Structural image and video understanding
Lou, Z.

Citation for published version (APA):
Lou, Z. (2016). Structural image and video understanding

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Image Alignment by Piecewise Planar Region Matching

6.1 Introduction

Image registration has been studied extensively in the literature as it is the building block of many computer vision tasks, such as image stitching [87, 88], medical image analysis [99], 3D reconstruction [1] or image editing [8]. A major challenge is that image pairs to be aligned in the process of image registration may be disturbed by differences in illumination, viewpoint changes, and camera settings. In general, image registration techniques assume a global geometric transformation which warps features from one coordinate system to the other. This is usually done by computing a $3 \times 3$ homography model on the two sets of corresponding features extracted (globally) from the image pairs. The homography-based approach is only suited to match images under constrained imaging conditions such as planar scenes and parallax free camera motion. Hence, the homography model is restricted to 3D planar motion or motion in which the optical center of the camera is fixed (rotation). However, these conditions may not hold, in general, for natural scenes and unconstrained recordings.

Apart from the use of homography models, other image registration methods may also be disturbed by the imaging conditions. For example, methods based on a global geometric transformation [115, 119] may suffer from complex camera motions. Non-parametric methods [11, 89, 124] may be negatively affected by repeating (texture) patterns. Methods using a single information modality, such as intensity [6, 92] or key-point descriptor information [87, 88], are limited in their robustness to shading patterns and large motions.

Therefore, in this paper, we propose a method considering a local geometric model (i.e. piecewise planar scenes) by deriving image regions approximated by planes. For each such plane, an affine model is computed. As the initialization step, our method uses a hierarchical figure-ground segmentation approach producing a large set of regions. Regions which can be approximated by planes are subsequently detected. Each pair of planes is matched using different cues such as intensity, gradient, texture and geometric information. A global constraint is used to avoid the overlap or separation of corresponding regions. The pair-plane matching is refined iteratively by re-segmenting and re-merging regions by an energy minimization process. Figure 6.1 describes the idea...
6. Image Alignment by Piecewise Planar Region Matching

![Image of two images with planar regions marked]

Figure 6.1: An example describing the proposed method. First, the method computes image regions which can be approximated by planes. Then, an affine model is computed for each planar region to align the two images.

of the proposed method. First, planar regions are detected. Then, an affine model is computed for each planar region to align the two images.

The method is designed according to the following criteria: (1) Local geometric model. Assuming piecewise planar scenes, the method allows for complex camera motions. (2) Smoothness. Since all pixels for each plane are assigned using a single affine parameter, the global image registration is smooth and repeating patterns are aligned consistently with each other. (3) Multiple cues. Combining different information cues enhances the robustness of the method to changes in the imaging conditions.

The paper is structured as follows: section 6.2 discusses the related work. The proposed method is described in section 6.3. Specifically, the initial segmentation step is discussed in section 6.3.1, section 6.3.2 focuses on the details of the cost function, while the minimization process is outlined in section 6.3.3. Section 6.4 presents the experiments and possible applications.

### 6.2 Related Work

Image registration is commonly based on key-points or appearance cues (e.g., intensity). Typically, the first class of methods computes a set of corresponding key-point features (or by dense sampling), such as SIFT [90], for the two images. Then, by computing a global transformation model (e.g., homography), the alignment of the two images is obtained [87, 88, 119]. The second type of algorithms provides a matching metric, such as the $L_2$ distance. By solving an optimization problem, these methods either compute a global transformation or flow fields to align the two images [118]. However, for images containing large variations in illumination and/or viewpoint, these methods may fail. For example, if the camera transformation is not purely rotational or when the scene cannot be approximated by a plane, a global homography model will often result in inaccurate pixel-by-pixel alignment.

More recent methods propose image alignment under different imaging conditions. Brox et al. [11] combine descriptors and intensity information in an optical flow setting. By minimizing an energy function, this method allows for the computation of dense optical flow fields. Since this method matches images sampled from a video sequence by
employing optical flow, it may be suffer from abrupt imaging variations. Liu et al. [89] define an energy function on densely sampled SIFT features. A flow field is obtained by minimizing the energy function in a coarse-to-fine manner. In the method of Lin et al [87, 88], a global affine transformation is obtained for two sets of unmatched features. Subsequently, the affine model is computed for each feature. By smoothly varying the deviation of the parameters of each feature in the global affine transformation, dense image matching is obtained. Bhat et al. [10] firstly find two sets of matched features. Then, by exploiting a geometrical constraint, various rigid motions (both foreground and background) are computed. After optimizing a Markov Random Field (MRF) using graph cuts, the two images are aligned pixel-by-pixel. While these methods align each rigid motion pixel-by-pixel, in contrast, our method aims at aligning images by considering the main motion.

Prokaj et al. [107] also use a piecewise model for image alignment. The difference to our approach is that we use more compact, robust regions to model planes instead of assuming a set of uniformly distributed triangles. Other algorithms based on piecewise models [5, 71, 94, 107] are designed to cope with specific types of images such as face and medical images. An image is divided into a set of triangles (or rectangles) based on sets of predefined key-points. Subsequently, points within each triangle share the same transformation. In this way, one triangle has a very high probability of spanning two planar regions. This drawback makes these methods inadequate for images with large viewpoint changes.

Instead of applying a global model [115, 119] or directly using triangles to approximate planar regions, we propose a method to detect planar regions and obtain the initial transformation parameters for each planar region. This strategy reduces the probability that a region will span various planes. We use a global constraint to ensure that regions are tightly connected reducing the influence of the outliers.

### 6.3 Image Registration by Piecewise Segment Alignment

The homography model assumes that the scene can be approximated by a plane. This assumption is too restrictive when the viewpoint of the images to be matched is different or when the scene contains more than one plane. Therefore, in this paper, we propose a piecewise planar model for matching two images.

#### 6.3.1 Initialization

To apply piecewise image matching, we first need to generate a number of planar regions. To this end, a large pool of (highly overlapping) segments is generated by applying the same segmentation method multiple times with different parameter settings. Then, the affine transformation is computed for each segment using pre-computed key-points [140]. After this, the matching residual of each transformation is measured by exploiting appearance and geometric cues. Finally, the initial transformation for each pixel is obtained by selecting the parameters of the segment which has the minimum matching residual. Those pixels which have similar transformation parameters are approximated by a plane.
Figure 6.2: The initialization phase. A set of matched points are computed using ASIFT [140] beforehand. After generating a large pool of segments by employing hierarchical segmentation on the reference image, we warp each segment by fitting an affine model $T_s$ to the corresponding points of the segment in the reference image and the target image. The matching residual $\varepsilon_s$ is computed using eq. 6.1 for each segment. Finally, for each pixel, the transformation having the minimum residual is retained. Pixels having the same parameters are approximated by planes.

**Generating A Large Pool of Segments**

A large pool of segments is generated by applying a hierarchical figure-ground segmentation method [14] on the reference image $I_r$. Segmentation seeds are uniformly chosen. For each seed, a parametric graph-cut method is applied with different scale values [14]. By applying this figure-ground segmentation for one seed and one scale value, one segment is generated. With multiple seeds (6x6 seeds uniformly chosen) for multiple scale values, thousands of segments are generated. The segments which are too small (e.g. less than 1 percent of the total image size) are discarded. Note that since we apply figure-ground segmentation multiple times for different seeds (i.e. location) and scales, the generated segments may overlap, as shown in Figure 6.2.

**Obtaining an Affine Model for Each Segment**

After a pool of segments is generated, each segment $s$ is warped by applying the affine model $T_s$, where $s \in \{1, 2, \ldots, D\}$ and $D$ is the total number of segments, as shown in Figure 6.2. This model is estimated for the two sets of matched points pre-computed for segment $s$ and the target image using ASIFT [140]. When one segment can be approximated by a plane and there are sufficient corresponding points, the segment can be matched well with the target image by applying an affine model. Note that if there are not enough matching points for one segment, this segment is deleted.
Evaluating the Transformation of Each Segment

After the affine model is obtained for each segment \( s \), the aim is to measure the performance of the affine transformation \( T_s \) for each segment \( s \), to identify segments which include several planes.

The matching residual of segment \( s \) is computed by:

\[
\varepsilon_s = \varepsilon_d(s) + \mu \cdot \varepsilon_g(s),
\]

where \( \varepsilon_d(s) \) is based on a descriptor cue and \( \varepsilon_g(s) \) is based on the epipolar geometry. \( \mu \) balances the importance of these two measurements and it is experimentally obtained.

To compute \( \varepsilon_d(s) \), we densely compute SIFT features [90, 127] for each pixel in segment \( s \) and in the target image. The average \( L_2 \) distance of each pair of corresponding pixels is obtained in feature space where \( \varepsilon_g(s) \) is the average geometric deviation of each pixel defined by:

\[
\varepsilon_g(s) = \frac{\sum_{i=1}^{n_s} D(T_s \ast x_i, F \ast x_i)}{n_s},
\]

In which \( T_s \) is the \( 3 \times 3 \) affine model for segment \( s \). \( F \) is the \( 3 \times 3 \) fundamental matrix between two images. \( x_i = [x_i, y_i, 1]^T \) are homogeneous coordinates of pixels in segment \( s \) and \( n_s \) is the number of pixels in segment \( s \). The fundamental matrix is computed using the method by [66]. \( T_s \ast x_i \) results in a point and \( F \ast x_i \) results in a line in the target image. Here, \( \ast \) is the cross product. \( D(p, l) \) is the minimum distance from the point \( p \) to the line \( l \). A drawback of using the fundamental matrix is that it is undefined for a pure rotation of the camera. However, for natural images of daily life, a 100\% pure rotation of the camera rarely occurs.

Obtaining the Initial Planes and Parameters

After matching, each segment \( s \) has an associated parameter \( T_s \) of the affine model and a measurement residual, \( \varepsilon_s \). Given that we have thousands of overlapping segments, each pixel in the reference image, \( I_r \), may be in a different segment. Therefore, for each pixel, the parameter of the transformation which has minimum \( \varepsilon_s \) is selected as the parameter of the affine model.

Two neighboring pixels/regions having similar transformation parameters are merged. To measure the similarity of the transformation parameters for two neighboring pixels/regions, we exchange their parameters and validate that the location of the warped pixels is similar. When the distance between the new (warped) coordinates and the original coordinates is below a certain threshold, the two pixels/regions are merged. The final planes are obtained as shown in the right bottom image of Figure 6.2.

6.3.2 Cost Function for Piecewise Alignment

In our framework, after initialization, several approximated planes are obtained for reference image \( I_r \), denoted by \( P_1, P_2, \ldots, P_M \) and their corresponding parameters denoted by \( T_1, T_2, \ldots, T_M \). Let \( x = [x, y, 1]^T \) be the homogeneous coordinate of a point in the
6. Image Alignment by Piecewise Planar Region Matching

Figure 6.3: The global constraint. Since different parameters are used for each plane, boundary points at the border of two planes are transformed to the same point using a different transformation of the two planes.

reference image. For image \( I_r \), each pixel belongs to only one plane:

\[
P_1 \bigcup P_2 \ldots \bigcup P_M = I_r, \tag{6.3}
\]
\[
P_1 \bigcap P_2 \ldots \bigcap P_M = \emptyset. \tag{6.4}
\]

A plane is a set of connected points: \( P_i = (x_1, x_2, \ldots x_{N_i}) \). Although only the homography model can fully model the transformation of a plane, too many degrees of freedom may lead to large distortions at the border of an image. Thus, we model the transformation of a plane by an affine model [107]. \( W(T_i, x) = T_i \ast x \) is the warped coordinate of \( x = [x, y, 1]^T \) using transformation \( T_i \). For each plane \( P_i \), there is a set of boundary points, denoted by \( B_i \), between the plane \( P_i \) and its neighboring planes. For a boundary point \( x \), \( \pi(x) \) denotes the index number of the plane containing \( x \). \( \xi(x) \) denotes the index number of its corresponding neighboring plane. Segmentation is only applied to the reference image. After the transformation parameters are estimated for each pixel of the reference image, the corresponding pixels in the target image are obtained by exploiting the transformation.

**Global Constraint**

By using the parameters obtained in section 6.3.1 independently for each plane, different planes may be matched to the same region in the target image. A global constraint is required to ensure that the transformation of planes is consistent. As shown in Figure 6.3, the boundary points between two planes should be transformed to the same point using the parameters of the two planes. For each boundary point, we minimize the \( L_2 \) distance between the warped coordinates using the transformation of two neighboring planes. This is defined by:

\[
E_{\text{global}} = \sum_{i=1}^{M} \sum_{b_j \in B_i} |W(T_{\pi(b_j)}, b_j) - W(T_{\xi(b_j)}, b_j)|^2, \tag{6.5}
\]

where \( T_{\pi(b_j)} \) is the parameter of the plane to which \( b_j \) belongs. \( T_{\xi(b_j)} \) is the parameter of its neighboring plane, and \( B_i \) are the boundary points of plane \( i \). Here, we assume that
each boundary point has a single neighboring boundary point. If there are many neighboring boundaries, we randomly select one.

**Cost Function**

To ensure robustness against orientation, illumination and occlusion, an energy function is defined for each plane similar to [11, 12]. Nevertheless, the planar models proposed here are affine transformations while the model used in [11, 12] is a nonparametric flow field. In fact, the energy of each plane is integrated to obtain the data term of the energy function.

\[
E_{\text{data}} = \sum_{i=1}^{M} \sum_{x \in P_i} \Psi(|I_t(W(T_i, x)) - I_r(x)|^2) + \beta \sum_{x \in P_i} \Psi(|\nabla I_t(W(T_i, x)) - \nabla I_r(x)|^2) + \gamma \sum_{x \in P_i} \delta(x) \Psi(|f(I_t(W(T_i, x))) - f(I_r(x))|^2),
\]

(6.6)

where \(I_t\) and \(I_r\) are the intensity values of the two images. \(\nabla I_t\) and \(\nabla I_r\) are the gradients of the matched images. \(f(I_t(W(T_i, x)))\) and \(f(I_r(x))\) are the HOG descriptors [29] of the warped target image and the reference image, respectively. \(\delta(x)\) is 1 if there is a descriptor available in image \(I_r\), otherwise, it is 0. The HOG descriptors are computed at every fourth pixel in the \(x\) and \(y\)-direction. The function \(\Psi(\cdot)\) is:

\[
\Psi(x^2) = \sqrt{x^2 + \epsilon^2}, \text{ where } \epsilon = 0.01.
\]

(6.7)

Finally, the energy function combines both the data term for each plane and the global constraint for all the planes is given by:

\[
E = E_{\text{data}} + \alpha \cdot E_{\text{global}}.
\]

(6.8)

6.3.3 Minimization

The aim is to minimize the data term and the global constraint term iteratively. During the minimization process, planes with high matching residuals are segmented and planes with similar parameters are merged. This process involves four steps which are executed iteratively until convergence or until a fixed number of iterations is reached (Algorithm 2).

1) **Minimizing the global constraint** \(E_{\text{global}}\): To minimize the global constraint, planes which are properly aligned are used to constrain other planes. For each plane \(P_i\), the following measurement is computed:

\[
\Phi(i) = \varepsilon_i + \lambda * \varphi(i),
\]

(6.9)
Algorithm 2: Minimization

**Input:**
Planes: $P_1, P_2, ..., P_m$
parameters for each plane: $T_1, T_2, ..., T_m$

**while** no converge or not reaching maximum iteration **do**

- Minimize the energy $E_{global}$
- Re-segment the plane with the largest residual
- Minimize Energy $E_{data}$
- Re-merge planes with similar parameters

**Output:**
Planes: $P_1, P_2, ..., P_n$
parameters for each plane: $T_1, T_2, ..., T_n$.

where $\varepsilon_i$ is defined by eq. 6.1 and $\varphi(i)$ is the distance between one plane and its neighboring planes:

$$
\varphi(i) = \sum_{x \in B_i} |W(T_{\pi(x)}, x) - W(T_{\xi}(x), x)|^2
$$

(6.10)

If $\Phi(i)$ of plane $P_i$ is larger than that of its neighboring plane $P_j$, all boundary points between these two planes are active boundary points of plane $P_i$ and inactive boundary points of plane $P_j$.

For plane $P_i$ and the group of corresponding active boundary points $A_i \subset B_i$, points $A_i$ are warped using the parameters of plane $P_i$ and the parameters of its neighboring planes to obtain two groups of warped points $C_1(i)$ and $C_2(i)$ respectively, as shown in Figure 6.4. The distance of these two groups of warped points needs to be minimized. In order to achieve this, an affine model is fitted between these two groups of warped points and the parameters of plane $P_i$ are updated:

$$
T_i \leftarrow H_g(i) * T_i,
$$

(6.11)

where $H_g(i)$ is a $3 \times 3$ matrix by fitting an affine transformation model between $C_1(i)$ and $C_2(i)$. For robustness, the parameter of a plane is updated only if the ratio of active boundary points is larger than a threshold (e.g. 0.7 in our experiments). To ensure that the aligned regions are not distorted by other regions, the planes are considered in the ranking order of the number of their active boundary points.

2) **Re-segment a plane with the largest residual:** A region is matched by computing the homography model if the region contains only one plane. However, larger regions may be constituted by more than one plane. Therefore, we use a matching residual to measure the matching score. The regions with the highest matching residual are segmented into smaller ones. The matching metric for each plane is defined by:

$$
\varepsilon_m(i) = \varepsilon_r(i) * \tau(P_i).
$$

(6.12)

$\varepsilon_r(i)$ is defined by eq. 6.1. $\tau(P_i)$ is the ratio between the area of plane $P_i$ and the size of the entire image.
6.3. Image Registration by Piecewise Segment Alignment

Figure 6.4: Minimizing $E_{global}$ for each plane. (a) Active boundary points. Given that the matching residual of the ground plane is larger than that of all its neighbors, all the boundary points are active boundary points for this plane. (b) Top image: the green points are transformed using their own parameters and the red points are transformed using the parameters of their neighbors. (b) Bottom image: result after fitting an affine model.

Given that a large pool of segments is generated during the initialization step, for the purpose of efficiency, one segment is selected from the pool to generate the new region. To select the new region, each segment is assigned a score:

$$\omega(j) = \exp\left(-\left(\frac{\text{sum}(P_i \cap S_j)}{\text{sum}(P_i)} - 0.5\right)^2\right),$$

(6.13)

where \(\text{sum}(P_i \cap S_j)\) is the size of the overlapping area of plane \(P_i\) and segment \(S_j\). \(\text{sum}(P_i)\) is the area of plane \(P_i\). Suppose a segment \(S_{max}\) which has the highest score is obtained, then plane \(P_i\) is segmented into two planes \(P_i \cap S_{max}\) and \(P_i - (P_i \cap S_{max})\).

3) Minimize the energy \(E_{data}\): the data term of each plane is minimized individually. For each plane \(P_i\), its corresponding region \(F_i\) is obtained in the target image by warping \(P_i\) employing the parameters computed in the previous iteration. By warping region \(F_i\) using the inverse transformation \(T_i^{-1}\) of plane \(P_i\), we obtain a region \(G_i\). Subsequently, a flow field is estimated by using the method of [11]. After the flow field is obtained, a set of corresponding points is generated for fitting an affine model \(H_d(i)\) using RANSAC [47]. The new parameters for plane \(P_i\) are defined by:

$$T_i \leftarrow T_i \ast H_d(i).$$

(6.14)

To provide robustness for the flow field, both planes \(P_i\) and region \(G_i\) are expanded by 20% in our experiment before computing the flow. Only the flow inside the original plane \(P_i\) is used to fit the affine model.

4) Re-merge planes with similar parameters: planes which have similar parameters are merged. Suppose there are two neighboring planes \(P_i, P_j\), the two planes are warped using the corresponding parameters \(T_i\) and \(T_j\) respectively resulting in two sets of warped coordinates \(K_i\) and \(K_j\). The parameters of the two planes are then exchanged to obtain
another two sets of warped coordinates $K_i', K_j'$. If the average distance between $K_i'$ and $K_i$ and the average distance between $K_j'$ and $K_j$ are both smaller than a threshold (e.g. 2 pixels in our experiment), planes $P_i$ and $P_j$ are merged and the parameters are updated by the parameters of the joint plane of $P_i$ and $P_j$. To avoid infinite loops, planes which were previously segmented are not merged. Planes are considered in order of their corresponding energy values (eq. 6.6).

6.4 Experiments

The proposed method is evaluated on two datasets with scales ranging from 0.5 to 2 and viewing angles from 0 to 90 degrees. In section 6.4.1, the datasets and the experimental setup are discussed. Section 6.4.2 analyzes the influence of the initialization step. In 6.4.3, the method is compared to the state-of-the-art. Finally, section 6.4.5 presents an application of the proposed method.

6.4.1 Datasets and Experimental Setup

The algorithm is tested on two datasets. The first dataset is obtained by Mikolajczyk et al. [95]. The other dataset is new and created by the authors (publicly available). This is because there is no public available dataset with ground truth annotations for image registration with large viewpoint variations. We have collected 20 pairs of images and manually labeled 100 to 300 corresponding locations for each pair. Figure 6.5 shows an example from our dataset. The selected images are collected from other papers [87, 88]
or from the Internet. These images are selected as they are taken under challenging imaging conditions including large changes in illumination and viewpoint, repeating patterns, moving objects and occlusion. The root-mean-square-error (RMSE) is used to measure the difference between the ground-truth coordinates and the warped coordinates for each location labeled in the images.

In our experiments, we set $\beta = 5$ and $\gamma = 300$ as suggested by [11]. For more details on the optimization of those parameters, please refer [11]. The parameter $\mu$ controls the balance between the descriptor and geometrical information and is set to $\mu = 0.1$. $\alpha$ is set to 1. The maximum iteration number is set to 10. These parameter settings work best in our experiments.

### 6.4.2 Influence of the Initialization

![Figure 6.6: The influence of the initial step on the final result. (a) Two input images to be matched. (b) The performance of our method for different initializations. The x-axis denotes the area size of the initial regions used. The y-axis is the matching error with respect to the ground-truth annotations. The initial error is the matching error after initialization. The final error is the matching error after performing minimization.](image)

We study the influence of the initialization step by changing the number of regions used as a starting point. Specifically, regions are selected with sizes larger than a threshold ranging from 0.01 to 1 (where the image size is normalized to 1). Hence, for a low threshold, the minimization process mainly consists of merging small starting regions into larger ones. For higher thresholds, the method starts with large regions (e.g., the whole image), and the minimization process mainly consists of splitting large starting regions into smaller ones.
As shown in Figure 6.6, the influence of the initialization depends on the image content. For images containing highly discriminative patterns, such as the car in Figure 6.6 (a)-top, the initial step is of minor importance. Even when using a global affine model (i.e., threshold = 1), the method provides high accuracy. On the other hand, for images containing many repeating patterns, such as the floor and wall in Figure 6.6 (b)-bottom, the initial step is of crucial importance. Patterns are aligned piece by piece during the minimization step. Nonetheless, it is more challenging to align regions with only repeating patterns.

6.4.3 Comparison to the State-of-the-Art: Ground Truth Dataset

In this section, the proposed algorithm is compared to four state-of-the-art methods: LDOF [11], SIFT-flow [89], Lin et al’s [88] and PiecewiseAAM [94]. LDOF [11] and SIFT-flow [89] are both non-parametric algorithms. LDOF [11] builds an energy function considering both intensity and texture descriptors. By minimizing an energy function with a smooth constraint, a flow field is obtained for each pixel. SIFT-flow [89] densely extract SIFT features for each pixel. By minimizing the $L_2$ distance in feature space in a coarse to fine manner, the method is able to compute a flow field for matching two images with a large difference in image appearance. Lin et al’s [88] propose a parametric algorithm to match two sets of unmatched key-points. By smoothly varying the deviation of the affine model of each key point by a global affine model, it computes a geometric transformation for each pixel. We also compare our method to the matching algorithm described in [94]. For AAM [94], each image is divided into a set of triangles based on a number of predefined key-points (e.g. eye corner). However, for general images, key-points are not predefined. We firstly use ASIFT [140] to find a set of matched pairs of key-points. Then triangulation is applied on those points. For each triangle, an affine model is obtained by fitting the three vertexes of the triangle. For regions with no key-points detected, the global affine model is used. We call this baseline PiecewiseAAM in our experiments.

In Table 6.1, the average matching error of the different algorithms are shown. We also show the error of a number of image pairs in Figure 6.7. From the results, it can be derived that the performance of LDOF [11] and SIFT-flow [89] is not very robust. For some of the tested images, although the viewpoint change is not very large, LDOF [11] and SIFT-flow [89] provide low accuracy, especially for pairs containing moving objects or repeating patterns. Since LDOF [11] and SIFT-flow [89] are nonparametric methods, these algorithms compute corresponding locations under a smoothness constraint. When there are moving objects or repeating patterns, they may end up in a local minimum. In Lin et al’s [88] method, the performance is reasonable when the viewpoint changes are relatively small. However, the performance decreases when the viewpoint changes drastically. When the viewpoint change is very large, the initial global affine model used in Lin et al’s [88] method is far from the actual matching problem. A smooth change from the global affine may fail to capture the real transformation. Since PiecewiseAAM [94] is designed for specific images (e.g. faces or medical images), it is hard to define the key-points for general images. If one defines the key-points to be Harris corners, then a small number of outliers may result in a large error of the final result. Furthermore, one triangle may span two planar regions. Since the proposed algorithm detects planar regions using figure-ground segmentation and applies an affine model to each planar region, it decreases the chance for regions to span planes. A global constraint is used
Table 6.1: Average error of algorithms. The average root-mean-square-error (RMSE) is used to measure the difference between the ground-truth coordinates and the warped coordinates (Unit: pixel).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>62.61</td>
<td>30.13</td>
<td>13.26</td>
</tr>
<tr>
<td>Methods</td>
<td>PiecewiseAAM [94]</td>
<td>Ours</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>8.30</td>
<td>4.72</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.7: The proposed method is compared to three state-of-the-art algorithms, LDOF [11], SiftFlow [89], Lin et al. [88], on the proposed dataset. For each pair of images, the distances between the ground-truth locations and the warped locations are shown.

to ensure that the regions are tightly connected, so that outliers of matching are reduced. With these two properties, the proposed algorithm outperforms other methods. A number of visual results are shown in Figure 6.8.

To get more insight in the algorithm, we study how the energy value changes with the iterations. Since the two terms in the energy function of eq. 6.8 is minimized in two steps iteratively, the energy may increase in one iteration. The energy values of three pairs of images are shown in Figure 6.9. The energy of each pair is subtracted by the minimum energy value of the pair. Figure 6.9(a) shows the energy value of each iteration. Since there are two steps of minimization for each iteration, we also show the energy after each step of each iteration in Figure 6.9(b). Figure 6.9(a) shows that after each iteration, the energy decreases gradually and remains constant after 10-20 iterations. During each iteration, since the two terms in the energy function eq. 6.8 are minimized iteratively,
Figure 6.8: Top to bottom: reference image, target image, final planes of our method, our warp of target image, overlay of [11], overlay of [89], overlay of Lin et al.[88] and overlay of our method. All overlay images are shown with the warped target image where the green channel is replaced by the reference image.
6.4. Experiments

(a) Total energy of each iteration

(b) Total energy of each step

Figure 6.9: The energy minimization process per iteration. Here, the minimization process of three pairs of images is shown. The energy of each pair is subtracted by the minimum energy value of the pair. Figure (a) shows the energy per iteration. Figure (b) shows the energy after each step of one iteration (since there are two steps per iteration, (b) has double points than (a)).

Table 6.2: Average time needed to align two images for different sizes (Small: 307×204, Middle: 614×408 and Big: 1228×816) (Unit: seconds).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>4</td>
<td>8</td>
<td>323</td>
</tr>
<tr>
<td>Middle</td>
<td>20</td>
<td>33</td>
<td>1742</td>
</tr>
<tr>
<td>Big</td>
<td>75</td>
<td>142</td>
<td>10803</td>
</tr>
<tr>
<td>Methods</td>
<td>PiecewiseAAM [94]</td>
<td>Ours</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>3</td>
<td>201</td>
<td></td>
</tr>
<tr>
<td>Middle</td>
<td>150</td>
<td>917</td>
<td></td>
</tr>
<tr>
<td>Big</td>
<td>161</td>
<td>1256</td>
<td></td>
</tr>
</tbody>
</table>

The energy may increase, as shown in Figure 6.9(b). But this does not influence the final result, to our experience, after 10 iterations, the matching result is visually good enough in general. To guarantee convergence, we indeed use a stop condition in the optimization process. If the energy starts to increase at iteration t, the algorithm is ended and the parameters of iteration t-1 are used. In our experiments, this was the case for 3 out of 20 pairs of images.

The average computation time is shown in Table 6.2. We run the algorithms on a computer containing an Intel Xeon CPU E31270 3.40GHz. The time shown is the average time. The pictorial complexity of the images may influence the matching time. For example, Lin et al. [88] need more time if there are more key-points detected.
6.4.4 Comparison to the State-of-the-Art on the dataset of Mikolajczyk.

We evaluate our method on the dataset of Mikolajczyk et al. [95] and compare our method with Lin et al.’s [88] method. Although we have shown that the advantage of our method is on matching images with large viewpoint changes, the results on this dataset show that our method performs also properly when matching images containing large rotation and scale changes. Some illustrative results are shown in Figure 6.10.

6.4.5 Application

Viewpoint Changing and Image Editing: The application chosen is automatic image editing. Imagine that someone is taking pictures of famous, known scenes. If the person is not satisfied with the viewpoint or the scene, it is possible to modify the picture using the proposed method. To this end, images could be retrieved from the Internet which have been taken from the same scene. Images are then matched by our method. By applying Poisson blending [105], a new image with a different viewpoint and/or a different scene is obtained. The result is shown in Figure 6.11.
Figure 6.11: Changing the viewpoint and editing image using images from the Internet. The top left is an image taken at Mogao Caves. We uploaded this image in the image search engine of Google and obtained another image of the same scene. Using our image matching method, we have aligned these two images and obtained a new image from a different viewpoint.

6.5 Conclusion

A method is proposed for matching two images containing large viewpoint changes. The method exploits different types of information, such as intensity, gradient, texture and epipolar geometry. By relaxing the global geometric transformation through a piecewise alignment approach, the proposed method is able to handle complex camera motions. The proposed method is more flexible than traditional parametric methods for matching images with a large parallax. Our approach can handle scenes with repeating patterns which is a crucial problem for non-parametric methods.

We compared the performance of our method to four other algorithms on two different datasets. The experiments show that our method obtains robust and accurate image alignment and outperforms state-of-the-art methods.