Selecting surface features for accurate multi-camera surface reconstruction

Thomas Popham
tpopham@dcs.warwick.ac.uk
Roland Wilson
rgw@dcs.warwick.ac.uk

Signal and Image Processing Group
Department of Computer Science
University of Warwick
Coventry, CV4 7AL, England

Abstract

This paper proposes a novel feature detector for selecting local textures that are suitable for accurate multi-camera surface reconstruction, and in particular planar patch fitting techniques. This approach is in contrast to conventional feature detectors, which focus on repeatability under scale and affine transformations rather than suitability for multi-camera reconstruction techniques. The proposed detector selects local textures that are sensitive to affine transformations, which is a fundamental requirement for accurate patch fitting. The proposed detector is evaluated against the SIFT detector [11] on a synthetic dataset and the fitted patches are compared against ground truth. The experiments show that patches originating from the proposed detector are fitted more accurately to the visible surfaces than those originating from SIFT keypoints. In addition, the detector is evaluated on a performance capture studio dataset to show the real-world application of the proposed detector.

1 Introduction

Establishing correspondences between different views of an object is a longstanding problem in computer vision and is a fundamental requirement in many computer vision applications, such as robotic navigation [19], camera calibration [23] and augmented reality [5]. One approach to the problem is to detect features in every view and then find correspondences by matching robust feature descriptors across different views [1, 11, 12, 25]. A second approach is to detect features in a reference image and then use standard stereo reconstruction techniques [20] to find the correspondences in the other images [15]. In the first approach, the key requirement upon the feature detector is that it should extract the same feature even with changes in scale and viewpoint. In the second approach, the key requirement is that it will select the best textures for surface reconstruction. This paper addresses this requirement, by asking the question: how should local image textures be selected, so that the visible surface can be accurately reconstructed using stereo techniques? In particular, we focus on planar patch fitting techniques [6, 17], as these offer a higher level of generality than techniques assume the surface surface is parallel to the image plane (the fronto-parallel assumption).

The contribution of this paper is a novel feature detector that extracts image textures for which a planar patch can be accurately fitted to the corresponding scene surface. The

BMVC 2009 doi:10.5244/C.23.107

proposed feature detector is based upon finding textures that are sensitive to shear transformations, which are part of the chain of projective transformations between two cameras via a plane [8]. The performance of the proposed detector is compared with a SIFT keypoint extractor [11] using a synthetic dataset and results are also shown on a real-world dataset. The remainder of the paper is organised as follows: section 2 reviews some of main feature detectors and descriptors as well as some planar patch fitting techniques; section 3 explains the planar patch fitting approach used in this paper; section 4 describes the proposed feature detector; section 5 shows some experimental results on synthetic and real datasets; and section 6 concludes the paper.

2 Related work

Numerous feature detectors have been proposed in the literature. The Harris [7] and KLT detectors [22] belong to the family of detectors which use a second moment matrix of image derivatives to find points for which the intensity change is significant in orthogonal directions. Another family of feature detectors is based upon the Hessian matrix of second derivatives [2] and these tend to detect 'blob-like' features. Since image features may exist at different scales within the image, many feature detectors adopt a multiresolution approach to find the characteristic scale of the feature [10]. Both Harris-Laplace and Hessian-Laplace detectors [13] are examples of extensions to earlier feature detectors. The difference-of-Gaussian detector also selects scale-space extrema and is an approximation to a Laplacian filter [9, 11]. More recently, affine-invariant detectors/descriptors have been proposed to improve the repeatability performance with changes in viewpoint [12, 13, 25].

In order to compare the various feature detectors, many performance measures have been suggested, including: Repeatability, Distinctiveness, Locality, Quantity, Accuracy and Efficiency [24]. Schmid *et al*. [18] evaluated interest point detectors using the repeatability rate and the information content and found that their improved version of the Harris detector [7] offered the best performance. Mikolajczyk *et al*. [14] compared the performance of six affine region detectors under changes in viewpoint, scale, illumination, defocus and image compression. Moreels and Perona [16] conducted a similar experiment using a set of 100 3d objects and found that Hessian-Affine feature detector performed best with changes in viewpoint and that no detector-descriptor combination performs well for viewpoint changes larger than 25 to 30 degrees.

As an alternative to affine/scale invariant features, this paper uses multi-camera reconstruction techniques to derive correspondences between features and therefore estimate their 3d position. In particular we review planar patch fitting methods which estimate a depth and surface orientation for a given rectangular region in the image. Carceroni and Kutulakos [4] fit a set of 'surfels' to the scene surfaces by partitioning the scene volume into a set of voxels and then searching for a surface element in each voxel. Birchfield and Tomasi [3] iteratively segment the scene into non-overlapping regions and estimate the affine parameters of each region in order to find a displacement map for scenes with slanted surfaces. Habbecke and Kobbelt [6] fit a dense set of planes to the surfaces by iteratively minimising the following objective function for each planar patch:

$$E = \sum_{c=2} \sum_{p \in \Omega} \left(I_1(p) - I_c(H(N)p) \right)^2 \tag{1}$$

where I_1 is the reference image, $I_2, \ldots, I_n$ is the set of comparison images, Ω is set of pixels

p belonging to the patch and $H(N)$ is the homography H as a function of the plane parameters N. Mullins *et al*. adopt a multiresolution approach to patch estimation by employing a particle filter to stochastically refine patch estimates in a coarse-to-fine manner [17].

3 Overview of plane fitting approach

The problem of fitting a planar patch to the scene surface from an imaged feature is now defined. It is assumed that each imaged feature corresponds to a locally planar surface patch within the scene. Let $g = (u,v,1)$ be the homogeneous co-ordinates of a feature point, let s be the scale at which the feature was detected and let w be the set of pixels belonging to a rectangular window around the feature point. A point $g = (u,v,1)$ on the image plane of a camera defines a unique ray from the camera centre into the scene, along which the centre of planar patch must lie. Therefore only one parameter, the depth d, must be determined in order to find the 3d location of the patch centre. In addition to the location of the patch centre, the surface orientation of the patch must also be estimated, adding another two parameters to the search: the angles θ_1 and θ_2. θ_1 is the angle between the surface normal and the z-axis and θ_2 is the angle between the x-axis and the projection of the surface normal onto the x-y plane. The notation $x = [d, \theta_1, \theta_2]$ is used for the state vector containing the parameters of the patch and $z = \{z_1, z_2, ..., z_c\}$ is the set of images from the c cameras surrounding the scene. The problem is to therefore find the parameters contained in the vector x with maximum probability given the input images:

$$\hat{x} = \underset{(d,\theta_1,\theta_2)\in S}{argmax}\ p(x|z) \tag{2}$$

where S is the space containing all possible combinations of (d, θ_1 and θ_2) and $p(x|z)$ is the probability of the state x given the images $z = \{z_1, z_2, ..., z_n\}$. Using Bayes' theorem, $p(z|x)$ is proportional to $p(x|z)$, if $p(x)$ is assumed to be uniform:

$$p(x|z) \propto p(z|x) \tag{3}$$

The conditional probability of the input images given the hypothesised state $p(z|x)$ is estimated by using the appearance consistency between the input images via the hypothesised plane. Two views of a plane may be related to each by using a homography to describe the mapping between corresponding points in each view [8]. A homography H is a 3x3 matrix which transforms homogeneous co-ordinates of a point in one view to the corresponding homogeneous co-ordinates of the point in a second view according to:

$$g_j = H_{ij} g_i \tag{4}$$

where g_i and g_j are the homogeneous co-ordinates $(u,v,1)^T$ of the points in camera i and camera j and H_{ij} is 3x3 matrix describing the homography between camera i and camera j. Since the cameras are calibrated, the homography only has three degrees of freedom. The homography H_{ij} may be directly computed from the camera geometry and the plane $n.X + d = 0$ with $n = (\pi_1, \pi_2, \pi_3)^T$ and $X = (x,y,z)^T$ [8]:

$$H_{ij} = K_j \left(R_j - t_j n^T / d\right) K_i^{-1} \tag{5}$$

where K_i and K_j are the 3x3 internal calibration matrices for cameras i and j, R_j is the rotation matrix for camera j and t_j is the camera centre of camera j. It is assumed in the

above expression that the centre of camera i lies at the origin, so that the 3x4 projection matrix is: $P_i = K_i[I|0]$, where I is a 3x3 matrix and 0 is a 3x1 matrix. In order to simplify the following notation, $H_{ij}(x)$ is used to denote the homography between cameras i and j via the plane parametrised by the state vector x. The reprojection error (using sum-of-squared differences) between two camera views of a plane parametrised by state vector x is therefore:

$$\varepsilon^2_{i\to j}(x) = \sum_{(u,v)\in w} \left(z_j(g_j) - z_i(H_{ji}(x)g_j)\right)^2 \tag{6}$$

where $z_j(g_j)$ is the interpolated intensity of the image in camera j at the point g_j. For clarity, the equation above assumes greyscale intensities, although colour information can be incorporated by summing over the 3 colour components. Where multiple cameras are available, the reprojection errors from the cameras are summed:

$$\varepsilon^2(x) = \sum_{j\in\mathscr{C}} \varepsilon^2_{i\to j}(x) \tag{7}$$

where $\mathscr{C}$ is the set of cameras. The reprojection error may be used to calculate the probability $p(z|x)$ by assuming that the reprojection error is normal i.i.d:

$$p(z|x) \propto e^{\left(-\frac{\varepsilon^2(x)}{2\sigma^2}\right)} \tag{8}$$

where σ is usually an empirically determined constant controlling the spread of reprojection errors. In this paper, the probability $p(z|x)$ is maximised by using a full search to ensure that the results are not affected by an optimization technique getting stuck in local maxima. However for practical applications, stochastic or gradient-descent strategies in [17] or [6] may be used. Since the task of estimating patches is ill-defined for surfaces which are perpendicular to the camera image plane, the estimation accuracy is poor for these patches. Therefore patches are removed if the angle between the patch and the camera image plane is greater than a threshold angle.

4 Detecting image features for accurate plane fitting from multiple images

In order to estimate the parameters x of the plane that minimises the objective function in equation (7), a natural requirement is that only the true plane parameters minimise the objective function. With multiple cameras, the depth of the feature is usually sufficiently constrained, but accurate surface normal estimation is often difficult, as different orientations of the plane may give rise to similar reprojection errors. The plane fitting task is therefore only possible when each combination of θ_1 and θ_2 transforms the feature texture to produce a unique texture. In other words, θ_1 and θ_2 should have an orthogonal effect upon the transformed texture. One possible way of ensuring that the surface orientation can be correctly estimated would be to perform a check to ensure that every combination of θ_1 and θ_2 produces a unique texture. However, this would be a very computationally intensive task, since it would have to be performed on a texture window for each pixel in the input image. The proposed method therefore checks the sensitivity of the local texture to an affine transformation, and in particular a shear transformation is used. Both x-shear and y-shear transformations are parameterised as follows:

$$H_{sx}(c_x) = \begin{bmatrix} 1 & c_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad H_{sy}(c_y) = \begin{bmatrix} 1 & 0 & 0 \\ c_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where both H_{sx} and H_{sy} are transformations of a point $(x,y,1)^T$ and c_x,c_y control the amount of shear in the x and y directions. The sensitivities $r_x(u,v)$ and $r_y(u,v)$ of the texture window f to the transformations are defined as:

$$r_x(u,v) = \left(\frac{\partial f}{\partial c_x}\right)^2 \qquad r_y(u,v) = \left(\frac{\partial f}{\partial c_y}\right)^2 \tag{10}$$

The gradients are estimated by comparing an original texture window $f(m,n)$ against a small transformation of the texture:

$$r_x(u,v) = \sum_{(m,n)} (f(m+u,n+v) - f(m+u+c_x n, n+v))^2 \tag{11}$$

$$r_y(u,v) = \sum_{(m,n)} (f(m+u,n+v) - f(m+u, n+v+c_y m))^2 \tag{12}$$

Since two parameters must be estimated for the surface orientation, we want the texture to be responsive to shear transformations in both the x and y directions. The actual response $a(u,v)$ at each pixel is therefore the square-root of the product of the two responses:

$$a(u,v) = \sqrt{r_x(u,v) r_y(u,v)} \tag{13}$$

Since the gradient of the texture response to the transform is being measured, c_x and c_y need to only cause a small change in the transformed texture. Through experimentation, it was found that the setting c_x and c_y to 0.1 caused a sufficient transformation of the texture for the estimation of the gradients in equations 11 and 12. In common with other detectors, a feature is added at location (u,v) if its response $a(u,v)$ is both a local maximum and above a threshold T. Since textures that are suitable for multi-camera reconstruction may be present at multiple scales in the image, the proposed detector is run at multiple levels of a Gaussian pyramid.

For the experiments in this paper, the responses r_x and r_y were calculated using a transformed texture window of 7x7 pixels, but for practical applications, an optimization could be made by taking a Taylor expansion around the centre point (u,v) of the texture window (with (m,n) as an offset from the centre point):

$$f(u+m,v+n) = f(u,v) + m\frac{\partial f}{\partial u} + n\frac{\partial f}{\partial v} \tag{14}$$

so that the responses $r_x(u,v)$ and $r_y(u,v)$ are:

$$r_x(u,v) = \sum_{(m,n)} \left(c_x n \frac{\partial f}{\partial u}\right)^2 \qquad r_y(u,v) = \sum_{(m,n)} \left(c_y m \frac{\partial f}{\partial v}\right)^2 \tag{15}$$

5 Experimental Results and Discussion

5.1 Experiment 1 - Evaluation on synthetic data

The proposed detector was compared with a SIFT detector for its ability to select suitable surface features for accurate multi-camera reconstruction. In order to obtain a meaningful comparison, experiments were conducted on a synthetic dataset so that the estimated patches could be compared against the ground truth. The synthetic dataset was created using 32 views of a textured cube and was rendered using OpenGL. The cube was textured using the benchmark 'graffiti' image from the feature performance comparison by Mikolajczyk *et al*. [14]. Figure 1 shows an example camera view of the scene and the general camera setup.

A single camera situated in the middle of the top-row (shown as yellow in figure 1) was used as the input to both feature detectors. The SIFT features were detected using the SIFT++ implementation available from [26]. 63 SIFT features were extracted using a scale-space threshold of 0.06 and an edge threshold of 10. The proposed detector produced 67 features and was run using $c_x = c_y = 0.1$ in equations 11 and 12, a threshold $T = 6$, and a window size of 7x7 pixels. Both feature detectors used the same Gaussian pyramid with a scale factor of 1.26 between layers to give 3 levels per octave.

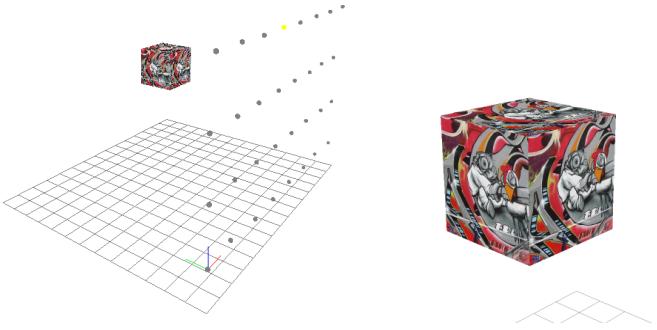
Patches were then estimated for both sets of features using the other 31 cameras to maximise the probability in equation 8. Out of the 63 features from the proposed detector, 50 patches were estimated and from the 67 SIFT features, 52 patches were estimated. The reason for having fewer patches than image features is that patches with an angle greater than 40 degrees to the camera image plane were removed, due to poor fitting results. In general, these removed patches corresponded to the features on the edge of the cube.

The patches originating from each detector were then compared against the ground truth to give a root mean square error $error_{rms}$ for each estimated parameter:

$$error_{rms} = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(\hat{x}_i - x_i)^2} \tag{16}$$

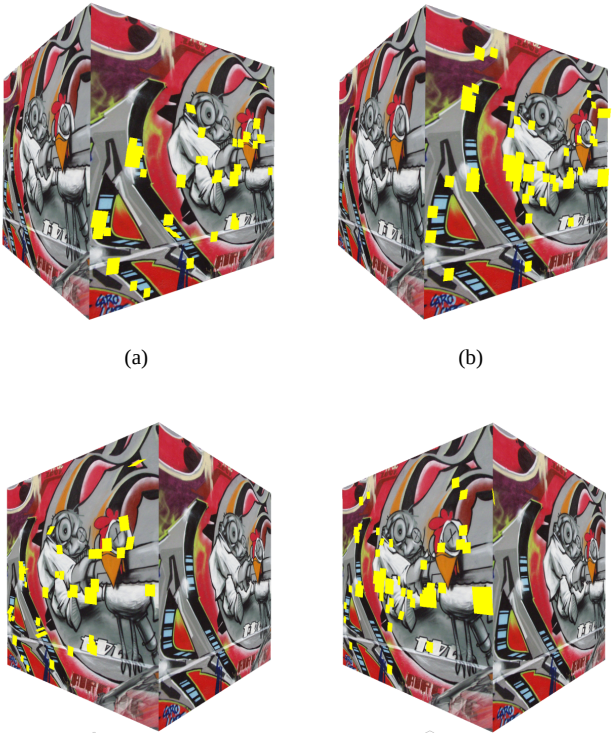
where N_p is the number of patches, $\hat{x}_i$ is the estimated parameter for patch i and x_i is the ground truth parameter for patch i.

Figure 2 shows the estimated patches from both the proposed detector and the SIFT detector and figure 3 shows zoomed-in views of the bottom left-hand corner of the cube. It can be seen from these images that the new detector gives a better fit than the SIFT features. This is born out by table 1, which shows the root mean squared errors against ground truth for both types of features. The surface orientation parameters are given in radians and the depth error measurements are relative to the cube which has sides of unit length. As can be seen from the table and figures, the patches originating from the proposed detector are more accurately fitted to the scene surface than patches originating from the SIFT keypoint extractor. The accuracy of depth estimation is slightly better for the patches originating from the proposed detector, but the real benefit of the proposed detector is shown in the improved surface normal estimation accuracy.



(a) (b)

Figure 1: a) Camera Setup b) Example view from one camera



(a) (b)

(c) (d)

Figure 2: (a,c) Estimated patches originating from SIFT features (b,d) Estimated patches originating from proposed detector

Feature Detector	Root mean squared errors		
	Depth d	θ_1 (radians)	θ_2 (radians)
Proposed method	0.0009	0.07079	0.04257
SIFT	0.0012	0.08986	0.15115

Table 1: Root mean squared errors estimated for depth (d) and surface orientation parameters (θ_1,θ_2). The depth errors are relative to the cube which has sides of unit length.

(a)

(b)

Figure 3: (a) Estimated patches originating from SIFT features (b) Estimated patches originating from proposed detector

5.2 Experiment 2 - Evaluation on real-world data

In order to show a real-world application of the proposed detector and plane-fitting algorithm, an evaluation is performed on a studio dataset captured with 32 firewire cameras at a resolution of 1024x768 pixels. The cameras have the same layout as those in figure 1, and were calibrated using the automatic package described by Svoboda *et al*. [21]. Since the shape of the subject is more complex than a simple cube, features are detected in the middle 4 cameras along the top-row (see figure 1). Figure 4 shows the patches overlaid onto the two different views of the scene. Although there is no ground truth for these data, it is clear that the estimated patches are close to the relevant surfaces.

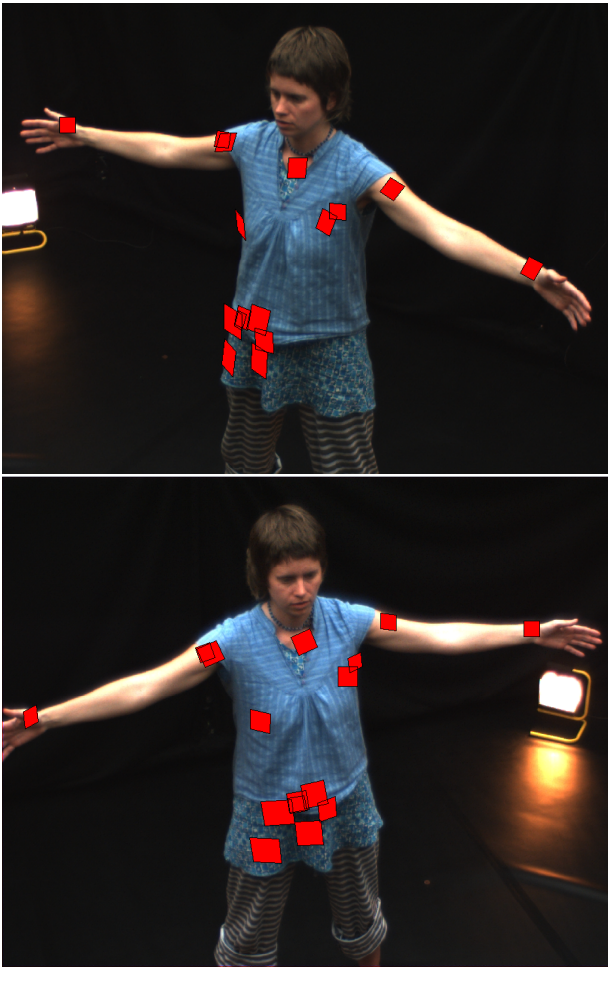


Figure 4: Two views of estimated patches from proposed feature detector

6 Conclusions

We have presented a novel feature detector that selects local textures for which planar patches can be fitted accurately to the corresponding scene surfaces. The performance of the proposed detector has been compared with a SIFT detector and it has been shown that patches are more accurately fitted when they originate from the proposed feature detector. There are many possible improvement or extensions to this work, including: testing on a wider range of synthetic objects and textures; a full comparison with other feature detectors; and finally, a more optimized implementation of the detector.

References

[1] A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings*, volume 1, 2000.

[2] P.R. Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, volume 579, page 583, 1978.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, 1999.

[4] R.L. Carceroni and K.N. Kutulakos. Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion, Shape and Reflectance. *International Journal of Computer Vision*, 49(2):175–214, 2002.

[5] V. Ferrari, T. Tuytelaars, and L. Van Gool. Markerless augmented reality with a real-time affine region tracker. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality*, pages 87–96, 2001.

[6] M. Habbecke and L. Kobbelt. A Surface-Growing Approach to Multi-View Stereo Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50, 1988.

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of applied statistics*, 1994.

[10] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[11] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[15] H.P. Moravec. *Robot spatial perception by stereoscopic vision and 3D evidence grids*. Carnegie Mellon University, The Robotics Institute, 1996.

[16] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[17] A. Mullins, A. Bowen, R. Wilson, and N. Rajpoot. Multiresolution particle filters in image processing. In *Proceedings of the Mathematics in Signal Processing Conference*, 2006.

[18] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.

[19] S. Se, DG Lowe, and JJ Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[20] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *Int. Conf. on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[21] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & Virtual Environments*, 14(4):407–422, 2005.

[22] C. Tomasi and T. Kanade. Detection and tracking of point features. *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91–132*, 1991.

[23] P. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *Computer Vision, 1998. Sixth International Conference on*, pages 727–732, 1998.

[24] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

[25] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[26] A Vedaldi. SIFT++: a lightweight C++ implementation of SIFT detector. `http://www.vlfeat.org/~vedaldi/code/siftpp.html`.