to produce a set of accurate output matches from each input match. Specifically, we have three goals:

1. Locate many new accurate local matches.
2. Substantially improve the local transformation estimation.
3. Eliminate false *phase-zero* matches.



(a) The overall proposed system.



(b) The flow of region expansion.

Figure 1: High-level and low-level flows: (Blue - Location info; Green - Transformation info.)

As we roughly illustrate in Figure 1b, the core idea behind region expansion is to treat each *phase-zero* match as a region match and try to expand the area of the matched region, while taking into account the knowledge of the local affine transformation (designated  $\hat{H}_0$ ). This will help us to achieve goals 1 and 2. Expanding the area (Algorithm 2) is done by first predicting the location of sample points around the initial region using the knowledge of the local transform, and then refining this prediction by scanning the image around the predicted locations (sub-Section 2.1), and thus locating new point matches. This scanning procedure, which also introduces contextual data of the initial region match, is used for rejecting false or inaccurate *phase-zero* matches and thus for achieving goal 3 (sub-Section 2.3). Finally, we use the new matches to re-estimate the transformation of the expanded region with significantly higher accuracy than the initial estimation (sub-Section 2.2), this re-estimation enables us to perform further expansion iterations, as long as the local affine model is preserved.

## 2.1 Expansion from Correct Matches

This section describes the initial analysis done by the proposed expansion algorithm. The motivation here is first to verify the given tentative region match, and then to improve every positively verified match and achieve higher accuracy and spatial coverage. The algorithm is formally described in Algorithm 2, and explained in this section.

Given a region in *the source* image, we sample the area around it with  $N$  evenly spaced points in

**Algorithm 1** The Expansion Algorithm (overview)

---

**Inputs:**  
 $I_{src}, I_{tgt}$  - Two gray-scale images (accessible to all methods)  
 $\{\pi_i^{(sin)}, \pi_i^{(tin)}\}_{i=1,\dots,K}$  - A set of 2D correspondences between  $I_{src}$  and  $I_{tgt}$  (“*phase-zero*” matches)  
 $\{Hs_i\}_{i=1,\dots,K}$  - The  $3 \times 3$  affine normalizing matrices describing the shape of the feature  $\pi_i^{(sin)}$ .  
 $\{Ht_i\}_{i=1,\dots,K}$  - The  $3 \times 3$  affine normalizing matrices describing the shape of the feature  $\pi_i^{(tin)}$ .

**Outputs:**  
 $\{\pi_i^{(sout)}, \pi_i^{(tout)}\}_{i=1,\dots,M}$  - Refined correspondences between  $I_{src}$  and  $I_{tgt}$  (“*phase-one*” matches)

- 1:  $\pi^{(sout)} = \emptyset, \pi^{(tout)} = \emptyset$
- 2: **for**  $1 \leq i \leq K$  **do**
- 3:   PA = extractPrincipalAxes( $Hs_i$ )
- 4:    $\hat{H}_0 = (Ht_i)^{-1} \cdot Hs_i$  ▷ Build affine transformation matrix
- 5:   Default number of samples  $N = 9$ . Default expansion factor  $\alpha = 1.5$ . Default further expansions FI=2.  
   (wasExpanded,  $[\pi_s, \pi_t]$ ) = regionExpansion( $\pi_i^{(sin)}$ , PA,  $\hat{H}_0$ ,  $N$ ,  $\alpha$ , FI) ▷ Algorithm 2
- 6:   **if** wasExpanded **then**
- 7:      $\pi^{(sout)} = \pi^{(sout)} \cup \pi_s$
- 8:      $\pi^{(tout)} = \pi^{(tout)} \cup \pi_t$
- 9:   **end if**
- 10: **end for**

---



(a) Expanding beyond a *phase-zero* point to the new samples  $\pi_{i=1..9}^{(s)}$  (b) Finding  $\pi_{i=1..9}^{(t)}$  as the centers of the green parallelograms. Red parallelograms are the prior estimations  $\hat{H}_0(\pi_{i=1..9}^{(s)})$ .

Figure 2: Example of region expansion and correction of point matches.  $N = 9$  *phase-one* matches are produced.

an area  $\alpha$  times larger than the given region (see illustration in Figure 2a for a rectangular shaped sampling, where  $N$  was set to 9, and  $\alpha$  was set to  $\sim 3$ ). In the implementation attached to this manuscript, we scatter the samples in an elliptic shape). Our aim is to match each sample point  $\pi_i^{(s)}$  in the source image, to a corresponding point  $\pi_i^{(t)}$  in the target image to produce what we designate a *phase-one* point match. Using the given estimated transformation of the region between the two images (designated  $\hat{H}_0$ ), we obtain an initial prediction of the location and pose of each of the corresponding points in the *target* image (Figure 2b - red parallelograms). For a given point  $\pi_i^{(s)}$ , this initial prediction allows us to scan around a predicted location in a constrained search window with the aim to find a better corresponding point  $\hat{\pi}_i^{(t)}$ . This search can be performed in many different ways, but since we have the estimated transformation of the search window, we can use a Normalized Cross Correlation (NCC) scanning algorithm. The template for the NCC procedure is deduced from a small patch around  $\pi_i^{(s)}$ . We use this patch as a template in order to find the maximal response

---

**Algorithm 2** The top-level region expansion procedure given a single *phase-zero* match

---

**procedure** regionExpansion( $X_0$ , PA,  $\hat{H}_0$ ,  $N$ ,  $\alpha$ , FI)

Inputs:

$X_0$  - A 2D vector denoting the location (in the source image) of the phase-zero point

PA - A 2D matrix built from the principal axes of the phase-zero region as columns

$\hat{H}_0$  - The initially estimated affine transformation induced by the given match

$N$  - A scalar parameter indicating how many new points to sample in each expansion

$\alpha$  - A scalar parameter indicating how much to expand (default value: 1.5)

FI - A scalar parameter indicating the number of further expansions

Outputs:

wasExpanded - Indicating whether the phase-zero match was successfully expanded

$\pi^{(s_{out})}, \pi^{(t_{out})}$  - Two corresponding lists of 2D points (phase-one matches)

- 1:  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N} = \text{expansionLoop}(X_0, \text{PA}, \alpha, \hat{H}_0, N)$
  - 2: Robustly re-estimate the affine transform  $\hat{H}$  from the pairs  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N}$   $\triangleright$  sub-Section 2.2
  - 3: **if** robust estimation fails **then**
  - 4: return (False, NULL)  $\triangleright$  Match is rejected in lack of affine consistency
  - 5: **end if**
  - 6:  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..K \leq N} \leftarrow$  the inlier subset of  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N}$
  - 7: **Repeat 7-11 FI times:**
  - 8:  $X_0 \leftarrow$  the centroid of the set  $\pi_i^{(s)}_{i=1,2..K}$
  - 9: PA  $\leftarrow$  the principle axes of the set  $\pi_i^{(s)}_{i=1,2..K}$
  - 10:  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N} = \text{expansionLoop}(X_0, \text{PA}, \alpha, \hat{H}, N)$
  - 11: Robustly estimate the affine transform  $\hat{H}$  from the pairs  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N}$
  - 12:  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..K \leq N} \leftarrow$  the inlier subset of  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N}$
  - 13: return (True,  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..K}$ )
- end procedure**
- 

**procedure** expansionLoop( $X_0$ , PA,  $\alpha$ ,  $\hat{H}$ ,  $N$ )

- 1:  $\Pi^{(s)} \leftarrow N$  evenly-spaced points around  $X_0$ , in an ellipse defined by  $\alpha \cdot \text{PA}$
  - 2: **for all**  $\pi_i^{(s)} \in \Pi^{(s)}$  **do**
  - 3:  $\pi_i^{(t0)} \leftarrow \hat{H} \cdot \pi_i^{(s)}$
  - 4: Crop a patch  $P$  around  $\pi_i^{(s)}$  in the source image, to use as a template
  - 5: Use  $P$  to scan around the corresponding  $\pi_i^{(t0)}$  in the target image
  - 6: Obtain  $\pi_i^{(t)}$  as the patch maximizing the NCC with  $P$
  - 7: **end for**
  - 8: return  $[\pi_i^{(s)}, \pi_i^{(t)}]_{i=1,2..N}$
- end procedure**
-

within the search window in the target image. We do this for all  $N$  sampled points and obtain  $N$  corresponding points in the target image (Figure 2b - green parallelograms). These points are considerably more accurate than the initial predictions, due to the contribution of the scanning algorithm. It should be noted that as long as the  $N$  templates are disjoint, *phase-one* matches are obtained independently of each other (given the initial affine transformation), while most of them (those that fall outside the *phase-zero* region) incorporate new information from the scene with respect to the *phase-zero* region. Thus, they are necessarily not redundant with respect to each other and with respect to the *phase-zero* match, while being considerably more accurate than the *phase-zero* match, since they maximized the NCC.

## 2.2 Re-Estimation of the Affine Transform

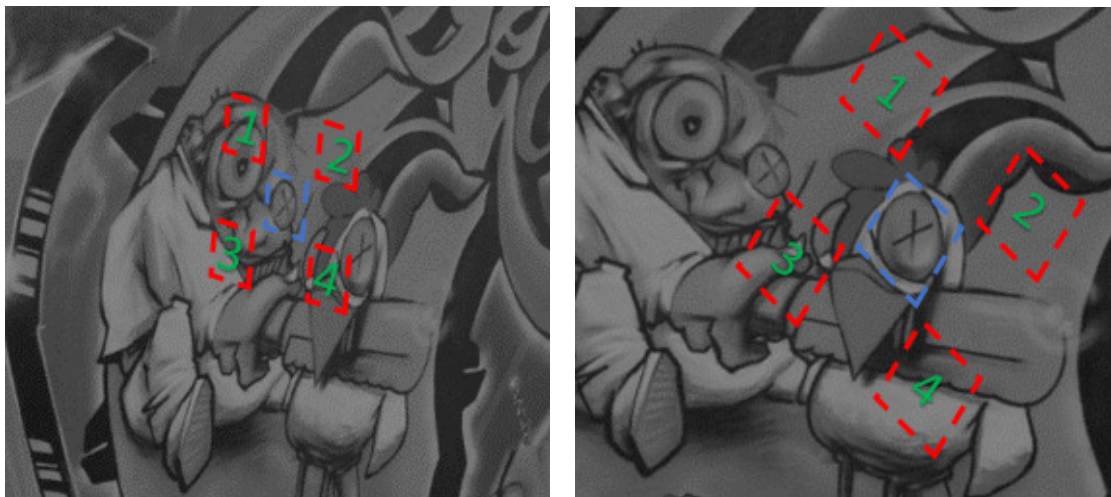
Following the expansion process described in sub-Section 2.1, we examine the contribution of the new *phase-one* matches to the estimation of the affine transformation. Since each  $2D$  location  $\pi_i^{(t)}$  in the target image is considerably more accurate than the initial prediction  $\pi_i^{(t_0)}$ , these matches can be viewed as a series of  $2D$  corrections to the initially estimated affine transform of the region, and can thus be fed back in order to re-estimate the affine transform of the expanded region. We note that while the proposed match scanning algorithm cannot optimize all the parameters of the affine transform in every scanned point separately, its ability to refine the location of each point relative to the prediction allows the re-estimation of the affine transformation from all located correspondences together (we need at least three, since there are six degrees of freedom). Relative to previous work [6, 15], this replaces a single non-linear six-dimensional optimization used to optimize the affine transformation estimation between two regions, with a series of  $N$  two-dimensional optimization problems that can be solved using a simple linear procedure such as NCC.

Given the corrected locations  $\hat{\pi}_{i=1..N}^{(t)}$ , the affine transformation between  $\hat{\pi}_{i=1..N}^{(s)}$  and  $\hat{\pi}_{i=1..N}^{(t)}$  is re-estimated using the Linear-Least-Squares estimator. This new estimation ( $\hat{H}_1$ ) is more accurate than the initial estimation ( $\hat{H}_0$ ). When the expansion factor  $\alpha$  is chosen properly, this improvement is proportional to  $\alpha$  [5]. This improved estimation can be used to extend the region even further in the same manner, as long as the affine model is preserved (see Algorithm 2).

We note the importance of an accurate affine transformation in predicting the locations of the corresponding points in the target image. The prediction error of a point increases linearly with the distance of the point from the original region. Thus, a prediction error of  $\pm 3$  pixels for predicting a location of a 4-pixel-distant point, becomes a prediction error of  $\pm 30$  pixels for a 40-pixels-distant point. Intuitively, a large prediction error reduces the chance of finding accurate point matches. Therefore, region expansion is limited by the accuracy of the affine estimation. Indeed, a very accurate estimation would allow larger expansions (as long as the affine model is preserved). Estimation correction is especially helpful in cases where the initial number of *phase-zero* matches provided is low due to a significant change of pose. An accurate affine estimation then serves for additional and further expansions in a similar manner to sub-Section 2.1, thus yielding many more new point matches and in turn re-improving the affine estimation.

## 2.3 Rejection of False Matches using Local Context

The expansion mechanism can also be used for rejecting false *phase-zero* matches. Given a false *phase-zero* match, it is very unlikely that the surroundings of the corresponding regions will be related by an arbitrary affine transformation. This fact is used to create a mechanism for rejecting false matches. When sampling around a *phase-zero* region in the source image with the aim to expand it, we expect the affine transformed samples to roughly correspond to points within the search window



(a) Sampling around the initial point (blue) to match new points (red). (b) Failure to match new points: The location of the new points is predicted (see numbers for correspondence), but the underlying image is very different from that expected.

Figure 3: A false match through the eyes of the expansion mechanism: The expansion process fails and the *zero phase match* (blue parallelograms) is declared false.

in the target image. When the *phase-zero* match is, in fact, a false match, these correspondences (*phase-one matches*) should be obtained with low probability (designated  $p_{fa} \ll 1$ ).

In order to approve a tentative *phase-zero* match in the process of expansion, we demand *at least four* approved *phase-one* matches among the samples (at least three for affine estimation + one for verification). In order to classify the matches (approve/disapprove), simple thresholding on the maximal NCC can be used. For better performance, we use the delta-criterion approach described in [5]. For  $N$  samples around a false *phase-zero* match, we assume that given a false match, the approval of different points is independent. Thus, the *joint* probability of at least four of them being approved in a typical case of  $N = 16$  and  $p_{fa} = 0.03$ , is  $P_{fa}^{joint} \simeq 0.013$ , which is given by a binomial distribution. In addition, for any set of approved *phase-one* matches around a tentative *phase-zero* match, we check the affine consistency (Algorithm 2), as we assume that matches are spatially close enough to be related by a single affine transformation. A set that is not affine *consistent* within itself is rejected in the expansion procedure. This procedure helps us reduce the rate of false matches even further. In our tests on a standard data-set, we observed that the probability of our expansion mechanism to approve a false *phase-zero* match is  $P_{fa}^{consistent} \simeq 0.003$ . In Figure 3, we present an illustration of a rejection scenario. The similarity of the *phase-zero* matches (patches within the blue parallelograms) is high and leads to a false match. Expanding beyond this initial match reveals a dissimilarity in the surroundings (corresponding patches in the red parallelograms are dissimilar) and the match is rejected.

We note that this mechanism might sometimes reject well-localized *phase-zero* matches with very inaccurate transformation estimation, because the search windows for locating *phase-one* matches in the target image will be very inaccurately localized. From a statistical standpoint, these cases are very rare compared to the amount of false *phase-zero* matches.

## 2.4 A Note on Algorithm Parameters

In the description of the expansion algorithm above, we have encountered three main parameters that influence the performance of each individual expansion procedure:



1. The expansion factor  $\alpha$ . This choice introduces delicate considerations which are beyond the scope of this manuscript and are available in the journal paper version of this work [5]. As a simple intuition, we note that while choosing a small  $\alpha$  ( $\sim 1$ ) restricts the positive effects of the expansion, choosing a large  $\alpha$  ( $\gg 1$ ) increases the error of predicting new samples, and the risk of exceeding the locality of the affine region.
2. The number of further expansions FI. This choice introduces a simple trade-off between *computational load* and *performance*, where performing more expansion iterations linearly increases computations, but also enables the location of more matches and better scene coverage (up to the affine region limitation)
3. The number of samples  $N$ . This choice again introduces a trade-off between *computational load* and *performance*, where increasing  $N$  means performing more match scanning operations, but also produces more new matches and linearly improves the precision of the affine estimations [5]. In practice, it is reasonable to relate the  $N$  to the actual area of the expanded region in order to provide a dense enough cover from one hand, and avoid redundancy on the other. Thus, we denote the more meaningful parameter here  $\rho \triangleq \frac{N}{A}$  where  $A$  is the area of the expanded region in pixels.

In the next section, we also include an empirical examination of the effects of  $\alpha$ , FI and  $\rho$  on the algorithm's result. For a more complete understanding of these parameters and their theoretical significance, we address the reader to [5].

### 3 Empirical Evaluation

We have tested the proposed algorithm while varying two crucial factors: 1) Using different state-of-the-art methods as “phase-zero” initialization of our algorithm; 2) By allowing the system to perform variable expansion steps in order to produce more point matches. For reference (a before/after test), we have compared the results to each phase-zero mechanism accordingly used. For examining the performance of point matching algorithms, there are two crucial criteria: the amount of correct matches (inliers) and their precision. As phase-zero initializers we have used *MSER* [10], for which we used code from [14], combined with a SIFT descriptor; and the *Harris-Affine* detector [11], using code from [14] (which outperformed the binaries supplied by the authors). We varied the amount of allowed expansions between 1 and 7. We have tested all methods and variants on 12 images [“Graffiti” + “Wall” sets] coming from the *Affine Covariant Features* website<sup>2</sup>, that focus on performance under various view angles. The ground truth homography was also taken from the same website and was used to measure the match error of every detected point. The match error is computed as the euclidean distance between the projection of the source point using the ground truth homography, and the location of the corresponding target point as detected by each of the algorithms. We treat all matches with pixel error smaller than 5, as “inliers”. We examine the performance across the different parameters of the algorithm and compare it to the performance of the respective phase-zero methods:

1. Varying FI. In Figure 4 we show the performance when allowing varying amounts of expansion iterations (the histogram count is the average under all image pairs in the data-set). In Figure 4a, we show the performance when using MSER as the phase-zero matching algorithm, while in Figure 4b we see the corresponding results when using Harris-Affine as the phase-zero algorithm. X1, X3, X5, X7 denote the allowance of 1, 3, 5 or 7 expansion steps of the proposed

<sup>2</sup><http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>, Accessed: 2014-01-29

algorithm (labeled REX). In these experiments  $\alpha$  was set to 1.5 in the first expansion step, and to 2 in all further expansion steps, while  $\rho$  was set to 0.0625. We observe how under both phase-zero initializations, the proposed algorithm was able to significantly improve the rate of inliers, while dramatically increasing the amount of matches. The shape of the histogram is related to the precision of the matches. We observe that while the REX match errors are concentrated around 0.5 pixels, the shape of the histogram is preserved under more expansion steps, indicating that the precision is preserved while the amount of correct matches is significantly increased. The slight drop of the inlier-rate when performing more expansion steps can be ascribed to a slight breakage of the affine transformation under the perspective transform introduced by some image pairs in the data-set.

2. Varying  $\alpha$ . In Figure 5 we show the performance when varying the expansion factor. In these experiments, the sample density  $\rho$  was set to 0.0625, while only one expansion step was allowed (FI=1). We note that as  $\rho$  remains constant, the amount of samples  $N$  per expanded region, actually increases with  $\alpha$ . Thus, we expectedly see an increase in the amount of matches when increasing  $\alpha$  up to a certain point. The small drop in performance between  $\alpha = 2.5$  and  $\alpha = 3$  can be ascribed either to the inaccuracy of the initial affine estimations, or to the breakage of the affine model under the perspective transform introduced by some image pairs in the data-set.
3. Varying  $\rho$ . In Figure 6 we show the performance when varying the sample density when locating new matches. In these experiments, the expansion factor  $\alpha$  was set to 1.5, while only one expansion step was allowed (FI=1). As expected, the amount of matches increases with sample density. We note that for many cases of small regions, setting  $\rho = 0.03125$  translates to taking  $N < 4$  samples, which is below the limit of the algorithm. In those cases we had to add samples to complete at least 4 samples. Thus the difference between  $\rho = 0.03125$  and  $\rho = 0.0625$  is well reflected in Figure 6. We conclude that choosing small values such as  $\rho = 0.03125$  for cases of small regions, is actually not recommended.

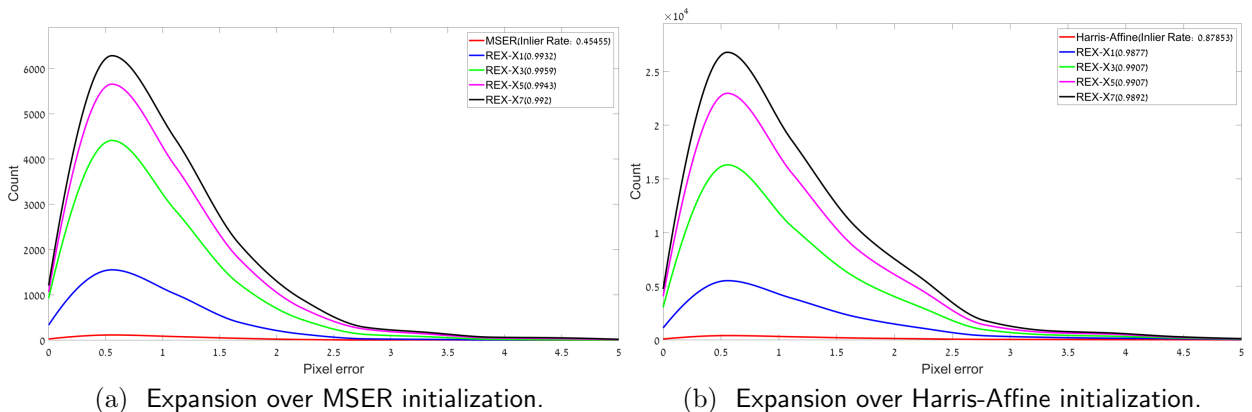


Figure 4: Histograms of point matching errors (only inliers are counted) with varying amounts of allowed expansion steps.  $\alpha$  is set to 1.5 in the initial expansion, and fixed to 2 in all further expansion steps, while  $\rho$  is fixed at 0.0625. The total amount of matches increases with the amount of allowed expansion steps, while the inlier-rate remains almost constant. For better visualization, the graphs are drawn as continuous plots, but represent discrete information.

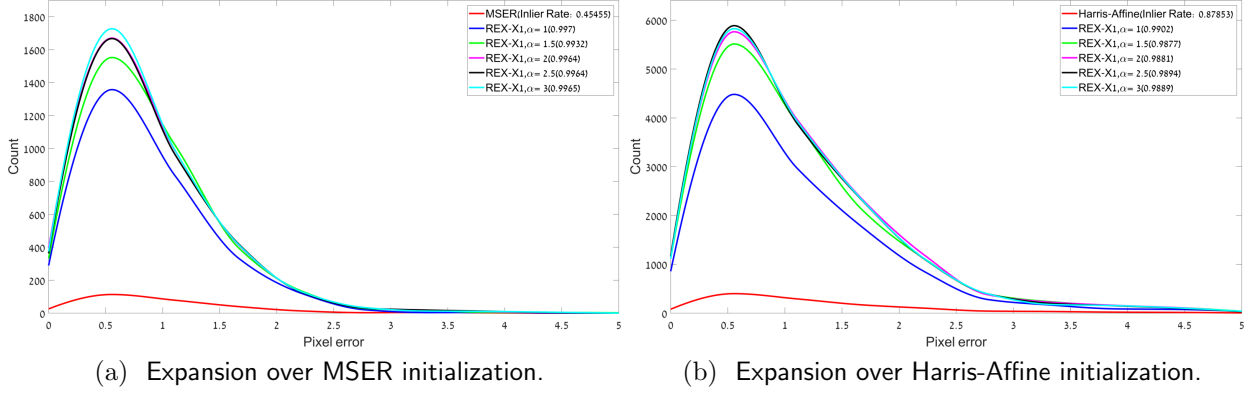


Figure 5: Histograms of point matching errors (only inliers are counted) with different expansion factors.  $\rho$  is fixed at 0.0625, and only one expansion step is allowed ( $FI=1$ ). The total amount of matches increases with  $\alpha$ , up to a point ( $\alpha = 2.5$ ), where increasing  $\alpha$  actually reduces the amount of matches. For better visualization, the graphs are drawn as continuous plots, but represent discrete information. .

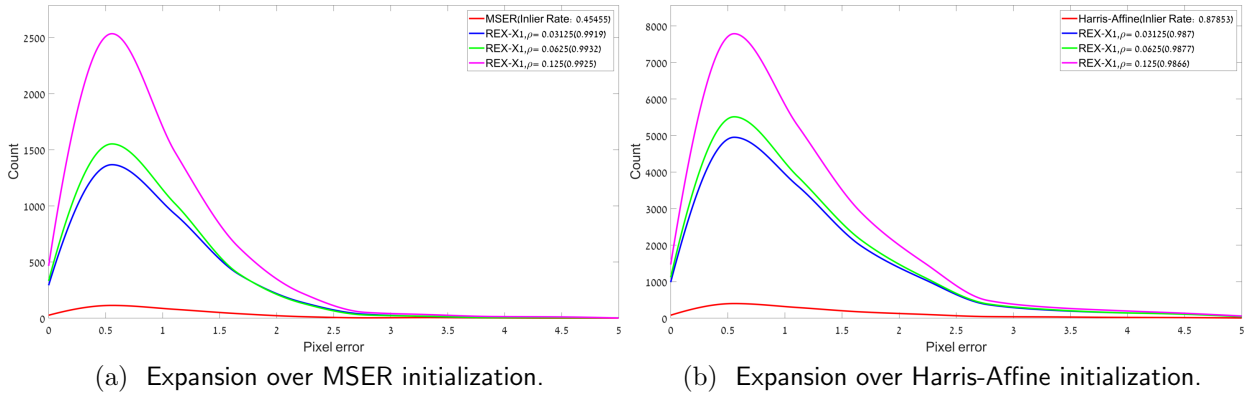


Figure 6: Histograms of point matching errors (only inliers are counted) with different sample densities.  $\alpha$  is fixed at 1.5, and only one expansion step is allowed ( $FI=1$ ). The total amount of matches expectedly increases with the sample density, but not linearly. For better visualization, the graphs are drawn as continuous plots, but represent discrete information.

## 4 Computational Analysis

The computational complexity of the proposed algorithm is significantly lower than that of ASIFT and comparable to that of the other tested algorithms. In this computational analysis, we assume that only one iteration of region expansion was performed for each phase-zero match. The first stage constitutes of applying MSER matching. We assume that MSER matching produces  $N_0$  *phase-zero* matches between the two images. Now, we apply expansion from each of the  $N_0$  regions to  $N_1$  new *phase-one* points. For each *phase-one* point, we apply the match scanning algorithm (thus running the scanning algorithm  $N_0 \times N_1$  times). The computational complexity of the correlation-based scanning algorithm is  $O((W_s \log_2(W_s))/(W_s - F_s))$ , where  $F_s$  is the area of the template, and  $W_s$  is the area of the search window. Thus, we have a complexity of  $O(N_0 \times N_1 \times (W_s \log_2(W_s))/(W_s - F_s))$ . In this work, we set  $F_s = 33 \times 33$ ,  $W_s = 49 \times 49$ , and typically  $N_1$  is between 9 and 16. Thus, in practice, we have added not more than  $N_0 \times 3.5 \times 10^3$  simple operations over MSER, which is near linear in the image size [10] and is considered a low complexity feature detector. Since  $N_0$  is typically linear in the image size, the complexity of the entire proposed algorithm is linear in the image size. To gain proportion, the total computation time (including MSER) is about twice the computation time of MSER matching alone.

## 5 Summary

The ability to estimate the affine transformation accurately, using simple initializers such as local region matches, and then considering larger regions, enabled us to obtain the accuracy benefits of considering the full affine set of transformations without the computational demand of state-of-the-art methods. Moreover, we showed a novel way to reject false matches, which is a crucial ability in many applications, without the need to consider cross-scene complex geometrical transformations. The results achieved by the proposed method are considerably better than those produced by any state-of-the-art technique. There are therefore three fundamental practical implications of this work: 1) making very accurate estimations of local affine transforms; 2) increasing the number of point matches; and 3) increasing the percentage of inliers. The ability to expand any point match to a larger region enabled us to analyze large portions of the scene, which could not have been analyzed by any existing local region detector. The expansion concept may also be generalized beyond affine or even planar regions to cover higher dimension surfaces of the scene, which will allow for better coverage of general scenes.

## Image Credits

All images in this manuscript were taken either from the “Affine Covariant Regions” dataset<sup>3</sup> or produced by the authors (license CC-BY-SA).

## References

- [1] A. ALAHI, R. ORTIZ, AND P. VANDERGHEYNST, *FREAK: Fast retina keypoint*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 510–517. <https://doi.org/10.1109/cvpr.2012.6247715>.
- [2] H. BAY, T. TUYTELAARS, AND L. VAN GOOL, *SURF: Speeded up robust features*, in Proceedings of European Conference on Computer Vision (ECCV), Springer, 2006, pp. 404–417. [https://doi.org/10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32).
- [3] M. CALONDER, V. LEPETIT, C. STRECHA, AND P. FUA, *BRIEF: Binary robust independent elementary features*, in Proceedings of European Conference on Computer Vision (ECCV), Springer, 2010, pp. 778–792. [https://doi.org/10.1007/978-3-642-15561-1\\_56](https://doi.org/10.1007/978-3-642-15561-1_56).
- [4] O. CHUM AND J. MATAS, *Matching with PROSAC-progressive sample consensus*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2005, pp. 220–226. <https://doi.org/10.1109/cvpr.2005.221>.
- [5] E. FARHAN AND R. HAGEGE, *Geometric expansion for local feature analysis and matching*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 2771–2813. <https://doi.org/10.1137/140997671>.
- [6] V. FERRARI, T. TUYTELAARS, AND L. VAN GOOL, *Simultaneous object recognition and segmentation by image exploration*, in Proceedings of European Conference on Computer Vision (ECCV), Springer, 2004, pp. 40–54. [https://doi.org/10.1007/978-3-540-24670-1\\_4](https://doi.org/10.1007/978-3-540-24670-1_4).

---

<sup>3</sup><http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>

- [7] M. A. FISCHLER AND R.C. BOLLES, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24 (1981), pp. 381–395. <https://doi.org/10.1145/358669.358692>.
- [8] C. HARRIS AND M. STEPHENS, *A combined corner and edge detector.*, in Alvey Vision Conference, vol. 15, Manchester, UK, 1988, p. 50. <https://doi.org/10.5244/c.2.23>.
- [9] D.G. LOWE, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision, 60 (2004), pp. 91–110. <https://doi.org/10.1023/b:visi.0000029664.99615.94>.
- [10] J. MATAS, O. CHUM, M. URBAN, AND T. PAJDLA, *Robust wide-baseline stereo from maximally stable extremal regions*, Image and Vision Computing, 22 (2004), pp. 761–767. <https://doi.org/10.5244/c.16.36>.
- [11] K. MIKOLAJCZYK AND C. SCHMID, *Scale & affine invariant interest point detectors*, International Journal of Computer Vision, 60 (2004), pp. 63–86. <https://doi.org/10.1023/b:visi.0000027790.02288.f2>.
- [12] J-M. MOREL AND G. YU, *ASIFT: A new framework for fully affine invariant image comparison*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 438–469. <https://doi.org/10.1137/080732730>.
- [13] C. TOMASI AND J. SHI, *Good features to track*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (1994), pp. 593–600. <https://doi.org/10.1109/cvpr.1994.323794>.
- [14] A. VEDALDI AND B. FULKERSON, *Vlfeat: An open and portable library of computer vision algorithms*, in Proceedings of the International Conference on Multimedia, MM '10, New York, NY, USA, 2010, ACM, pp. 1469–1472. <https://doi.org/10.1145/1873951.1874249>.
- [15] A. VEDALDI AND S. SOATTO, *Local features, all grown up*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, IEEE, 2006, pp. 1753–1760. <https://doi.org/10.1109/cvpr.2006.176>.