An Image-Based System for Urban Navigation

Duncan Robertson and Roberto Cipolla
Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
dpr20@eng.cam.ac.uk

Abstract

We describe the prototype of a system intended to allow a user to navigate in an urban environment using a mobile telephone equipped with a camera. The system uses a database of views of building facades to determine the pose of a query view provided by the user. Our method is based on a novel wide-baseline matching algorithm that can identify corresponding building facades in two views despite significant changes of viewpoint and lighting. We show that our system is capable of localising query views reliably in a large part of Cambridge city centre.

1 Introduction

This research is motivated by a conceptual mobile telephone navigation application. Using a handset with a built-in camera, the idea is that the user could transmit a photograph of his surroundings to a central computer system. Given approximate knowledge of the handset's position (obtained by signal triangulation or similar), the system would compare the query image with a database of stored images in order to determine pose. Navigation information could then be transmitted back to the user and projected into the image. Compared with radio signal triangulation-based schemes[1], this approach gives significantly better accuracy and determines camera orientation as well as position.

1.1 Existing Work

This is essentially similar to the *global localisation* problem familiar from the mobile robot navigation literature. *E.g.* in the system described by Se *et al*. [11], a robot uses a calibrated trinocular stereo rig to locate *visual landmarks*, which are simply image features with associated 3D positions. By matching visible landmarks with previously identified ones, the robot can determine its pose even if it is moved. To cope with large changes of viewpoint, image features are characterised in a way that is invariant with image rotation and scale [9]. This is *wide-baseline matching*.

In related work, wide-baseline matching has been used to identify similar views using corresponding image features [10, 12]. By characterising the features in a way that is invariant with affine transformations, it is possible to identify similar views despite substantial changes of viewpoint. For example, given a photograph of a building, the system

[1] Existing signal triangulation-based schemes (such as http://www.cursor-system.com) claim an accuracy of a few 10's of meters although performance may be reduced by multi-path effects in urban environments.

BMVC 2004 doi:10.5244/C.18.84

described by Shao *et al*. [12] can identify more photographs of the same building in a large database of photographs obtained from a wide range of viewpoints. However, their system does not determine the pose of the query.

Elsewhere, vanishing points have been used automatically to determine camera orientation in urban environments [2, 7, 13]. For example, Coorg and Teller [2] describe a system that uses vanishing points to determine the orientation of dominant building facades relative to a panoramic camera. However, a combination of specialist hardware is used to determine camera position, including GPS and inertial sensors. Here, only the image data is required. Stamos and Allen [13] have also used vanishing points as part of a pose determination algorithm. However, their system requires a detailed geometric model obtained by laser range finding and they have only localised views in the vicinity of a single building.

1.2 Approach

By contrast to the robot localisation system described by Se *et al*. [11], the idea here is to determine the pose of a query view by reference to a much simpler model comprising a database of rectified views of building facades. Building facades can be associated with a meaningful 3D coordinate system using readily available map data.

This is essentially an image database retrieval problem. Given a query view, the first step is to identify a nearby database view. Then the pose of the query view is obtained from the plane-to-plane transformation that relates it to the building facade. Our approach is based on a novel, wide-baseline matching algorithm that can identify corresponding building facades in two views in a way that is invariant with significant changes of viewpoint and lighting and robust to clutter.

We describe a system that has been shown to be capable of localising query views reliably in a large part of Cambridge city centre.

1.3 Review and notation

Under perspective projection $\tilde{\mathbf{u}} \sim \mathbf{P}\tilde{\mathbf{X}}$, homogenous pixel coordinates $\tilde{\mathbf{u}}$ are related to homogenous world coordinates $\tilde{\mathbf{X}}$ by a 3×4 projection matrix $\mathbf{P}$, where $\sim$ means equality up to scale. A projection matrix may be decomposed as $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad -\mathbf{R}^\top \mathbf{t}]$ where $\mathbf{R}$ is a 3×3 rotation matrix that describes camera orientation, $\mathbf{t}$ is the Euclidean camera position, and $\mathbf{K}$ is an upper triangular camera calibration matrix.

2 Wide-baseline matching

We begin by describing our wide-baseline matching algorithm. Given two views, the aim is to identify corresponding image features despite significant changes of viewpoint and lighting, and in a way that is robust to clutter.

2.1 Canonical views

The algorithm works by assuming that both views will contain a dominant plane in the form of a building facade[2]. By determining the orientation of the camera with respect to this plane, views may be transformed into a canonical frame by *metric rectification* [8] (see Figure 1c).

Camera orientation is determined using the vanishing points belonging to the principal horizontal and vertical directions that define the facade. By assuming that a significant proportion of imaged edges are aligned with these directions, the associated vanishing points can found automatically using the approach described by Kosecka and Zhang [7]. One problem is to decide which vanishing points belong to vertical and horizontal directions. To identify the vertical vanishing point $\tilde{\mathbf{v}}_v$, it is assumed that the camera is held approximately 'right way up'. Thus, $\tilde{\mathbf{v}}_v$ is chosen such that the vanishing direction $\mathbf{K}^{-1}\tilde{\mathbf{v}}_v$ is the one most nearly parallel to the vertical $[0 \quad 1 \quad 0]^\top$. A horizontal vanishing point $\tilde{\mathbf{v}}_h$ will be associated with a perpendicular direction, *i.e.* it should obey the constraint $(\mathbf{K}^{-1}\tilde{\mathbf{v}}_v)^\top(\mathbf{K}^{-1}\tilde{\mathbf{v}}_h) \approx 0$. If more than one horizontal vanishing point is detected, we select the one most strongly supported by line segments in the image.

Without loss of generality, a local coordinate system $X_F Y_F Z_F$ may be aligned with the building facade defined by the vertical and horizontal vanishing points. Hence, by considering points at infinity corresponding to the associated vanishing directions, it is simple to derive the following constraint on the elements of the projection matrix $\mathbf{P}_F$:

$$[\lambda_v\tilde{\mathbf{v}}_v \quad \lambda_h\tilde{\mathbf{v}}_h] = \mathbf{P}_F \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1}$$

where λ_v and λ_h are unknown scale factors and the subscript F denotes quantities in the local coordinate system. Writing $\mathbf{P}_F = \mathbf{K}[\mathbf{R}_F \quad -\mathbf{R}_F^\top\mathbf{t}_F]$, this equation can be rearranged and expressed in terms of camera calibration matrix $\mathbf{K}$ and camera orientation $\mathbf{R}_F$:

$$\mathbf{K}^{-1}[\lambda_v\tilde{\mathbf{v}}_v \quad \lambda_h\tilde{\mathbf{v}}_h] = \mathbf{R}_F \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{2}$$

Given $\mathbf{K}$, and by exploiting the properties of a rotation matrix, equation 2 can be solved simply for scale factors λ_v and λ_h, and camera orientation $\mathbf{R}_F$. In our conceptual mobile telephone navigation application, we assume for the time being that it would be possible to retrieve an approximate camera calibration matrix for each query view from a database of handset models. Camera calibration can be determined in the laboratory by photographing a suitable chessboard pattern [14].

Having determined camera orientation $\mathbf{R}_F$ we can rectify the view [8]. This is equivalent to rotating the camera by ${\mathbf{R}_F}^{-1}$ so that image plane is aligned building facade. Pixel coordinates $\tilde{\mathbf{u}}_f$ in the rectified view may be related to pixel coordinates $\tilde{\mathbf{u}}$ in the original view by the following equation:

$$\tilde{\mathbf{u}}_f \sim \mathbf{H}_f\tilde{\mathbf{u}} \tag{3}$$

[2]This assumption is not too restrictive in our mobile telephone navigation application, since the user can easily be instructed to point the camera at a building.

where $\mathbf{H}_\mathrm{f}$ is a 3×3 homography given by $\mathbf{H}_\mathrm{f} = \mathbf{K}_\mathrm{f}\mathbf{R}_\mathrm{F}{}^{-1}\mathbf{K}^{-1}$. Here, $\mathbf{K}_\mathrm{f}$ defines the origin and scale of the coordinate system for the canonical view and has the form:

$$\mathbf{K}_\mathrm{f} = \begin{bmatrix} \alpha_\mathrm{f} & 0 & u_{0\mathrm{f}} \\ 0 & \alpha_\mathrm{f} & v_{0\mathrm{f}} \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\mathbf{u}_{0\mathrm{f}} \sim [u_{0\mathrm{f}} \quad v_{0\mathrm{f}}]^\top$ is the origin and α_f is the scale. Usually it will be convenient (i) to set the α_f such that the transformation $\mathbf{H}_\mathrm{f}$ preserves the vertical dimension of the pixel at the centre of the original image (so the average change of scale is minimised), and (ii) to choose $\mathbf{u}_{0\mathrm{f}}$ such that the quadrilateral boundary of the transformed image fits in a bounding rectangle with top left corner $[0 \quad 0 \quad 1]^\top$.

Between rectified views, a building facade will be related by a simple scale-plus-translation transformation. Thus, pixel coordinates $\tilde{\mathbf{u}}'_\mathrm{f}$ in the first view may be related to pixel coordinates $\tilde{\mathbf{u}}_\mathrm{f}$ in the second by the following equation:

$$\tilde{\mathbf{u}}'_\mathrm{f} \sim \mathbf{H}_\mathrm{s}\tilde{\mathbf{u}}_\mathrm{f} \tag{5}$$

where $\mathbf{H}_\mathrm{s}$ is has the form:

$$\mathbf{H}_\mathrm{s} = \begin{bmatrix} \alpha_\mathrm{s} & 0 & u_{0\mathrm{s}} \\ 0 & \alpha_\mathrm{s} & v_{0\mathrm{s}} \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Here α_s is the scale factor and $[u_{0\mathrm{s}} \quad v_{0\mathrm{s}}]^\top$ is the translation in pixels.

Finally, let the *horizon line* be defined as the line of intersection of the horizontal plane defined by the camera's optical centre with the image plane, *i.e.* the camera's 'eye level'. In canonical views, the horizon line is a horizontal line that passes through the point $\mathbf{K}_\mathrm{f}[0 \quad 0 \quad 1]^\top$ (see Figure 1).

2.2 Feature detection and characterisation

Because the canonical views are free from perspective distortion, it is simple to detect and characterise image features in a way that is invariant with changes of viewpoint.

Firstly, interest points are located at the maxima of the Harris corner response, which is given by $\det(\mathbf{C}) - 0.05\mathrm{trace}^2(\mathbf{C})$ with:

$$\mathbf{C}(\mathbf{u}, \sigma, \overline{\sigma}) = G(\mathbf{u}, \overline{\sigma}) * \begin{bmatrix} L_u^2(\mathbf{u}, \sigma) & L_u L_v(\mathbf{u}, \sigma) \\ L_u L_v(\mathbf{u}, \sigma) & L_v^2(\mathbf{u}, \sigma) \end{bmatrix} \tag{7}$$

where $G(\mathbf{u}, \overline{\sigma})$ is a symmetric 2D Gaussian with standard deviation $\overline{\sigma}$, and L_u and L_v are image derivatives in the u and v directions respectively [5]. Differentiation is carried out by convolution with the differential of another symmetric 2D Gaussian with standard deviation σ. Here, smoothing and detection scales $\overline{\sigma}$ and σ are set to 1.5 and 1.0 pixels respectively.

Next, local image regions in the vicinity of the interest points are characterised by sampling RGB pixel values in a square grid pattern centred on the interest point. Good results have been obtained using a grid with dimensions 8×8 and a spacing of 2 pixels. Some amount of robustness to scale variation and feature localisation error is achieved by sampling the pixels from a version of the image that has been smoothed by convolution with a 2D Gaussian of standard deviation 2 pixels.

(a) (b) (c)

Figure 1: Obtaining canonical views. (a) Straight line segments are detected in left and right views and dominant vanishing points are recovered automatically. Line segments associated with the horizontal and vertical vanishing points are shown in (b). Finally, the images are transformed into a canonical frame corresponding to a rectified view of the dominant plane (c). In the rectified views, the horizon line is horizontal.

To account for larger changes of scale, feature detection is repeated at multiple image scales like in [3]. Here, the detector described above is applied to each level of a pyramid of scaled images. The base of the pyramid is the original image and successive levels are obtained from their predecessors by bicubic interpolation. Good results have been obtained using a pyramid with 5 levels, with each level 1.2 times smaller than its predecessor.

In the subsequent matching stage (Section 2.3, below), features are compared by evaluating the sum of squared differences between RGB pixel intensities [6]. To achieve robustness to lighting variation, pixel intensities $\mathbf{I}_{uv} = \begin{bmatrix} R & G & B \end{bmatrix}^\top$ are first normalized according to the following equation:

$$\hat{\mathbf{I}}_{uv} = \frac{\mathbf{I}_{uv} - \bar{\mathbf{I}}}{\sqrt{\frac{1}{N}\sum_{u,v} |\mathbf{I}_{uv} - \bar{\mathbf{I}}|^2}} \tag{8}$$

where $\hat{\mathbf{I}}_{uv}$ is the normalised value, $\bar{\mathbf{I}}$ is the mean, and N is the number of pixels (64 in this case). To compare features efficiently, a coarse-to-fine approach is used (after Burt and Adelson [1]). The idea is that a large proportion of candidate matches can be rejected within comparatively few operations by correlating smaller versions of the full-size template.

2.3 Feature matching

A robust, multi-scale matching approach is used, similar to the one described by Dufournaud *et al*. [3]. The idea is to match features detected at one scale in the first image with

features detected at another scale in the second, repeating matching for a succession of candidate scale relationships.

The complete matching algorithm is outlined below. Without loss of generality, the second image is considered to be the higher resolution one. In case the first image is the higher resolution one, this sequence is repeated with the first and second images reversed:

1. Detect and characterise features in the first image at a single scale $s = 1$. Detect and characterise features in the second image at a range of scales $s = 1.2^{-\rho}$, where detection level $\rho \in \{0, ..., 4\}$ (see Section 2.2).

2. Match features detected at level 0 in the first image with features detected at a candidate level ρ_c in the second, where $\rho_c \in \{0, ..., 4\}$.

3. For each candidate level, use RANSAC [4] robustly to estimate the scale-plus-translation transformation $\mathbf{H}_\mathrm{s}$. Redo matching constrained by this estimate. Count matches.

4. Finally, select the estimated transformation with the most matches.

At step 3, an initial set of correspondences could be obtained by unguided matching. However, computational efficiency can be improved by an order of magnitude by assuming that the views have been obtained from similar heights. This assumption is sufficient to fix one of the degrees of freedom of $\mathbf{H}_\mathrm{s}$ since the horizon lines must be aligned. Thus, for a particular candidate scale relationship, the search for a match in the second image can be restricted to a narrow margin surrounding a 1D scan line. In consequence, the proportion of outliers is usually greatly reduced.

3 Database formulation

The basis of our global localisation system is a database of views of building facades. At present, database views are obtained by photographing buildings with a hand-held digital camera[3]. Because our wide-baseline matching algorithm is effective despite significant changes of viewpoint, a single view of each facade is sufficient to localise query views obtained from a wide range of viewpoints.

Building facades in the database are associated with a meaningful 3D coordinate system using readily available map data. A point on the facade $\begin{bmatrix} X_\mathrm{F} & Y_\mathrm{F} & 0 & 1 \end{bmatrix}^\top$ may be related to pixel coordinates $\tilde{\mathbf{u}}_\mathrm{f}$ in the rectified view by a simple scale-plus-translation transformation:

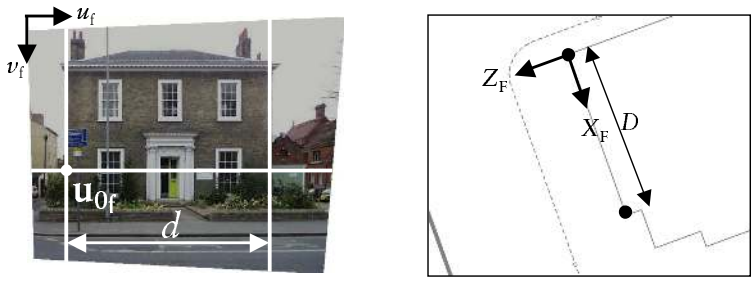
$$\tilde{\mathbf{u}}_\mathrm{f} \sim \mathbf{H}_\mathrm{a} \begin{bmatrix} X_\mathrm{F} & Y_\mathrm{F} & 1 \end{bmatrix}^\top \tag{9}$$

where

$$\mathbf{H}_\mathrm{a} = \begin{bmatrix} \alpha_\mathrm{a} & 0 & u_{0\mathrm{a}} \\ 0 & \alpha_\mathrm{a} & v_{0\mathrm{a}} \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Here α_a is a scale factor and $\begin{bmatrix} u_{0\mathrm{a}} & v_{0\mathrm{a}} \end{bmatrix}^\top$ is the translation. We find $\mathbf{H}_\mathrm{a}$ by identifying two vertical lines in each rectified view and their corresponding points on a map (see Figure

[3] We anticipate that this process could be automated by attaching a camera to a vehicle equipped with an inertial car navigation system. A navigation system could be used to provide camera pose and the map to provide the positions of building facades.



(a) (b)

Figure 2: Building facades in the database are associated with a meaningful coordinate system using a map. (a) Two vertical lines are identified manually in each database view (the horizon line is also shown). (b) These lines correspond to points on the map (circles). The point $\mathbf{u}_{0f}$ is the projection of origin of the facade coordinate system $X_F Y_F Z_F$.

2). Then $\alpha_a = D/d$ and $\begin{bmatrix} u_{0a} & v_{0a} & 1 \end{bmatrix}^\top$ is the projection of the origin of the facade's coordinate system $X_F Y_F$.

4 Global localisation

Having registered a set of database views, the pose of a query view can be determined automatically using the homography that relates it to a building facade in the database.

Database retrieval. The first step is to identify a nearby database view. Our system works by conducting two-view matching between the query view and each database view in turn using the method described in Section 2. We select the database view with the most robustly estimated matches in common with the query view. By assuming that a nearby database view has been obtained from a similar height to the query view, two-view matching may be conducted efficiently using the scan line constraint described in Section 2.3. To increase the speed of the database retrieval, rectification and feature detection is conducted in advance for database views. In our mobile telephone navigation application, prior knowledge of handset position (*e.g.* from cell location) could be used to constrain the database search to a few hundreds of views.

Pose determination. Having determined the scale-plus-translation transformation $\mathbf{H}_s$ that relates a building facade between the database and query views, all that remains is to find the pose of the query. In the local facade coordinate system, the required projection matrix $\mathbf{P}_F$ relates 3D points $\begin{bmatrix} X_F & Y_F & Z_F & 1 \end{bmatrix}^\top$ to pixel coordinates $\tilde{\mathbf{u}}$:

$$\tilde{\mathbf{u}} \sim \mathbf{P}_F \begin{bmatrix} X_F & Y_F & Z_F & 1 \end{bmatrix}^\top \tag{11}$$

Combining equations 3, 5, and 10, it is possible to write the following relationship between X_F, Y_F coordinates in the local coordinate system and image coordinates in the original query view:

$$\tilde{\mathbf{u}} \sim \mathbf{H}_f{}^{-1} \mathbf{H}_s \mathbf{H}_a \begin{bmatrix} X_F & Y_F & 1 \end{bmatrix}^\top \tag{12}$$

where $\mathbf{H}_a$ relates points in the facade coordinate system to pixel coordinates in the rectified view, $\mathbf{H}_s$ relates the dominant plane between the query and database views, and $\mathbf{H}_f{}^{-1}$ relates pixel coordinates in the canonical view to pixels coordinates in the original view.

Comparing equation 12 with equation 11, we can see that that it defines three columns of the projection matrix $\mathbf{P}_F$ (up to a single unknown scale factor). Since $\mathbf{K}$ and $\mathbf{R}_F$ are known, it is simple to recover the remaining column (up to the same scale factor). Finally, the camera pose estimate $\mathbf{R}_F$, $\mathbf{T}_F$ can be related from the facade's coordinate system to the map coordinate system by applying a coordinate system transformation.

5 Evaluation

To test our global localisation system, we constructed a database of views by photographing all the buildings in the main shopping street in Cambridge's city centre, as well as a number of other buildings of interest from around the city. Our database comprises 200 views in total (one per building typically) and spans at least 2 km of building facades. The area covered by the database spans an area several times greater than the positional uncertainty associated with existing mobile positioning systems.

We also revisited the area at different times of day and obtained query views from a variety of viewpoints. Compared to the database views, the query views were obtained at different distances from the dominant building facades and/or with different camera orientation. Usually distance differed by at least 30% and orientation by at least 30°. There were 97 query views in total. Many of our query images contained significant clutter (pedestrians, traffic, etc.) representative of that experienced in a city centre urban environment.

Using the framework described earlier, we attempted to determine the pose of each of 97 query views. Each query took around 10 s using a 1.4 GHz desktop PC (although our present implementation is not especially efficient). Pose determination results were verified by sketching building facade outlines in the database views and projecting them into the query views using recovered homographies (since camera focal length is known, this method gives a good indication of the accuracy of the pose estimates). Overall, 93 out of 97 queries were registered correctly. Representative results are shown in Figure 3. Because of our robust matching strategy, usually only one or two matches were found between a query view and incorrect database views. When the system did fail, this was because the database contains some photographs of buildings (or parts of buildings) that are virtually identical to each other.

6 Conclusions

We have described the prototype of a system designed to allow a user to navigate in an urban environment using a mobile telephone equipped with a camera. The system facilitates efficient determination of the pose of a query view by reference to a database of views of building facades.

One limitation is that some buildings (and parts of buildings) are very similar. This means that the system might be unable to distinguish between some viewpoints without more information, *e.g.* extra query views. Another limitation is that conducting two-view matching between the query view and every nearby database view is slow. A more

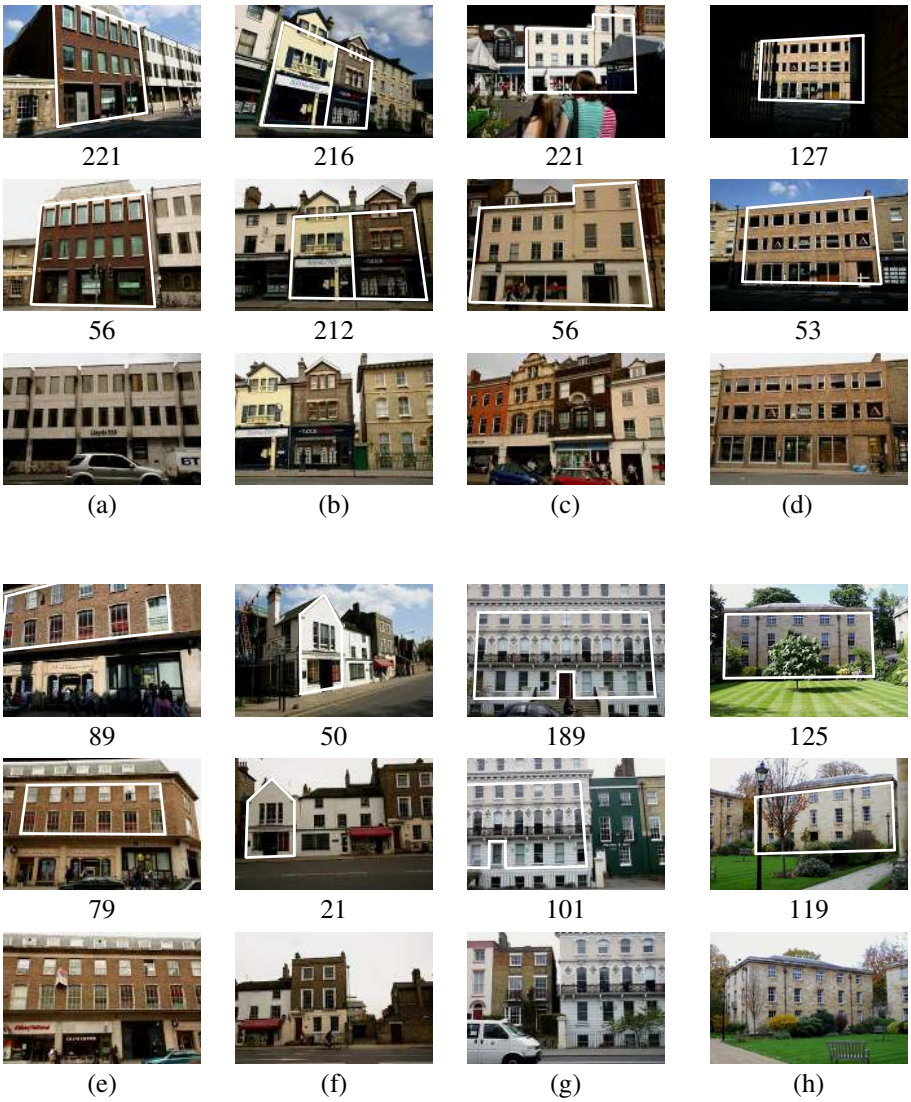


Figure 3: Illustrative database retrieval results. The first rows show query views. The second and third rows show the best and second best database retrieval results together with the number of robustly estimated correspondences. The first six images (a-f) show correct database retrieval results. Transferred building outlines demonstrate the accuracy of the recovered pose estimates. In (g), the correct database view has been recovered but the scale-plus-translation transformation is wrong. In (h), the wrong view has been retrieved because the two buildings are identical.

efficient strategy might be to use more 'global' image properties such as most frequent colours to eliminate unlikely database views in advance.

In this paper, camera intrinsic parameters have been assumed known. In recent work, we have extended our system to compute the focal length and coefficients of radial distortion for the query view automatically using vanishing points. In future research, we will explore the possibility of acquiring database views using a camera attached to a moving vehicle. Using an inertial car navigation system, it should be possible to register views automatically in the world coordinate system. Then Ordnance Survey map data could be used to provide the approximate location of dominant building facades.

References

[1] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[2] S. Coorg and S. Teller. Automatic extraction of textured vertical facades from pose imagery. Technical Report TR-759, Massachusetts Institute of Technology, 1998.

[3] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pages 612–618, 2000.

[4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381–395, 1981.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 189–192, 1988.

[6] T. Kanade and M. Okutomi. A stereo matching algorithm with an adapative window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[7] J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision (ECCV'02)*, pages 476–490, 2002.

[8] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Eurographics*, volume 18, pages 39–50, 1999.

[9] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, 1999.

[10] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision (ICCV'01)*, pages 525–531, 2001.

[11] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[12] H. Shao, T. Svoboda, T. Tuytelaars, and L. Van Gool. HPAT indexing for fast object/scene recognition based on local appearance. In *Computer Lecture Notes on Image and Video Retrieval*, pages 71–80, 2003.

[13] I. Stamos and P. K. Allen. 3D model construction using range and image data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pages 531–536, 2000.

[14] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.