Epipolar Geometry Estimation for Urban Scenes with Repetitive Structures

Maria Kushnir and Ilan Shimshoni, *Member, IEEE*

***Abstract*—Algorithms for the estimation of epipolar geometry from a pair of images have been very successful in dealing with challenging wide baseline images. In this paper the problem of scenes with repeated structures is addressed, dealing with the common case where the overlap between the images consists mainly of facades of a building. These facades may contain many repeated structures that can not be matched locally, causing state-of-the-art algorithms to fail. Assuming that the repeated structures lie on a planar surface in an ordered fashion the goal is to match them. Our algorithm rst recti es the images such that the facade is fronto-parallel. It then clusters similar features in each of the two images and matches the clusters. From them a set of hypothesized homographies of the facade is generated, using local groups of features. For each homography the epipole is recovered, yielding a fundamental matrix. For the best solution, it then decides whether the fundamental matrix has been recovered reliably and, if not, returns only the homography. The algorithm has been tested on a large number of challenging image pairs of buildings from the benchmark ZuBuD database, outperforming several state-of-the-art algorithms.**

***Index Terms*—Fundamental matrix, repeated structures, SIFT**

I. Introduction

REPEATED structures are common in many types of scenes. They are especially prevalent in objects such as buildings, as can be seen, for example, in Fig. 1. For reasons which will be explained shortly, algorithms for epipolar geometry estimation from two images tend to fail on such scenes. The goal of this paper is to present an algorithm to deal with these cases. We focus on building facades, which are one of the most common repeated structures.

In recent years much progress has been made in developing algorithms for epipolar geometry estimation for wide baseline image pairs. Such algorithms are usually given as input two images and a feature detection algorithm (e.g. SIFT [1]) is run on both, yielding a set of features and their associated descriptors. The two feature sets are then matched, yielding a set of pairs of similar features from the two images. On this set of putative matches a robust algorithm from the RANSAC [2] family is run. This results in a model which in some cases is the fundamental matrix or a homography in others. The matches are also classi ed as inliers or outliers.

In this general framework many advances have been made, resulting in wide baseline stereo image registration systems which are successful in many hard cases with very low inlier match percentages. However, for scenes with repeated structures they often fail because repeated structures yield sets of similar local features that humans and automated systems fail to match correctly. Thus, each feature from such a set in the rst image is matched to many features from its correspondin g set in the second image with similar local matching scores. As the scores are local there is no way to detect the correct match. When general algorithms encounter this situation they choose at random one of the matches, or all of the matches, or discard these possible matches completely. When the images consist mainly of repeating structures, in the rst two case s the algorithm will fail due to very low inlier match percentages. In the last case, the algorithm will fail due to a very low number of correct matches. It is therefore needed to develop an algorithm which is able to deal with this case.

In this paper we present an approach that exploits the structural regularities in the image and uses them as supplementary information during the matching procedure. We rst extract all the SIFT keypoints and their descriptors in both views. We then distinguish and handle separately repeated structures and other features. This is done by nding all the clusters of sim ilar points in each image and incorporating them independently in the registration process. Moreover, in contrast to most image registration algorithms, in which repeated features are neglected, in our algorithm they play a major role in both the estimation and the veri cation steps.

We propose an algorithm that correctly matches repeated structures placed on planar or close-to-planar surfaces. The main application of such an algorithm is for image pairs where the overlap between them is comprised mostly of building facades. Without such an algorithm, systems with an image registration component will unexpectedly fail from time to time when most of the image overlap involves these facades. This is since these algorithms will fail to match in the initial stage repeated features on these facades.

Our algorithm exploits ve important characteristics of urban scenes with repeated structures.

1) A large number of the repeated structures lie on a planar surface in an ordered fashion and if they can be matched correctly the geometric relationship between the image pair can be recovered.
2) Similar local features detected in the image can be clustered.
3) Clusters from one image can be matched to clusters from another image, without determining initially how the individual members of a matched cluster are matched.
4) The fact that repeated elements are partially organized horizontally and vertically is suf cient for building a lis t of hypothesized keypoint correspondences.
5) A small number of non-repeating matches can also be found on the planar surface.

M. Kushnir and I. Shimshoni are with the Department of Information Systems, University of Haifa, 31905 Haifa, Israel. (e-mail: mkushn01@campus.haifa.ac.il, ishimshoni@mis.haifa.ac.il)

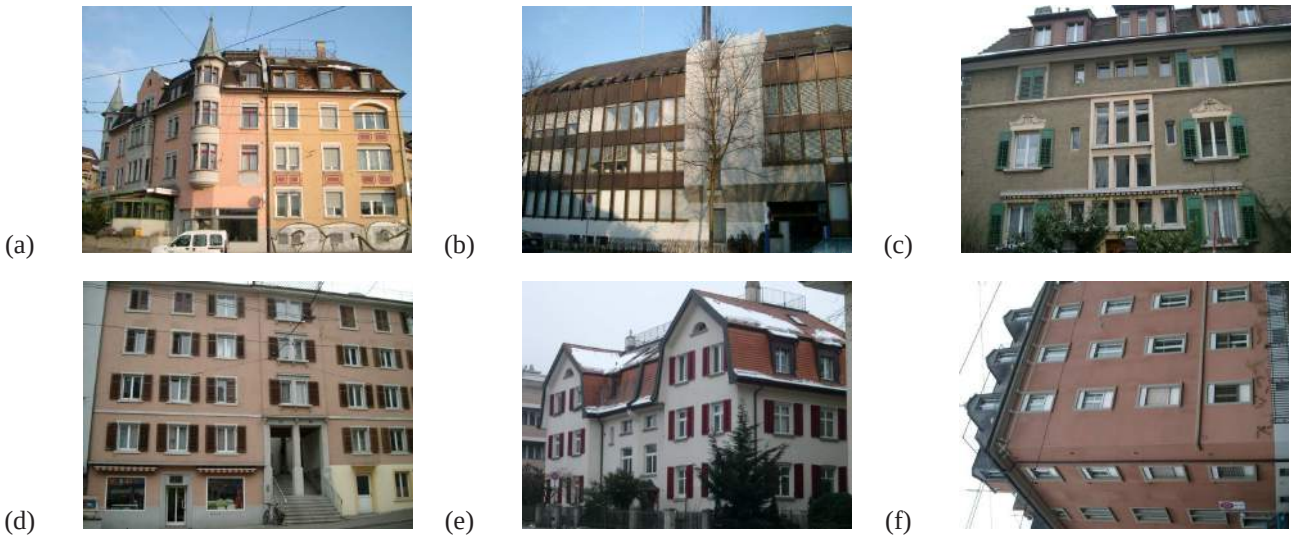


Fig. 1. Images with repeating structures. (a) The non-periodic case: a building with repeating elements appearing on the same vertical and horizontal lines. (b) Grid of repeated structures: a building with repeating elements appearing periodically on the same vertical and horizontal lines. (c) Building with different window sizes and locations. (d) Partial facade. (e) Multiple facades. (f) Building photographed with a roll angle of approximately 90°.

We developed the following algorithm which takes these characteristics into account. Exploiting the first characteristic we recover from each image the main building facade and rectify it so the facade is fronto-parallel. This is done for two reasons. It reduces the number of unknowns that have to be recovered to estimate the fundamental matrix. As this procedure eliminates the perspective effects on the image of the facade, the repeated features on the facade in each image and between the two images become much similar. As a result, repeated features on both images are clustered and clusters in the two images are matched (characteristics 2 and 3). Using the geometric relationships between the repeated features on a facade, small groups of features are created in each image and matched yielding hypothesized transformations between the rectified facades (characteristic 4). The transformations are ranked according to the number of matches that support them, giving a higher weight for non-repeating matches (characteristic 5). Once the transformation has been recovered a RANSAC process is performed to recover the epipole, completing the recovery of the fundamental matrix. In the final step the algorithm decides whether the fundamental matrix has been recovered reliably and if not returns only the homography.

The algorithm was tested on all the image pairs from the ZuBuD database of images of buildings from Zurich [3], for which general purpose state-of-the-art algorithms usually fail, succeeding in most cases.

The paper continues as follows. Section II briefly summarizes the related work. In section III we present our general approach and assumptions. Detailed explanations of our algorithm are introduced in sections IV-VII. In Section VIII we present experimental results on challenging image pairs from the ZuBuD database. We compare our method to three state-of-the-art wide baseline registration algorithms, BEEM [4], BLOGS [5] and USAC [6]. We also show that our method outperforms the Generalized RANSAC [7] algorithm, designed to solve problems caused by ambiguities due to repetitive scene structures. Conclusions and plans for future work are discussed in Section IX.

II. Related work

Since our method incorporates repeated elements into the epipolar geometry estimation process, we first survey general methods for epipolar geometry estimation. Then, we discuss repeated elements studied in different contexts. Finally, we focus on works involving repeated patterns in the process of image registration.

A. Epipolar geometry estimation

Standard techniques for epipolar geometry estimation usually start with keypoint detection on both input images, followed by descriptor representation of every extracted keypoint. The next step is keypoint matching, which is based on descriptor similarity. It is generally accepted that keypoint matching cannot avoid producing incorrect correspondences (outliers). The RANSAC algorithm has become the method of choice for outlier removal in epipolar geometry estimation.

Different approaches have been suggested to speed up the RANSAC algorithm. Several algorithms guide the selection of subsets selected by the RANSAC process [8], [9], [6], [5]. For example, in [8] random sampling is replaced by guided sampling. The guidance of the sampling is based on the correlation score of the correspondences. PROSAC [9] exploits the linear ordering defined on the set of correspondences by the similarity function used in establishing putative correspondences. PROSAC samples are drawn from progressively larger sets of top-ranked correspondences. LO-RANSAC [10] exploits the fact that the model hypothesis from an uncontaminated minimal sample is often sufficiently near the optimal solution and a local optimization step is applied to selected models. In USAC [6] progressive sampling and sequential probabilistic hypothesis testing, are combined with local optimization to yield a comprehensive solution.

Finally, in BLOGS [5] a probability distribution is randomly sampled. However, unlike most algorithms that characterize inlier and outlier error distribution separately, this algorithm uses a conditional characterization of the probability space of correspondence based on joint feature distributions.

Other methods were suggested to reduce the number of putative matches selected at each iteration, resulting in a much faster algorithm that can deal with a much higher percentage of outliers [11], [12], [4]. In [11], [12] three af ne region - to-region matches are used to estimate the epipolar geometry in each RANSAC sample. In BEEM [4] only two similarity transformations are required.

When a large subset of inliers is consistent with a degenerate epipolar geometry, standard epipolar geometry estimation algorithms often return a completely incorrect epipolar geometry with many inliers. In [13], [14] RANSAC-based algorithms for robust estimation of epipolar geometry in the possible presence of a dominant scene plane are presented.

The rst step (or given as input) of all these methods generates putative matches and assigns to each one them a score. When the scene contains mainly repeated structures these algorithms are not able to match the features correctly. This is since as the scores are local there is no way to detect the correct match. When these algorithms encounter this situation they choose at random one of the matches, or discard these possible matches completely. In all these cases the algorithm may fail due to very low inlier match percentages or a very low number of correct matches.

B. Repeated elements

In this work we deal with image registration of repeated elements, but repeated elements have been studied extensively in other contexts, such as detection and grouping of similar elements [15], [16], [17], classi cation and identi catio n [18], matching [19], [20], geo-tagging and location recognition [21], [22], [23], urban reconstruction and scene modeling [24], as well as structure from motion methods [25].

We would like to elaborate on several of these papers. In [25] the problem of recovering the structure from a large number of images (SfM) when the scene contains multiple instances of the same object was addressed. The challenge was to eliminate the incorrect fundamental matrices from the set of fundamental matrices recovered from matching all image pairs. This was done using geometric and image-based cues.

Perhaps the papers most closely related to our work on image registration and repeated patterns are [20], [21], [23], [26], [27], [28], [7], as they present different approaches for matching images of building facades, without analyzing or modeling the entire structure, as was done in [15], [16], [17], [29], [30].

In [7] a guided RANSAC algorithm is presented. A large number of putative matches is generated by matching all possible similar points but giving repeated features low probabilities. Thus, they are not used in the model generation step but only in the veri cation step. When the number of correct unique correspondences is small, the running time of the algorithm can be long. Our method while limited to urban scenes exploits this fact to yield a much faster algorithm.

In [20] it is assumed that the objects investigated are comprised of planar quadrilaterals bounded by straight lines. For each hypothesized match between a pair of quadrilaterals, the homography between images is calculated. The score of the homography is given by counting the number of corresponding Harris corners within the region. It should be noted that unlike our method there is no descriptor extraction for the detected Harris points, and that this method results in a projective homography that matches two building facades without estimating the full epipolar geometry.

In [23] the position of a mobile robot is recovered by matching building facades. The algorithm exploits the fact that the views were obtained from similar heights, thus restricting the matches to a narrow margin surrounding a 1D scan line. Similarly, in [26], invalid correspondences are eliminated on the basis of geometric constraints generated from approximate knowledge of internal and external camera calibration parameters. Our algorithm is more general and does not use such information.

Scenes with multiple objects are dealt with in [27] and [28], using an a-contrario approach. The authors focus on the post-processing step, in which the algorithm has to decide which of the matches belong to the current solution.

There are several works based on the general idea of ASIFT [31]. In [32], [33] ASIFT has been adapted to deal with repeated structures by deriving a score which takes into account repeated features. The algorithms are run on a large number of simulated image pairs. The main difference between ASIFT based approaches and our approach is that we generate only a small number of (simulated) recti ed images and exploit the fact that for the correct image pair the transformation between the recti ed images is simple.

Finally, there are several works which present geo-tagging algorithms. [21] extracts calibrated images from an existing database and matches them to an input image. The transformation supported by the maximal number of matches is returned. A shifted solution might therefore be returned by the algorithm. In [34], an indexing scheme for scenes with repeated structures. While the goal in these cases is to nd the matching image in a database, the goal of our algorithm is to recover the epipolar geometry between the images.

III. Overview

Scenes with repeated structures are very common. In this paper we will focus on the special case where many of the repeated elements lie on planar or close-to-planar surfaces such as building facades. In this section we will present our general approach as well as a list of observations and assumptions that guided the design of the algorithm.

Urban scenes often contain many repetitive structures, with dominant repetitions lying mostly along the vanishing point directions, as can be seen for example in Fig. 1. In our algorithm we consider two typical cases describing most of the urban architecture nowadays.

- *The non-periodic case.* In the rst case the repeated elements are partially organized horizontally and vertically in 3D (Fig. 1(a), (c)-(f)).

- *Grids of repeated structures.* In the second case there exists a 2D grid of repeated elements with constant repetition intervals between them, in both directions. Consider for example Fig. 1(b). The repeated 2D element consists of a window and part of structure between the oors.

Therefore, our algorithm has two variants, one dealing with the general repeating structure case and another one which exploits the grid structure when it exists. In that case a simpler model has to be recovered. As the general algorithm can be applied for both cases, the algorithm for grids of repeated structures is only applied when it is able to detect the grid structure reliably. Consider for example Fig. 1(f). Even though it can be considered an example of the second case, our automatic method did not detect it as such (due to the small number of repeated grid elements) and the general algorithm was successfully applied to it.

Our algorithm relies on certain assumptions about the buildings; however, it should be noted that these assumptions are not "global" but "local," as we are dealing with real world images. In particular, we designed our algorithm for the non-periodic case to require the alignment of only a small fraction of the repeated elements. This makes it tolerant to different window sizes and locations, such as those shown in Fig. 1(a) & (c). In the case of grids of repeated structures, we allow the 2D grid to cover only part of the facade. Different additional elements on the facade, such as the white area appearing in Fig. 1(b), are disregarded by our algorithm. This is in contrast to algorithms that require the complete facade to be visible. In [24], for example, the boundaries of the facade are required to be visible in the image and the algorithm described in [20] does not tolerate occlusions well. Occlusions and partially visible facades, as shown in Fig. 1(b) & (d) and which are very common in real images, are handled by our algorithm. In addition, images of buildings with multiple facades, as presented in Fig. 1(a), (e) & (f), can also be handled by our algorithm. The common case where the original images are taken with a roll angle of approximately $90°$, as shown in Fig. 1(f), are also taken into account by our method.

(a) (b)

Fig. 2. Typical results of matched feature clusters from two views. The clustering is only partial in both images.

The algorithm is described as follows as given in pseudo-code in Algorithm 1.

When two images I_1 and I_2 containing a planar surface with repeated objects are given, the rst step of the algorit hm (described in Section IV-A) is to nd, for each image, a homography which will transform it into a fronto-parallel view. This step is performed for two reasons.

Algorithm 1 Algorithm for recovering epipolar geometry from urban images with repeated structures

1: Input: images I_1 and I_2
2: Apply *Image recti cation* : yield recti ed images RI_1 and RI_2
3: Extract SIFT features from RI_1 and RI_2
4: Cluster SIFT features from each image based on descriptor similarity yielding clusters of features
5: Match clusters from the two images, yielding cluster pairs
6: **if** the building is of the type of a grid of repeating elements **then**
7: Estimate scale s from horizontal and vertical interval ratios
8: Generate a set of hypothesized homographies $\{H_i\}$, each one computed from s and a non-repeating match
9: **else**
10: Generate a set of hypothesized homographies $\{H_i\}$ from minimal subsets of matches
11: **end if**
12: Rank homographies according to supporting matches
13: **repeat**
14: Take next homography H_i from list
15: Estimate epipole e' for I_2
16: $F_i = [e']_\times H_i$
17: Compute score for F_i as the number of supporting matches
18: **until** score decreases
19: **if** F is supported by at least 10 non-planar matches **then**
20: **return** F
21: **else**
22: **return** H
23: **end if**

- Eliminating the projective distortion makes the descriptors recovered from the repeated features in an image more similar to each other and thus easier to cluster. The corresponding feature sets recovered from the two images also become more similar making it easier for corresponding clusters to be found.
- When given two fronto-parallel images of a planar surface, the transformation between them is much simpler. All that has to be recovered is a 2D translation and a scale factor.

SIFT features are extracted from each of the recti ed images, RI_1 and RI_2 and features with similar descriptors are clustered. We then match pairs of clusters from the two images. There are, of course, features which do not cluster; these will be called non-repeating features.

Typical results of matched feature clusters from two views are shown in Fig. 2. It can be seen in both images that the clustering is only partial. There are missing points in the clusters due to occlusion, to the clustering process itself, and due to other reasons. This is one of the challenges that our algorithm deals with.

In Section V we describe how to generate a set of hypothesized transformations $\{H_i\}$ of the recti ed plane appearing in both recti ed images. In this case all that needs to be

recovered is the relative scale s and the relative translation (t_x, t_y). As stated earlier, our algorithm deals with two types of buildings separately. In the non-periodic case, hypothesized transformations are found by locally matching minimal subsets of features from a cluster generated from the rst image to a subset of features from its corresponding cluster from the second image. In the case of a grid of repeated structures, periodicity is assumed. We estimate the horizontal and vertical repetition intervals for each recti ed image, after which t he relative scale s between the two recti ed images is calculated. The unknown translation is then hypothesized from a single non-repeating match.

The hypothesized transformations are ranked by the number of matched features $\{\mathbf{x}, \mathbf{x}'\}$ that approximately satisfy

$$\mathbf{x}' \cong H_i \mathbf{x}. \tag{1}$$

I.e., $||\mathbf{x}' - H_i\mathbf{x}/(H_i\mathbf{x})(3)|| < thresh$, where $thresh$ was empirically determined to be 5 pixels.

In Section VII we exploit the fact that the fundamental matrix F can be factored into

$$F = [\mathbf{e}']_\times H. \tag{2}$$

Therefore for each homography in the list F can be computed by estimating the epipole $\mathbf{e}'$. Once F has been found, the algorithm decides whether there is enough evidence to support it. This is done by counting the number of supporting matches not lying on the plane. If this number is greater or equal an empirically determined value of 10, F is returned, and if not it returns only H.

IV. Preprocessing

A. Image recti cation

In our algorithm, we use the Canny edge detector to detect edges and extract from them line segments in the image. In order to detect vanishing points we apply RANSAC [2] to nd points in the image plane which lie closest to the extensions of a large number of lines in the maximum likelihood sense [35, Chapter 8.6.1]. The two points with the highest number of supporters are assumed to be the vertical and horizontal vanishing points $\mathbf{V}_{\mathbf{p_v}}$ and $\mathbf{V}_{\mathbf{p_h}}$ respectively.

Under the standard assumptions of square pixels, zero skew, and that the principal point is at the image center, and using the fact that the vanishing points represent orthogonal directions in space, the internal calibration matrix K can be recovered [35, Chapter 8]. When two orthogonal facades can be seen in the image, there exists a third vanishing point candidate $\mathbf{V}_{\mathbf{p_3}}$ with a large number of supporting lines. We can use K to verify whether this vanishing point represents a direction in 3D space which is close to being orthogonal to the directions of the other two vanishing points. This is done by computing the angle between them as follows

$$\cos^{-1}\left(\frac{\mathbf{V}_{\mathbf{p_3}}\omega\mathbf{V}_{\mathbf{p_h}}}{\sqrt{\mathbf{V}_{\mathbf{p_3}}\omega\mathbf{V}_{\mathbf{p_3}}}\sqrt{\mathbf{V}_{\mathbf{p_h}}\omega\mathbf{V}_{\mathbf{p_h}}}}\right),$$

where $\omega = K^{-T}K^{-1}$ is the image of the absolute conic.

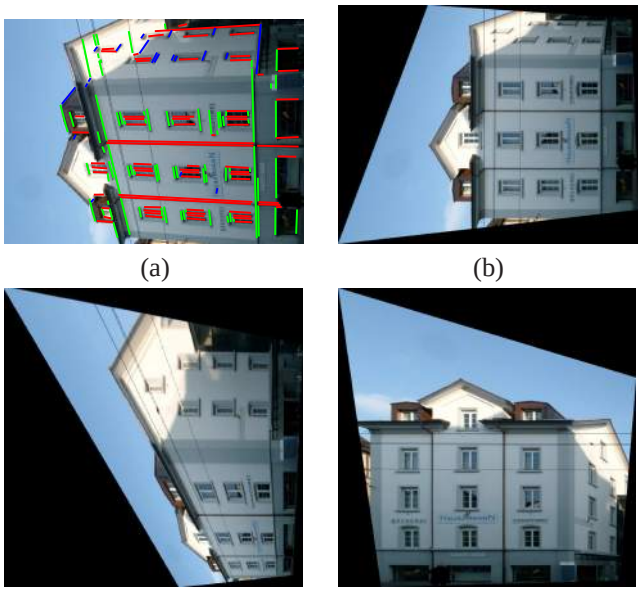
Given the vertical vanishing point (the one with the largest support) and one of the other two vanishing points, a rotation matrix R can be recovered such that the directions of the vanishing points in space are transformed into directions parallel to the x and y coordinates. R is computed as follows:

$$R = \begin{pmatrix} (K^{-1}\mathbf{V}_{\mathbf{p_h}})/|(K^{-1}\mathbf{V}_{\mathbf{p_h}})| \\ (K^{-1}\mathbf{V}_{\mathbf{p_v}})/|(K^{-1}\mathbf{V}_{\mathbf{p_v}})| \\ (K^T(\mathbf{V}_{\mathbf{p_h}} \times \mathbf{V}_{\mathbf{p_v}}))/|(K^T(\mathbf{V}_{\mathbf{p_h}} \times \mathbf{V}_{\mathbf{p_v}}))| \end{pmatrix}.$$

When applying the homography

$$H = KRK^{-1}, \tag{3}$$

to the original image a fronto-parallel view of the facade is generated. An example of the results of this procedure can be seen in Fig. 3. The procedure returns three recti ed images (the two others are rotations of $\pm 90^\circ$ of the rst) in order to deal with the case when the vanishing points have been swapped. When three orthogonal vanishing points have been recovered six recti ed images are generated, three for each of the orthogonal facades. The algorithm will then be applied to a single recti cation of the rst image with each of the three or six recti ed views of the second image.



(a) (b) (c) (d)

Fig. 3. Image recti cation. (a) The original second image (Fig . 2(b)) with detected line segments, which are consistent with the three vanishing points. (b) First fronto-parallel recti ed image, which is recovere d from $\mathbf{V}_{\mathbf{p_v}}$ and $\mathbf{V}_{\mathbf{p_h}}$. (c) Second fronto-parallel recti ed image, which is recove red from $\mathbf{V}_{\mathbf{p_v}}$ and $\mathbf{V}_{\mathbf{p_3}}$. (d) The recti cation and orientation of the second image which is correctly aligned to the recti cation of the rst ima ge (Fig. 2(a)).

B. Keypoint extraction and clustering

For each recti ed image we extract SIFT features and descriptors (using the implementation provided by [36]). As recommended for example in [23], [15], [21], [37], this step is performed on the recti ed images, since in the case of repeat ed features, descriptors are more similar due to the reduction of the projective distortion.

In general, each SIFT keypoint can be assigned an orientation, based on the local image gradient direction, which is the key step in achieving invariance to rotation. In our case, we use

upright SIFT keypoints, whose orientation is set to be vertical. The single xed orientation for all features is a natural cho ice, given that the rotation is compensated for by the recti cati on. Moreover, it prevents features such as, for example, window corners of different orientations, to be considered as the same feature.

We then cluster the SIFT keypoints within a single image based on their appearance similarity. When choosing a clustering algorithm we take into account the fact that the repeated features have undergone perspective distortion which has been only partially corrected by the recti cation. Thus , the further the features are in space, the more different they are. We therefore assume that the shape of the clusters in the descriptor space is non-isotropic. Moreover, the number of clusters changes from case to case and therefore can not be given as input to the algorithm which is the case in the k-means algorithm.

Points are therefore clustered using the agglomerative clustering method, with the single linkage criterion. This criterion was chosen in order to be able to deal with the non-isotropic nature of the clusters. We de ne the distance measure betwee n two keypoint descriptors as the Euclidean distance between their normalized descriptors.

The clustering process terminates when the clusters are too dissimilar to be merged, with the distance threshold set to 0.45. This value was chosen in order to prevent over-segmentation, which is less preferable in our case, as will be explained in Section V-A. For each cluster, we select the medoid of the repeating points' descriptors as the cluster descriptor.

This feature extraction and clustering process is performed for one of the three rotations of the plane. For the other rotations all that needs to de done is to transform the coordinates and apply a permutation on the feature descriptor vectors.

The result of this step is not perfect. Not all clusters represent real repeating objects and not all repeating objects are represented by a cluster of repeating features, as can be seen in Fig. 2. Both images show ten windows. Thus, in theory, ten features are expected in the matched clusters of both views. However, in Fig. 2(a) there are seven features in the cluster, whereas in Fig. 2(b) only six features were found and clustered.

Nonetheless, the number of recovered clusters can be used to differentiate between image pairs with or without repetitive objects, by counting the number of clusters. Fig. 4(a) shows a typical example of an image without repetitive structures: 4 clusters of features, marked by different colors, are shown. Fig. 4(b) shows a typical example of an image with repetitive structures: there, 38 clusters of features, also marked with different colors, are shown. We de ne an image as having repetitive structures if there are at least 10 clusters in both recti ed images in the pair.

C. Cluster matching

For each pair of recti ed images, we match keypoint clusters from the previous stage. Here this process has to be repeated for each rotation of the second image due to the change in descriptors. We check all possible cluster pairs from the two images and compute the Euclidean distance between the cluster descriptor vectors of each pair. As in Lowe's approa ch, we de ne the best match as the one with the minimal distance. We determine the probability that a cluster match is correct by taking the ratio of distances from the closest neighbor to the distance of the second closest. Small clusters (smaller than 5 points), or those that do not have any good match (distance ratio larger than 0.8), are discarded.

(a) (b)

Fig. 4. The number of recovered clusters can be used to differentiate between image pairs with or without repetitive structures. (a) Image without repetitive structures (4 clusters). (b) Image with repetitive structures (38 clusters).

V. Planar transformation estimation

As we consider two separate cases of urban architecture, based on different assumptions, we will divide this section into two independent parts. We will rst describe, in Section V-A, the algorithm for the non-periodic case and then, in Section V-B, the treatment for grids of repeated objects.

A. The non-periodic case

We start the image registration process by searching for a speci c transformation H, induced by the recti ed plane with repeating elements on it, that maps one recti ed image t o another. For that purpose we assume that repeating keypoints appear on the same vertical and horizontal lines, without speci c requirements about distances or periodicity. From them we build a list of candidate transformations.

When searching for all possible homographies, we use the recti ed images extracted previously. As a result, both tra nsformed images are fronto-parallel. Thus, instead of searching for eight degrees of freedom of a general projective transformation H, we are left with only three, namely the two coordinates of a relative translation (t_x,t_y) and the relative scale s. This is a simpli ed case of a similarity transformation

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \implies \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

To detect H candidates, we check all the feature points from the two images. For each point from the rst image $\mathbf{x}_c = (x_{cx}, x_{cy}, 1)^T$ we look for its vertical $\mathbf{x}_v = (x_{cx}, x_{vy}, 1)^T$ and horizontal $\mathbf{x}_h = (x_{hx}, x_{cy}, 1)^T$ nearest neighbors within the same cluster, if they exist. Each such point triplet will be denoted $\mathcal{T}$. We perform an identical procedure on the second image; the resulting point triplets are each denoted $\mathcal{T}'$.

Since we work with point triplets within the same clusters, it is clear now why we prefer under-segmentation. If there is over-segmentation, possible triplets might be split between several clusters, reducing our chances of finding the correct ones. Although under-segmentation might introduce additional triplets that combine points from different clusters, our algorithm eliminates these hypotheses.

After the point triplets have been obtained for both rectified images in the pair, we match pairs of point triplets that belong to the matched clusters. This results in three matched points. Every such matched triplet

$$\mathcal{MT} = \{\mathcal{T}, \mathcal{T}'\} \tag{5}$$

is used to compute a transformation H, which is given by:

$$s = \frac{x'_{hx} - x'_{cx}}{x_{hx} - x_{cx}} = \frac{x'_{vy} - x'_{cy}}{x_{vy} - x_{cy}}, \tag{6}$$

$$t_x = \frac{x'_{cx}x_{hx} - x'_{hx}x_{cx}}{x_{hx} - x_{cx}}, \tag{7}$$

$$t_y = \frac{x'_{cy}x_{vy} - x'_{vy}x_{cy}}{x_{vy} - x_{cy}}. \tag{8}$$

The decision to work with triplets is a compromise between two contradictory preferences:

- Relying on the minimal subset of features from a given cluster is preferable due to partial clustering that might result from occlusions and noise; namely, points might be missing from the clusters. Generally, if the scale constraint in Eq. 6 is used, the minimal requirement for transformation estimation is two feature point pairs, which could even be from different clusters.
- Relying on triplets from the same cluster gives rise to fewer candidates and much more accurate results. The scale constraint is exploited to eliminate transformations that do not satisfy it.

When the triplet strategy fails due to lack of candidates, we resort to using pairs of feature points from each image.

To illustrate the feature subset selection process, we present in Fig. 5 a typical result. In both images two point triplets are marked by blue stars. These two point triplets can be used to compute the correct H. The red points in Fig. 5(a) indicate additional points that support that H. The green squares in Fig. 5(b) represent an alternative point triplet from which an additional (incorrect) H candidate is computed.

(a) (b)

Fig. 5. Typical results, when building a list of all possible homographies. (a) Blue stars: An arbitrary point with its vertical and horizontal nearest neighbors within the same cluster. Red points: Additional points from the same cluster, that support the same H. (b) Blue stars: correct point match and its nearest neighbors. Green squares: An alternative point match.

The result of this step is a list $\{H_i\}$ of hypothesized transformations of the rectified plane.

B. Grids of repeated structures

Here we confront a more challenging case than that discussed earlier, the case of periodic repeating elements.

As in the non-periodic case, a general projective transformation H that maps one rectified image to another is reduced to Eq. 4, with only three unknowns left: the relative translation (t_x, t_y) and the relative scale s. In the case of a grid of repeated objects we address this problem differently. We compute the relative scale s using the periodicity, whereas the relative translation (t_x, t_y) is extracted from the non-repeating keypoint correspondences.

1) Computation of the relative scale s: During the first step of image rectification, additional information can be extracted using the detected line segments. Assuming periodicity, we estimate the optimal horizontal I_x and vertical I_y repetition intervals separately for each rectified image. The relative scale s is then calculated from those repetition intervals, as will be explained below.

For all vertical and horizontal lines detected in the rectified image, their intersections are computed with the horizontal and vertical axes respectively. We define the difference between every pair of those intersections as a possible horizontal or vertical repetition interval, and search for the best one. We describe here in detail this search for the horizontal interval. The second case of the vertical interval is identical.

Consider, for example, two vertical line segments, we denote their intersection points with the "x" axis x_1 and x_2, respectively. If those line segments support a horizontal interval I, then

$$x_1 - x_2 = kI \tag{9}$$

for some integer k. Counting how many line pairs support interval length I requires $O(N^2)$ computations. Instead we can use the following equivalent equation.

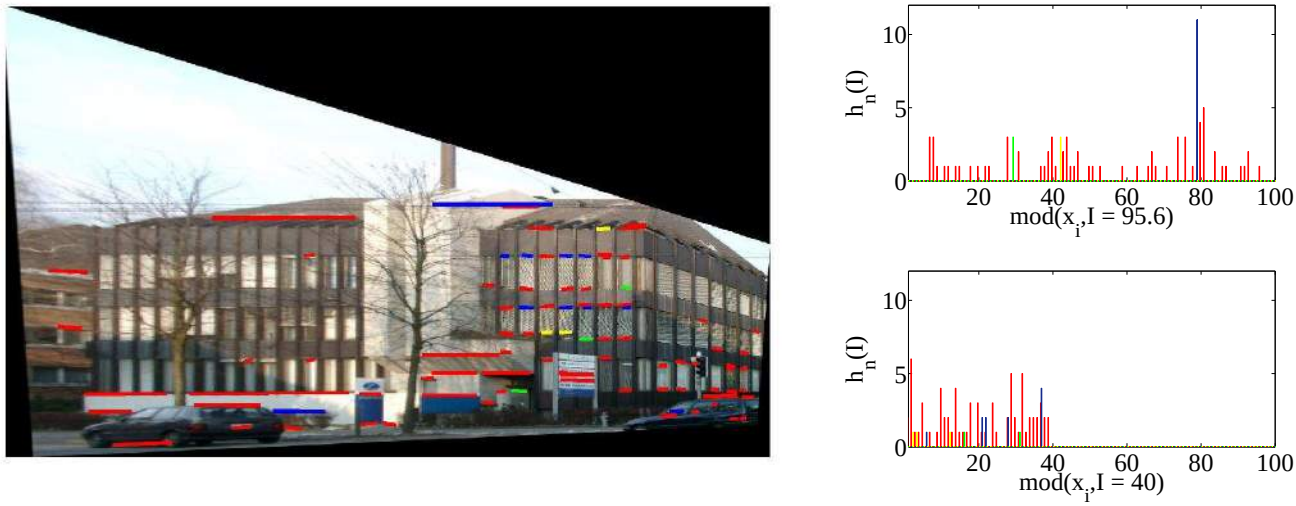
$$\text{mod}(x_1, I) = \text{mod}(x_2, I). \tag{10}$$

Thus, for every possible interval I, we build a histogram h_n of $\{\text{mod}(x_i, I)\}$. Thus, the number of supporting pairs of lines for an interval will be

$$N_I = \sum_{n=0}^{\lceil I-1 \rceil} h_n(I)(h_n(I) - 1)/2. \tag{11}$$

If an interval I is a good candidate, we expect to have sharp peaks in the histogram that come from the unified intersections of the repeating lines.

An example of such a procedure can be seen in Fig. 6. In Fig. 6(a) a rectified image with all the detected horizontal line segments is shown. On the bottom of Fig. 6(b) we present a typical histogram for a bad candidate of I_y. In this case all the bins are relatively uniform, without any preference for a specific position. On the top of Fig. 6(b), a typical histogram for a good candidate of I_y is shown. There is a peak at 79, indicating the unification of repeating lines, which are colored in blue in Fig. 6(a). Another peak colored in green also exists.



(a) (b)

Fig. 6. Estimation of optimal I_y. (a) Recti ed image with all the detected horizontal line seg ments. (b) Histograms for good ($I_y = 95.6$) and bad ($I_y = 40$) candidates of I_y.

However, for intervals $\left\{\frac{I}{2}, \frac{I}{3}, \frac{I}{4} \ldots\right\}$, we expect to have even sharper histograms, as bins of the histogram of I are merged together. Thus, the score for interval I is set to

$$S_I = N_I I \tag{12}$$

to induce a preference for I and not for its fractions. The algorithm builds a list of several (three in our implementation) candidates for I with the maximal scores, so that even when the correct interval obtains the second or third highest score it will be found.

An example of the function $S_I(I)$ can be seen in Fig. 7(a). The multiples and fractions of I also obtain quite high scores. In this case, three candidates for I with the maximal scores are $I_{max} = 84.7, 42.4, 169.2$.

In addition, the value of $S_I(I_{max})$ can be used to detect images with a grid of repeating structures. In general, we expect that the value of $S_I(I_{max})$ will be higher in the case of a grid of repeated objects than in the non-periodic case, as shown in Fig. 7. Thus we can distinguish between the two cases by thresholding it.

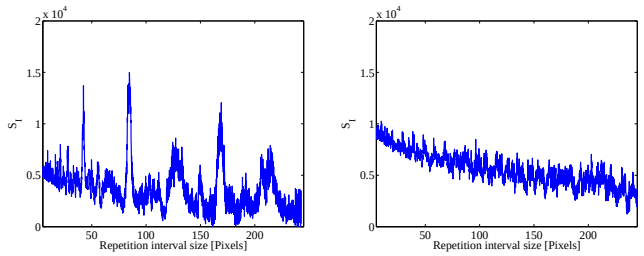
During the next step, when building the list of all possible transformations between the two recti ed images, we exploi t the list of horizontal and vertical repetition intervals extracted previously. We compute the relative scale s from Eq. 4, by:

$$s = I_{x_2}/I_{x_1} = I_{y_2}/I_{y_1}, \tag{13}$$

where I_{x_i} and I_{y_i} are the horizontal and vertical repetition intervals in the two recti ed images which obtained the maximal scores. This process succeeds when I_{x_1} and I_{x_2} represent the same horizontal distance in the 3D scene. This also has to hold for I_{y_1} and I_{y_2} but the vertical distance might be different.

Eq. 13 is used to select sets of consistent interval values and the scale s. Thus, when estimating the transformation between the two recti ed images, we are left with only two out of the eight degrees of freedom of a general projective transformation H: the two coordinates of the relative translation t_x and t_y.

2) Extraction of the relative translation t_x, t_y: A single keypoint correspondence is suf cient for the estimation of the relative translation, which yields a transformation H. Assuming that this H is correct, it should be supported not only by repeating keypoints, but also by corresponding locations of non-repeating SIFT keypoints. Thus, it is possible to build a candidate for H from each non-repeating keypoint correspondence. As in the non-periodic case, the output of this step is also a list of candidate transformations $\{H_i\}$ of the recti ed plane.



(a) (b)

Fig. 7. The combined interval score S_I vs. repetition interval size I. (a) The case of a grid of repeated objects. (b) The non-periodic case.

VI. Homography Improvement and Ranking

From this point on, all the homography improvement and ranking steps, as well as those for image registration, are identical for the non-periodic case and for grids of repeated structures and is performed on the original images.

A. Homography improvement

For each candidate transformation H_i, other feature point pairs from different cluster pairs which satisfy the transformation relation are accumulated and are considered point matches which support the transformation. Relying on them, we improve the accuracy of H_i using LO-RANSAC [10] as

follows. We iteratively calculate a homography by randomly selecting half of the supporting point matches, and compute H_i using a non-linear method which minimizes Sampson's approximation to the geometric re-projection error. As a result we obtain a more accurate homography that relies on many points, instead of a single triplet in the non-periodic case or a relative scale and a point pair in the case of grids of repeated structures. This calculation is immune to the inaccuracies of the recti cation procedure and the proximity between the points in the triplet. See Fig. 8 for all the point matches which support the transformation relation H. The result of this step is a list of candidate homographies $\{H_i\}$ and the hypothesized matches which support each one of them.

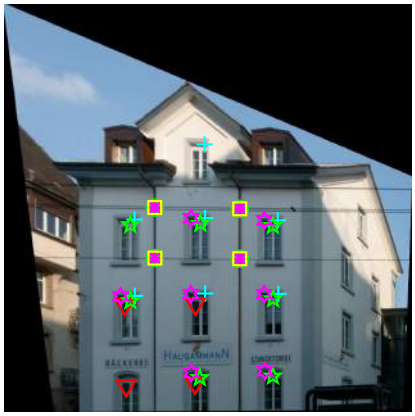


Fig. 8. Feature points from different clusters which satisfy the transformation relation H. Different markings indicate feature points from different clusters.

B. Homography ranking

Once the set of homographies has been generated, the next task is to rank them. The main challenge of this step is to differentiate between the correct homography and homographies which are translations of it, since they will also have a large number of supporting matches. In order to better deal with this task, we exploit the fact that not only repeating elements appear on the plane, but also several non-repeating keypoints. Although there might not be many such matches we use them to detect the correct homography. The algorithm exploits the fact that the probability that a non-repeating keypoint candidate match will approximately satisfy one of a limited number of incorrect homographies is quite low.

We start by matching the SIFT keypoints from the two images, using the standard technique proposed by Lowe. We rank each homography from the list by the number of keypoint matches which are consistent with it. If the homography H is the correct one, it should return not only repeating keypoints from several clusters, but also corresponding locations of non-repeating SIFT keypoints. Therefore, we rank the homographies by means of the following score:

$$S_H = N_{rep} + \alpha N_{non-rep}, \tag{14}$$

where α is a weight constant and N_{rep} and $N_{non-rep}$ are the numbers of supporting correspondences from repeating and non-repeating keypoints respectively. In our experiments we set $\alpha = 100$ (the algorithm works well for $\alpha > 10$), to emphasize that we mainly rely on a small number of non-repeating keypoint matches to rank the homographies.

One of the advantages of our method is that we can empirically tune this weight constant α, by changing the preference of non-repeating over repeating keypoints. In [20], [21], [23], this would be impossible, since only the number of matches is counted. As a result, for an image pair with partial occlusion of the repeating elements, the homography H having the maximal overlap would be chosen, as there are naturally more repeating keypoints in the images. In our method, on the other hand, a few highly weighted non-repeating keypoints would be suf cient to detect the correct H, regardless of occlusion or a partially non-overlapping scene. This step results in a list of candidate homographies, sorted in descending order according to the score S_H.

VII. Image Registration

After the homographies have been ranked, a RANSAC process will be run on each candidate homography H to estimate the epipole $\mathbf{e}'$ on I_2. Combining it with the homography H yields F, as stated in Eq. 2. This process is run on homographies from the ranked set until no improvement is obtained. When looking for the correct fundamental matrix F, we assume that the repeating elements lie on the underlying plane, and therefore they are not considered in this step. Correspondences of non-repeating keypoints however, can appear on or off the plane. Thus, we select off-plane non-repeating keypoints for the estimation of F. Those point pairs $\{\mathbf{x}_i, \mathbf{x}'_i\}$ must satisfy:

$$\|H\mathbf{x}_i - \mathbf{x}'_i\| = \|\mathbf{x}''_i - \mathbf{x}'_i\| = |\rho_i| > d_{proj}, \tag{15}$$

where $\mathbf{x}''_i = H\mathbf{x}_i$, ρ_i is the projective depth, relative to the underlying plane, which is equal to zero for points on the plane. d_{proj} is a constant distance threshold. In our experiments d_{proj} was set to ve pixels.

Another problem, often overlooked in the literature and rarely addressed is demonstrated in Fig. 9. This is the problem of putative matches (pairs of type $\mathbf{x}'$ and $\mathbf{x}''$ are shown), which are due to incorrect matches between repetitive features that were not detected as such during the clustering phase. In general RANSAC is able to deal with outliers. However, when the feature pairs lie on a horizontal or vertical line on the facade, as can be seen in the gure, these incorrect matches will vote together for an incorrect epipole, the horizontal or vertical vanishing point. This choice usually produces an incorrect solution. We therefore remove these putative matches from consideration. These removed matches satisfy

$$H\mathbf{x} - \mathbf{x}' \cong \mathbf{V_p}, \tag{16}$$

where $\mathbf{V_p}$ is one of the vanishing points.

All the remaining matches, termed candidate F supporters, are used in the RANSAC step to recover the epipole $\mathbf{e}'$. This is done as follows. Substituting Eq. 2 into the epipolar constraint yields

$$\mathbf{x}'^T [\mathbf{e}']_\times H\mathbf{x} = 0. \tag{17}$$

Fig. 9. Wrong matches of non-repeating keypoints that result from repetitive elements. Original image with non-repeating keypoints $\mathbf{x}'_i$ marked by yellow circles and $\mathbf{x}''_i$ by red crosses. Green lines are proportional to the projective depth ρ_i.

Two candidate F supporters are suf cient to estimate $\mathbf{e}'$. The candidate H and the recovered $\mathbf{e}'$ will then be combined to yield the fundamental matrix $F = [\mathbf{e}']_\times H$.

In this step the putative matches come from two sources:

- Matched features extracted from the recti ed images; these features come mainly from the parallel planes consisting of the building's facade.
- Matched features extracted from the original images. These matches usually come from off-plane 3D points, since they become too distorted in the recti cation process to be matched using the recti ed images.

Once the RANSAC step has been completed, all the matches that support the fundamental matrix F (including the ones that support the homography) are given to a nal RANSAC; this step accurately recovers the homography H.

The question that remains is whether the algorithm should return F or whether there is not enough evidence to support a fundamental matrix (when, for example, the overlap between the two scenes is close to planar) and only H should be returned. We answer this question by counting the number of matches that support F and do not support H. If there are more than a certain number of supporters (10 in our experiments), F is returned by the algorithm and if not, only H is returned.

VIII. Experimental results

We will now present experimental results of our implementation of the algorithm. We ran experiments with the same parameters on all the results included in this work. These parameters were automatically selected to produce optimal results. We used the publicly available ZubuD database [3] to test our method. The database contains 1005 color images of 201 buildings (5 images per building) from Zurich, taken from different viewpoints and under different illumination conditions, yielding 2010 image pairs. A typical set of ve images of the same building are shown in Figure 10. The gradual change in the viewpoint yields a signi cant rotatio n between the two most distant views as can be seen. The state of the art algorithms failed to correctly match these two images, while our algorithm matched them correctly. The success can be attributed to two reasons. When the images are recti ed, the projective distortion is eliminated making the images more similar. These images also contain a large number of repeated features which are dealt with by the algorithm.

As we were interested in the additional value that our method can contribute, we compared it to the state-of-the-art wide baseline registration algorithms BLOGS [5], USAC [6], and BEEM [4], which can estimate the epipolar geometry in many dif cult cases. In addition we implemented the Generalized RANSAC algorithm [7], developed especially for scenes containing similar repetitive structures and compared its performance to the others. For BLOGS, USAC and BEEM we used the original implementations, available on the Internet, including all the algorithms' parameters, as proposed by th eir authors. Our implementation of the Generalized RANSAC algorithm, was based on all the details speci ed in the paper . The number of iterations was signi cantly increased and set to 100,000 instead of 2000 that was used in the paper.

In order to choose a set of challenging image pairs, we automatically selected all the image pairs, for which BEEM found less than 30 inliers, and then manually checked the resulting fundamental matrices. As a result 139 image pairs were found, for which BEEM failed to nd a correct fundamental matrix. This set is denoted “ZuBuD1 set”. In addition, it is also important to test how does the method perform on “easy” pairs compared to other algorithms. We therefore randomly selected 139 image pairs, out of the remaining pairs from the ZuBuD database. This set is denoted “ZuBuD2 set”. Following BLOGS, for each image pair we manually marked 16 correspondences, different from the SIFT features used to estimate the epipolar geometry. These serve as the ground truth and the mean of roots of the Sampsons distances of these hand marked correspondences serve as the quantitative performance measure. In the cases where there was not enough evidence to support a fundamental matrix and our algorithm returned H, performance evaluation was performed on the in-plane ground truth correspondences only.

In general we tested two different schemes for running our algorithm. It can be run as a standalone method denoted by “our method alone”. It can also be run as combination of our method with any other standard algorithm. In that scheme our algorithm is ran only when the standard method fails to nd a convincing solution. A solution which is supported by less than a certain number of matches is assumed to be a failure. Here we report the results of BLOGS combined with our method, denoted by “our method + BLOGS”, since this combination produced the best results. The failure threshold was set at 50 supporting matches.

We compare the performance of different registration algorithms on the two sets in Figure 11. For every algorithm on each image pair we check the mean of roots of Sampsons distances of the hand marked correspondences and consider it as a success when the performance measure is smaller than a threshold. We present percentage of correct epipolar geometry estimations on each set of 139 image pairs as a function of the threshold. As can be seen from the results, for a threshold greater than seven pixels our method alone outperforms all the others on the dif cult “ZuBuD1 set”, while there is considerable degradation relative to all others on the easy

Fig. 10. A typical set of ve images of the same building.

"ZuBuD2 set". When combining our method with BLOGS, the results somewhat improve on the dif cult set and dramatical ly improve on the easy dataset, achieving the best performance for every threshold on both datasets. Considering the two graphs, it seems that the notion of easy and dif cult cases is very different for our method and the general algorithms. Therefore, even though our algorithm alone performs well on hard cases, it is recommended to always use the combined scheme.

When the reasons for the failures of our method were analyzed, we found that the main cause for failure was when one of the original images was taken with a large view angle relative to the normal of the fronto-parallel plane. There are of course other reasons such as occlusion, or cases when our assumptions about the architecture were not satis ed (buildings almost without repetitions or non planar buildings), but in the vast majority of cases our algorithm failed due to the large view angle (larger than $40°$). A typical example of such an image is presented in Fig. 12(a). Due to the perspective distortion, even when the image is recti ed as can be seen in Fig. 12(b), there is not enough information for the algorithm to use. As a result it fails.

(a) (b)

Fig. 12. A typical example of an image taken with a large view angle relative to the fronto-parallel. (a) The original image. (b) Part of the recti ed image.

To show the typical numerical results of the various steps of our algorithm, as well as a comparison to the other registration algorithms, we chose 20 image pairs of different buildings out of image pairs for which BEEM or BLOGS algorithms failed to nd a correct fundamental matrix, whereas our algorithm ran successfully. We summarize the statistics of the main parameters on the typical runs in Table I.

Due to the low number of non-repeating feature points in those images, our implementation of the Generalized RANSAC algorithm succeeded on only 3 image pairs, whereas BEEM succeeded on 4. We also veri ed that USAC failed on all image pairs. The best performance from the compared algorithm was achieved by BLOGS on the chosen set of 20 image pairs, it succeeded in 6 of the cases.

We provide in Table I for each image pair its object number as it appears in the database. In the second column we indicate the type of output result obtained by our algorithm. "H" stands for the fully planar case, "F" for cases when the fundamental matrix has been found, and "P" for cases decided by our algorithm as grids of repeated structures. In the third column we present the manually veri ed ground truth ("F" or "H").

Next we show that the typical number of SIFT keypoints extracted from the rst image of each image pair is around 3000. Those points are grouped in clusters if similar, or combined in non-repeating match pairs otherwise. In the following three columns we show that the algorithm typically handles about 700 non-repeating match pairs and approximately 40 different clusters of keypoints. These are used to generate a set of candidate homographies. The next two columns in Table I are H inliers and F supporters (number of matches that are inliers of F but not of H), which were described in sections VI-B and VII . The last four columns indicate which of the other registration algorithms succeeded.

In Figure 13 we present four representative results. For each image pair we show the non-repeating matches that are inliers of the fundamental matrix F. The matches, which are also inliers of the homography H, are connected by green lines, and those that were considered as F supporters (matches that are inliers of F but not of H) are connected by cyan lines. The large number of repeating matches are not shown in the gure in order not to clutter it.

In Fig. 13(a) we can see an example of a fully planar case, since there is only one building facade in the left image. As a result, an in nite number of fundamental matrices could be chosen, one of which is shown here. In that case, as discussed earlier, the con dence in F is low due to the small number of its supporters (cyan lines), and we report only the recovered H with its inliers (green lines).

In Fig. 13(b), on the other hand, both buildings have two facades. As a consequence, we obtain a large number of keypoints at different depths and report a fundamental matrix F along with its inliers. We can clearly see a color differentiation between on-plane and off-plane keypoints. Keypoints located on the plane are connected by green lines, whereas off-plane matches are in cyan.

A building with two parallel planes on the same facade is presented in Fig. 13(c). In this situation, the correct H maps only one of the planes, but the keypoints from the other plane have different depths and are colored in cyan. In this example the correct H maps keypoints from the inner plane. The matches from the other plane are used to estimate the fundamental matrix correctly.

In Fig. 13(a) and 13(c) we demonstrate the non-periodic

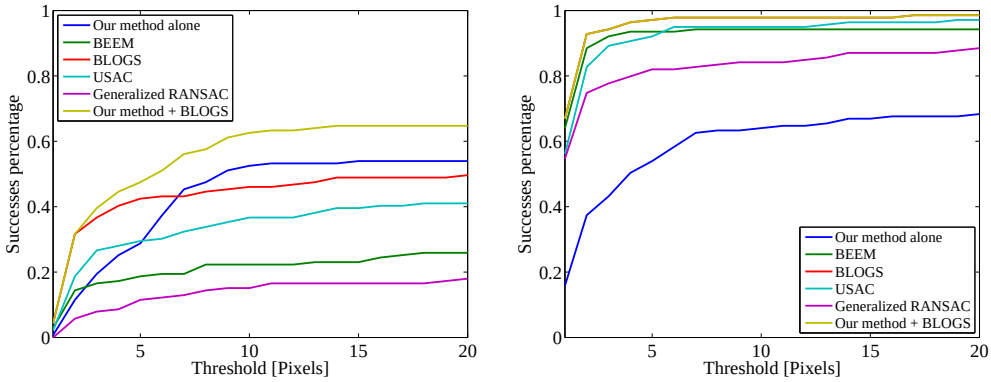


Fig. 11. Performance comparison of the different registration algorithms. (a) "ZuBuD1 set". (b) "ZuBuD2 set".

ZuBuD	Type	GT	Points	Matches	Clusters	Hs	H-in.	F-supporters	USAC	BLOGS	BEEM	Generalized RANSAC
2	P-F	H	3338	648	43	431	14	20				
7	H	H	2616	511	26	4	30	2				
8	H	H	3600	781	53	15	78	7		✓		
10	H	H	2882	706	39	18	91	4			✓	
39	H	H	3073	671	48	11	61	7				
41	H	H	3680	772	26	164	78	5				
53	H	H	2084	481	15	2	47	8				
57	F	F	3707	785	44	195	92	13		✓		
61	P-F	H	4515	874	36	619	17	21				
66	F	F	2956	795	36	1	106	24		✓		✓
67	H	H	3167	727	29	2	95	7			✓	
92	F	F	3992	898	46	7	102	10			✓	
101	F	H	4065	712	36	285	16	16				
110	F	F	2618	700	28	20	42	13		✓		
112	F	F	3482	1029	30	10	314	16			✓	✓
116	H	H	2549	721	20	1564	183	7		✓		✓
120	H	H	3472	809	21	3	215	3		✓		
131	P-F	F	3974	799	37	575	19	30				
162	F	F	4085	785	62	43	13	31				
184	H	H	3247	682	38	33	88	5				

TABLE I
STATISTICS SUMMARY OF THE MAIN PARAMETERS.

case, whereas in Fig. 13(b) and 13(d) grids of repeated structures are presented.

Finally, in Fig. 13(d), an incorrect result is shown. Our method decided that the image is a P-F type rather than a H type in accordance with the ground truth. In this case a homography between two views of the building facade is accurately calculated and indicated by the green lines. However, not only is the incorrect epipole selected during the RANSAC search, as indicated by the cyan lines, but it has enough supporters to be reported as correct.

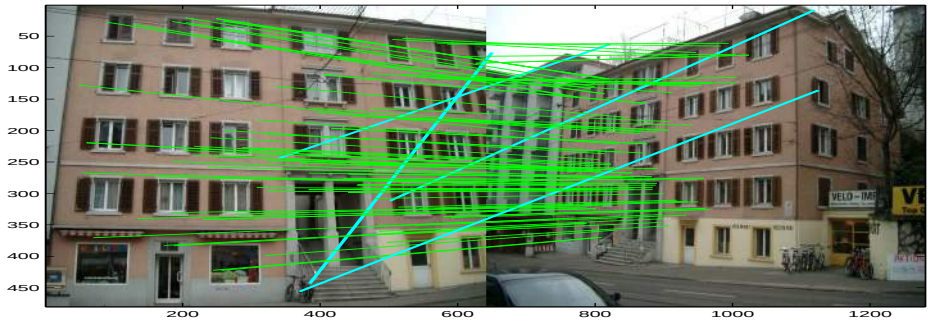
IX. Conclusions and Future Work

In this paper we presented a registration algorithm for scenes of building facades with repeating structures. Even though the algorithm assumes that the image contains a large global structure, it is local in nature and thus able to handle noisy, partially occluded scenes.

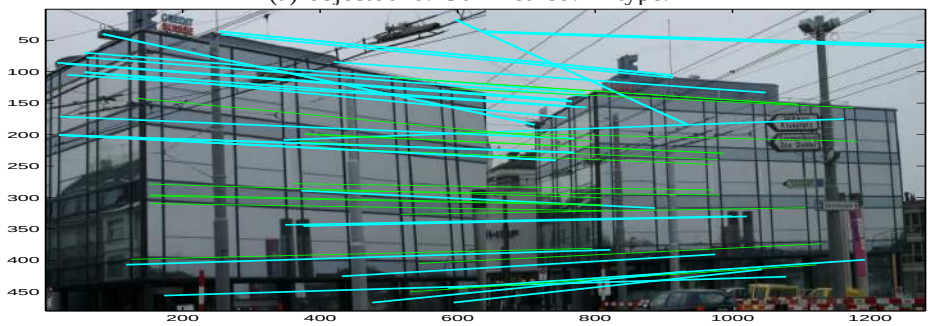
The algorithm divides the task of epipolar geometry estimation into three steps. It rst recti es the images. Then it recovers the homography associated with the recti ed facad e and nally recovers the epipole. Whereas most image registration algorithms neglect repeated features, they play a major role in ours, both in the estimation and veri cation steps. I n our method similar features are clustered in each of the two images, after which clusters of features are matched. From these cluster pairs, a set of hypothesized homographies of the building facade are generated and ranked. For each candidate homography the epipole is recovered in a separate step.

Due to the recti cation step, the algorithm is able to deal with wide baseline image pairs. It does however have problems in dealing with images where the angle between the viewing direction and the normal to the recti ed plane is large.

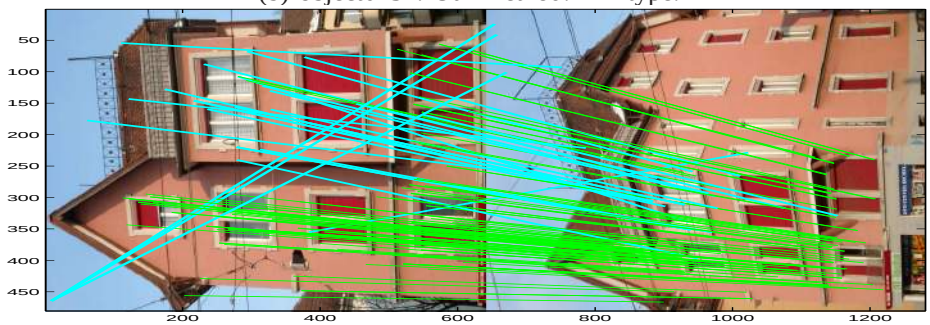
The algorithm was implemented and tested on two sets of image pairs from the publicly available ZuBuD database, a set



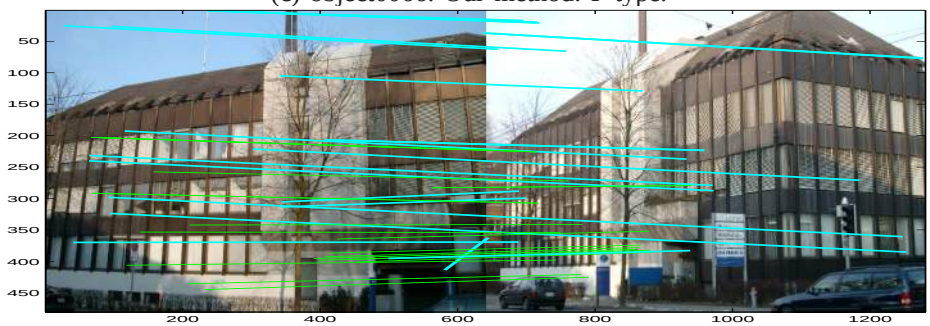
(a) object0010. Our method: H type.



(b) object0131. Our method: P-F type.



(c) object0066. Our method: F type.



(d) object0061. Our method: P-F type.

Fig. 13. Experimental results of our method on object0010, object0061, object0066 and object0131 from the ZuBuD database.

of hard cases and a set of easy cases. Analyzing the results, it seems that the notion of easy and dif cult cases is very different for our method and the general algorithms. Therefore, even though our algorithm alone performs well on hard cases, it is recommended to rst run one of the general algorithms and on automatically reported failures run our algorithm. This scheme produced superior results.

Future research will be dedicated to developing an algorithm also capable of dealing with images of man-made, non-planar objects and natural scenes with repeating objects. In addition we intend to address the case of planar textures.

ACKNOWLEDGMENT

This research was supported by the VULCAN Consortium of The Israeli Ministry of Industry and Commerce.

REFERENCES

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model tting with applications to image analysis and automa ted cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[3] H. Shao, T. Svoboda, and L. Van Gool, "ZuBuD Zurich Buildings Database for Image Based Recognition." in *Technical Report 260, CVL, ETH Zurich*, 2003.
[4] L. Goshen and I. Shimshoni, "Balanced exploration and exploitation model search for ef cient epipolar geometry estimation," *PAMI*, vol. 30, no. 7, pp. 1230–1242, 2008.
[5] A. Brahmachari and S. Sarkar, "BLOGS: Balanced local and global search for non-degenerate two view epipolar geometry," in *ICCV*, 2009, pp. 1685–1692.
[6] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: a universal framework for random sample consensus," *PAMI*, vol. 35, no. 8, pp. 2022–2038, 2013.
[7] W. Zhang and J. Kosecka, "Generalized RANSAC framework for relaxed correspondence problems," in *3DPVT*, 2006, pp. 854–860.
[8] B. Tordoff and D. Murray, "Guided sampling and consensus for motion estimation," in *ECCV*, 2002, pp. 82–98.
[9] O. Chum and J. Matas, "Matching with PROSAC progressive sample consensus," in *CVPR*, 2005, pp. 220–226.
[10] O. Chum, J. Matas, and J. Kittler, "Locally optimized random sample consensus," in *German Pattern Recognition Symposium*, 2003, pp. 236–243.
[11] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?"," in *ECCV*, 2002, pp. I: 414–431.
[12] O. Chum, J. Matas, and S. Obdrzalek, "Enhancing RANSAC by generalized model optimization," in *ACCV*, 2004, pp. II: 812–817.
[13] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *CVPR*, 2005, pp. 772–779.
[14] J.-M. Frahm and M. Pollefeys, "RANSAC for (quasi-)degenerate data (QDEGSAC)," in *CVPR*, vol. 1, 2006, pp. 453–460.
[15] C. Wu, J. Frahm, and M. Pollefeys, "Detecting large repetitive structures with salient boundaries," in *ECCV*, 2010, pp. 142–155.
[16] S. Wenzel, M. Drauschke, and W. Forstner, "Detection of repeated structures in facade images," *PRAI*, vol. 18, no. 3, pp. 406–411, 2008.
[17] N. Jiang, P. Tan, and L. Cheong, "Multi-view repetitive structure detection," in *ICCV*, 2011.
[18] Y. Liu, R. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," *PAMI*, vol. 26, no. 3, pp. 354–371, 2004.
[19] J. Hays, M. Leordeanu, A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in *ECCV*, 2006, pp. 522–535.
[20] J. Lee, K. Yow, and A.-S. Chia, "Robust matching of building facades under large viewpoint changes," in *ICCV*, 2009, pp. 1258 – 1264.
[21] G. Baatz, K. Koser, D. Chen, R. Grzeszczuk, and M. Pollefeys, "Handling urban location recognition as a 2D homothetic problem," in *ECCV*, 2010, pp. 266–279.
[22] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert, "Detecting and matching repeated patterns for automatic geo-tagging in urban environments," in *CVPR*, 2008, pp. 1–7.
[23] D. Robertson and R. Cipolla, "An image-based system for urban navigation," in *BMVC*, 2004, pp. 819–828.
[24] G. Wan, N. Snavely, D. Cohen-Or, Q. Zheng, B. Chen, and S. Li, "Sorting unorganized photo sets for urban reconstruction," *Graphical Models*, vol. 74, no. 1, pp. 14 – 28, 2012.
[25] R. Roberts, S. Sinha, R. Szeliski, and D. Steedly, "Structure from Motion for Scenes with Large Duplicate Structures," in *CVPR*, 2011, pp. 3137–3144.
[26] E. Serradell, M. Ozuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer, "Combining geometric and appearance priors for robust homography estimation," in *ECCV*, vol. 6313, 2010, pp. 58–72.
[27] J. Rabin, J. Delon, Y. Gousseau, and L. Moisan, "MAC-RANSAC: a robust algorithm for the recognition of multiple objects," in *3DPVT*, 2010.
[28] F. Sur, N. Noury, and M.-O. Berger, "Image point correspondences and repeated patterns," INRIA, Tech. Rep. RR-7693, 2011.
[29] M. Bansal, K. Daniilidis, and H. Sawhney, "Ultra-wide baseline facade matching for geo-localization," in *ICCV*, 2012, pp. 175–186.
[30] R. Tylecek and R. Sára, "Spatial pattern templates for recognition of objects with regular structure," in *Proc. GCPR*, 2013.
[31] J. Morel and G. Yu, "ASIFT: a new framework for fully af ne invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.
[32] C. L. Brese, J. J. Zou, and B. Uy, "An improved ASIFT algorithm for matching repeated patterns." in *ICIP*, 2010, pp. 2949–2952.
[33] N. Noury, F. Sur, and M.-O. Berger, "How to overcome perceptual aliasing in ASIFT?" in *ISVC (1)*, 2010, pp. 231–242.
[34] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, 2013, pp. 883–890.
[35] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, March 2004.
[36] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.
[37] W. Changchang, B. Clipp., L. Xiaowei, J.-M. Frahm, and M. Pollefeys, "3D model matching with viewpoint-invariant patches (VIP)," in *CVPR*, 2008, pp. 1–8.

Maria Kushnir received the BSc and MSc degrees in physics from the Technion-Israel Institute of Technology, Haifa, Israel. Currently, she is a PhD candidate in the Department of Information Systems at Haifa University. Her current research interests are computer vision and machine learning.

Ilan Shimshoni received the BSc degree in mathematics and computer science from the Hebrew University in Jerusalem, the MSc degree in computer science from the Weizmann Institute of Science, Rehovot, Israel, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign. Currently, he is a professor in the Department of Information Systems at Haifa University. His research interests are computer vision, robotics, and computer graphics. He is a member of the IEEE.