# Leveraging 3D city models for rotation invariant place-of-interest recognition

Int J Comput Vis (2012) 96:315–334
DOI 10.1007/s11263-011-0458-7

# Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition

**Georges Baatz · Kevin Köser · David Chen · Radek Grzeszczuk · Marc Pollefeys**

Received: 15 October 2010 / Accepted: 5 May 2011 / Published online: 27 May 2011


**Abstract** Given a cell phone image of a building we address the problem of place-of-interest recognition in urban scenarios. Here, we go beyond what has been shown in earlier approaches by exploiting the nowadays often available 3D building information (e.g. from extruded floor plans) and massive street-level image data for database creation. Exploiting vanishing points in query images and thus fully removing 3D rotation from the recognition problem allows then to simplify the feature invariance to a purely homothetic problem, which we show enables more discriminative power in feature descriptors than classical SIFT. We rerank visual word based document queries using a fast stratified homothetic verification that in most cases boosts the correct document to top positions if it was in the short list. Since we exploit 3D building information, the approach finally outputs the camera pose in real world coordinates ready for augmenting the cell phone image with virtual 3D information. The whole system is demonstrated to outperform traditional approaches on city scale experiments for different sources of street-level image data and a challenging set of cell phone images.

**Keywords** Location recognition

G. Baatz (✉) · K. Köser · M. Pollefeys
Department of Computer Science, ETH Zurich, Zurich, Switzerland
e-mail: gbaatz@inf.ethz.ch

K. Köser
e-mail: kevin.koeser@inf.ethz.ch

M. Pollefeys
e-mail: marc.pollefeys@inf.ethz.ch

D. Chen
Department of Electrical Engineering, Stanford University, Stanford, CA, USA
e-mail: dmchen@stanford.edu

R. Grzeszczuk
Nokia Research at Palo Alto, Palo Alto, CA, USA
e-mail: radek.grzeszczuk@nokia.com

## 1 Introduction

In recent years, due to the ubiquitousness of cell phones and cameras, the demand for real-time localization and augmentation of virtual (3D) information arose and several systems have been proposed to solve the location recognition problem (Robertson and Cipolla 2004; Schindler et al. 2007; Wu et al. 2008b; Irschara et al. 2009; Zhang and Kosecka 2006; Zhu et al. 2008; Cao and McDonald 2009; Zamir and Shah 2010; Knopp et al. 2010) or the closely related image retrieval problem (Sivic and Zisserman 2003; Nistér and Stewénius 2006; Jegou et al. 2008; Philbin et al. 2007; Perdoch et al. 2009). A commonly used scheme that we also follow extracts local features (e.g. Lowe 2004; Bay et al. 2008) from a collection of reference images, vector-quantizes the feature descriptors to visual words and stores images as documents of these words in a database. Then for a query image techniques from web text search are applied to find the closest documents in the database, followed by a reranking of the result list based on geometric constraints.

We specifically look at the problem of place-of-interest recognition and camera pose estimation in urban scenarios, where we want to see how far we can get with visual information only. However, in contrast to general object recognition or image retrieval scenarios that cannot assume much about geometry and image content, we propose a tailored solution to the localization problem from cell phone images in a city (see Fig. 1). Here, often


**Fig. 1** A cell phone picture is taken to localize the user and to retrieve point-of-interest information from the database. We assume to know the camera's focal length and that we can identify the vertical and horizontal facade direction by means of vanishing points to allow for rotation invariant recognition

- massive amounts of calibrated street level data are available for training[1]
- rough 3D city models exist[2]
- facades are planar and structures are vertically and horizontally aligned
- the camera's focal length is known approximately

By projecting the offline training views to the facades' surfaces, we can completely factorize out rotation from the recognition problem (in photometric matching and geometric verification). This enables the storage of gravity-aligned orthophotos (facade parts) in the database as opposed to densely sampling the space of all possible viewing poses. Query images can be transformed accordingly by finding the vertical and horizontal vanishing points of the given building. For recognition, matching and verification this reduces the problem to finding purely homothetic transformations, i.e. a scale and 2D offset on the building's surface. We show that this increases the discriminative power as compared to previous approaches on the one hand and allows to replace the computationally expensive verification of estimating an affine model or a homography now by simply estimating just three position related parameters. For this we propose a novel stratified homothetic parameter estimation, i.e. we perform three subsequent 1D estimates for distance, horizontal and vertical offset with respect to the building surface. Here the algorithm was designed in a way that e.g. window-to-window matches support the correct distance estimate through their scale ratio even if the match is from a different window instance on the facade's window grid. After having obtained the distance from the facade, horizontal and vertical offsets can be computed in the same way and we observe that using this reranking strategy is very effective in boosting the correct document to the first positions of the tested short list. As a side effect, we obtain the 6 DOF camera pose in world coordinates.

The key novel contributions are the orthophoto representation in the database allowing also for a more discriminative feature descriptor (upright SIFT), the fast homothetic verification scheme and the exploitation of 3D building geometry so as to provide an absolute camera pose. While these have been sketched already in a preliminary version presented at a conference (Baatz et al. 2010), here we provide a more detailed presentation of the voting scheme, add extensive evaluations on the homothetic transform and parameter estimation as well as we provide many examples and analyze the failure cases. In the next section we will relate the approach to previous work, before we go into details of the overall system and demonstrate its performance on different sources of cell phone and street level data.

## 2 Previous Work

Location recognition at the city scale is closely related to image search and large scale object recognition for which a huge amount of previous work exist. A commonly used approach builds on top of the bag-of-features approach of Sivic and Zisserman (2003) and the vocabulary trees (VT) of Nistér and Stewénius (2006). In the image retrieval scenario, usually the camera intrinsics and object geometry are unknown. It can therefore be difficult to find strong geometrical constraints for filtering the initial visual-word based results, although recent approaches look at (locally) consistent orientations and feature shapes (Jegou et al. 2008; Philbin et al. 2007; Perdoch et al. 2009) and exploit the fact that pictures are usually not taken upside down. Location recognition approaches (Zhu et al. 2008; Zhang and Kosecka 2006; Irschara et al. 2009) usually know the intrinsic parameters of the camera, but do not exploit dense 3D models of the scene since these are difficult to obtain for larger environments.

The closest earlier works to ours are probably by Robertson and Cipolla (2004), Wu et al. (2008b) and Schindler et al. (2007). The first one uses vanishing points, but works purely in 2D with local patch matching on a relatively small set of images (<100) and does not obtain 6 DOF pose in the city coordinate system since 3D information is missing. The concept of vanishing point rectification in general is well

[1]Nowadays several sources of image data taken from vehicles exist, e.g. Google's "Street View" or Microsoft's "Streetside". We use Earthmine's "3D street level imagery" for database creation and Navteq's "Enhanced 3D City Models" for testing.

[2]In this contribution we use extruded building outlines from Sanborn data (cf. to http://www.sanborn.com/products/citysets.asp), but such information is also available from OpenStreetMap (cf. to http://www.openstreetmap.org).

**Fig. 2** *Left*: Panoramic image near the San Francisco Ferry Building grabbed by Vehicle. *Right*: Extruded building outline of Ferry Building

known, e.g. rectifying features according to vanishing points has been presented recently and independently of our work in Cao and McDonald (2009), where the authors however focused on single images. Exploiting 3D geometry on the other hand has been proposed in Köser and Koch (2007) and Wu et al. (2008a), however these approaches require depth information for both images to be matched. Building on top of that, Wu et al. (2008b) uses 3D information from local reconstructions of streets of houses for database creation, but can only handle query images taken at fronto-parallel perspective relative to the building and cannot cope with out-of-plane rotations. In the field of systems using image data only Schindler et al. (2007) presented a large scale recognition system with impressive results also based upon a vocabulary tree. However, only 2D image data is used and in our experiments we show that in urban scenarios with mainly building facades, 3D rotation invariant matching and recognition outperforms 2D methods.

While the trend in the last years went towards building increasingly larger databases and generating even synthetic views to densely sample the space of all different viewpoints (Irschara et al. 2009), we go into a different direction and represent only the building facades (upright orthophotos). Also related to this Schindler et al. (2008) worked on detecting repeated elements on planar facades. The goal was however not large scale recognition but to extract periodic textures and to obtain the lattice parameters by clustering descriptors. In contrast, our work does not rely on repeated patterns but rather uses vanishing points and building geometry to rectify the image data. An interesting effect of this rectification is that it enables the usage of upright features, for which the feature orientation is obtained from vertical building axes, avoiding multiple descriptors for the same keypoint, avoiding potential bias of standard SIFT descriptors towards the bins of canonical orientations and allows distinguishing local structures differing by rotation. It has already been observed in face recognition (Dreuw et al. 2009) that exploiting the knowledge of aligned patches and reducing the invariance requirements can increase the recognition performance. Already for the SURF detector (Bay et al. 2008), rotation invariance could be disabled, however this was mainly motivated by performance reasons, while we show that leveraging rotation information helps recognition.

## 3 Offline Creation of the Recognition System

### 3.1 Data Acquisition and Selection

For creating the database we exploit two sources of information (see Fig. 2):

- Calibrated image data: Panoramic images captured by a vehicle driving systematically through the streets. For each of these images camera position and orientation is known from GPS and sensor data.
- 2D Building floorplans as available from land registration or fire insurance companies as well as building heights. The 2D maps can be extruded to piecewise planar 3D models approximating the buildings (see Fig. 2) and each of these buildings is assigned a place-of-interest ID.

For the dataset of San Francisco, panoramic images have been taken roughly every 10 meters and 14896 places of interest have been covered.

### 3.2 Sparse Representation of all Places-of-Interest in a City

Given that we know the 3D building geometry, we can reproject the panoramic images onto the 3D models. Assuming planar lambertian surfaces, these should give rise to the same textured 3D model (apart from image resampling), so there is a huge redundancy in the captured panoramic images. While it might be beneficial to fuse multiple views of the same features, we leave the optimal redundant sampling of the facades from multiple overlapping panoramas for future work. Instead, we improve both memory requirements and execution speed by eliminating the redundancy as follows: For each place of interest (POI), we find the panoramic images within 50 m distance to the building outline and extract perspective images with a 60° field of view every 20°. We prune those that look away from the POI or see it at a very oblique angle. The others are selected or rejected so as to represent all the POI surface subject to minimal overlap and maximal orthophoto resolution, when projecting the view onto the facade (see Fig. 3). We obtain 58601 perspective images with a $1024 \times 1024$ resolution on the San Francisco dataset.

**Fig. 3** Bird's eye view of Ferry Building. Portions of the panoramic images that are used to sparsely cover all facades of the POI are highlighted

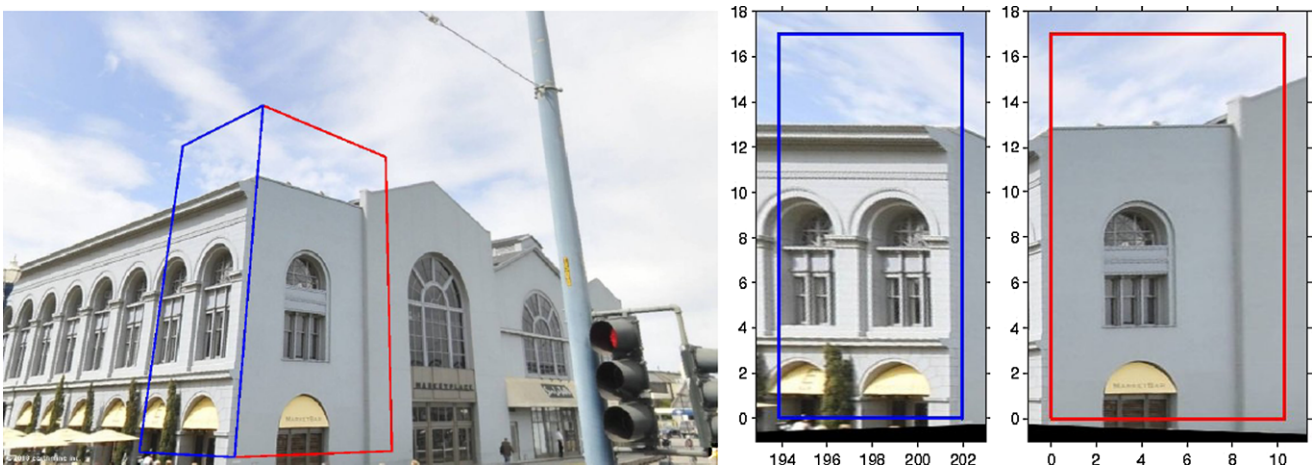


**Fig. 4** *Left*: Building geometry projected into an image. *Right*: Two orthophotos generated from this image with overlaid geometry. The *axes* show the known scale in meters. As the building height might be inaccurate we add a safety margin to incorporate also facade parts that are slightly higher than predicted by the building height. We assume that having some sky on the frontal view will not hurt recognition, because usually there are no features on it

### 3.3 Geometric Rectification

Using the building height information we extrude the building outlines to 3D. We then project the reference images onto these 3D surfaces and render synthetic orthoviews (see Fig. 4). For each of the planar facade parts we generate orthophotos and use GPU-SIFT[3] to extract DoG keypoints and SIFT descriptors. Generally, for descriptor computation, previous approaches estimate keypoint orientations from the local gradient histogram. Rotating the local patch however in a way that the dominant peak is in the zero degree direction potentially makes the descriptors less discriminative, since all of them might have now significant mass in the zero degree descriptor bins and purely rotated local patches can no longer be distinguished. Instead, we project the gravity direction onto the facade and align the keypoints with this direction (upright SIFT). Effectively, by computing a gravity-compatible orthophoto, we remove all effects of 3D rotation and perspective from the image data.[4] Matching such features reduces the 6 DOF perspective recognition problem to a homothetic problem involving only scale and offset ambiguities in the 2D plane.

### 3.4 Vocabulary Tree Indexing

Based upon the extracted descriptors we use hierarchical $k$-means clustering to learn a vector quantization and build a visual vocabulary. We choose a random subset of 16M descriptors from the whole set of about 130M. We build a tree with the following parameters: split factor $k = 10$, depth $d = 6$ which leads to one million leaf nodes.

We choose the bags of visual words to consist of all the features stemming from one perspective image. In that way, we avoid very small bags of words corresponding to small rectified patches. We then index these bags of words using an inverted file system (IFS) for fast retrieval.

[3] C. Wu: "SiftGPU" (Version 0.5.360) http://cs.unc.edu/ccwu/siftgpu.

[4] Apart from non-planarities and image resolution issues due to interpolation.

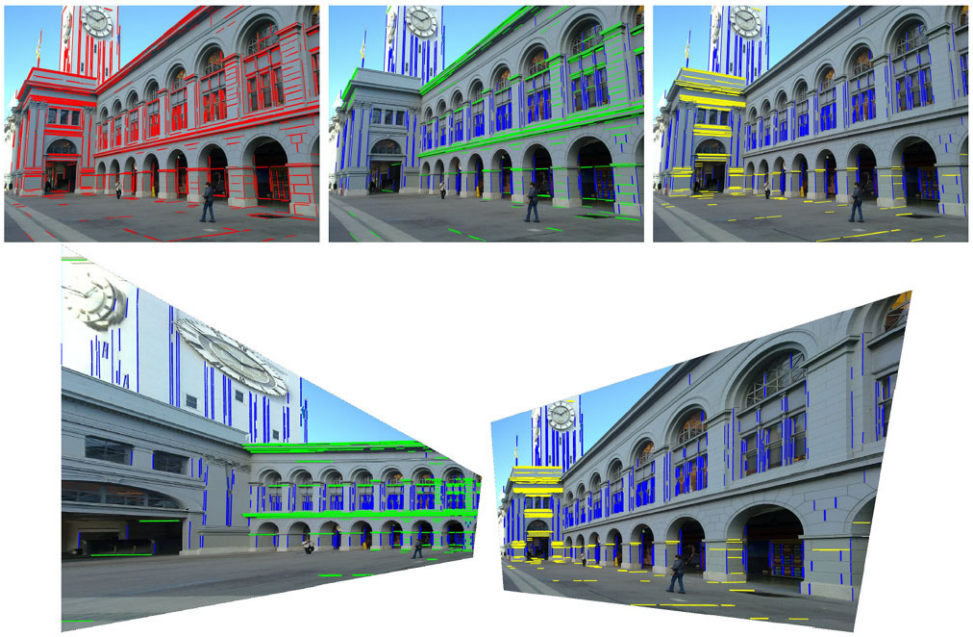

**Fig. 5** *Top row*. *Left*: Query image with detected line segments. *Middle* and *right*: Lines belonging to the same vanishing point have been given the same color. Each image shows only the lines corresponding to one pair of orthogonal vanishing points. *Bottom row*: Two rectifications of the query image according to the two chosen pairs of vanishing points

## 4 Recognition of Places of Interest

The incoming query image is assumed to be taken by a cell phone for which a rough estimate of the orientation (e.g. landscape versus portrait) is usually available from the cell phone's sensors, so that we can correctly assign vanishing points to real-world directions. We also expect that the focal length of the cell phone's camera is known (as typically provided in the EXIF data of jpeg files or from the phone manufacturer), so that we can reason about orthogonality of directions in 3D space (see Sect. 4.1 for more details). With increasing computational power of mobile devices, some of the steps required for recognition could be run on the cell phone, e.g. to save bandwidth by transmitting only a set of sparse descriptors (Chandrasekhar et al. 2009; Takacs et al. 2010). However, in our implementation we focus on the general recognition system and not on a specific mobile device implementation. Consequently, the image is transmitted to a backend server where further processing is performed, before the results are returned to the mobile device.

### 4.1 Removing 3D Rotation Effects from Query Image

We detect line segments in the image using a method[5] based on Kosecka and Zhang (2002), estimate vanishing points as intersections of these lines, followed by a subsequent refinement step. Since the camera calibration is known, we can backproject the presumed vanishing points to rays in 3D space, which should be orthogonal. Every pair of points that does not fulfill this orthogonality constraint is no longer considered for rectification.

In case there are still multiple pairs of vanishing points left, we try to reduce the number of candidate pairs further. We estimate the importance of a plane by taking into account the number of lines on it and the closeness of lines corresponding to different vanishing points. We stretch the lines by 15% on both ends and then count the number of intersecting lines. For the plane with the highest number and all those within 95% of it, we generate an orthoview while discarding all the other planes.

We rectify the image by applying a homography that maps the vanishing points to vertical respectively horizontal infinity (see Fig. 5). Here, we choose the vanishing point which (interpreted as a ray) is closest to the known gravity vector to become vertical and the other one to become horizontal.

### 4.2 Retrieving Candidate Matches

On these rectified images, we then compute upright SIFT features. Following the approach of Sivic and Zisserman (2003), the features are quantized into visual words using the vocabulary tree and the inverted file is used to retrieve a list of all images having at least one of those words in common. The list is then reranked according to the L1-norm of histogram differences, such that images with a visual word distribution similar to the query move closer to the top. The top $c$ candidates (e.g. $c = 50$) are further examined by geometric verification while the remaining ones are discarded.

[5]D. Hoiem: "Finding Long, Straight Lines" http://www.cs.illinois.edu/homes/dhoiem/software/index.html.

### 4.3 Geometric Verification Voting Scheme

So far, ranking only used frequencies of visual words for POI identification. As usual, geometrical verification of the feature configurations can be used to improve the ranking. Unlike previous approaches, which usually perform RANSAC with an affine or epipolar geometry model, we leverage the fact that we are solving a homothetic problem.

Since we are matching the rectified query image (upright orthophoto) to the most promising geometry-aligned database images (also upright orthophoto), the difference between the images are only in scale and offset (related to camera distance and position with respect to the facade).

First we observe that for all true correspondences $\{(S_{\text{facade},j}, S_{\text{query},j})\}$ the scale ratios $\rho_i := \sigma_{\text{query},i}/\sigma_{\text{facade},i}$ of corresponding DoG keypoints should be equal. Please note that an approximate consistency of scale ratios has been proposed as a geometric verification heuristics for general scenes by Jegou et al. (2008), however, now in the homothetic verification a constant scale ratio complies exactly with the model.

Still, due to a noisy feature scale estimate from the detector, the ratio of the query feature scale and the facade feature scale can still deviate slightly from the true image scale ratio. When swapping the roles of the images, the same argument applies for the inverse ratios, since the problem is symmetric. Consequently, we transfer it to the logarithmic domain, and require the differences of logarithmic scale ratios to agree up to a threshold $\log t$ that depends on the expected scale estimation uncertainty of the SIFT detector (e.g. $\log t = 0.15$):

$$|\log \rho_i - \log \rho_j| \leq \log t\,. \tag{1}$$

In order to determine the scale ratio $\rho^*$ with the most support, we find the argmax of the function from kernel density estimation (Bishop 2006):

$$\rho^* = \arg\max_{\rho} \sum_i e^{-\frac{(\rho-\rho_i)^2}{2\sigma^2}}\,, \tag{2}$$

with $\sigma = \log t$. This can be viewed as a continuous version of histogram binning or of a 1D-Hough transform (Duda and Hart 1972). All the datapoints within a certain distance of $\rho^*$ (e.g. $2\sigma$) are considered inliers.

Using the estimated scale ratio, we transform the feature coordinates of both images to a common scale. Since we know the true scale of the database image, we can have all the coordinates expressed in meters. Truly matching feature points now differ only by a global translation. The $x$ and $y$ components of this translation are estimated independently. We define the coordinate differences $\xi_i := x_{\text{query},i} - x_{\text{facade},i}$ and $\nu_i := y_{\text{query},i} - y_{\text{facade},i}$. As before, true correspondences should exhibit a consistent coordinate difference:

$$|\xi_i - \xi_j| \leq d \quad \text{and} \quad |\nu_i - \nu_j| \leq d\,. \tag{3}$$

Since all of the coordinates are expressed in terms of a known unit, we can again derive in a principled way a reasonable value for translation tolerance $d$, completely independently of image resolutions (e.g. $d = 0.3$ m). We vote for $x$- and $y$-displacement separately using the same scheme as before (without transforming to log-space). The intersection of the two resulting inlier sets constitutes the final inlier set of the geometric verification (see Fig. 6) and its cardinality is used to generate a new ranking of all the candidates under consideration.

This scheme has several advantages over previous approaches: RANSAC on top of an essential matrix, affine or projective transformation estimates 5, 6 or 8 parameters respectively. In contrast, our approach only needs to determine three degrees of freedom total, which means that the search space has a lower dimension. On top of that, each degree of freedom is estimated separately, thus further reducing the search space, which increases reliability and efficiency. In fact, we can afford exhaustively testing every hypothesis rather than sampling just some of them.

Every feature correspondence provides three constraints (scale, $x$- and $y$-coordinate). Thus, a single correspondence is enough to generate a complete hypothesis. Earlier, RANSAC-based approaches usually ignore scale and require outlier-free subsets of 5, 3 or 4 correspondences respectively. In order to hit such a set reliably, one needs to draw a number of samples which is essentially exponential in the number of required correspondences.

Finally, even wrong correspondences can still contain partial information about the solution. For instance, consider facades with many repeated window elements: if one window in the query image is (mistakenly) matched to the window below in the database image, this correspondence will vote for the right scale ratio and for the correct $x$-coordinate.

### 4.4 Pose Estimation from 2D-2D Correspondences

Since we used vanishing points to rectify the original query image, we obtain the camera orientation with respect to the facade directly from the vanishing points. As the rectified image plane is parallel to the facade, the only remaining parameters are those obtained in the previous section: Since we know the facade texture in meters the scale ratio can directly be used to compute a (perpendicular) distance $\text{pos}_z$ of the camera from the facade. Assuming the camera is calibrated with focal length 1 pixel and principal point at zero, then

$$\text{pos}_z = \text{res}_{\text{facade}} \cdot \sigma_{\text{facade}}/\sigma_{\text{query}}, \tag{4}$$

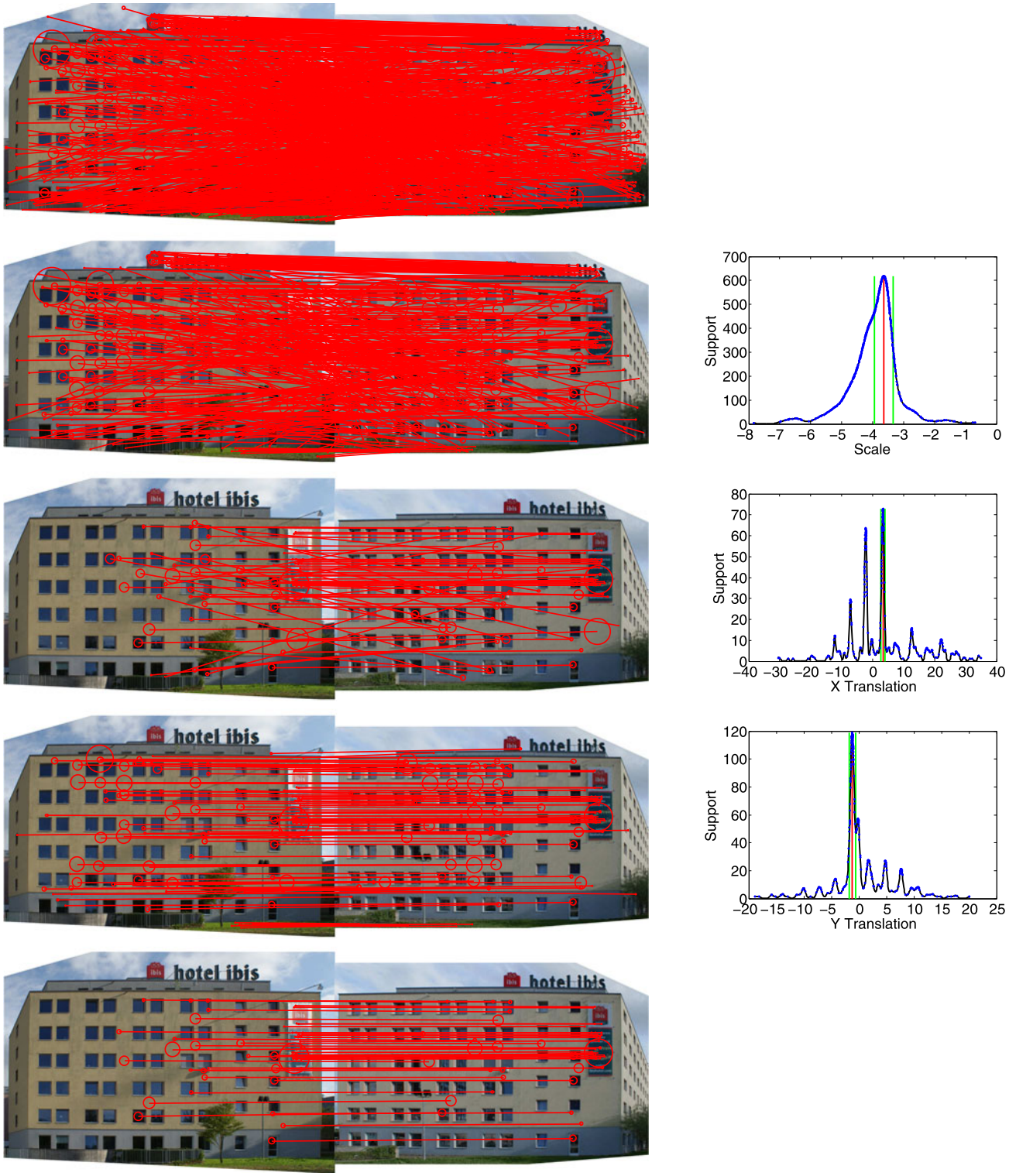


**Fig. 6** (Color online) Illustration of our voting scheme. *Top row*: Raw correspondences. *Second row*: Scale inliers. The plot on the *right* shows the distribution of scale ratios (log scale). *Blue dots* represent individual scale ratios, the *red line* marks the maximum and the two *green lines* show the range of inliers. *Middle row*: X-inliers. In the plot, note the secondary local maxima occurring at a regular interval. They correspond the repeating window structure. *Fourth row*: Y-inliers. Again, there are local maxima following a regular pattern, but they are less pronounced. *Bottom row*: Final inliers. They are computed as the intersection of the $x$- and $y$-inliers. Also note that there are no false positives left

where $\text{res}_{\text{facade}}$ represents the resolution of the orthophoto in pixel/meter. The cell phone's $\text{pos}_x$-offset (parallel to the facade) can directly be computed from the feature position

$$\text{pos}_x = \text{res}_{\text{facade}} \cdot (x_{\text{facade}} - \sigma_{\text{facade}}/\sigma_{\text{query}} \cdot x_{\text{query}}), \quad (5)$$

and $\text{pos}_y$ in an analogous way. The local camera orientation with respect to the wall is simply the inverse vanishing point rotation. Finally, the relative coordinates with respect to the facade can be converted to absolute world coordinates using the facade's pose in the world.

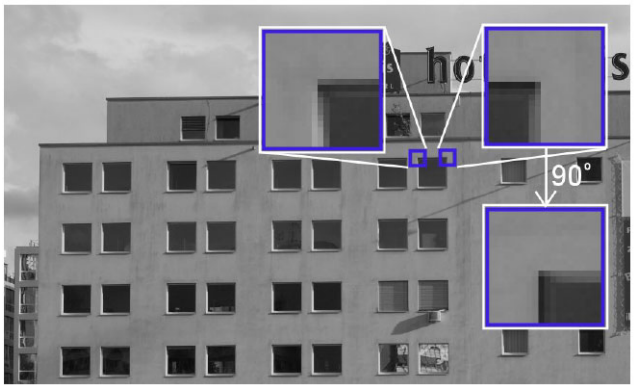


**Fig. 7** The two patches in the top region of the image differ mostly by a 90° rotation. When such patches are rotated into a canonic orientation (as is done in traditional SIFT), the resulting descriptors cannot be distinguished. When the descriptor/patch is extracted with respect to gravity direction the patches differ significantly

## 5 Experiments

### 5.1 Upright SIFT Versus Traditional SIFT

As can be seen schematically from Fig. 7 the traditional SIFT approach is unable to distinguish between patches that differ only by their orientation (since it is designed to be invariant to rotation). In order to test whether the SIFT descriptor's discriminative power improves if we do not rotate it into the dominant gradient orientation a simple experiment has been run (see Fig. 8) on the image sequences for descriptor evaluation provided by Mikolajczyk and Schmid (2005). Here we warp all 5 images of such a sequence to the first image, so that orientations are the same for corresponding SIFT keypoints.[6] Features at the same position ±50% feature size, same scale ±20% and same orientation ±30° are assumed to be a geometrical ground truth correspondence, other features are assumed to be not in correspondence. By comparing every descriptor of image 1 to every descriptor in the other images we generate the precision-recall diagram for the three sequences bark, wall and graffiti (see Fig. 8) as

[6]For this experiment, we used A. Vedaldi and B. Fulkerson's vlfeat (v0.94 available from http://vlfeat.org) for detector and descriptor.

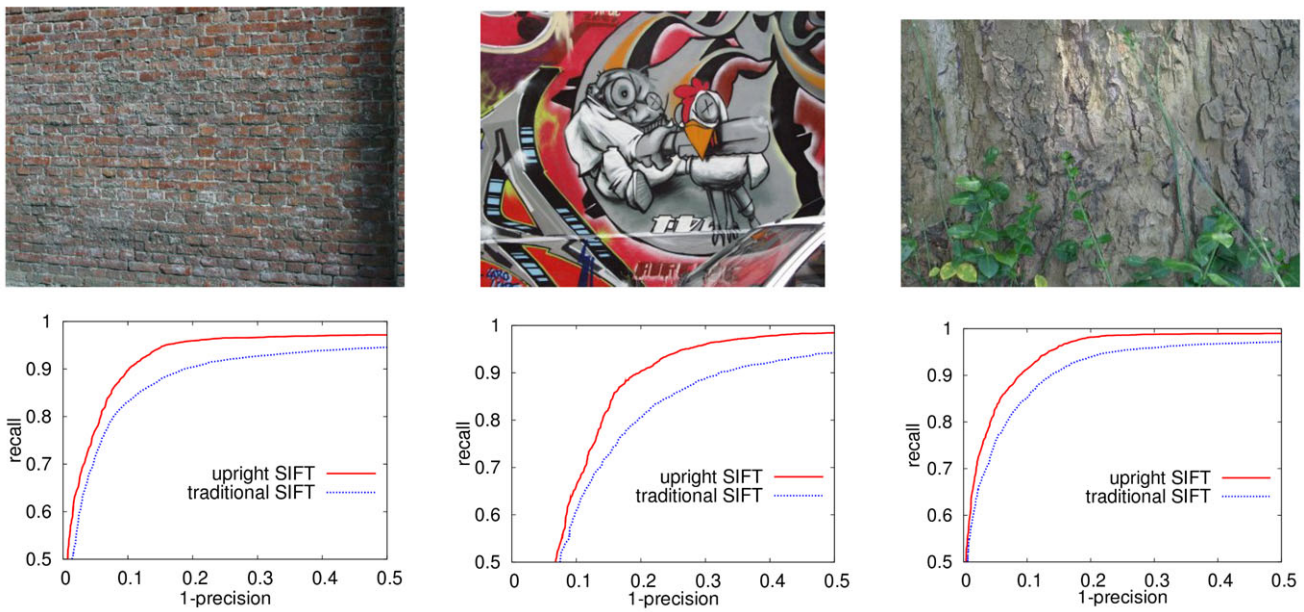


**Fig. 8** Upright-SIFT vs. traditional SIFT with orientation estimation: All 5 images of the wall, graffiti and bark sequences (Mikolajczyk and Schmid 2005) are warped to the first image of their sequence before DoG keypoints are extracted. We now compare the descriptiveness of upright-SIFT (with zero-orientation) and standard SIFT which estimates orientation from the local gradient histogram (Lowe 2004). For a given precision (fraction of correct matches within all obtained matches) we get a higher recall rate (fraction of correct matches with respect to the set of geometrical ground truth correspondences)

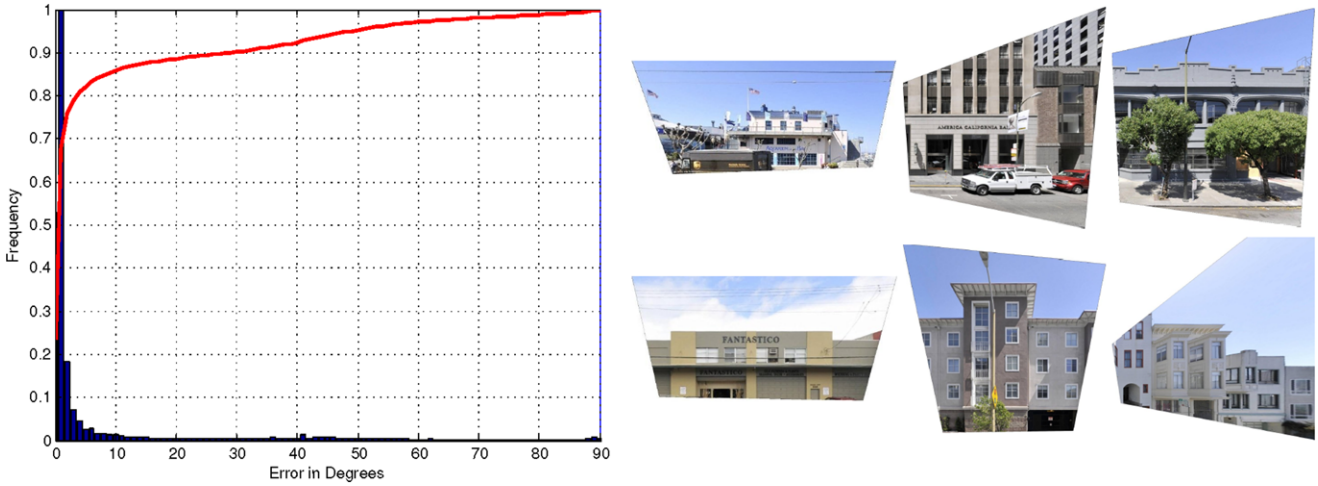


**Fig. 9** *Left*: Histogram of orientation errors from vanishing points in degrees (*bars*) and cumulative distribution (*curve*), histogram scaled to the range [0, 1]. *Right*: Some rectified cell phone images

has been done in Mikolajczyk and Schmid (2005). Note that this experiment compares upright SIFT to traditional SIFT in general and is independent of the recognition pipeline that we propose.

In all of these sequences upright produces a significantly higher precision for a given recall fraction of the geometrical ground truth matches. A possible explanation is that when rotating the SIFT descriptor to the dominant orientation some gradient orientation histogram entries are more likely to obtain responses than others (e.g. those of the dominant orientation). On top of that traditional SIFT cannot distinguish local regions that mostly differ by a rotation whereas this is possible using upright SIFT (see also Fig. 7).

## 5.2 Vanishing Point Detection

For 31034 Earthmine images, we ran the vanishing point detection algorithm. In order to measure the error, we computed the angles between the directions that were found and the horizontals/verticals of known building surfaces. The distribution of these angles is shown in Fig. 9. 75% of the time, the vanishing points are estimated correctly up to 2 degrees, the median error is 0.9°.

## 5.3 Geometric Verification

We compared the performance of our proposed voting scheme to that of the traditional approach (RANSAC with a homography or an affine model). The nine image pairs shown in Fig. 10 were used. First, they were rectified automatically based on vanishing points. In the resulting images, we clicked manually correspondences to determine the ground truth homography which maps one image to the other. Then we extracted upright SIFT features in vanishing-point rectified images and computed putative correspondences based upon descriptor similarity as in Lowe (2004). These correspondences were separated into inliers and outliers according to the ground truth homography. Borderline correspondences were discarded entirely.

By removing an appropriate number of inliers and/or outliers, we generated correspondence sets with any desired outlier ratio. For every outlier ratio from 0%–100%, we ran both our voting algorithm and the following three variants of RANSAC 1000 times each.

- **Homography Ransac**: The correspondences were transformed back into the original images. We estimated a homography and we used 1000 iterations.
- **Affine Ransac**: The correspondences were transformed back into the original images. We estimated an affine transformation (with loose outlier thresholds) and we used 1000 iterations.
- **Homothetic Ransac**: We used the original correspondences between the rectified images and estimated a homothetic transformation from a single correspondence. The number of iterations was at most 1000. When there were less than 1000 correspondences, we deterministically iterated through all of them.

### 5.3.1 Speed and Complexity

Figure 11 shows a run-time comparison. *Homography Ransac* is by far the slowest method. This is because constructing the minimal solution from 4 correspondences is more expensive than the lighter-weight models. *Voting* is the fastest.

It is interesting to note that for a low number of correspondences (less than 100), *Homothetic Ransac* is about as

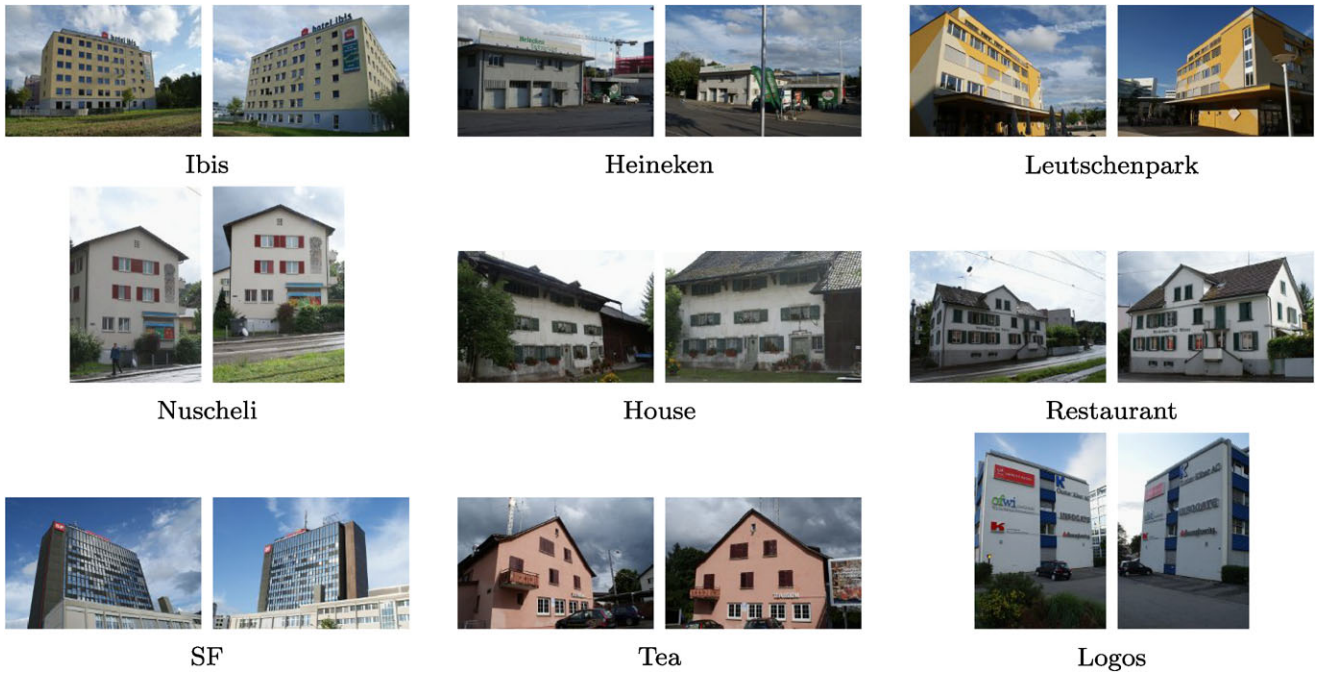


**Fig. 10** Pairs of images used for testing

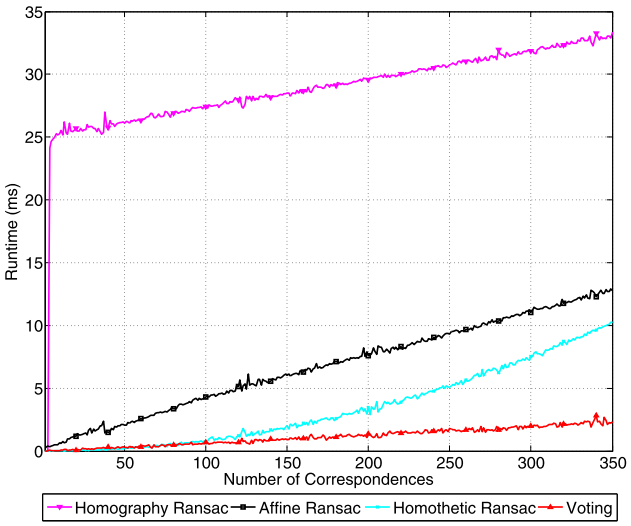


**Fig. 11** Time required for estimating the transformation

fast as *Voting*, but gets worse for higher numbers due to its quadratic runtime. In contrast, *Voting* has near-linear runtime, because it is a series of three successive 1D problems, so it can take advantage of pre-sorting the scalar values to speed up support calculation.

### 5.3.2 *Qualitative Comparison of Models*

In Fig. 12 we analyzed how often a method does not find a transformation at all. This happens when none of the candidate models gets sufficient support.[7] Note however, that a zero probability of failure does *not* mean "all went well": It merely means that the algorithm found a model (with some support) which may or may not be accurate. There is a trade-off to be made: Either one is more picky in accepting a solution, which increases the probability of failure, or one is more tolerant, which decreases not only the probability of failure, but also the accuracy of the solutions found that way (see Fig. 13).

Except for image pair "Heineken" (with significant off-plane structures), *Voting* succeeded almost always. *Homothetic Ransac* only fails in 2 out of 9 image pairs, and only at 90% outliers. *Homography* and *Affine Ransac*, on the other hand, start failing much earlier in all pairs: *Affine Ransac* at 20%–80% outliers and *Homography Ransac* at 50%–80%.

Figure 13 shows the average reprojection error of the estimated transformation. For low outlier ratios, *Homography Ransac* performs quite well: In 7 image pairs it is the best of the 4 methods. For higher ratios however, it becomes very unstable: Either RANSAC fails to find a solution at all (see Fig. 12), or the error fluctuates wildly with the mean exceeding several hundred meters and a standard deviation of several thousand. This high error could be avoided by increasing the required number of inliers, but that would further reduce the chances of finding a solution.

*Affine Ransac* starts off with a high reprojection error, which is to be expected since the underlying affine trans-

[7] We empirically determined optimal inlier thresholds: For a homothetic model we require 3 inliers, for affine 8 and for homography 12.

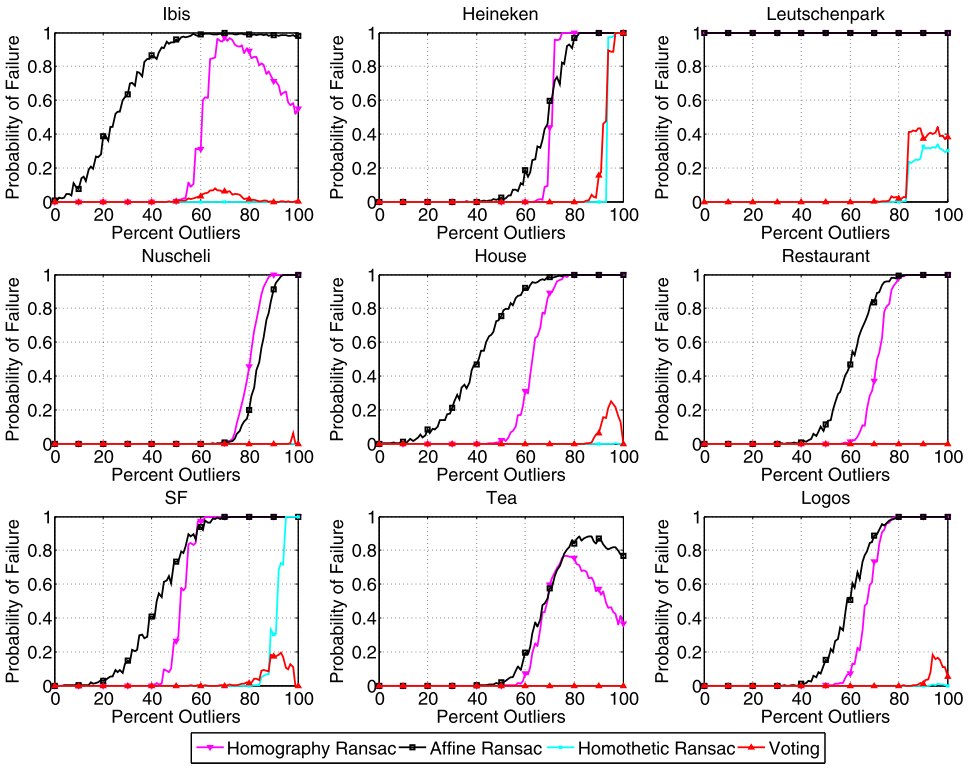


**Fig. 12** Probability that the given algorithm does not find a transformation with sufficient support

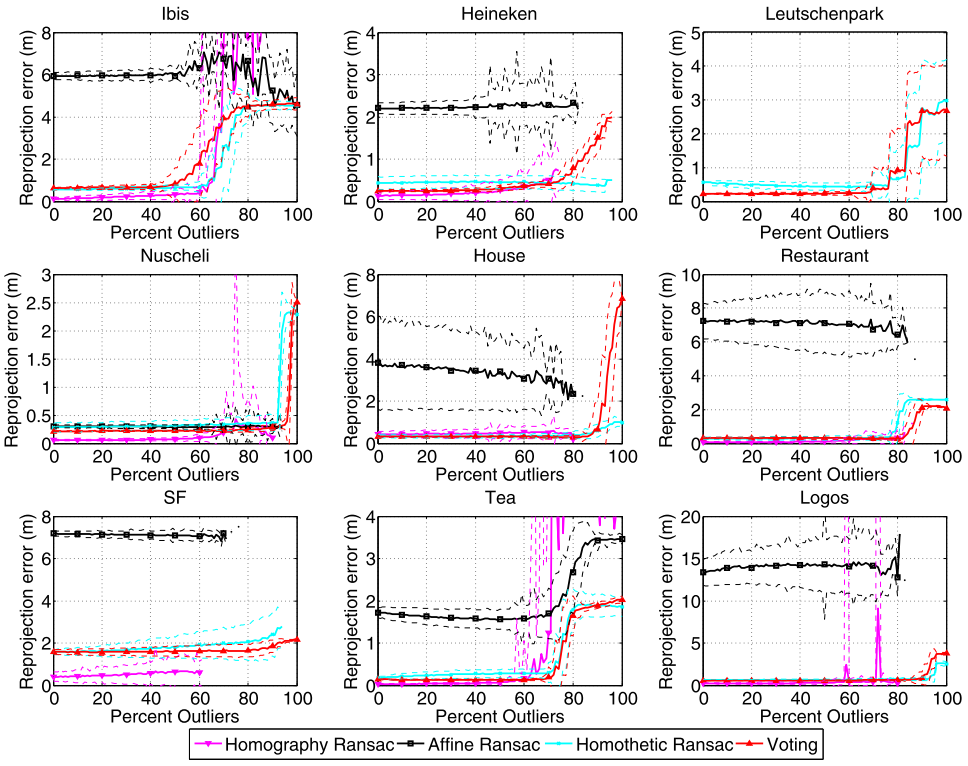


**Fig. 13** Average reprojection error of estimated transformation (if any). Missing parts in the curves indicate that no transformation has been found. *Dotted lines* show standard deviation

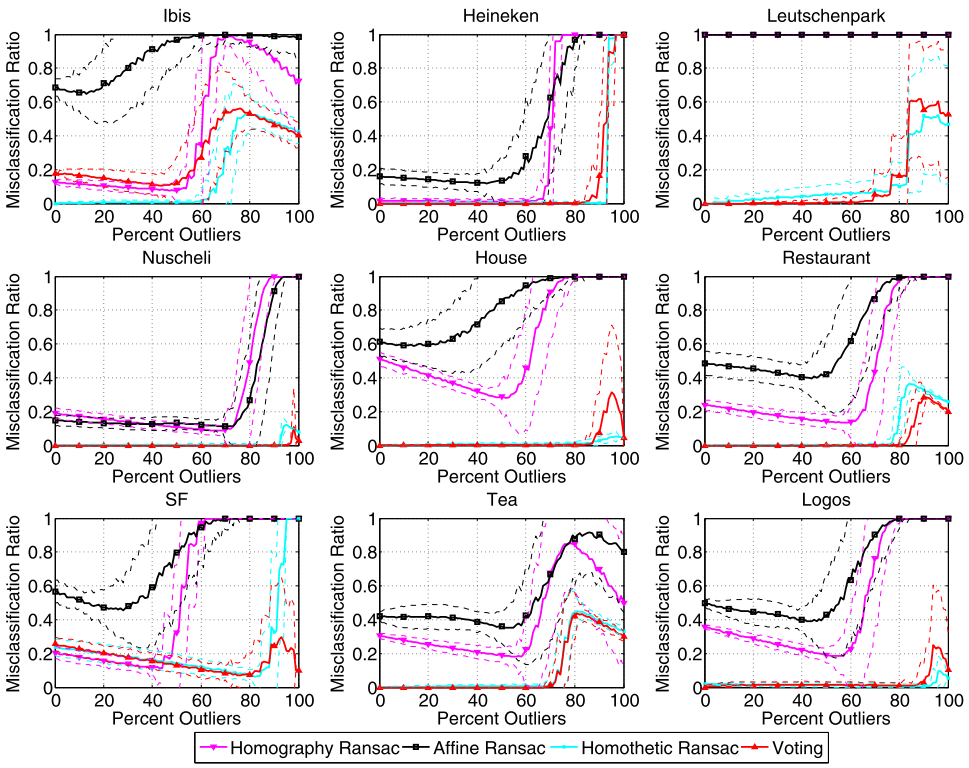


**Fig. 14** Number of false positives plus false negatives divided by the number of (putative) correspondences presented to the algorithm. *Dotted lines* show standard deviation

formation requires loose inlier thresholds to cope with perspective effects. On the positive side, it is very insensitive to varying outlier ratios. For higher outlier ratios, however, it still fails to find a solution in 7 image pairs.

*Homothetic Ransac* and *Voting* perform quite similar. At low outlier ratios, they are slightly worse than *Homography Ransac*. This is most likely due to an imperfect rectification or slight radial distortion, which can be better compensated by the homography's much higher number of degrees of freedom. However, for increasing outlier counts, *Homothetic Ransac* and *Voting* degrade much more gracefully, both in terms of being able to find a solution and in terms of reprojection error of that solution.

Since reprojection error as a measure of quality may not be completely fair towards *Affine Ransac* (which cannot model perspective effects well), and in order to fuse quality and success rate into a single plot, we explored an additional measure of quality: All of the tested algorithms partition the correspondences into inliers and outliers. In Fig. 14 we investigated how well this partition matches the ground truth classification. This number is important, since the number of inliers after geometric verification is often used to decide whether a given pair of images matches or to rerank a list of possible candidates. More precisely, the figure shows the fraction of mistakes (false positives plus false negatives) a given method made. Whenever the method failed to return a transformation, we assumed a fraction of 1.

*Homography* and *Affine Ransac* make many classification errors. Especially for high outlier ratios of 70% and above, the number of misclassifications is very high. But even for perfect inlier sets, *Affine Ransac* only recognizes half of them as such in 7 of 9 images, *Homography Ransac* is only slightly better.

*Homothetic Ransac* and *Voting* perform much better. For 7 image pairs, the misclassification ratio remains close to zero for up to 70%–80% outliers. But even for high outlier ratios, the classification error reaches 100% only rarely (in 1 pair for *Voting* and in 2 pairs for *Homothetic Ransac*).

### 5.3.3 Kernel Density Voting Versus RANSAC

Figure 15 compares the reprojection error of the two methods. Except for image pair "Ibis", *Voting* has an error smaller or at least similar to *Ransac*. Also note how *Voting* has a smaller standard deviation, which indicates that the results are more stable across multiple runs.

A possible explanation for the behavior of "Ibis" is the off-plane sign on the roof which, due to parallax, is classified as outlier: *Voting* is more prone to averaging it with the true correspondences on the facade while *Ransac* chooses either one or the other (and more often the facade).

Figure 16 shows how the reprojection error evolves for various fixed outlier ratios as the total number of correspondences increases. It is interesting to note that the error changes (usually it decreases). Again, with the exception of

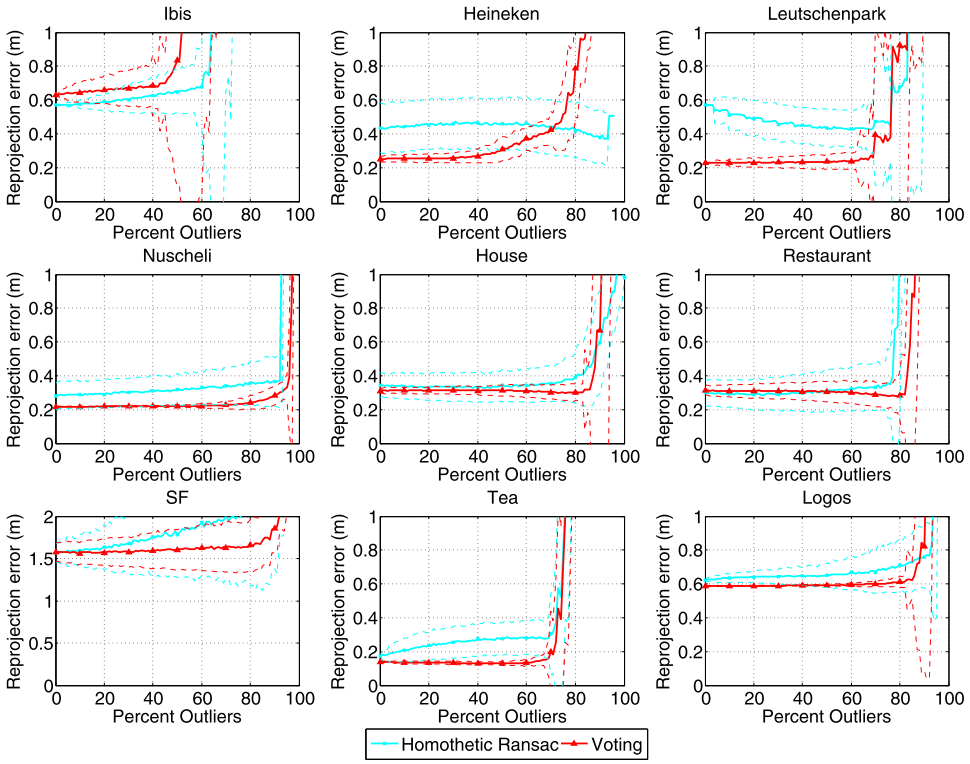


**Fig. 15** Magnified details of Fig. 13: Average reprojection error of estimated transformation. *Dotted lines* show standard deviation

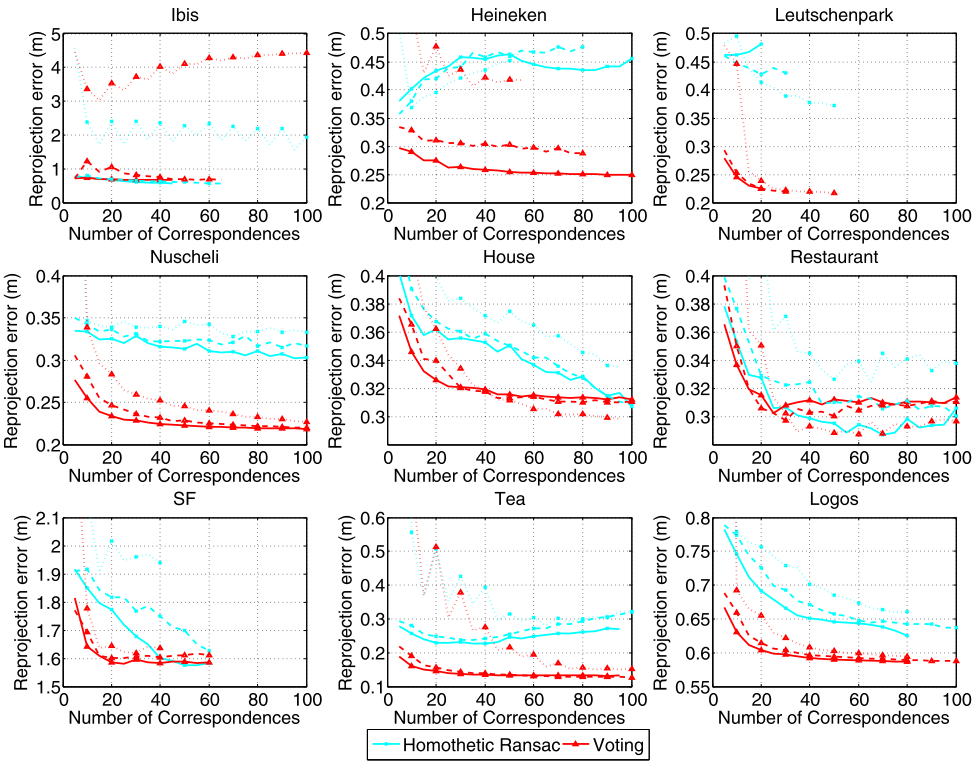


**Fig. 16** Average reprojection error of estimated transformation as a function of the number of correspondences for a fixed outlier ratio of 30% (*solid*), 50% (*dashed*) and 70% (*dotted*)

"Ibis", the curves for *Voting* are generally lower than the corresponding curve for *Ransac*.

We conclude that just rectifying the images already allows for faster and more robust methods than the traditional approaches. But even among the rectified methods, *Voting* has a better time complexity than RANSAC and on medium to large correspondence sets is significantly faster (e.g. 4× for 350 correspondences). In applications where the trans-

formation between images is of interest, the built-in averaging of *Voting* leads to more accurate estimates. While *Homography Ransac* may offer some benefits in the low outlier range, in any case its runtime is far too high, which makes it unsuitable for an online application such as ours.

### 5.4 Recognition

In this section we evaluate RANSAC with an affine model and our stratified voting scheme (plus some intermediate methods) as part of a recognition pipeline. For the sake of readability of the figures we had to limit the number of curves, so we decided not to include RANSAC with a homothetic model because in terms of quality it is very similar to voting. As our reference implementation we chose only the affine model because it is common in image search (Perdoch et al. 2009; Philbin et al. 2007) as it constitutes a good compromise between generality and efficiency, i.e. (with loose thresholds) it is not significantly weaker but more stable to compute and much faster than a homography.

The different variants of the recognition pipeline used in our comparison are:

- **Affine**: This is our reference implementation. The VT and IFS are trained and built on the raw survey images. As feature descriptor we use traditional SIFT. For geometric verification we use the affine model with loose thresholds.
- **Masked**: Same as before, except that for survey images we use geometric models to discard all features that do not lie on a building. This variant uses the same regions of the original images as the following variants. Its interest lies in testing how discarding background features affects recognition.
- **Rectified**: Survey images are rectified using known 3D models of the buildings and query images are rectified using estimated vanishing points. The feature descriptor is still standard SIFT. Geometric verification is our proposed 3-degrees-of-freedom plane alignment using stratified histogram voting.
- **Upright**: Survey and query images are rectified as before, but in addition we use upright SIFT. Geometric verification is again 3DOF plane alignment.

We evaluated each of these four implementations on three different query sets:

- **Earthmine**: This dataset consists of 31034 Earthmine images that were *not* selected for the training set. However, they stem from the same day and have been taken under the same conditions as the training set so that they must be considered as very easy. The images were automatically chosen such that they point towards a building. Whether or not this building is partially or completely occluded by vegetation was not a factor.

**Table 1** Frequency of the top-ranked image being correct. For each dataset the best percentage has been highlighted

| | Affine | Masked | Rectified | Upright |
|---|---|---|---|---|
| Earthmine | 84.3% | 83.0% | 82.6% | **85.0%** |
| Navteq | 33.9% | 26.3% | 25.2% | **38.9%** |
| Cellphone | 30.2% | 23.2% | 25.2% | **32.1%** |

- **Navteq**: This dataset consists of 182 images, sampled at angles of 70° to 120° degrees (with respect to driving direction) and 0° to 20° (tilt) from panoramic image data from Navteq, where panoramic images have been chosen such that buildings could be seen reasonably well. This data has been taken more than one year later than the Earthmine training data and with different equipment.
- **Cellphone**: This dataset consists of 1180 images taken by various people with different camera phones (Nokia N95, N97, N900, N86) having between 5 and 8 megapixel resolution. During rectification, they have been downsampled to 1–2 megapixel. These images are from pedestrians' perspective partially under extreme angles and constitute the most challenging dataset.

We examined how frequently a correct building is returned as one of the top $n$ candidates for $n$ ranging from 1 to 50. This information was recorded for both the ranking before and after geometric verification and for all combinations of implementations and query sets. The results are shown in Fig. 17. Since we are targeting augmented reality applications, we are mainly interested in the percentages for the top ranked image. These numbers are summarized in Table 1.

We observe that the performance is generally better on the Earthmine query set than on the other two, which is to be expected since these images have been taken under the same conditions as the database images.

We notice that *Affine* generally outperforms *Masked*. The difference between the two is that the database for the former contains features from both buildings and surroundings, while the latter uses only features from buildings. This indicates that features from the surroundings help recognition rather than distract. This is probably the main reason why the pre-verification curves of the other two methods are lower than *Affine*. They suffer from the same disadvantage as *Masked*: having ignored the features from the surroundings.

With respect to the pre-verification curves, *Rectified* does slightly worse than *Masked*. On the other hand, the post-verification curve for *Rectified* is flatter. This means that rectifying the images may hurt performance in the VT part, but it allows for a stronger geometric verification (3DOF homothetic vs. affine).

It also paves the way for using upright SIFT. As already stated before, upright SIFT is more discriminative because it can distinguish image patches that differ only by a rota-

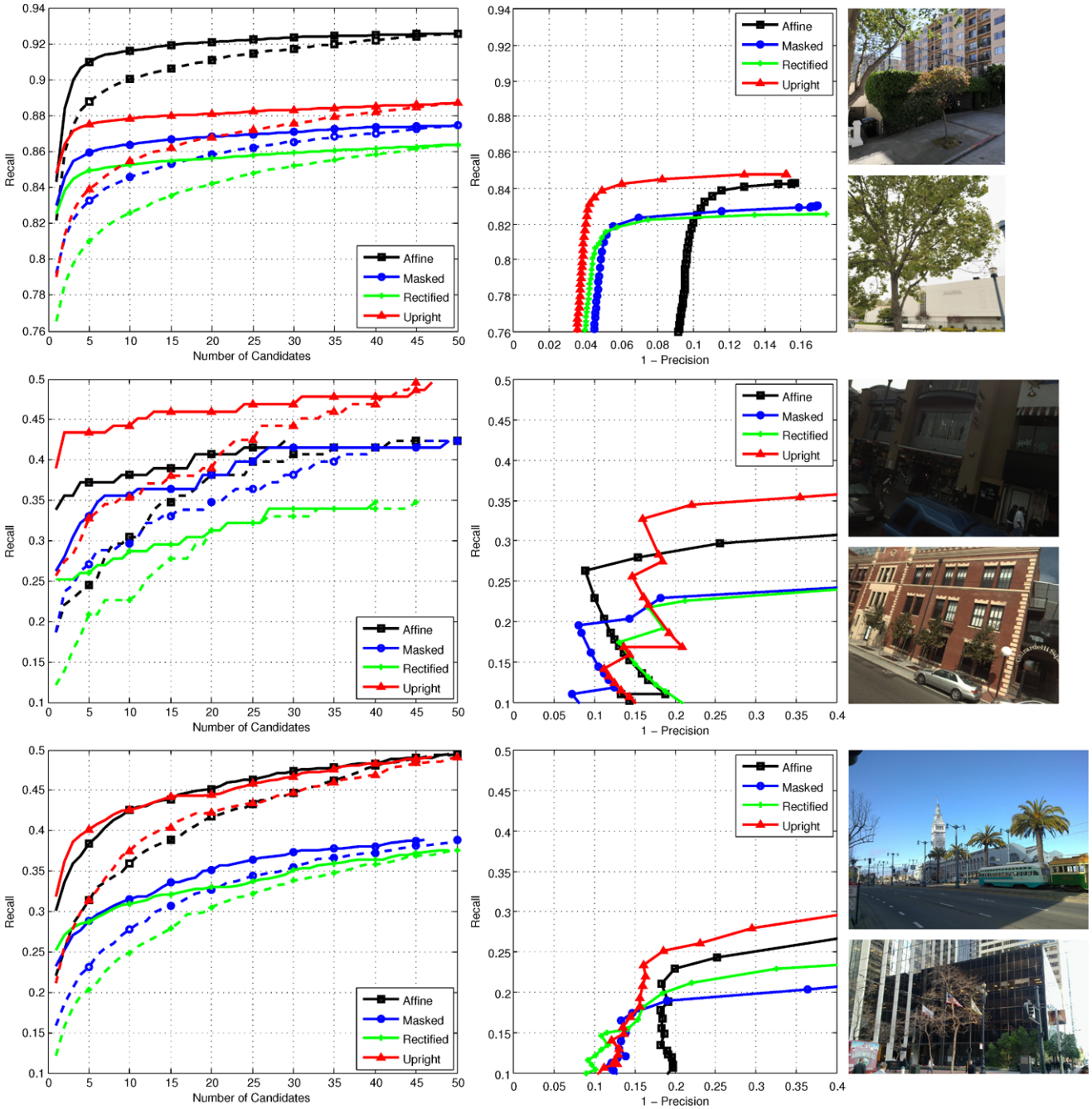


**Fig. 17** *Left column*: Frequency of correct building being among top *n* candidates. (*Dashed lines*: before geometric rectification, *solid lines*: after.) *Middle column*: Precision-vs.-recall curve based on the number of inliers for accepting a candidate answer. *Right column*: Sample query images. *Top row*: Earthmine. *Middle row*: Navteq. *Bottom row*: Cellphone

tion. We see that already the pre-verification curve for our proposed method (*Upright*) is higher than for *Masked* and *Rectified*. Combined with the strong 3DOF verification, it outperforms the other methods on all three datasets with respect to the top-ranked candidate (see Table 1). On top of that, this advantage gets bigger on the more challenging datasets.

We have seen that *Affine* has the highest pre-verification curve due to the inclusion of background features. Even though *Upright* is the better overall system, combining the advantages of both methods might yield even better results. We plan to address this in future work.

We also examined the precision-recall trade-off. The number of inliers for the top candidate is compared to a

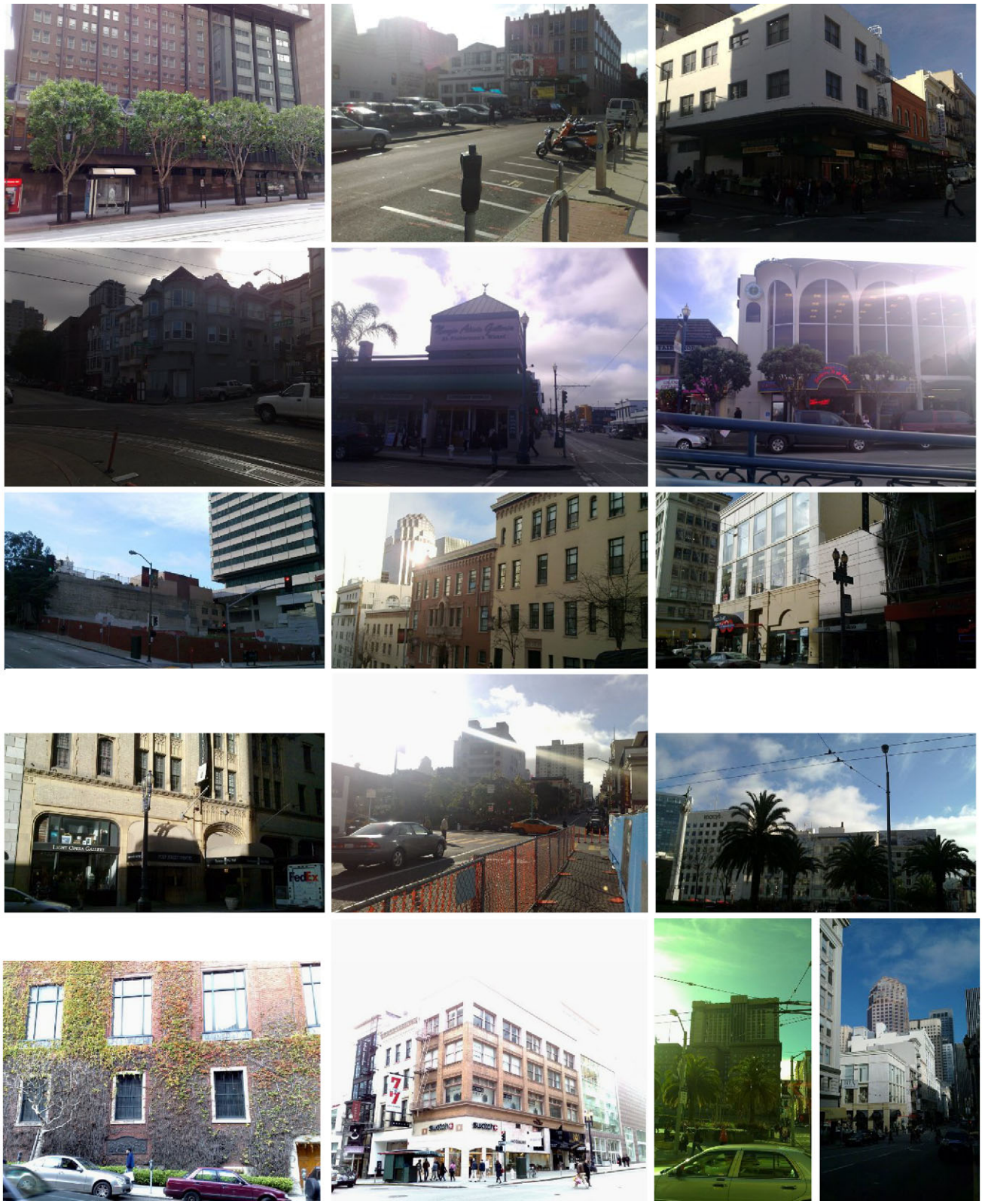

**Fig. 18** Example queries which could not be recognized. Dominant error sources are illumination, occlusion and distance. We decided to include such images in the data set since such problems will always occur in real cell phone based scenarios, where you would not wait for better weather or until the delivery truck moves away

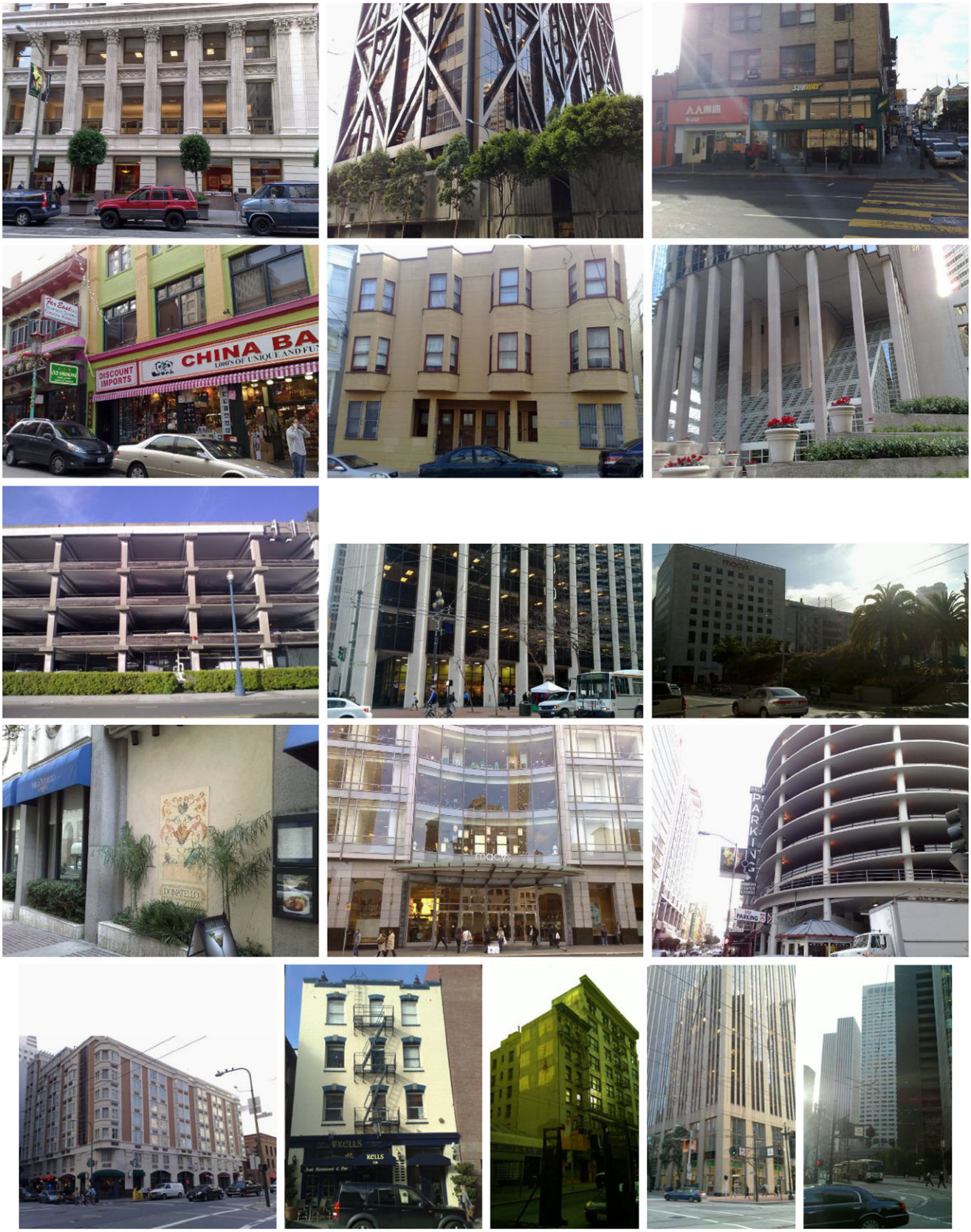


**Fig. 19** Example queries correctly recognized. We deliberately show also some successful cases where the planarity assumption was not strictly fulfilled


**Fig. 20** Analysis for three examples of failed recognition. *Top Row*: Query cell phone image. *Center Row*: Closest image (manually chosen) in database. *Bottom Row*: standard SIFT matching as proposed in Lowe (2004) on rectified query and database to illustrate the difficulty already in 2-image feature matching without quantization: Often the street level window decoration changes for shops (e.g. *left*). Reflections in windows are also a major problem, since they change strongly with viewpoint. Also cars, pedestrians, vegetation and other things can occlude parts of the facade at the street level. The street level is however often the disambiguating part of the facade since the higher levels often only contain grids of windows. For some images the cell phone query has been taken from the opposite side of the street so the covered facade parts largely vary between database and query (e.g. *right*)

threshold. If the number is below, the system returns "no-answer", otherwise it returns the top candidate. By setting this threshold to lower values, one achieves a higher recall (how often a query gets a correct answer), but also lower precision (how often an answer is actually correct). By choosing a higher threshold these spurious matches can be reduced at the cost of losing some correct matches as well.

For all three query sets *Masked* and *Rectified* share a similar precision-recall curve with a better precision than *Affine*, but a worse recall. For the Earthmine and Cellphone datasets, *Upright* is clearly the better choice, while for Navteq it depends on how one wants to trade precision for recall.

In Figs. 18 and 19 a small subset of the cell phone images is shown. Please note that these sets should resemble an unexperienced user taking a picture of some building with his or her cell phone in order to obtain location information. Consequently we included also very difficult images with bad illumination and weather conditions, occlusions and pictures showing multiple buildings or from further away.

#### *5.4.1 A Closer Look at Some Unrecognized Images*

The dataset is really challenging and contains also images which are very difficult to uniquely recognize for a trained local human observer, such as entrances of office towers and houses in residential area that do not look different from other houses. On top of that the data set contains images with bad lighting (sun in the image or image very dark) and with vegetation occluding large parts of buildings and queries that look along a street with many facades (not one big facade but many small ones). Some of these close to impossible pictures are displayed in Fig. 18.

Among the images that do not seem so difficult for a human, we analyze three cases in Fig. 20. Here, due to the glass facades there are many reflections that change with viewpoint, and weather/lighting and texture and geometry has changed locally because the shops changed decoration. From matching the images against the ground truth database images we observe that SIFT features alone are not good in describing the content, which can be seen from the fact that we obtain no reasonable matches in the two-image correspondence problems without considering descriptor quantization and the whole database pipeline. For such images other information such as color or text recognition might help.

## 6 Conclusion

We presented an approach for recognizing places of interest in cell phone images. By exploiting approximate 3D city models it was possible to convert street level data to an orthophoto-like representation of the facades of the city. In this representation also the gravity direction is known which enabled the use of upright SIFT features which have been proven more discriminative than classical SIFT on the standard feature descriptor test sets as well as in the location recognition pipeline. The given system can be seen as 3D rotation invariant matching and allowed for estimating homothetic transformations between a rectified cell phone image and a building facade, where the parameters scale and 2D offset of the homothetic transformation can be estimated separately. This allows for an efficient 1D voting scheme related to kernel density estimation and the resulting reranking has been shown to be very effective in boosting the true image to a top position in the reranked list.

**Acknowledgements** We would like to thank Ramakrishna Vedantham and Sam Tsai for their valuable help with the software and database infrastructure.

## References

Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., & Pollefeys, M. (2010). Handling urban location recognition as a 2D homothetic problem. In *ECCV*.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, *110*(3).

Bishop, C. M. (2006). Pattern recognition and machine learning. ISBN 0-387-31073-8, Section 2.5.1, p. 123.

Cao, Y., & McDonald, J. (2009). Viewpoint invariant features from single images using 3D geometry. In *IEEE Workshop on Applications of Computer Vision*.

Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., & Girod, B. (2009). CHoG: compressed histogram of gradients. In *CVPR*.

Dreuw, P., Steingrube, P., Hanselmann, H., & Ney, H. (2009). SURFface: face recognition under viewpoint consistency constraints. In *BMVC*.

Duda, R. O., & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11–15.

Irschara, A., Zach, C., Frahm, J.-M., & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *CVPR*.

Jegou, H., Douze, M., & Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*.

Knopp, J., Sivic, J., & Pajdla, T. (2010). Avoiding confusing features in place recognition. In *ECCV*.

Kosecka, J., & Zhang, Wei (2002). Video compass. In *ECCV*.

Köser, K., & Koch, R. (2007). Perspectively invariant normal features. In *Workshop on 3D Representation for Recognition, ICCV*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–1630.

Nistér, D., & Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR*.

Perdoch, M., Chum, O., & Matas, J. (2009). Efficient representation of local geometry for large scale object retrieval. In *CVPR*.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

Robertson, D., & Cipolla, R. (2004). An image based system for urban navigation. In *BMVC*.

Schindler, G., Brown, M., & Szeliski, R. (2007). City-scale location recognition. In *CVPR*.

Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., & Dellaert, F. (2008). Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*.

Sivic, J., & Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *ICCV*.

Takacs, G., Chandrasekhar, V., Tsai, S., Chen, D., Grzeszczuk, R., & Girod, B. (2010). Unified real-time tracking and recognition with rotation-invariant fast features. In *CVPR*.

Wu, C., Clipp, B., Li, X., Frahm, J.-M., & Pollefeys, M. (2008a). 3D model matching with viewpoint invariant patches (VIPs). In *CVPR*.

Wu, C., Fraundorfer, F., Frahm, J., & Pollefeys, M. (2008b). 3D model search and pose estimation from single images using VIP features. In *Workshop on Search in 3D, CVPR*.

Zamir, A., & Shah, M. (2010). Accurate image localization based on Google maps street view. In *ECCV*.

Zhang, W., & Kosecka, J. (2006). Image based localization in urban environments. In *3DPVT*.

Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., & Sawhney, H. S. (2008). Real-time global localization with a pre-built visual landmark database. In *CVPR*.