QuadSIFT: Unwrapping Planar Quadrilaterals to Enhance Feature Matching

Marco Filax
Chair of Software Engineering
Otto-von-Guericke University of Magdeburg
marco.filax@ovgu.de

Tim Gonschorek
Chair of Software Engineering
Otto-von-Guericke University of Magdeburg
tim.gonschorek@ovgu.de

Frank Ortmeier
Chair of Software Engineering
Otto-von-Guericke University of Magdeburg
frank.ortmeier@ovgu.de

ABSTRACT

Feature matching is one of the fundamental issues in computer vision. The established methods, however, do not provide reliable results, especially for extreme viewpoint changes. Different approaches have been proposed to lower this hurdle, e. g., by randomly sampling different viewpoints to obtain better results. However, these methods are computationally intensive.

In this paper, we propose an algorithm to enhance image matching under the assumption that an image, taken in man-made environments, typically contains planar, rectangular objects. We use line segments to identify image patches and compute a homography which unwraps the perspective distortion for each patch. The unwrapped image patches are used to detect, describe and match SIFT features.

We evaluate our results on a series of slanted views of a magazine and augmented reality markers. Our results demonstrate, that the proposed algorithm performs well for strong perspective distortions.

Keywords
Projective Transformation, Perspective Distortion, Feature Detection, SIFT

1 INTRODUCTION

Feature matching is one of the fundamental issues in computer vision. It was successfully applied in a variety of applications, e. g., Recognition [18], Tracking [6], Stitching [2] or Visual Odometry [23]. These feature matching algorithms are typically divided into three steps: detection, description, and matching.

There exists a variety of methods to detect features, e. g., Maximally Stable Extremal Regions [8], Scale Invariant Feature Transform (SIFT) [19], or Speeded Up Robust Features [1]. After that, during the description phase, spacial information of features is extracted - SIFT is one of the most popular methods. In the matching step, features descriptors of two images are compared by calculating the distance, e. g., the Euclidean distance for SIFT features. The goal is to find corresponding features

If the viewpoint change is reasonable small, all of the above-mentioned methods typically produce good results. However, if the viewpoint change is large enough, the problem of detecting, describing, and matching features becomes challenging. An example is illustrated in Fig. 1. There, a square S is viewed from two different viewpoints ψ_1 and ψ_2. The resulting images depict two different quadrilaterals: one for every viewpoint. State-of-the-art descriptors do not compensate perspective distortion as the projective transformation, wrapping S_{orig} to S_1 or S_2, is unknown in advance. Morel and Yu [22, 26] demonstrated that the established meth-

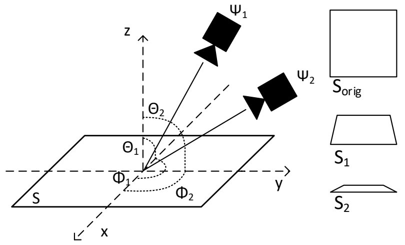


Figure 1: A planar square is viewed from two different viewpoints ψ_1 and ψ_2. S_{orig} represents a frontal view of a square. S_1 and S_2 demonstrate the projection of the square to different viewpoints. The square has been projected to two different quadrilaterals depending on the longitude ϕ and latitude θ values of the view-sphere [26].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ods, e. g., SIFT, do not provide suitable results, if the viewpoint change $|\theta_1 - \theta_2|$ is larger than 60°.

In this paper, we extend the well-known SIFT [19] framework. We increase the capabilities of SIFT by inverting the projective transformation introduced through a slanted view. Therefore, we focus on planar objects in man-made environments. Typically, objects in man-made environments represent some sort of structure, e. g., walls, windows or pictures - these structures form rectangular objects. This is especially the case for indoor environments. We exploit that observation to increase the capabilities of SIFT. We search for quadrilaterals, determined by a set of four non-collinear line segments. Given a valid quadrilateral, we determine the homography, that projects the quadrilateral into a square to unwrap the projective transformation. Finally, we utilize the SIFT framework to find correspondences. We evaluate our algorithm by matching different slanted views of a planar magazine and compare our approach with a well-known affine invariant extension of SIFT [22, 26].

The paper is structured as follows: In Sec. 2 related work of different authors is presented. We describe our approach in Sec. 3 and evaluate its performance in Sec. 4.

2 RELATED WORK

In this section, we summarize existing approaches aiming at detecting features in perspective distorted scenes.

Morel and Yu proposed an affine invariant feature matching approach - an extension of the well-known SIFT approach - called Affine SIFT (ASIFT) [22, 26]. ASIFT is designed to cope with viewpoint changes larger that 60°. To achieve this, different viewpoints of the original images are simulated by sampling the longitude and latitude of the view-hemisphere. SIFT features of two simulated images are matched and the highest amount of matches represents the result.

However, ASIFT is computationally intensive. For every parameter simulation features have to be detected, described and matched. To speed up the simulation, the authors proposed a two-resolution procedure. The parameter simulation is done with two low-resolution versions of the input images. In the case of success, the best set of parameters is used to compute SIFT Features on the original resolution images.

Cai et al. proposed an approach quite similar to ASIFT, called Perspective SIFT (PSIFT) [3]. The authors used a similar two-resolution procedure to increase the efficiency. In contrast to ASIFT, PSIFT unwraps perspective distortion by calculating a homography according to the longitude and latitude parameters of the view-hemisphere. The authors double the amount of parameters that have to be sampled, which additionally increases the computational effort.

To relax with the computational intensity, an iterative approach, iterative SIFT (ISIFT), has been proposed [28]. ISIFT is quite similar to ASIFT: the approach transforms one image to another view, more similar to the other image, by simulating viewpoint changes. First, features in both images are detected, described and matched. A homography is estimated which maps points in one image to points in the other image. Then, a view is simulated, by inverting the homography matrix, such that it is more similar to the other image. Additionally, illumination differences are detected and eliminated. This process is repeated until it converges to a stable amount of matches.

This approach has a significant drawback: its success is based on the initial matching of the two images. If, e. g., because of a strong viewpoint change, the feature matching stage fails, the algorithm is not able to produce reliable results. Thus, other authors used regions to increase the performance of SIFT.

Chen et. al. proposed to extract regions with the popular maximally stable extremal region (MSER) [20] detector and fitted them into ellipses [4]. These ellipses are assumed to be circular in a frontal view. Thus, they are transformed into circular areas according to the ellipses parameters and their second-order moment. Finally, the authors used SIFT to detect, describe and match features in the circular area. However, we believe circular areas are not common in man-made environments. In our opinion, man-made environments are more likely to contain line segments. Many lines in the scene are typically parallel and others are orthogonal [24]. This is satisfied, e. g., for doors, windows, shelves, buildings, and signs. This assumption led us to the approach, proposed in this paper. A key concept is the detection of quadrilaterals.

Different authors tackled the problem of quadrilateral detection. Typically, a set of constraints is used to detect if line segments form a quadrilateral [30, 17, 9]. Leung et al. proposed an algorithm to detect a planar quadrilateral in order to project an image on a squared surface. The authors measured the overlap of a detected line segment and segments of the quadrilateral hypothesis. Further, they imposed different constraints: e. g., opposite sides are of similar lengths or the angles of adjacent line segments are within a specific range. In [30, 9] the authors required opposite sides to be almost parallel. However, these approaches are too restrictive to serve as quadrilaterals detectors for strong viewpoint changes. If the camera moves to a position nearly parallel to the planar rectangle, most of the constraints do not hold, as illustrated in Fig. 4.

3 QUADSIFT: QUADRILATERAL SIFT

Feature detection algorithms have been a subject of scientific research for decades. Although there has been a

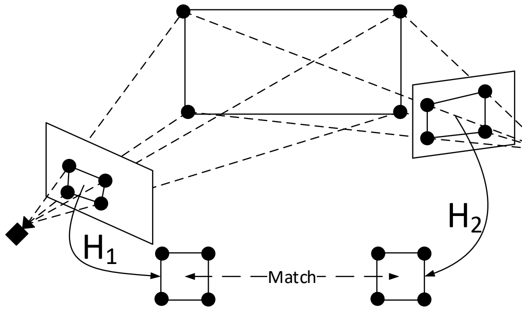


Figure 2: QuadSIFT: we detect quadrilaterals in two slanted views of a planar rectangle to compute two homography matrices that project the quadrilaterals to squared patches and finally detect, describe and match SIFT features.

lot of progress in that area, most feature detection algorithms suffer from the same problem: A strong viewpoint change results in a dramatic decrease of matching performance [22, 26]. This is because existing methods do not adhere to the perspective distortion introduced by a viewpoint change.

In this paper, we propose an extension of the well-known SIFT framework to revert the perspective distortion. To achieve this goal, we rely on additional information: rectangular structures. We make the following assumption to make that possible: The images used to detect, describe, and match features contain man-made objects. Based on the observation that objects in man-made environments typically contain line segments, we can assume that such an object most likely represent some rectangular structure, e. g., windows, walls or pictures. Given a perspective distorted rectangular object, we can revert the perspective distortion by wrapping the quadrilateral to a rectangle. Therefore, we propose an algorithm as illustrated in Fig. 2. We detect perspective distorted rectangles - quadrilaterals - in an image taken by a camera. We estimate the perspective distortion by calculating the homography that wraps a quadrilateral into a rectangular patch. In specific, we wrap the quadrilateral into a square as the aspect ratio of the physical rectangular structure is not known in advance. We detect, describe and match SIFT features in the squared patches.

Different problems have to be solved to reach this goal. First, we need to detect straight, linear contours: line segments. Second, scattered straight lines have to be grouped. Third, a quadrilateral has to be selected and a homography, unwrapping it into a squared patch, has to be computed. Finally, the unwrapped image area can be used to detect, describe and match SIFT features. We tackle every problem individually in the following sections.

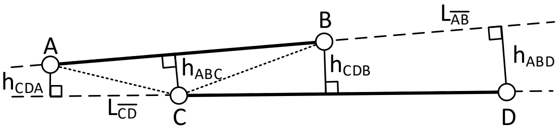


Figure 3: The line segment collinearity distance metric is defined as the absolute sum of the line segment endpoints heights to another line. The sum of all Euclidean distances (heights) $h_{ABC} + h_{ABD} + h_{CDA} + h_{CDB}$ defines the total absolute error.

3.1 Line Segment Detection

In the recent years, the scientific community made a huge progress in line detection algorithms - they became fast and reliable. There exists a variety of algorithms to detect lines and line segments, e. g., Hough Lines [7], probabilistic approaches [16], the Binary Descriptor [29], or Line Segment Detector [10, 11]. Choosing an appropriate detection algorithm is typically up to the specific use-case. Through a preliminary empirical evaluation, we found that the LSD detector yields better results than others with respect to the computation time for the given purpose. Thus, we used its OpenCV implementation to detect line segments longer than 75 pixels.

3.2 Detecting Scattered Line Segments

The detected line segments shall be used to rectify a quadrilateral. Therefore, we need to detect valid quadrilaterals from the set of line segments by selecting four line segments and evaluating their geometrical feasibility. However, if we view a perfectly rectangular object, e. g., a sheet of paper, from a viewpoint nearly collinear to the surface, its surface might be bent and thus, the line segments might be scattered into smaller segments or hidden by other objects.

To gain robustness for a quasi-planar object, we group nearly collinear line segments. We use an agglomerative clustering algorithm [5] to group the line segments. For this bottom-up approach, we need to define a distance metric specifying that two clusters have to be merged together. We use the relaxed collinearity property of two line segments as the distance metric.

We derive the distance metric as follows: Determining the collinearity of two line segments is a boolean operation. Either the line segments are collinear or not. We have to relax the collinearity property to use it as a distance metric, such that it is unaffected by the length, absolute positions and collinear distances of the line segments. Relying on the slope of a line segment does not account to parallel segments. Further, we cannot rely on the convex area defined by four endpoints of two line segments as it is affected by the length of the segments. Instead, we consider every combination of three endpoints as a triangle: If two line segments are nearly collinear, the height of the four triangles, defined by line segments as its base and another line segments endpoint, becomes close to zero. Thus, we sum up the absolute heights of every triangle defined a line segment endpoints and another line segment endpoint.

An example for the proposed distance metric is depicted in Fig. 3. $\overline{AB}$ and $\overline{CD}$ represent two line segments with the endpoints A,B,C,D. $L_{\overline{AB}}$ and $L_{\overline{CD}}$ are the two lines, defined by the finite line segment. The endpoints define four triangles ABC, ABD, CDA and CDB. Note, that every triangle has to have one of the lines $L_{\overline{AB}}$ or $L_{\overline{CD}}$ as its base. The total distance is $dist_{\overline{AB}\,\overline{CD}} = h_{ABC} + h_{ABD} + h_{CDA} + h_{CDB}$ If $dist_{\overline{AB}\,\overline{CD}}$ is suitable small, we consider these line segments as collinear and group them to a single segment.

3.3 Quadrilateral Detection

The next step is to select a valid perspective distorted rectangle - a not self-intersecting convex quadrilateral. A convex quadrilateral is defined by four line segments. However, if we view a perfectly rectangular object under an extreme viewpoint, especially if we position the camera nearly collinear to the surface as shown in Fig. 1, we experience a series of problems.

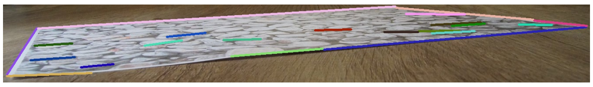


Figure 4: Slanted view of a sheet of paper. Note that, although the sheet is perfectly rectangular, opposite edges are not similar in length. The angles of two adjacent line segments vary strongly. Further, the edges of the paper are scattered into smaller line segments.

First, we detect quadrilaterals in man-made environments, based on line segments. Typically, quadrilaterals are detected with line segments, grouped by their vanishing points [25, 21]. However, if the set of line segments is sparse or includes a large set of outliers, these algorithms yield insufficient results. Given a minimal example, as shown in Fig. 4, vanishing point based grouping algorithms would fail, because there are only four line segments present. Reliably calculating two vanishing points is not possible given four line segments. Thus, we can not rely on vanishing points.

Second, we cannot impose constraints, that are typically used to detect quadrilaterals [15, 17, 12], e. g., opposite segments are of similar length, the angle of two intersecting line segment is within a specific range or two line segments have a similar slope. These conditions do not hold if the camera moves to a position nearly collinear to the planar surface. An example is depicted in Fig. 4.

To account for these difficulties, we evaluate the intersections of the lines, defined by the detected line segments. Other approaches simply evaluate intersection points *within* a line segment or rely on vanishing points. While using intersections within a line segment restricts to rely on perfectly planar objects, detecting two or more vanishing points requires a sufficient amount of line segments. Our algorithm succeeds even for the minimum set of line segments and is robust against bent surfaces. The algorithm to detect valid self-intersecting convex quadrilaterals is summarized as follows: First, select four lines from the set of detected line segments. Second, compute the intersections of the lines to cope with scattered lines segments. Finally, remove outliers if there are more than four intersection points, by evaluating the overlap of line and line segments to eliminate invalid intersection points.

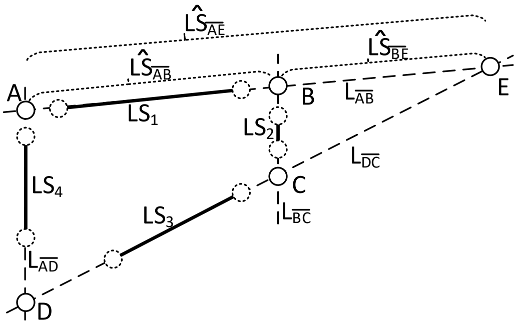


Figure 5: Given four line segments $LS_1,..,LS_4$ in general order: we compute their intersections A,B,C,D,E and eliminate outliers by evaluating the overlap ratio of detected line segments $LS_1,..,LS_4$ and virtual line segments, e. g., $\hat{LS}_{\overline{AB}}$, $\hat{LS}_{\overline{AE}}$, $\hat{LS}_{\overline{BE}}$.

To demonstrate our approach we depict an example in Fig. 5. We selected four line segment $LS_1,..,LS_4$ from the set of all detected line segments. Then, based on their collinear lines $(L_{AB}, L_{BC}, L_{DC}, L_{AD})$, we can calculate the five intersections A, B, C, D and E. It is clearly visible that $ABCD$ represents the valid quadrilateral defined by $LS_1,..,LS_4$. This, however arises a problem: there exist multiple subsets of four intersection points describing a quadrilateral, e. g., $ABEC$. Thus, we have to detect the invalid intersection point E.

Other algorithms try to calculate vanishing points to rectify a given image [25, 21]. Given vanishing points it is possible, to cluster the line segment according to their geometrical feasibility. Then, one can detect valid quadrilaterals. However, as we have a set of exactly four line segments, where every intersection is defined by two lines, e. g., $L_{AB} \times L_{BC} = B$ and $L_{AB} \times L_{DC} = E$, it is not possible to reliably calculate vanishing points. Thus, we evaluate the feasibility of intersection points by maximizing the overlap ratios of detected line segments and their virtual line segments. We will use the example configuration depicted in Fig. 5 to describe the approach.

We identify valid quadrilateral points with the overlapping ratio of a detected line segment and a virtual line segment. Virtual line segments are defined by their intersection points on a line collinear to a detected line segment and other lines. We exemplary depict the three virtual line segments for LS_1 in Fig. 5 $\hat{LS}_{\overline{AB}}$, $\hat{LS}_{\overline{AE}}$ and $\hat{LS}_{\overline{BE}}$. It is visible that $\hat{LS}_{\overline{AB}}$ and $\hat{LS}_{\overline{AE}}$ overlap LS_1 whereas $\hat{LS}_{\overline{BE}}$ does share an overlapping region with LS_1. Thus, $\hat{LS}_{\overline{BE}}$ does not represent a candidate. Further, we observe the order $O_{LS_1,\hat{LS}_{\overline{AB}}} > O_{LS_1,\hat{LS}_{\overline{AE}}}$ whereas $O_{LS,\hat{LS}}$ is the overlap ratio. This indicates that $\hat{LS}_{\overline{AB}}$ with its intersection points A and B represent a possible quadrilateral candidate. Four candidates are required to detect a quadrilateral. For the given example we found that $\hat{LS}_{\overline{AB}}$, $\hat{LS}_{\overline{BC}}$, $\hat{LS}_{\overline{CD}}$ and $\hat{LS}_{\overline{AD}}$ with the points A, B, C and D share the largest overlap with $LS_1,...,LS_4$ and thus, $ABCD$ describes the valid quadrilateral.

3.4 Homography Estimation

In order to unwrap an image, we need to estimate a homography that transforms a given set of at least four points to into another set of points [13]. Given a valid quadrilateral in an image, defined by four points, we need to find four rectangular points.

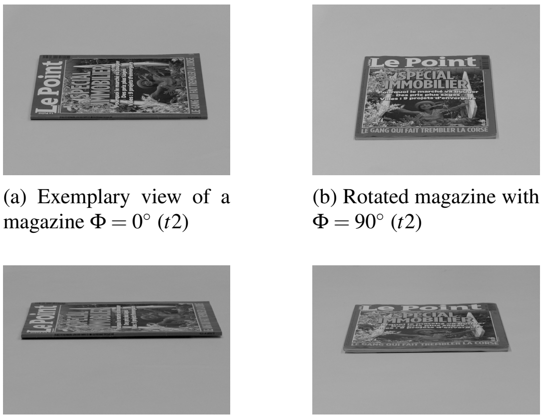
To determine four rectangular points with the help of a quadrilateral, Hua et al. [14] estimated a rectangle, whereas the aspect ratio of the physical rectangle is known. However, it is impossible to determine the aspect ratio for a quadrangle if the physical rectangle is unknown in advance. It is not possible to estimate the physical aspect ratio directly from a single image. The measured aspect ratio in an image can differ from the physical aspect ratio, especially in slanted views.

To overcome this issue, we chose to unwrap a detected quadrangle to a square. This eliminates the need of detecting the aspect ratio of an arbitrary quadrangle. We empirically chose to unwrap quadrangles into squares with 500-pixel length on each side.

Given four corner points of a detected quadrilateral and four corner points of a square, we calculate the homography mapping the quadrilateral to a square [13]. We use SIFT [19] to detect and describe features in two squared patches. We employ a brute force matching strategy [19]. To detect weather a descriptor in one squared patch matches some descriptor in the other squared patch we apply Lowe's ratio test: If the nearest distance of the best match for a descriptor is smaller than k times the second best match for that descriptor, the best match is considered to be valid, with k=0.8. Finally, we remove duplicated matches.

4 EVALUATION

We evaluate our method with the test set used by Morel and Yu [22]. We chose this test set because it is commonly used by other works in the domain. The test set consists of different slanted and zoomed views of a magazine and other objects. As an evaluation on scale invariance of SIFT is not in the scope of this paper, we omit the zoomed views. We focus on the different slanted views of a magazine - resulting in two different test sets called $t2$ and $t4$. Fig. 6 depicts examples from the database. We determine the quality of our approach by determining the total number of feature correspondences found with the SIFT framework. The better our framework corrected the perspective distortion introduced through a viewpoint change the higher is the total number of correspondences. We compare the proposed approach with a well-known affine invariant extension of SIFT: ASIFT [22][1].



(a) Exemplary view of a magazine $\Phi = 0°$ ($t2$)

(b) Rotated magazine with $\Phi = 90°$ ($t2$)

(c) Exemplary view of a magazine $\Phi = 0°(t4)$

(d) Rotated magazine with $\Phi = 90°$ ($t4$)

Figure 6: Examples from the database: (a) depicts a slanted view of a magazine from $t2$. In (b) the magazine was rotated. (a) and (b) shall be matched later - SIFT is unable to detect matching features. (c) depicts a view of the magazine from $t4$. In contrast to (a), the camera was moved nearly collinear to the planar object. (d) depicts a rotated version of (c). (c) and (d) shall be matched later. Again, SIFT is unable to detect matching features.

We implemented our method, described in Sec. 3, with OpenCV in C++. We used the OpenCV implementation of the Line Segment Detector [11] to find line segments in the images. With these segments, we detect quadrilaterals as described in Sec. 3.3. We used a greedy strategy to select a single quadrilateral by maximizing the quadrilateral area in an image. Then, we unwrapped the perspective distortion, introduced through a slanted view, by mapping the detected quadrilateral to a squared patch. We detect, describe and match SIFT features in the squares. We used a brute force matching algorithm, an outlier suppression proposed by Lowe [19] with k=0.8 and eliminated duplicated

[1] Implementation available online: `http://demo.ipol.im/demo/my_affine_sift/`

Φ	SIFT	ASIFT	QuadSIFT
10°	284	3496	1457
20°	22	2185	1399
30°	0	1729	1339
40°	0	1264	1215
50°	0	964	1142
60°	0	961	1054
70°	0	921	984
80°	0	879	918
90°	0	789	933

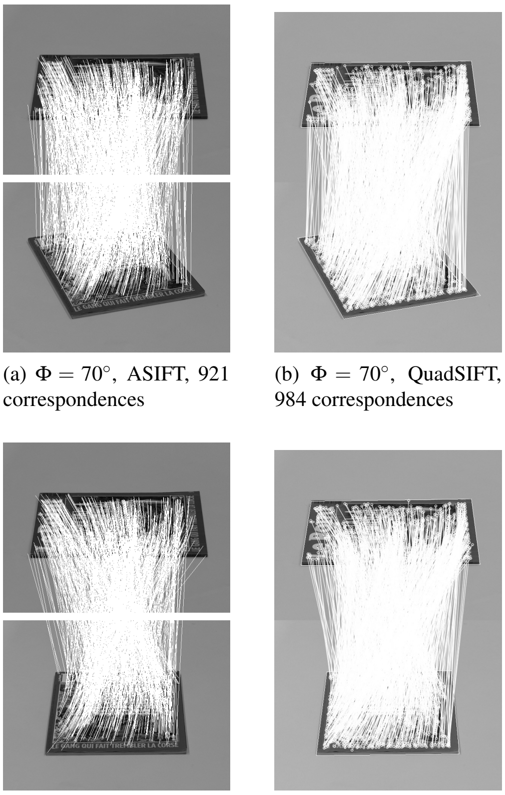
Table 1: Evaluation based on the test set $t2$ taken from [22]. Every row represents another rotation of the magazine denoted by Φ. We depict the number of correspondences when matched with Fig. 6(a) for every approach.

matches. Finally, we wrapped the matches to the original images and displayed the results.

We evaluate our algorithm by matching slanted views of a magazine. Examples of the test set $t2$ are depicted in Fig. 7. Table 1 depicts the result for test set $t2$. The first column depicts the rotation angle Φ of the magazine - every rotated magazine is matched against Fig. 6(a) with $\Phi = 0°$. The second column presents the total matches with the SIFT algorithm. The third column depicts ASIFT results. The last column depicts the results of QuadSIFT.

QuadSIFT is more efficient: QuadSIFT requires six seconds per image for the given test set on average. ASIFT required nearly ten seconds on average. For a rotation of the magazine from $\Phi = 10°$ to $\Phi = 30°$ ASIFT provides better results than QuadSIFT. This observation is present in every test set evaluated in this paper. This is because the implementations of ASIFT and QuadSIFT differ. While Morel and Yu used a pure C++ implementation of SIFT, we used the implementation available in OpenCV. If the rotation of the magazine is large enough QuadSIFT produces slightly more correspondences than ASIFT. In combination with the first observation, this means that QuadSIFT produces better results than ASIFT, if the perspective distortion is large enough.

The test set $t4$ is quite similar to $t2$. The major difference is that the camera moved to a viewpoint nearly collinear to the magazine plane - the perspective distortion is higher than the distortion in the test set $t2$. Examples from the test set are depicted in Fig. 6 (c) and (d). The results of the tests are shown in Tab. 2 and examples in are shown Fig. 8. Starting from $\Phi = 40°$ the number of correspondences calculated with QuadSIFT is higher than those calculated with ASIFT. Taking the differing implementations into account, this means, that QuadSIFT outperformed ASIFT. Again, ASIFT produces better results than QuadSIFT if the rotation angle is small.



(a) $\Phi = 70°$, ASIFT, 921 correspondences

(b) $\Phi = 70°$, QuadSIFT, 984 correspondences

(c) $\Phi = 90°$, ASIFT, 789 correspondences

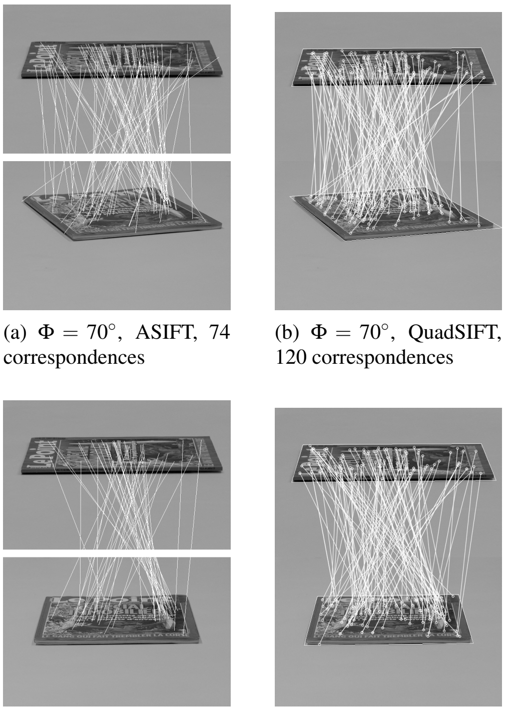
(d) $\Phi = 90°$, QuadSIFT, 933 correspondences

Figure 7: Evaluation based on the dataset $t2$ from [22]. Every row depicts the matching results of Fig. 6(a) and a rotated magazine. SIFT (not shown) found 0 correspondences.

Finding nearly twice as many correspondences, for the two images of the magazine with a rotation of $\Phi = 90°$, we gain the following observation: If the distorted object is of a nearly quadratic shape, we achieve better results. We believe, that this observation is clarified if we consider the aspect ratio of the planar object. While ASIFT does not directly affect the aspect ratio of the

Φ	SIFT	ASIFT	QuadSIFT
10°	15	1079	380
20°	0	589	341
30°	0	310	293
40°	0	232	262
50°	0	130	198
60°	0	98	150
70°	0	74	120
80°	0	70	106
90°	0	43	104

Table 2: Evaluation based on the test set $t4$ taken from [22]. Every row represents another rotation of the magazine denoted by Φ. We depict the number of correspondences when matched with Fig. 6(c) for every approach.



(a) $\Phi = 70°$, ASIFT, 74 correspondences

(b) $\Phi = 70°$, QuadSIFT, 120 correspondences

(c) $\Phi = 90°$, ASIFT, 43 correspondences

(d) $\Phi = 90°$, QuadSIFT, 104 correspondencess

Figure 8: The matching results of Fig. 6(c) and a rotated magazine from testset $t4$. SIFT (not shown) did not produce correspondences.

planar object, our approach is designed to map an object into a square. If we now consider quadratic objects captured by the camera, e. g., augmented reality markers, this drawback is negligible. If we consider non-quadratic objects, we can overcome this drawback if the aspect ratio is known in advance.

To clarify these claims, we evaluate our approach on an additional test set. We took ten images of a squared marker in a similar manner as Morel and Yu. We rotated the marker in between every image by 10°. In a similar manner, we took ten images of a rectangular marker with an aspect ratio of 3:1. We executed the proposed approach on both markers whereas the aspect ratio of the objects is unknown. The results are depicted in Tab. 3 and Tab. 4. Additionally, we used our approach with a known aspect ratio on the rectangular marker. We compared the results with ASIFT.

ϕ	SIFT	ASIFT	QuadSIFT
10°	498	4712	2681
20°	201	3778	2563
30°	23	2710	2418
40°	0	2424	2307
50°	0	2202	2189
60°	0	2080	2121
70°	0	1832	2073
80°	0	1701	2076
90°	0	1711	2021

Table 3: Evaluation based on a squared marker: Φ denotes the rotation of the marker. The markers are matched to $\Phi = 0$ (Shown in Fig. 9(a)). We depict the number of correspondences calculated with SIFT, ASIFT, and our approach.

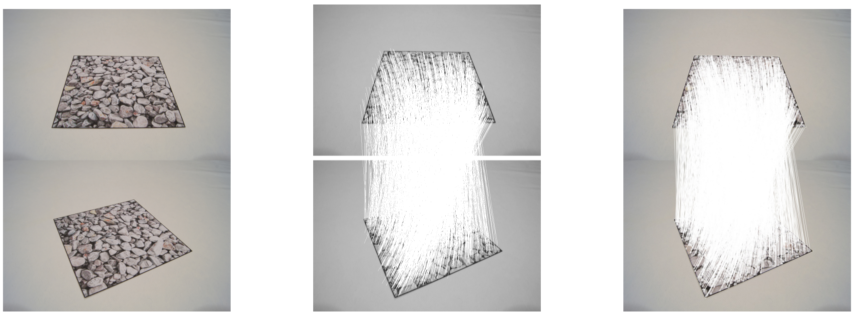
ϕ	ASIFT	QuadSIFT (1:1)	QuadSIFT (3:1)
10°	5450	1616	2469
20°	3702	1487	2410
30°	2382	1270	2145
40°	2292	1239	2065
50°	1885	1266	2006
60°	1530	1230	1827
70°	1379	1260	1815
80°	1257	1197	1641
90°	1348	1267	1727

Table 4: Evaluation based on a rectangular marker: Φ denotes the rotation of the marker. Further, we depict the number of correspondences calculated with SIFT, ASIFT, and the proposed approach with unknown (we assume an aspect ratio of 1:1) and a known aspect ratio (here 3:1).

Tab. 3 depicts the number of correspondences calculated with SIFT, ASIFT, and QuadSIFT for the squared marker. We depicted an example in Fig. 9. It is clearly visible, that QuadSIFT produces better results than ASIFT. Comparing these results to the results of the test set $t4$, we come to the following observation: Increasing the rotation of the non-squared magazine from $\Phi = 0°$ to $\Phi = 90°$ decreases the number of found correspondences by 73%. Increasing the rotation of the squared marker from $\Phi = 0°$ to $\Phi = 90°$ decreases the number of found correspondences by only 25%. This is because of the squared shape of the marker. If the planar objected is close to a physical square our algorithm is superior.

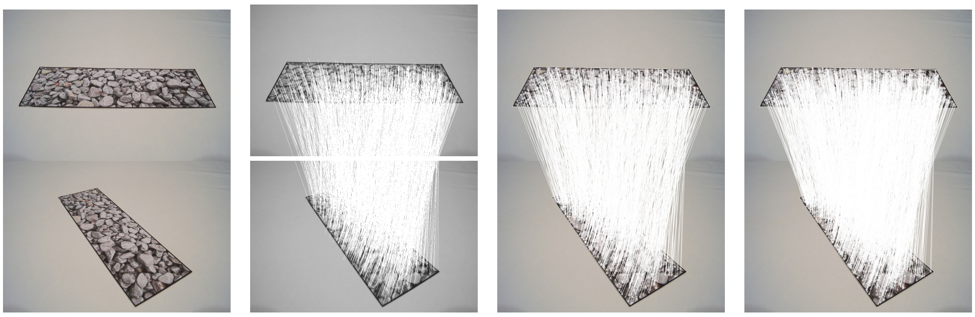
To determine the influence of the aspect ratio, we calculated the absolute number of correspondences of a non-squared marker with known and unknown aspect ratio. The results are depicted in Tab. 4. Φ denotes the amount of rotation applied to the marker in between two images. The second column depicts the absolute number of correspondences calculated with ASIFT. The third and the fourth column depict the results of the proposed approach, once with an unknown aspect ratio - mapping the quadrilateral to square - (cf. third column) and one with known aspect ratio - mapping the quadrilateral into a 500x1500px rectangle (cf. fourth column). It is visible, that if the aspect ratio of the object is known in advance, the results of QuadSIFT are superior if the viewpoint change is large. However, if the aspect ratio is unknown in advance and the object is not of a squared shape, ASIFT produces slightly better results.

Based on the different tests evaluated in this paper we can formulate the following observations. Detecting



(a) A rotated, squared marker with $\Phi = 60°$ (b) ASIFT, 2080 Correspondences found (c) QuadSIFT, 2121 Correspondences found

Figure 9: A comparison of ASIFT and QuadSIFT using a squared object. QuadSIFT produces more correspondences as ASIFT.



(a) A rotated, rectangular marker with $\Phi = 60°$ (b) 1530 ASIFT Correspondences (c) QuadSIFT (1:1), 1230 Correspondences (d) QuadSIFT (3:1), 1827 Correspondences

Figure 10: A comparison of ASIFT and QuadSIFT using a rectangular object. QuadSIFT produces more correspondences as ASIFT if the aspect ratio is known in advance.

quadrilateral objects to unwrap the perspective distortion enhanced feature matching results for extreme viewpoints. QuadSIFT performs similarly to ASIFT if the viewpoint change is reasonably large. If the viewpoint change is small, ASIFT outperforms QuadSIFT in terms of found correspondences. If perspective distortion is large enough QuadSIFT outperforms ASIFT. QuadSIFT seems to be more efficient than ASIFT. Knowing the aspect ratio in advance is beneficial. If the aspect ratio is unknown QuadSIFT performs similarly to ASIFT.

5 CONCLUSION

In this paper, we proposed a new method to enhance feature matching in slanted views of planar rectangular objects. Planar rectangular objects are transformed into quadrilaterals due to the perspective distortion introduced by a slanted view. We designed a method to revert the perspective distortion: Based on line segments, we detected valid quadrilaterals, select a single quadrilateral, and unwrapped the perspective distortion by computing a homography mapping a quadrilateral to a square. Finally, we detect, describe and match features using SIFT. We want to point out, that the feature extraction method is not restricted to SIFT. QuadSIFT can be realized with other feature matching frameworks.

As shown in the evaluation, our algorithm is capable of detecting, describing and matching features, especially in slanted views of planar rectangular objects. We found that QuadSIFT produces results comparable to the state-of-the-art algorithm ASIFT. If the physical, planar object is of a squared form, our approach is superior. Further, our results indicate that QuadSIFT is more efficient than ASIFT.

The proposed approach is designed to detect quadrilaterals. If an image does not contain line segments, e. g., in nonman-made environments, it will fail. Further, we restricted the quadrilateral selection to a greedy approach. We always select the largest quadrilateral. However, we plan to extend the proposed approach: we want to detect multiple quadrilaterals. This is especially interesting if there are multiple planar objects on different planes, e. g., in a corridor with planar objects on the walls or floor. Thus, we plan to incorporate a better quadrilateral detection, e. g., using an approach as described in [27].

6 REFERENCES

[1] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Comput.

Vis. Image Underst. 110(3), 346–359 (2008)

[2] Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. In: Int. J. Comput. Vis. vol. 74, pp. 59–73. Kluwer Academic Publishers-Plenum Publishers (2007)

[3] Cai, G.R., Jodoin, P.M., Li, S.Z., Wu, Y.D., Su, S.Z., Huang, Z.K.: Perspective-SIFT: An efficient tool for low-altitude remote sensing image registration. Signal Processing 93(11), 3088–3110 (2013)

[4] Chen, M., Shao, Z., Li, D., Liu, J.: Invariant matching method for different viewpoint angle images. Appl. Opt. 52(1), 96–104 (2013)

[5] Chidananda Gowda, K., Krishna, G.: Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern Recognit. 10(2), 105–112 (1978)

[6] Donoser, M., Riemenschneider, H., Bischof, H.: Shape guided Maximally Stable Extremal Region (MSER) tracking. In: Proc. - Int. Conf. Pattern Recognit. pp. 1800–1803 (2010)

[7] Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM 15(1), 11–15 (1972)

[8] Extremal, M.S., Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust Wide Baseline Stereo from. Br. Mach. Vis. Conf. pp. 384–393 (2002)

[9] Fung, H.K., Wong, K.H.: A Robust Line Tracking Method based on a Multiple Model Kalman Filter Model for Mobile Projector Systems. Procedia Technol. 11(2), 996–1002 (2013)

[10] Grompone Von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. 32(4), 722–732 (2010)

[11] Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a Line Segment Detector. Image Process. Line 2, 35–55 (2012)

[12] Hartl, A., Reitmayr, G.: Rectangular Target Extraction for Mobile Augmented Reality Applications. IEEE Int. Conf. Pattern Recognit. pp. 81–84 (2012)

[13] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)

[14] Hua, G., Liu, Z., Zhang, Z., Wu, Y.: Automatic Business Card Scanning with a Camera. 2006 Int. Conf. Image Process. pp. 373–376 (2006)

[15] Jagannathan, L., Jawahar, C.: Perspective Correction Methods for Camera Based Document Analysis. Proc. First Int. Work. Camera-based Doc. Anal. Recognit. pp. 148–154 (2005)

[16] Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic Hough transform. Pattern Recognit. 24(4), 303–316 (1991)

[17] Leung, M.C., Lee, K.K., Wong, K.H., Chang, M.M.Y.: A Projector-based Movable Hand-held Display System. CVPR (2009)

[18] Lowe, D.G.: Object recognition from local scale-invariant features. Proc. Seventh IEEE Int. Conf. Comput. Vis. 2([8), 1150–1157 (1999)

[19] Lowe, D.G.: Distinctive image features from scale invariant keypoints. Int'l J. Comput. Vis. 60, 91–11020042 (2004)

[20] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing 22(10), 761–767 (2004)

[21] Mičušík, B., Wildenauer, H., Vincze, M.: Towards detection of orthogonal planes in monocular images of indoor environments (2008)

[22] Morel, J.M., Yu, G.: ASIFT: A New Framework for Fully Affine Invariant Image Comparison. SIAM J. Imaging Sci. 2(2), 438–469 (2009)

[23] Nister, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. J. F. Robot. 23(1), 3–20 (2006)

[24] Rother, C.: A new approach for vanishing point detection in architectual environments. Proc. BMVC. 20(9), 382–391 (2000)

[25] Shaw, D., Barnes, N.: Perspective Rectangle Detection. Perspect. Rectangle Detect. pp. 1–152 (2006)

[26] Yu, G., Morel, J.: A fully affine invariant image comparison method. Int. Conf. Acoust. Speech Signal Process. 26(1), 1597–1600 (2009)

[27] Yu, S.X., Zhang, H., Malik, J.: Inferring spatial layout from a single image via depth-ordered grouping. In: 2008 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. (2008)

[28] Yu, Y., Member, S.S., Huang, K., Member, S.S., Chen, W., Member, S.S., Tan, T.: A novel algorithm for view and illumination invariant image matching. IEEE Trans. Image Process. 21(1), 229–40 (2012)

[29] Zhang, L., Koch, R.: An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. J. Vis. Commun. Image Represent. 24(7), 794–805 (2013)

[30] Zhang, Z., Wu, Y., Shan, Y., Shafer, S.: Visual panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. Proc. 2001 Work. Percetive user interfaces (2001)