Ultra-wide Baseline Aerial Imagery Matching in Urban Environments

Hani Altwaijry
haltwaij@cs.ucsd.edu
Serge Belongie
sjb@cs.ucsd.edu

Department of Computer Science and Engineering
University of California, San Diego
California, USA

Abstract

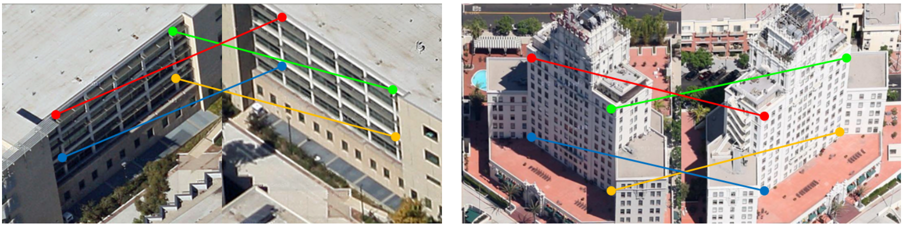
Correspondence matching is a core problem in computer vision. Under narrow baseline viewing conditions, this problem has been successfully addressed using SIFT-like approaches. However, under wide baseline viewing conditions these methods often fail. In this paper we propose a method for correspondence estimation that addresses this challenge for aerial scenes in urban environments. Our method creates synthetic views and leverages self-similarity cues to recover correspondences using a RANSAC-based approach aided by self-similarity graph-based sampling. We evaluate our method on 30 challenging image pairs and demonstrate improved performance to alternative methods in the literature.

1 Introduction

Correspondence matching is a fundamental problem in computer vision, with critical importance in structure from motion [17] and stereo disparity estimation [27]. Correspondence estimation also plays a key role in image registration [3, 11] and pose-estimation [15].

Today, a large amount of aerial imagery is available online via mapping services such as Google Maps [19] or Bing [10]. These images are typically tied to location and orientation metadata. If we were to pick any pair of images from two different aerial views (see Figure 1 for an example), and perform SIFT-based [23] correspondence matching, we would find ourselves with a large number of mismatches due to the large distortions between the images. Even when augmenting these methods with robust approaches such as RANSAC [14] and its variants, we would still fail at finding correct correspondences since RANSAC has difficulty calculating the correct model without a large ratio of correct matches to outliers. These difficulties – large distortions, and low ratio of correct matches to outliers – together render traditional methods ineffective. This problem has been called "Ultra-wide" baseline correspondence matching because the distance and angle from which these two images were taken is extremely large and cannot be explained by small translations or rotations [30].

In this paper, we consider the problem of correspondence matching for aerial imagery in urban environments. Our approach builds on multiple ideas in the literature. Namely, A-SIFT [34], patch-based methods [29], Generalized RANSAC framework [35], self-similarity [18, 28], graph-based image matching [20], and geometric-invariance [24]. The main idea behind this work is to combine view-synthesis with multiple point correspondences under a



(a) (b)

Figure 1: Two pairs of aerial images with correspondences are shown. Notice the large affine transforms and repeated structure exhibited in the two, as well as the varying lighting conditions.

RANSAC-based scheme. Robust model estimation is supported by self-similarity principles and graph-based modeling that drives the sampling process in a restricted manner that allows the correct model to be extracted. Each of these ideas was chosen to deal with specific problems that cause failures in the earlier approaches as we will now briefly describe.

Synthesis vs. Normalization: As described in [34], features usually employ two techniques to achieve certain invariance properties. Those two techniques are synthesis and normalization. In the first case, different possibilities are synthesized to make up for certain changes. For example, in A-SIFT [34] and in [21], different affine transformations were synthesized to capture appearance changes. However, when normalization is used, the calculated feature is projected to some nominal standard, which can be difficult to produce, such that different instances could be projected to that same standard. We believe the ultra-wide baseline nature in aerial images calls for view-synthesis, and hence, we follow in the footsteps of A-SIFT and adopt affine synthesis.

Patches: In feature-based approaches, a detector is implemented to find points or regions that are salient. A descriptor is then built by using a support region around given keypoints. In the case of aerial imagery, the images exhibit similar scale that allows us to disregard scale changes to a certain degree. Therefore, a fixed-size patch is likely to yield good results under this assumption, especially when augmented with affine transforms that include small scale changes.

Multiple-Correspondence RANSAC: In the Generalized RANSAC framework of [35], multiple point correspondences are allowed by having points that satisfy a distance threshold as viable candidate matches, as opposed to match uniqueness criteria as with the SIFT [23]. By allowing multiple correspondences we overcome the case of repeated structure, however, it gives rise to ambiguities that need to be resolved. When we incorporate view-synthesis to the system, a combinatorial explosion of possibilities arise. This requires more guided sampling for RANSAC.

Self-similarity and graph-based representation: In [18], textures comprising repeated elements were detected by correlating regions around different keypoints with each other. In a similar sense, repeated structure also arises in buildings' facades. This signals the need of a method to disambiguate our possible matches. We see a number of graph-based approaches [4, 20] used in image correspondence matching. We connect these two ideas by creating a graph of self-similar patches in our images which we use to drive the Multiple-Correspondence RANSAC sampling process.

1.1 Dataset

We collected 30 aerial image pairs showing buildings from different aerial vantage points from Google Maps [19]. As far as we know, there are no previous datasets dedicated for ultra-wide baseline aerial imagery. The examples were hand picked to be representative for most aerial scenes of urban environments, and such that buildings exhibit a dominant plane.

2 Related Work

2.1 Correspondence Matching

Scale-invariant Feature Transform (SIFT) [23] presented a large step in feature based matching. A large body of work has appeared since then, including many other feature descriptors such as SURF [2], BRISK [22], and FREAK [25]. These feature descriptors usually perform badly under extreme viewpoint changes, leading to failure even when applied in a RANSAC framework.

A-SIFT [34], integrates affine-invariance to SIFT by synthesizing affine views of the two scenes under consideration. The different synthesized images are then passed through the standard SIFT keypoint detection and description process. While this approach sounds applicable to our problem, the huge number of matches and the ambiguous repeating structures defeat the approach. In A-SIFT, the affine transformations applied to the images are discarded after extracting descriptors. This leads to a heavy dependence on the matching and robust estimation approach, because random sampling cannot be prevented from mixing different affine transformations in a local region.

D-Nets [33] take a different approach in finding correspondences. Their method generates lines between keypoints or grid points and calculates descriptors for each line. The line segments from the two images are matched through a hashing and voting scheme. Their method delivers both good performance and accuracy. Their departure from conventional patch-based approaches offers good insight into correspondence matching and therefore we compare our approach to D-Nets.

2.2 Ultra-wide Baseline Matching

There have been several works in wide baseline stereo matching [26, 31]. However, in those cases the distortions exhibited in the pair of images are not very large. For the case of “Ultra-wide” baseline matching, several works have been presented.

The “scale-selective self-similarity” S^4 descriptor was presented by Bansal *et al.* [1] which they used in performing geolocalization of street view images through facade matching against labeled bird’s-eye view aerial images. In their case, the aerial images were labeled by marking the existing building facades for rectification purposes.

Chung *et al.* [9] present a method for building recognition by employing semantically rich sketch representations that are matched with a spectral graph matching procedure. Their method uses MSERs [13] to detect affine-regions that are then used to find repeating structure, which in turn are used to create a sketch representation. One interesting aspect of their work is their use of geometrical invariants based on node relationships. Our method shares the spirit of this approach, as we will describe shortly.

In [36], Zhang *et al.* present a visual phrases approach to image retrieval. It is supported by imposing geometric constraints over the different visual words in a given scene. The

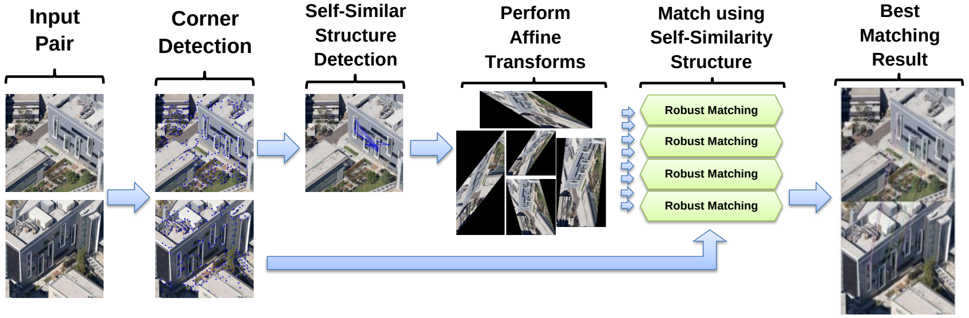


Figure 2: An overview of the matching pipeline is shown here. The details of the approach are discussed in Section 3.

geometry preserving notions they present highlights the importance of respecting geometry between keypoints occurring together spatially.

2.3 Robust Estimation

Correspondence matching is often supported by robust estimation approaches such as RANSAC [14] to extract the correct model representing the underlying geometry. Many variants of RANSAC exist to solve different problems, such as the existence of multiple models (Multi-RANSAC [37], Sequential-RANSAC [32]) as in multiple facets of a building. In these variants, the notions of multiple point correspondences is not considered. Other variants such as PROSAC [8] perform guided sampling to increase robustness to outliers.

The Generalized RANSAC framework [35], incorporates the notion of many-to-many matching in RANSAC as an effort to overcome repeated structure or self-similarity problems. However, it still based on random sampling which does not respect spatial structure. This leads to many draws that give rise to incorrect models.

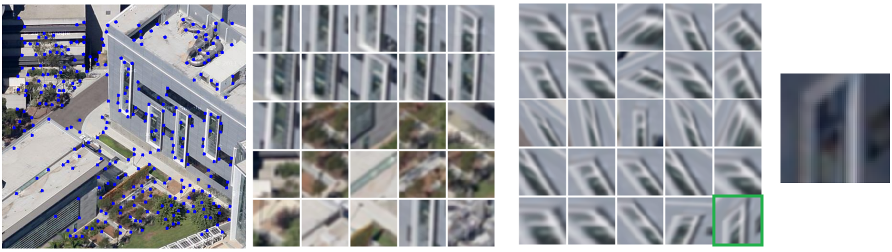
In the literature, there are other approaches to perform matching under the many-to-many paradigm. Namely, spectral methods such as [6] and optimization based methods such as [7], however, space limitations do not permit their discussion.

3 Approach

3.1 Feature Extraction and Description

For keypoint detection, we employed the standard Harris corner detection procedure [16] after smoothing the image with a Gaussian kernel. Our goal was to obtain the corner points covering most of the features on building facets.

We describe our keypoints by placing a window of size $p \times p$ around each keypoint making a square patch, and then we compute the Histogram of Oriented Gradients (HOG) [12]. Our use of HOG was due to its power of capturing the gradient structure and its wide success in the object recognition literature. Figure 3(a) shows an example of detected keypoints and sample patches.



(a) (b)

Figure 3: In (a), detected keypoints are shown along a subset of the patches representing them. We can see that the extracted keypoints represent good corners that are likely to be encountered in another view of the building. In (b), a sample of affine transformed patches corresponding to the keypoint shown on the right. A correctly matching patch is highlighted with a green borderline.

3.2 Affine Synthesis

The aerial imagery under consideration seems to obey the affine camera model to some extent, as the camera is very distant from the imaged objects, and the field-of-view is small. This leads us assume affine local regions, and therefore following in the spirit of A-SIFT [34], we synthesize affine transformations. However in A-SIFT, the transformations are applied to both input pairs, and follows a different sampling procedure. We apply our transformations to one of the input pairs only, and as follows:

$$Scale = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \; Shear = \begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \; Rotation = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\forall S_{x,y} \in [S_{begin} : S_{step} : S_{end}], \forall Sh_{x,y} \in [Sh_{begin} : Sh_{step} : Sh_{end}], \forall \theta \in [\theta_{start} : \theta_{step} : \theta_{end}] \quad (2)$$

$$A = Scale \times Shear \times Rotation, \qquad I_{S_x,S_y,Sh_x,Sh_y,\theta} = A \times I \quad (3)$$

where I is an image.

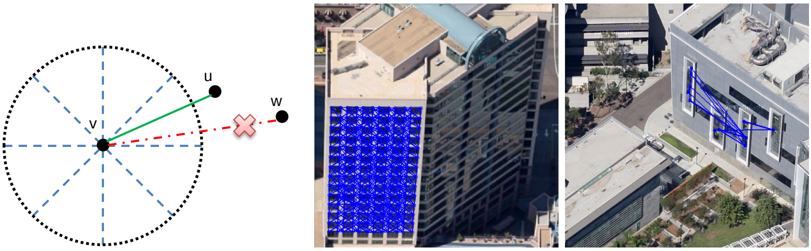
The transformations applied belong to a subset of the affine transformations group. The different variable ranges for, *e.g.* $S_{x,y}$, are chosen to cover a wide variety of affine transformations that should capture the expected distortions in the aerial imagery. Figure 3(b) shows instances of affine transformed patches, and a corresponding patch from the target image.

3.3 Self-Similarity Graph

Buildings, in general, exhibit features that are similar to one another which is due to architectural designs with repeating patterns of windows, balconies, railings, etc. This is leveraged by forming a graph over similar patches in one of the input images. Note that we only consider one image from the input pair for the self-similarity information and not both. The reason will become clear during the matching stage.

We begin our self-similarity computation by calculating the distance matrix D for all pairs of patches by comparing their HOG descriptors using the l_2 norm, i.e.:

$$D_{ij} = ||h_i - h_j||_2 \quad (4)$$



(a) (b)

Figure 4: In (a), we illustrate the angular binning around a given vertex v, and show how we assign the vertex u as the appropriate neighbor, as opposed to choosing the vertex w. The choice is made based on geometrical distance. In (b), two examples of the largest connected component in its simplified form using 9 angular bins.

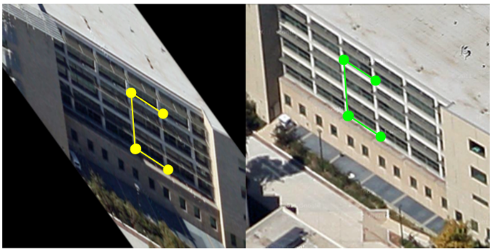


Figure 5: An instance of a transformed input image, and the sampled minimum set. Although the correspondences shown are incorrect in this instance, the spatial configuration is respected, which is the goal behind using the self-similar graph sampling strategy.

where h_i is the HOG descriptor of the patch i. Afterwards, we proceed by constructing a graph $G(V,E)$ with the adjacency matrix M, such that:

$$M_{ij} = \begin{cases} 1 & \text{if } D_{ij} \leq \tau_1, i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where τ_1 is a distance threshold. Using the adjacency matrix M, we find all connected components $C_i(V',E')$ such that:

$$\forall v,u \in V' \iff \text{a path exists between } v \text{ and } u \tag{6}$$

After finding all connected components within G, we select the connected component with the largest cardinality of vertices after passing a non-collinearity test. Then, we simplify it by introducing geometric relations. First, around each vertex, we divide the space into k angular bins. A vertex $v \in V'$ is allowed to have up to k neighbors, such that an angular bin can only have a single neighbor u. We select u as the geometrically closest neighbor to v falling into that angular bin. An example of this step is illustrated in Figure 4(a).

The result of this step is a connected component that describes the structure of self-similar patches. A sample of a simplified self-similar structure is shown in Figure 4(b).

3.4 Matching and Robust Model Estimation

When proceeding to the matching stage, approaches like A-SIFT [34] discard the affine transformations they used. We believe that if two patches match as caused by affine trans-

Algorithm 1 Affine Synthesis

```
Require: Affine Transforms $A^*$, Connected Component $C^*$, Patch Set $P_1$, Patch Set $P_2$
  $AllModels \leftarrow \phi$
  for $A \in A^*$ do
      $P_1' \leftarrow ApplyAffine(A, P_1)$
      $C \leftarrow ApplyAffine(A, C^*)$
      $h_i \leftarrow HOG(p_i)\ \forall p_i \in P_1'$
      $D_{ij} \leftarrow ||h_i - h_j||_2, \forall p_i \in P_1', p_j \in P_2$
      $S \leftarrow \{(i,j) : D_{ij} \leq \tau_2\}$
      $NewModel \leftarrow RANSAC(S, C, P_1', P_2, nIter)$    // Execute RANSAC Algorithm 2
      $AllModels \leftarrow AllModels \cup NewModel$
  end for
  return $BestSet := max_{AllModels}\ |BestSet|$
```

forming one of them, then this affine transform gives us hints about the underlying local geometry that could lead to such matching. Therefore, we explicitly incorporate our affine transformations as part of our RANSAC-based robust estimation method.

Let us call our input images I_1 and I_2. We begin by calculating keypoints on both of I_1 and I_2, and the self-similarity graph obtaining the connected component C^* from I_1. Let P_1 be the set of patches defined by the vertices of C^*, and let P_2 be all patches from I_2. We proceed with Algorithm 1, which applies all affine transformations under consideration to the input data. When an affine transform A is applied, we calculate our matches, and transfer control to Algorithm 2, which is a Multiple-Correspondence RANSAC that samples the data according to the transformed connected component C.

In its essence, the algorithm samples points in the input pair that respect a certain spatial configuration. That configuration ensures that points sampled in the transformed I_1, and in the target I_2 will have *the same geometric relationship*. This enforcement is achieved by maintaining the angular binning relationships of the pairs of points in the current sample. As a result, this decreases the number of random samples to be taken as opposed to randomly picking correspondences. An example of a sample following geometric constraints is shown in Figure 5.

Currently a single homography is estimated, which is clearly a hurdling limitation. However, for an initial test of our approach we believe this is sufficient as we aim to capture the dominant plane in the scene. A final note on our implementation, the best homography guess is passed through a final RANSAC round seeded with the best homography. If RANSAC produced a larger consensus set, we choose the new model, otherwise, we keep the older one. This seemed to increase the robustness of the estimation.

The algorithm performs $O(nIter \cdot |A^*|)$ RANSAC runs, and in each run, it performs $O(nm + |C^*||S| + n)$ operations where nm account for matching n points from I_1 with m points in I_2, and $|C^*||S|$ account for worst case neighbor matching, and finally n for model evaluation.

4 Experiments

4.1 Implementation Details

In our implementation, we used a patch size of 50×50. Each cell in the HOG descriptor covered 5×5 pixels. The number of iterations $nIter$ is set to 5000. The affine transforms ranges were chosen reasonably to cover possible transformations occurring in the aerial imagery.

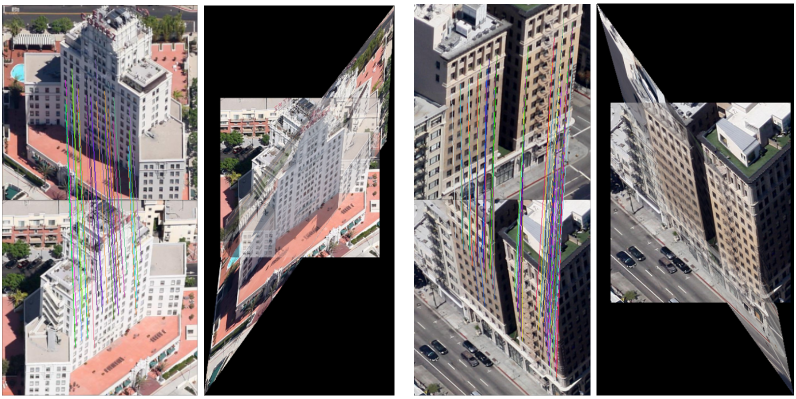
Algorithm 2 RANSAC with Graph Sampling

```
Require: Match Set $S$, Connected Component $C^*$,
    Point Set $P_1$, Point Set $P_2$,
    RANSAC Iteration Count $nIter$

    $BestModel \leftarrow \phi$, $BestSet \leftarrow \phi$
    for $iter \leq nIter$ do
        Pick $v \in C^*$ randomly
        Pick $v' \in P_2$, such that $(v, v') \in S$
        $MinSample \leftarrow$ NeighborMatch($v$, $v'$, [$v$ $v'$])
        $Homography \leftarrow HomographyDLT(MinSample)$
        $ConSet \leftarrow$ EvaluateModel(Homography, $P_1$, $P_2$)
        if $|ConSet| > |BestSet|$ then
            $BestModel \leftarrow Homography$
            $BestSet \leftarrow ConSet$
        end if
        return $(BestModel, BestSet)$
    end for
```

```
function NEIGHBORMATCH($v$, $v'$, $Q$)
    /* The goal of neighbor match is to find
    correspondences that exhibit the same spatial
    layout by looking at matching angle bins*/
    loop    // over neighbors of $v$
        Pick $u \in C^*$, such that $(v, u) \in C^*$
        Pick $u' \in P_2$, such that $(u, u') \in S$
        if angleBin($v$,$u$) = angleBin($v'$,$u'$) then
            if $|Q| \leq minCount$ then
                return NeighborMatch($u$,$u'$, $Q \cup [u\ u']$)
            else
                return $Q \cup [u\ u']$
            end if
        end if
    end loop
    return FAILURE
end function
```

HomographyDLT(): is the Direct Linear Transform algorithm for estimating homographies.
EvaluateModel(): calculates the re-projection error of the identified matches.
minCount: minimum sample size required, for homographies it is 4.
angleBin(*v*,*u*): looks up the angle bin relating these two verticies.



(a) (b)

Figure 6: An example pair is matched using our method, and the recovered homography is used to stitch the two images together.

Certain assumptions were made when choosing these values, *e.g.* we cannot have a 90 degree rotation present in the aerial imagery. We ran our implementation in two configurations, to measure its sensitivity to parameter change. Their details are as follows:

In the first configuration we set $\tau_1 = 6$, $\tau_2 = 7$. The scale factors were chosen as $S_{x,y} \in [0.5, 2]$. The shear factors were chosen as $Sh_{x,y} \in [-1.5, 1.5]$. The rotation angles were between $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. The blur kernel was 3×3, with $\sigma = 0.4$. The Harris detector, had a window size of 7×7, and a threshold of 0.001.

In the second configuration we set $\tau_1 = 6.5$, $\tau_2 = 7$. The scale factors were chosen as $S_{x,y} = 1$. The shear factors were chosen as $Sh_{x,y} \in [-1.75, 1.75]$. The rotation angles were between $\theta \in [-\frac{\pi}{12}, \frac{\pi}{12}]$. The blur kernel was 3×3, with $\sigma = 1.5$. The Harris detector, had a window size of 7×7, and a threshold of 0.01.

Method	Correct Homography	Shifted Homography	Different Plane	Failure	Success Rate
Our Approach-1	3	5	1	21	30%
Our Approach-2	5	4	1	20	33%
A-SIFT	1	0	5	24	20%
D-Nets	4	3	2	21	30%

Table 1: Results of finding homographies using two configurations of Our Approach, A-SIFT, and D-Nets.

4.2 Experimental Setup and Results

To evaluate our approach, we input each pair of the aerial images to: (1) Our approach, (2) A-SIFT, and (3) D-Nets. For A-SIFT [34], we used the implementation provided on their website with a slight modification to estimate a homography using OpenCV [5] instead of a fundamental matrix. We believe that A-SIFT encapsulates SIFT by definition, and therefore we do not compare with the standard SIFT. For D-Nets [33], we use the implementation provided on their website in a straight forward manner employing the FAST keypoint detector.

We measure whether each of these methods find the correct homography, or finds a shifted version of the correct one, or finds a correct but different plane, or completely fails. The results are shown in Table 1. A correct homography is tested against a human labeled homography, and is considered correct if the number of correct matches exceed 75%. Shifted versions and other planes are judged empirically using visualizations. Figure 6(a) shows the result of our matching algorithm on two pairs, and a visualization of the recovered homography by stitching the two images. [1]

Between the two runs, there were 7 unique correct homographies. We see that our method finds a lot of shifted homographies, especially in the cases with numerous repeated structures. In these cases, the typical cause is not finding corresponding keypoints due to the Harris threshold, or too few iterations.

Relative to D-Nets, the results are highly comparable. The issue becomes computational cost vs. memory cost. Our method is computationally intensive. On the other hand, D-Nets requires a lot of memory; they recommend about 32GB of RAM. Our Matlab implementation occupies about 0.8 GB of RAM when running, which can be greatly reduced under a different language implementation. The machine we used had a 3.4 GHz Intel Core i7 processor with 12 GB of RAM.

The failure cases we exhibit are mainly due to two main issues: (1) a self-similar connected component is not found, or poorly constructed with collinearity issues. (2) keypoints are not detected properly due to image blur. Therefore factors such as the size of the employed Gaussian blur, the Harris threshold, or HOG distance threshold have a great impact on the performance. We believe performance can be greatly enhanced by tweaking the connected component discovery by introducing similarity-transitivity resulting in strongly connected components that suffer less collinearity issues, which improves the sampling.

5 Conclusions and Future Work

In conclusion, our proposed approach provides a step forward in the challenging real world problem of ultra-wide baseline image matching for urban environments. Through our use

[1] Visual examples of all pairs are shown in the supplementary material.

of affine synthesis along with the self-similarity graph, we greatly reduce the number of RANSAC iterations needed to find a solution. In our future work, we will pursue the following improvements: (1) reducing the number of affine transformations needed, (2) improving the graph operations, (3) improving the angular binning approach by including distance bins, and (4) including the support of multiple planes.

Acknowledgements

We would like to thank Oscar Beijbom, Mohammad Moghimi, and Eric Christiansen for their valuable discussions and comments. This work was supported by the KACST Graduate Studies Scholarship.

References

[1] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. Ultra-wide baseline facade matching for geo-localization. In *ECCV 2012*.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV 2006*.

[3] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin. An automatic image registration for applications in remote sensing. *Geoscience and Remote Sensing 2005*.

[4] Robert C. Bolles. Robust feature matching through maximal cliques. 1979.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] M. Carcassoni and E.R. Hancock. Point pattern matching with robust spectral correspondence. In *CVPR 2000*.

[7] Ouk Choi and In So Kweon. Robust feature point matching by preserving local geometric consistency. *Computer Vision Image Understanding 2009*.

[8] Ondrej Chum and Jiri Matas. Matching with PROSAC - progressive sample consensus. In *CVPR 2005*.

[9] Yu-Chia Chung, T.X. Han, and Zhihai He. Building recognition using sketch-based representations and spectral graph matching. In *ICCV 2009*.

[10] Microsoft Corp. Bing maps. URL http://maps.bing.com/.

[11] Xiaolong Dai and S. Khorram. A feature-based image registration algorithm using improved chain-code representation combined with invariant moments. *Geoscience and Remote Sensing 1999*.

[12] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*.

[13] Maximally Stable Extremal, J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from. In *BMVC 2002*.

[14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*.

[15] R.M. Haralick, Hyonam Joo, D. Lee, S. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *Systems, Man and Cybernetics 1989*.

[16] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference 1988*.

[17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[18] James Hays, Marius Leordeanu, Alexei A. Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In *ECCV 2006*.

[19] Google Inc. Google maps. URL http://maps.google.com/.

[20] David Lavine, Barbara A. Lambird, and Laveen N. Kanai. Recognition of spatial point patterns. *Pattern Recognition*, 16(3):289 – 295, 1983.

[21] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR 2005*.

[22] S. Leutenegger, M. Chli, and R.Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV 2011*.

[23] David G. Lowe. Object recognition from local scale-invariant features. *ICCV 1999*.

[24] Joseph L. Mundy and Andrew Zisserman, editors. *Geometric invariance in computer vision*. MIT Press, Cambridge, MA, USA, 1992.

[25] Raphael Ortiz. Freak: Fast retina keypoint. In *CVPR 2012*.

[26] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV 1998*.

[27] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV 2002*.

[28] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR 2007*.

[29] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV 2012*.

[30] Dennis Tell and Stefan Carlsson. Combining appearance and topology for wide baseline matching. In *ECCV 2002*. Springer Berlin Heidelberg.

[31] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC 2000*.

[32] T. Vincent and R. Laganiere. Detecting planar homographies in an image pair. In *ISPA 2001*.

[33] Felix von Hundelshausen and Rahul Sukthankar. D-nets: Beyond patch-based image descriptors. In *CVPR 2012*.

[34] Guoshen Yu and Jean-Michel Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line 2011*.

[35] Wei Zhang and Jana Kosecka. Generalized RANSAC framework for relaxed correspondence problems. In *3DPVT 2006*.

[36] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR 2011*.

[37] Marco Zuliani, Charles S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *ICIP 2005*.