We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 7

Generation of 3D Sparse Feature Models Using Multiple Stereo Views

Matthew Watson, Asim Bhatti, Hamid Abdi and Saeid Nahavandi

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/45905

1. Introduction

Augmented Reality (AR) renders virtual information onto objects in the real world. This new user interface paradigm presents a seamless blend of the virtual and real, where the convergence of the two is difficult to discern. However, errors in the registration of the real and virtual worlds are common and often destroy the AR illusion. To achieve accurate and efficient registration, the pose of real objects must be resolved in a quick and precise manner.

An augmented world is presented to a user through an interface such as a head mounted display or tablet computer. To achieve the AR illusion, the relationship between the viewing interface and the anchor on which to render information in freespace (the real 3D environment) must be found. This calculation of **pose** (position and orientation relative to the user) enables the world coordinates of the virtual content to be translated to match the real world coordinates of the render anchor so that the virtual content can be aligned or *registered* into reality. The term 'registration' refers to the precise alignment of one or several virtual coordinate system(s) to real world entities.

Vision sensors offer a passive, detailed, non-invasive and low cost method for establishing a pose estimate for AR applications (Lepetit & Fua, 2005). Two common vision based approach's are:

1. Egomotion, and
2. Recognition

Egomotion establishes the 3D motion of a camera in freespace by monitoring visual flow or tracking salient but uncorrelated features in a scene frame by frame. Conversely, recognition estimates the pose of specific entities based on locally related and known features. Egomotion is a scene-based technique used to localise the pose of a camera from an arbitrary

initial point, where as recognition detects and tracks local coordinate systems of independent, known entities relative to a current perspective. Egomotion-based systems only allow information to appear in user specified regions, with no synchronicity with objects in the real world. Through recognition, a system can **perceive** specific entities in an environment, and seamlessly augment information that **directly corresponds** to those entities. When a system knows what it is looking at, it can deliver contextual information to a user.

Pre-learnt information is termed *a priori* knowledge and can assist a vision system to recognise object in freespace. A priori knowledge is assumed to be an accurate representation of the object, requiring no validation or justification by further experience. Imparting a computer system with a priori knowledge requires some anterior experience with the object. Typically, an offline learning stage is used to sample information from an object, which is stored in a database as a true representation of the object. When a recognition system runs online, the current data it is sampling from the world is referenced back to this database to see whether the object exists in the current environment. If recognised, the pose of that object can be determined through further processing. The accuracy of the pose estimate directly corresponds to the quality of registration attainable.

Generating a priori data for this purpose requires some careful considerations as to the type of information present in the dataset. Characterising an object with naturally occurring local features produces a distinct object representation. This form is generally considered (Lepetit & Fua, 2005) to be a robust method of classifying and recognising multiple objects with a vision sensor. (Rothganger et al., 2003) note that building this type data from multiple views offers a more complete and robust data set than a representation built from any single view.

View clustering was introduced by (Lowe, 2001) to create a complete object representation by blending a set of training images captured from different locations around a view sphere. Lowe grouped similar images by the quality of the feature matches between the images. Similar to Lowe's view clustering methodology, (Schaffalitzky & Zisserman, 2002) spatially organised multiple unordered views of a scene into clusters based on the similarity between the views. Using the 'now standard' wide baseline stereo approach, invariant descriptors were matched between images using a binary space partition tree. After filtering for outliers and incorrect matches, a greedy algorithm was used to join the subset of images together into a complete model.

(Gordon & Lowe, 2006) built upon Schaffalitzky and Zisserman's framework, to generate a 'metrically accurate 3D model of an object and all its feature`e locations'. The model was built by matching highly descriptive SIFT features (Lowe, 2004) between multiple views in an unordered image set. The greedy algorithm of (Schaffalitzky & Zisserman, 2002) was used to construct a spanning tree to cluster similar views together. Multiple 2D feature correspondences were found by traversing this tree. From those matches, they recovered the projective parameters between views and estimated the 3D locations of the 2D features.

Monocular wide baseline stereo techniques such as (Gordon & Lowe, 2006) and (Schaffalitzky & Zisserman, 2002) can offer more spatial information than any single view

systems, however these algorithms have to compensate for a high risk of viewpoint related occlusions and less accurate interest point localisation (Bay, 2006). A short baseline stereo system simplifies the correspondence problem considerably and has few viewpoint related occlusions meaning that they have the potential to deliver a denser feature match set. Segmenting features based on their relative depth also allow a short baseline system to be robust against incorrect foreground/background matches.

This chapter investigates the generation of a priori data. In the proposed methodology, detailed features of an object are first matched between multiple short-baseline stereo pairs to produce dense depth maps. Several stereo pairs are then fused together to form a single model representation of an object, producing a dense model with higher resolution than it's wide baseline counterparts termed the *Sparse Feature Model* (SFM).

2. A priori data and the sparse feature model

We classify n objects of interest as $\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_n$. For the k-th object of interest, a group F_k of features $\mathbf{f} = [f_1, f_2, ..., f_m]^T$ is extracted, where m is the dimension of the feature vector. Figure 1 shows the features $\mathbf{f}$, grouped as F_k, with reference to the k-th object's coordinate system $(\mathrm{O_k}, \vec{i}_k, \vec{j}_k, \vec{k}_k)$ and the imaging device coordinate system $(\mathrm{C}, \vec{i}_c, \vec{j}_c, \vec{k}_c)$ in freespace.

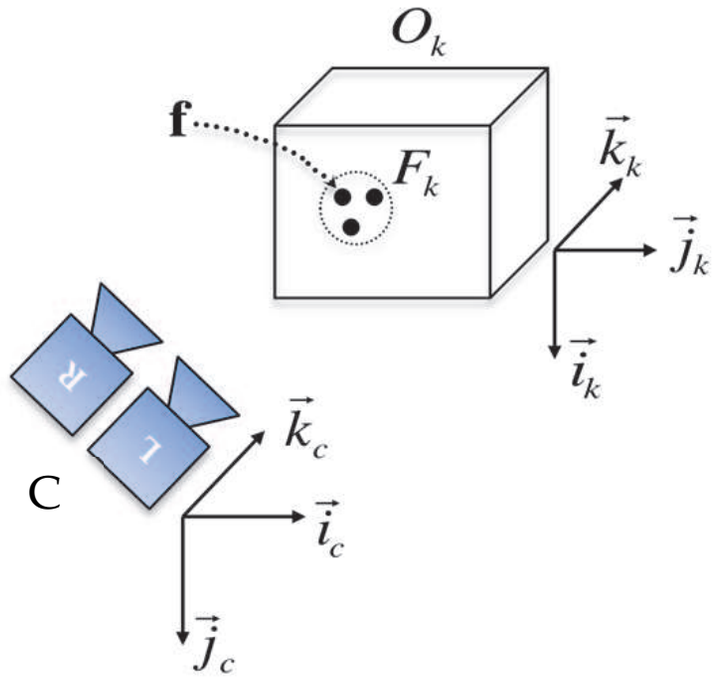

Figure 1. Features, feature set, k-th object coordinate system and imaging device coordinate

This chapter introduces a methodology to generate *a priori* data in the form of a Sparse Feature Model (SFM). A SFM is a concise representation of an object, where each point in model represents the 3D location of a highly descriptive 2D image features. To construct this model, an object $\mathbf{O}_k$ is imaged from multiple perspectives using a short baseline stereo camera $\mathbf{C}$. For each stereo pair, a feature extraction method locates robust and repeatable

interest points to generate feature sets $F^L_{k,i}$ and $F^R_{k,i}$, where L and R represent left and right images, and i is the i-th view of the k-th object. Correspondence between features in $F^L_{k,i}$ and $F^R_{k,i}$ is established for each i-th view. These corresponding features are triangulated to generate a 2.5D perspective view $M_{k,i}$. Finally, a 3D shape registration technique merges each 2.5D perspective view $M_{k,i}$ into a unified 3D representation M_k, termed the Sparse Feature Model.

If the multi-view merging process is shown by $\bigcup$ then

$$M_{k,i} = F^L_{k,i} \bigcup F^R_{k,i} \tag{1}$$

$$M_k = \bigcup_i M_{k,i} \tag{2}$$

Where M_k is the SFM representation of the k-th object O_k. This procedure is shown graphically in Figure 2, where the operator $\bigcup$ is merger operator.

Note that the merging operator $\bigcup$ is different from the normal mathematic operator of union due of the correspondence and the matching process. During correspondence, any two matched features might be exactly similar or a little bit different from each other. With the merger operator $\bigcup$ a hybrid feature calculated from the two matched features is carried forward. In a traditional union, both would be carried forward.

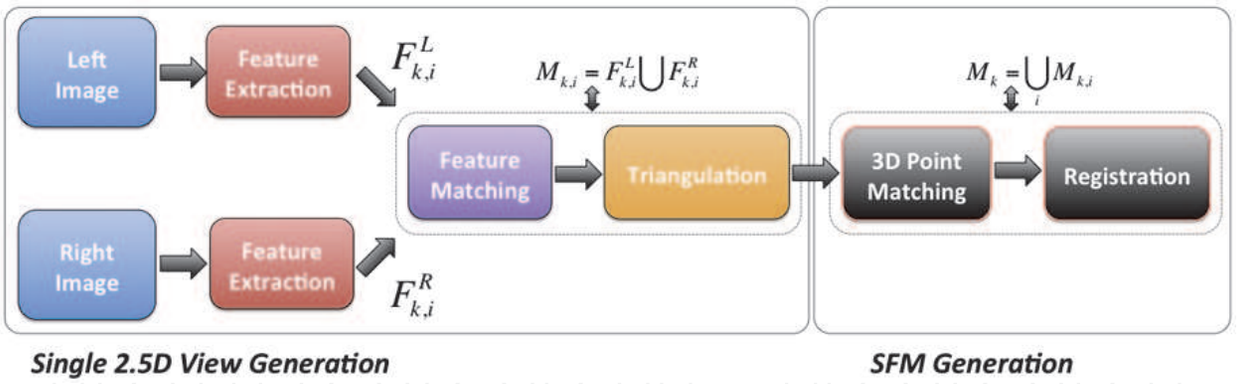

Figure 2. Block diagram of the 3D SFM generation for the k-th object

2.1. Assumptions

There are n number of objects of interest that we want to generate spare feature model from multiple ordered pairs of short baseline stereo images. SURF features (Bay et al., 2008) are extracted in each stereo pair. The SURF feature algorithm builds a feature vector from appearance of local neighbourhood of pixels surround a feature of interest. Therefore, this method is suitable for textured objects. When producing a sparse feature model we assume that a textured object is imaged in an uncluttered environment to ensure that SFM contains a set of features that only represent that object of interest. The 3D principles of a calibrated short baseline stereo system is used to segment the object from the foreground and

background to ensure that only features generated from the appearance of the object appear in the SFM. These assumptions help build a sparse feature models for different objects that accurately represents the unique arrangement of local features of each object, and are therefore suitable for pose estimation via recognition.

3. Short baseline stereo imaging

From Figure 2, the first step of our methodology is to use a short baseline stereo camera system to synchronously capture two images, left and right, from slightly different perspectives. Figure 3 shows the stereo capturing system that is used in this study.

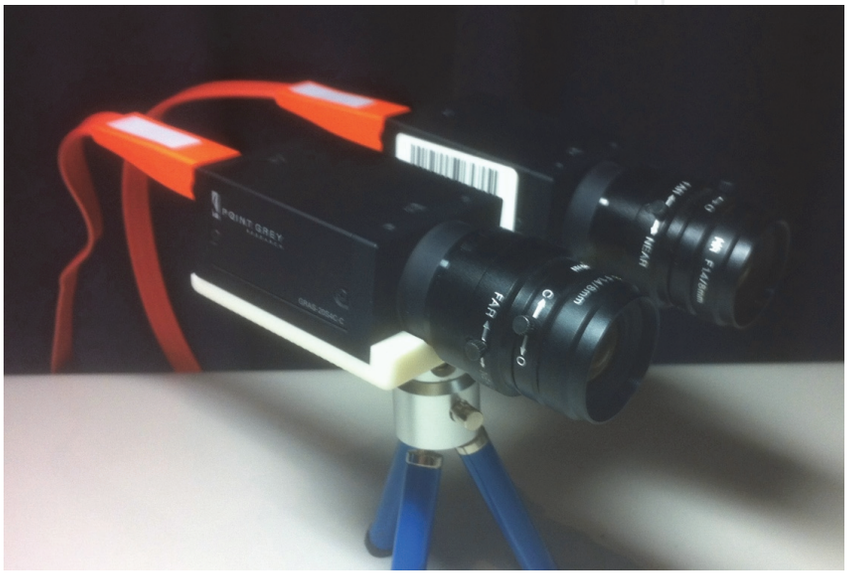

Figure 3. Stereo camera setup for the study and its calibration parameters

3.1. Camera calibration

The calibration of two pinhole type cameras in a fixed baseline stereo arrangement as in Figure 3 is a common procedure. There are many freely available toolkits, including the camera calibration toolbox for Matlab (Bouguet, 2010) and calibration routines in OpenCV (Vezhnevets et al., 2011). We assume that the stereo cameras used in the imaging device are pre-calibrated and that the intrinsic and extrinsic matrices are known. For more information on stereo calibration, see (Hartley and Zisserman, 2003). The camera calibration parameters for the stereo rig in Figure 3 are listed in Table 1. The stereo rig was calibrated using Jean-Yves Bouguet Camera Calibration Toolbox for Matlab (Bouguet, 2010).

3.1.1. Extrinsic parameters (Bouguet, 2010)

- **Om** relates to a rotation R of the left camera relative to the right by the Rodrigues' formula R = Rodrigues(om).
- **T** is the translation of the right camera with respect to the left, signifying that the camera centre of the right camera is situated 68mm away from the left.

3.1.2. Intrinsic parameters (Bouguet, 2010)

- **Focal Length (L and R)** are the focal lengths of each camera
- **Principle Point (L and R)** are the 2D image coordinates of the camera centres.
- **α (L and R)** is the angle of skew of a pixel. In this case the pixels of the cameras were estimated to be perfectly square.
- The 5x1 **distortions** vector holds the coefficients for the radial and tangential distortions of the camera lenses.

Parameter	Value
Extrinsic Parameters	
Om	[0.0045 ; 0.0066 ; 0.0006]
T	[68.0796 ; 0.0041 ; -0.0003]
Intrinsic Parameters	
Focal Length L	[1901.4 ; 1901.8]
Focal Length R	[1893.0 ; 1894.2]
Principle Point L	[811.3492 ; 611.1065]
Principle Point R	[805.1364 ; 649.4665]
α L (pixel skew)	0
α R (pixel skew)	0
Image Distortions L	[-0.1168 ; 0.3025 ; 0 ; 0 ; 0]
Image Distortions R	[-0.1106 ; 0.1934 ; 0 ; 0 ; 0]

Table 1. Camera calibration parameters

3.2. Two-view geometry

The mathematical nature of multiple-view computer vision is a mature topic of research (Faugeras, 1993; Faugeras & Luong, 2001; Hartley and Zisserman, 2003). The axioms of two-view geometry describe the intrinsic relationship between two images taken from slightly different perspective views of a 3D scene highlighted in Figure 4. In this figure, the left and right image planes are shown in a 3D coordinate system X,Y,Z. A 3D interest point of $\mathbf{p} = (x_k, y_k, z_k)$ of the k-th object has a 2D projection in the left and right images denoted as (u_i, v_i) and $(\hat{u}_i, \hat{v}_i)$ where the ray intersects the image plane on a path towards the camera centre. These 2D projections are obtained from the two projection matrices that map the interest point $\mathbf{p}$ on both images. These projection matrices come from the camera calibration parameters. If $\mathbf{P}_L$ and $\mathbf{P}_R$ are the two 3x4 projection matrices for the left and right images, then

$$\varsigma_L \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{P}_L \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad \text{for the left image} \tag{3}$$

and

$$\varsigma_R \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \\ 1 \end{bmatrix} = \mathbf{P}_R \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \text{ for the right image.} \quad (4)$$

where ς_L and ς_R is the distance of the interest point from the focal plane of the left and right cameras respectively.

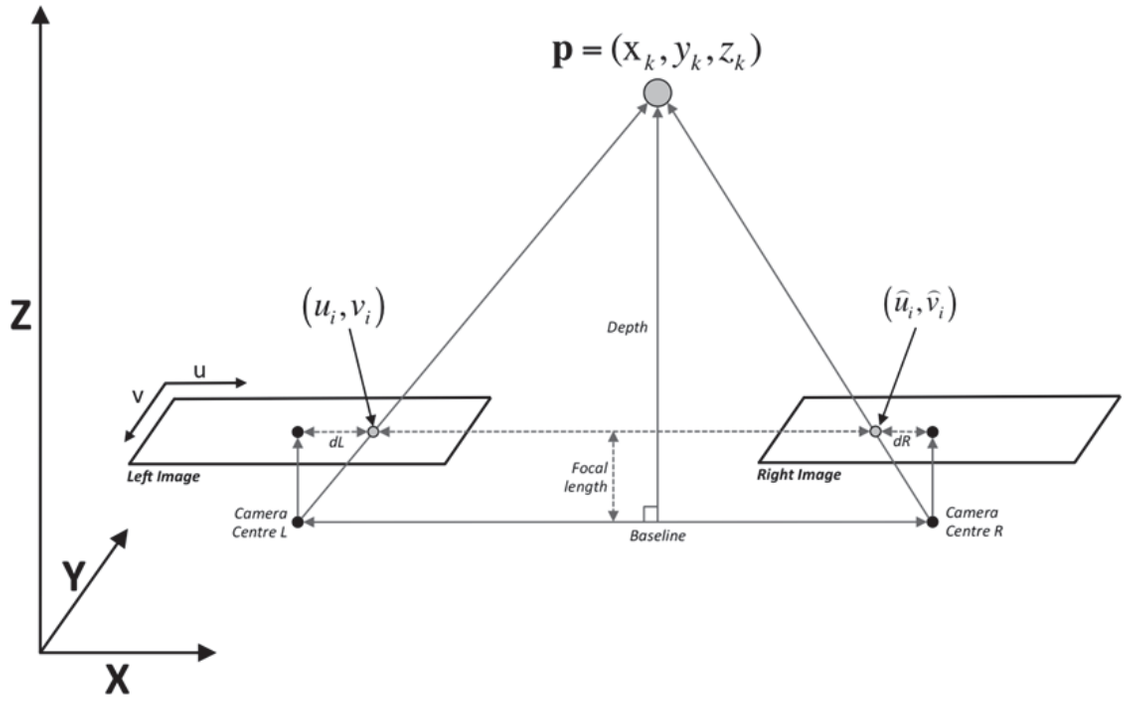

Figure 4. Geometry of 2D views and stereo cameras

4. Feature extraction

Once a stereo pair has been captured, the next stage of the block diagram in Figure 2 is to perform feature extraction. There are various considerations when selecting a suitable feature extraction method, including accuracy, distinctiveness and repeatability. The features should be robust to rotation, scaling, illumination and perspective distortion. To achieve a more discernable and repeatable feature, researchers have looked at ways of adding extra information after feature detection. A description stage constructs a high dimensional feature vector by sampling the pixel neighbourhood around a detected feature. If the vector is unique enough compared to the rest of the feature neighbourhoods, a descriptor is appended to the sampled feature. Substantially increasing the uniqueness of a detected feature with a descriptor returns a higher likelihood of a positive match during correspondence, however at a cost of time through the extra processing.

One such detector and descriptor scheme is Speeded Up Robust Features (Bay et al., 2008) or SURF for short. SURF has demonstrated remarkable repeatability, distinctiveness, robustness and efficiency when compared (Bay et al., 2008; Cattin et al., 2006) to other such

features types like SIFT (Lowe, 2004). Though SIFT was the forbearer for descriptive feature matching, SURF leverages off short comings of SIFT to produce a more robust and efficient description algorithm. For these reasons, SURF has been chosen as the feature extraction method in this work.

SURF uses a Hessian matrix based detector to find blob like textures in an image, and a distribution based descriptor to construct high dimensional vectors around detected interest points. The SURF descriptor is explained in (Bay et al., 2008), and is summarised in the following sections.

4.1. SURF's Hessian matrix based detector

4.1.1. Integral images

The fast computation time of SURF interest points is largely contributed to the use of integral images. The intensity calculations for the box type convolution filters used in SURF are easily calculated once an integral image has been computed. An integral image Im_Σ for an input image Im is generated by

$$\mathrm{Im}_\Sigma(x,y)=\sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y}\mathrm{Im}(i,j) \tag{5}$$

The value of any pixel in the integral image $\mathrm{Im}_\Sigma(x,y)$ at each point (x,y) is the sum of pixels above and to the left of that point (Viola & Jones, 2001; Bay et al., 2008).

4.1.2. Hessian matrix

SURF detects blob-like structures at locations and scales where the determinate of the Hessian matrix is maximum (Bay et al., 2008). Given a point $\mathbf{p}=(x,y)$ in an integral image Im_Σ, the Hessian matrix $\mathrm{H}(\mathbf{p},\sigma)$ in the space $\mathbf{p}$ and at scale σ is:

$$\mathrm{H}(\mathbf{p},\sigma)=\begin{bmatrix} l_{xx}(\mathbf{p},\sigma) & l_{xy}(\mathbf{p},\sigma) \\ l_{xy}(\mathbf{p},\sigma) & l_{yy}(\mathbf{p},\sigma) \end{bmatrix} \tag{6}$$

where $l_{xx}(\mathbf{p},\sigma)$ is the convolution of the Gaussian second order derivative with the integral image Im_Σ in point $\mathbf{p}$, and similarly for $l_{xy}(\mathbf{p},\sigma)$ and $l_{yy}(\mathbf{p},\sigma)$ (Viola & Jones, 2001; Bay et al., 2008). These Gaussian second order functions in xx,yy and xy are shown in Figure 5 (left to right).

These functions are convolved with integral images to produce $l_{xx}(\mathbf{p},\sigma)$, $l_{xy}(\mathbf{p},\sigma)$ and $l_{yy}(\mathbf{p},\sigma)$ in the hessian matrix. Although the Gaussian second order functions are optimal for scale space analysis, they are discretised and cropped for the approximate SURF algorithm to make the calculations more efficient.

The SURF uses an approximate for the second order Gaussian functions, denoted by d_{xx}, d_{yy} and d_{xy}, and are re shown in Figure 6.

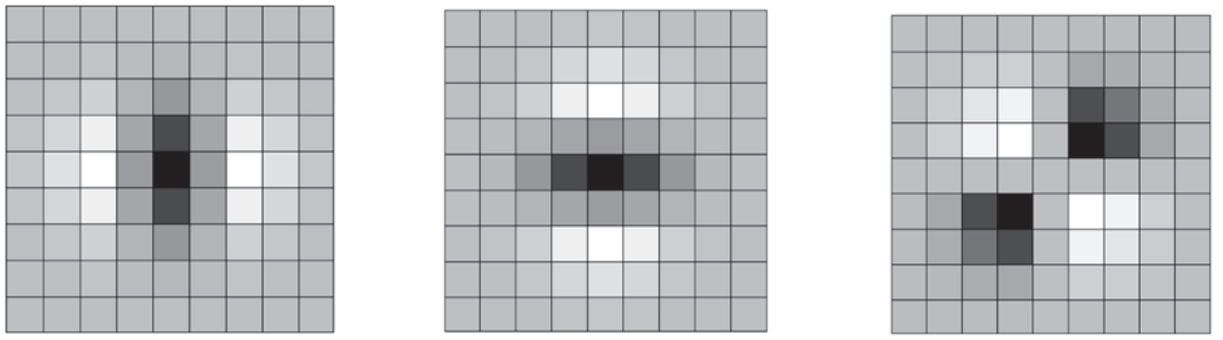

Figure 5. Second order Gaussian functions in xx, yy and xy directions (Bay et al., 2008)

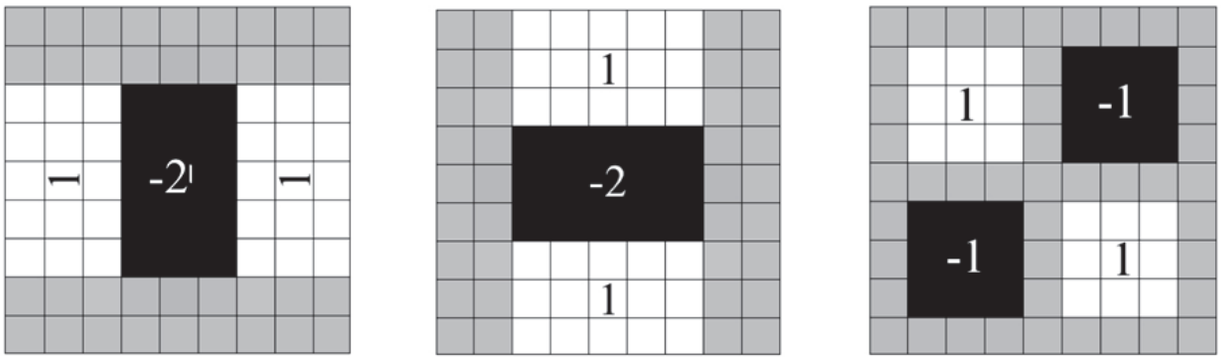

Figure 6. Approximation of second order Gaussian functions in xx, yy and xy directions (Bay et al., 2008)

The approximation of second order Gaussian functions over the integral image using box filters allows computing the hessian matrix at very low computation cost. The approximation for the Hessian matrix $\tilde{\mathrm{H}}$ is obtained by applying a simple relative weight to the hessian matrix as:

$$\tilde{\mathrm{H}} = \begin{bmatrix} d_{xx}(\mathbf{p},\sigma) & wd_{xy}(\mathbf{p},\sigma) \\ wd_{xy}(\mathbf{p},\sigma) & d_{yy}(\mathbf{p},\sigma) \end{bmatrix} \tag{7}$$

where w is a relative weight.

The relative weight of the filter responses is used to balance the expression for the Hessian's determinant. This is needed for the energy conservation between the Gaussian kernels and the approximated Gaussian kernels. It has been shown in that the appropriate value for the relative weight is 0.912 (Bay et al., 2008), therefore

$$\det(\tilde{\mathrm{H}}) = d_{xx}d_{yy} - (0.9d_{xy})^2 \tag{8}$$

The above determinant of the approximated Hessian represents the blob response in the image at location $\mathbf{p}$ (Bay et al., 2008).

4.2. SURF's distribution based descriptor

4.2.1. Orientation assignment

The description stage in SURF samples the pixel neighbourhood surrounding a detected feature to create a high dimensional vector. This vector greatly increases the uniqueness associated with detected features, and allows like features to be filtered out of the final data set. To assign a descriptor to a blob feature, the Haar wavelet responses in the x and y directions within a circular neighbourhood of radius 6s around the interest point $\mathbf{p} = (x, y)$ is calculated for different scales of σ, where s is the scale at which the interest point is detected. Figure 7 shows the Haar wavelet filters that are applied to the integral image, where the response in x or y direction is quickly calculated.

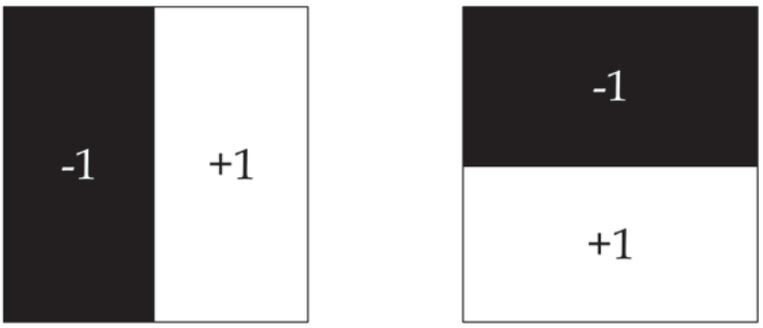

Figure 7. Haar wavelet filters to compute response for the x (left) and y (right) directions (Bay et al., 2008)

The wavelet responses are weighted by a second order Gaussian with $\sigma = 2s$. The responses are represented as points in a coordinate system centred at the interest point, with the horizontal and vertical directions aligned to the image coordinate system. The dominant orientation is estimated by calculating the sum of all responses within a 60º sliding orientation window (Bay et al., 2008), as shown in Figure 8. In this figure, the scattered blue points are the Haar wavelet responses for different scales. The red arrow indicates the assigned direction.

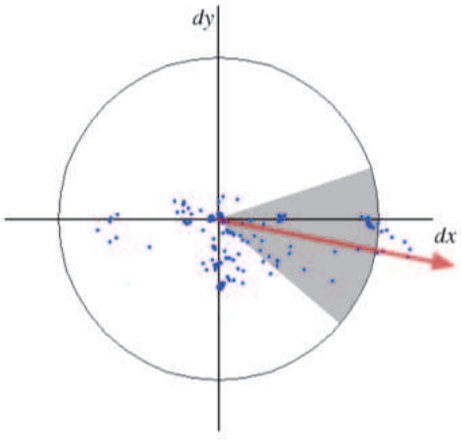

Figure 8. Orientation assignment (Bay et al., 2008)

4.2.1. Generation of the SURF descriptor

To build a 64 dimensional SURF descriptor, a quadratic grid with 4x4 square sub-regions is laid over the interest point. The quadratic grid is aligned to the orientation estimate calculated in the previous section. Each square of the quadratic grid is further divided into 2x2 sub-divisions, as shown in Figure 9, where the sub region squares and sub division squares are indicated.

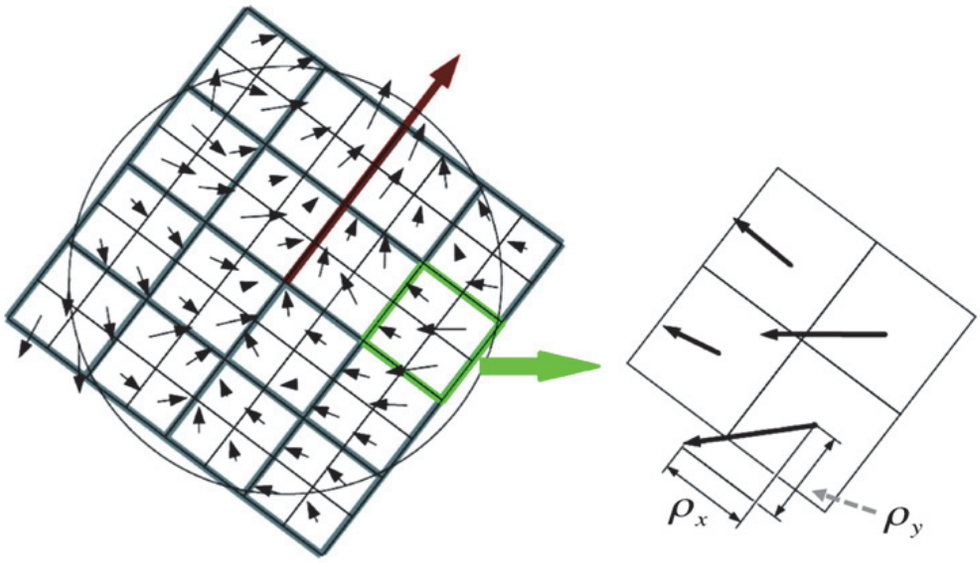

Figure 9. The 4x4 quadratic grid consisting of 16 sub-regions (left), and a 2x2 sub-division of a sub-region (right) (Bay et al., 2008)

For each sub-division, the x,y response of the Haar wavelet filters are calculated to obtain a vector located at the centre of each square. The horizontal and vertical components of these vectors in the coordinate system of the quadratic grid are depicted as ρx_i and ρy_i, where $i = 1,2,3,4$. Based on these components, four values are calculated as

$$\sum \rho x_i \ , \ \sum \rho y_i , \ \sum |\rho x_i| \text{, and } \sum |\rho y_i| . \tag{9}$$

These four values represent the actual fields in the SURF descriptor for one sub-region. With 16 sub-regions of the quadratic grid there will be 64 individual values for the SURF descriptor for any sampled interest point.

5. Generation of 2.5D views

Data from any single view of a three-dimensional object is not representative of the object as a whole (Rothganger et al., 2003). This is a consequence of self-occlusion, where the object's geometry inherently obstructs information from a single perspective. Due to occlusion, we term the 3D data obtained from a single stereo pair as a 2.5D representation (or view). To construct a 2.5D view, features are extracted from the stereo pair, matched between each image and then triangulated to localise their position in 3D space.

5.1. Generation of a feature set for single images

For each i-th stereo pair, the SURF algorithm is used generate feature sets $F_{k,i}^{L}$ and $F_{k,i}^{R}$, for the left (L) and right (R) images. As mentioned in the SURF overview section, each salient feature in any of the left and right images is assigned a 64 dimensional descriptor. We use the SURF algorithm in Matlab 2012b. An example of SURF feature extraction for one stereo pair is shown in Figure 10 for a textured cube structure. The left and right images have been concatenated into a single figure and coloured accordingly. The position of the extracted features in the left and right images are indicated with a circle and plus marks respectively.

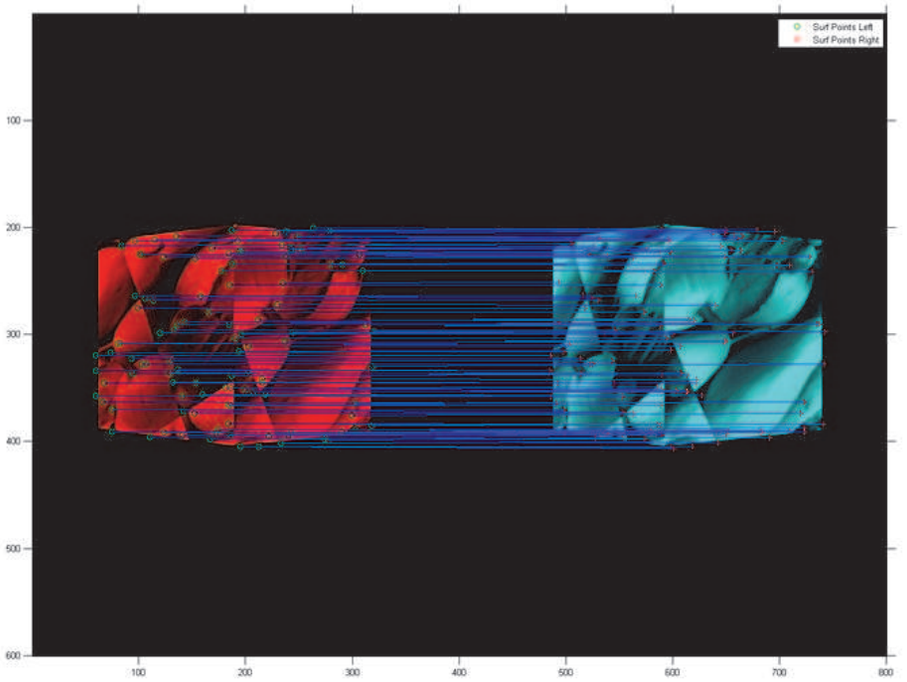

Figure 10. SURF feature extraction for a left and right image

5.2. Feature correspondence

After extracting features for each of the left and right images, the feature correspondence block of Figure 2 finds feature matches between each image of the stereo pair. There are different methods to calculate correspondence, however as mentioned previously matching high dimensional data like the SURF descriptor is time consuming. The previously established methods for correspondence of simple feature do not perform efficiently for high dimensional data.

Linear methods try to establish the best match for each feature, for example, in the left image with all features in the right. For a small number of simple features, linear methods will return the best answer, however they become extremely time consuming when dealing with large amounts of features (Gordon & Lowe, 2006), especially if the matching stage has to deal with large vectors. More advanced binary search structures like *k*-d trees and variants (Beis & Lowe, 1997; Gordon & Lowe, 2006) allow searches in large data sets to be implemented with great efficiency for simple features. These structures often have trouble

dealing with high dimensional data, potentially deteriorating to a time cost equivalent to a liner method.

Approximate nearest neighbour searches can run significantly faster for high dimensional vectors than linear and nearest neighbour methods. Muja and Lowe's (Muja & Lowe, 2009) Fast Library for Approximate Nearest Neighbour matching (FLANN) has been designed to automatically select either a hierarchal k-means structure or a randomise kd-tree with optimal parameter based on the input data. Although FLANN can return matches for large data sets many orders of magnitude faster than a linear search, the matches are less than optimal. This library is ideal for real time feature matching of many high dimensional features, however this benefit is not critical in the execution of this methodology. Finding the highest number of optimal matches is important; hence we implement a linear search with some modifications.

A useful product of the SURF feature detection stage is the trace of the Hessian matrix (sign of the Laplacian). This is calculated automatically during the detection phase. It distinguishes light blogs on dark backgrounds and vice-versa. During correspondence, we first check if the signs of the traces of the Hessian matrices match for the pair of features being compared, which can significantly reduce the time it takes for correspondence. This is a unique feature of the SURF detector; an advantage that the SIFT feature descriptor (Lowe, 2004) does not have. In addition to this check, we enforce a best to second best threshold to ensure that a current match is somewhat better than the previous estimated match.

For the i-th matched pair of features $\mathbf{f}_i^L$ and $\mathbf{f}_i^R$ in the feature sets $F_{k,i}^L$ and $F_{k,i}^R$, we generate an estimate for the descriptor to be appended to the matched points in the stereo pair based on weighted average of the matched descriptors. The weight is obtained from the strength value in the description stage of the SURF algorithm by

$$\mathbf{f}_i = \frac{s_i^L \mathbf{f}_i^L + s_i^R \mathbf{f}_i^R}{s_i^L + s_{i\ i}^{RL}} \tag{10}$$

where $\mathbf{f}_i$ is the descriptor chosen to represent the matched points. s_i^L and s_i^R are the strength values of the descriptors in the left and right image.

We performed feature matching on the stereo pairs and the result of the matched descriptors for a sample pair is indicated in Figure 10. The correspondence for each matched pair is shown with a horizontal blue line.

5.3. Triangulation

Triangulation localises a point in 3D space by analysing its 2D projections in a stereo pair (see Figure 4). The projection points for an interest point $\mathbf{p} = (x_k, y_k, z_k)$ for the k-th object were shown in equations (3) and (4) as (u_i, v_i) and $(\hat{u}_i, \hat{v}_i)$ respectively. Using the intrinsic and extrinsic parameters from the calibration of the stereo camera rig, we can use triangulation to calculate the position of $\mathbf{p} = (x_k, y_k, z_k)$ from the locations of (u_i, v_i) and $(\hat{u}_i, \hat{v}_i)$, and the difference in disparities from the camera centres (dL and dR in Figure 4).

The triangulation of sparse salient 2D image features is a little bit different from general dense disparity estimation in stereo image processing. Following the same rules, the sparse triangulation procedure should estimate the depth of matched points that have been localised with sub pixel accuracy. This can be achieved by merging equations (3) and (4) in a homogenous equation of $\mathbf{Ax}=0$, where $\mathbf{x}=\begin{bmatrix}\hat{\mathbf{p}}^T & w\end{bmatrix}^T$, $\hat{\mathbf{p}}$ is a scaled 3D pose of the point, scaled by w. The homogenous linear equation $\mathbf{Ax}=0$ can be simply obtained noting the cross product of any vector with itself is a zero vector. Therefore,

$$\left[u_i, v_i, 1\right]^T \times \mathbf{P}_L \begin{bmatrix}\mathbf{p}\\ 1\end{bmatrix} = 0 \tag{11}$$

$$\left[\hat{u}_i, \hat{v}_i, 1\right]^T \times \mathbf{P}_R \begin{bmatrix}\mathbf{p}\\ 1\end{bmatrix} = 0 \tag{12}$$

The expansion of cross products in equations 11 and 12 will result to

$$\mathbf{A} = \begin{bmatrix} u_i\mathbf{p}_L^{3T} - \mathbf{p}_L^{1T} \\ v_i\mathbf{p}_L^{3T} - \mathbf{p}_L^{2T} \\ \hat{u}_i\mathbf{p}_R^{3T} - \mathbf{p}_R^{1T} \\ \hat{v}_i\mathbf{p}_R^{3T} - \mathbf{p}_R^{2T} \end{bmatrix} \tag{13}$$

where the first two rows of $\mathbf{A}$ are associated to the left image and the second two rows are associated with the right image. The vectors of $\mathbf{p}_L^{jT}$ and $\mathbf{p}_R^{jT}$ are obtained from the j-th rows of the known projection matrices $\mathbf{P}_L$ and $\mathbf{P}_R$.

The non-zero solution of the equation $\mathbf{Ax}=0$ is the eigenvectors of $\mathbf{A}$ that are associated to the non-zero eigen values of $\mathbf{A}$. If there is more than one eigen value, then the eigen vector associated to the minimum eigen value will be selected for the parameter of $\mathbf{x}$. Hence,

$$\mathbf{x} = \begin{bmatrix}\hat{\mathbf{p}}\\ w\end{bmatrix} = eigv(\mathbf{A}) \text{ for the minimum eigen value of } \mathbf{A} \tag{14}$$

Finally the unscaled 3D position of the corresponding points of (u_i, v_i) and $(\hat{u}_i, \hat{v}_i)$ is obtained by

$$\mathbf{p} = \frac{1}{w}\hat{\mathbf{p}} \tag{15}$$

5.4. Constructing all 2.5D perspective views

Applying the triangulation procedure from equations 13-15 for any corresponding pair in a feature set $F_{k,i}^L$ and $F_{k,i}^R$, a 2.5D perspective view $M_{k,i}$ can be produced, as in Equation 1. Each point will represent the 3D coordinates of a highly distinctive 2D SURF descriptor, relative to the imaging device. The descriptor for this 3D point is obtained with Equation 10.

An example of the 2.5D view based on the stereo pair represented in Figure 10 is shown in Figure 11. Figure 11 shows a view of the XZ plane from the estimate, to highlight the surface contours of the captured data. The red crosses in Figure 11 belong to the 3D locations of the corresponding features shown in Figure 10. They highlight the two faces of the cube pointing towards the camera.

Clearly, the structure of the cube has been reconstructed in Figure 11. However, the variation of the apparent distribution points can be attributed to the SURF point detection scheme. SURF detects blob like structures that have a certain width and height. Therefore, the resultant perspective distortion from the angle at which the faces were imaged distorts the blobs, shifting the centroid for each point. Errors in camera calibration and the triangulation routines can also contribute to these variations. We chose this image set as an example of an extreme 2.5D generation scenario, due to the angle of the object being sampled. On faces with shallower angles compared to the image plane, this method produces 2.5D views with lower variations in depth estimates.

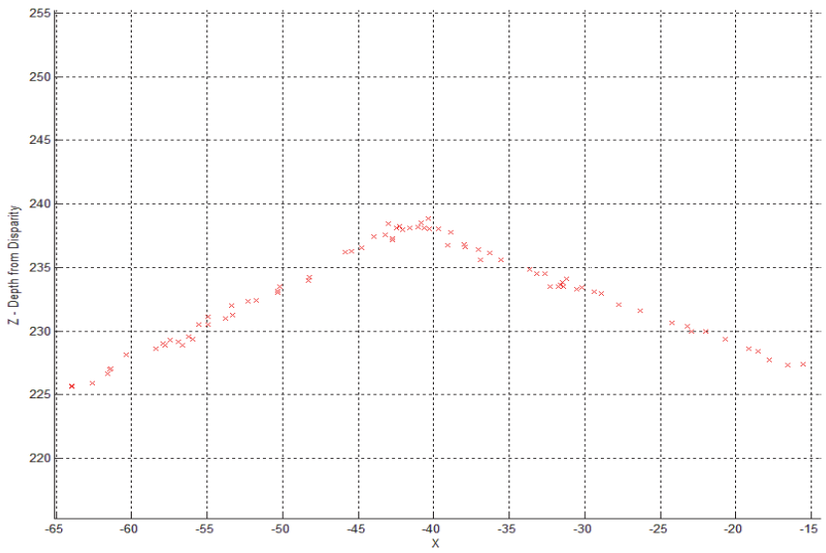

Figure 11. An XZ perspective of the 2.5D view generated from the stereo pair in Figure 10

6. 2.5D-view registration

Once a series of i 2.5D perspective views $M_{k,i}$ have been built from an ordered set of stereo images, each 2.5D must be registered into a single coordinate space, following Equation 2. To achieve this, correspondence must be established between matching features of overlapping 2.5D views. To merge one 2.5D perspective view on to another, an error metric is assigned to estimate an initial coarse geometric transformation of the two clouds. Minimising this error metric brings these clouds into alignment. Fine adjustment of the merger is achieved using an iterative refinement routine. Once two views are merged, this process is repeated for the initial merged set and another similar view so that all perspectives are registered into a single coordinate system. These procedures are explored in the following sections.

6.1. 3D Point correspondence

Identical to the correspondence problem in section 5.2, the goal is to find which points in two overlapping 2.5D perspective views match each other. We define one 2.5D cloud the model $M_{k,i}^{M}$ and the 2.5D cloud we wish to merge on to the model the as data $M_{k,i}^{D}$. Correspondence of 3D points is quite often more difficult that 2D feature matching, as the primary data in the cloud are single points with only 3D coordinates. Similarities in the arrangement of these points can be used to drive some method of surface matching, however with sparse data this becomes challenging. One advantage of this methodology is that every point in $M_{k,i}^{M}$ and $M_{k,i}^{D}$ has been triangulated from a highly descriptive 2D image features. Given that the model and data should have overlapping regions, it can be assumed that they have been taken from similar perspectives. Therefore, as every point in the 2.5D perspectives has a high dimensional feature vector appended it, we can use this extra information to identify matching points.

The same linear correspondence technique in section 5.2 is used to find SURF features in the model feature set $F_{k,i}^{M}$ that match to SURF features in data feature set $F_{k,i}^{D}$. Again, we can take advantage of the sign of Laplacian to reduce the breadth of the search. With the addition of 3D displacement of points, a geometric constraint is used to reject pairs with a distance greater than a measure of the median distance, as in (Masuda et al., 1996). Outliers can have a substantial affect when performing the following least squares minimisation, therefore the aforementioned filtering steps are essential in reducing the prevalence of outliers in the final correspondence set.

6.2. Registration

Registration is an iterative procedure that merges the points of the data ($F_{k,i}^{D}$) onto the model ($F_{k,i}^{M}$). The geometric relationship between corresponding points $\mathbf{f}^{M}$ and $\mathbf{f}^{D}$ in $F_{k,i}^{M}$ and $F_{k,i}^{D}$ is given (Eggert et al., 1997) by:

$$\mathbf{f}^{M} = \mathrm{R}\mathbf{f}^{D} + \mathrm{t} \tag{16}$$

where R is a 3x3 rotation matrix, t is a translation vector.
We can estimate the optimal rigid transformation parameters $\left[\hat{\mathrm{R}},\hat{\mathrm{t}}\right]$ between the two clouds by minimising the distance error Ψ (Eggert et al., 1997), in:

$$\Psi = \min_{\hat{\mathbf{R}},\hat{\mathbf{t}}} \sum \left\| p_i - \hat{\mathrm{R}} q_j - \hat{\mathrm{t}} \right\|^2 \tag{17}$$

We explicitly minimise equation 6 using the singular value decomposition (SVD) approach in (Eggert et al., 1997).

6.3. Registration result

Figure 12 below shows the final output of the registration methodology explained in section 5. This cloud has been generated from eighteen 2.5D perspective views from the sampled

object shown in the top right corner. One such perspective was shown in Figure 11, generated from the stereo pair shown in Figure 10.

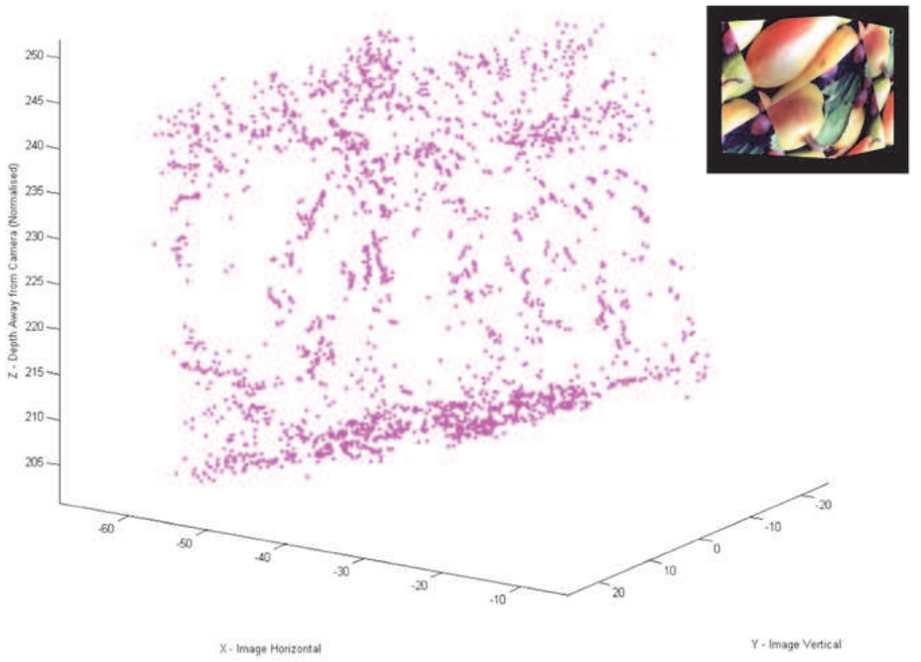

Figure 12. A final sparse feature model generated using this methodology

7. Conclusion

This chapter examined the generation of a priori data for freeform objects using multiple stereo views and 3D point registration. By unifying features from multiple short base line stereo pairs, a compact yet highly descriptive cloud termed the *sparse feature model* was developed. A sparse feature model can help estimate the position and orientation of an object in freespace quickly and accurately, and is useful for augmented reality.

The triangulation of descriptive 2D features in multiple stereo pairs was performed to produce multiple 2.5D perspective views of an object. Each 2.5D view was then merged into a single 3D cloud using 3D-to-3D point matching and registration. Every point in the final cloud represents the precise 3D position of highly descriptive 2D image features in a unified coordinate system. The generated sparse feature model contains robust and repeatable features, invariant to rotation, scaling, and illumination. As it was built from multiple perspectives, the SFM represents a sparse yet complete 3D representation of the object.

In future work, we will apply this methodology to generate a database for different objects of interest. This database will then be used for a pose estimation system via recognition in an augmented reality application.

Author details

Matthew Watson, Asim Bhatti, Hamid Abdi and Saeid Nahavandi
Centre for Intelligent Systems Research, Deakin University, Australia

8. References

Bay, H., Ess, A., Tuytelaars, T., and Gool, L.V., SURF: Speeded Up Robust Features, *CVIU*, pp. 346-359, 2008

Bay, H., From Wide-baseline Point and Line Correspondences to 3D, *Doctoral Dissertation*, Swiss Federal Institute of Technology, ETH Zurich, 2006

Beis, J. S., and Lowe, D. G., Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, *CVPR*, pp. 1000–1006, 1997

Bouguet, J., 2010, Matlab Camera Calibration Toolbox, available at: http://www.vision.caltech.edu/bouguetj/calib_doc/

Cattin, P.C., Bay, H., Van Gool, L.J., and Székely, G., Retina mosaicing using local features, *MICCAI*, pp. 185-192. October 2006

Eggert, D.W., Lorusso, A., and Fisher, R.B., Estimating 3D rigid body transformations: a comparison of four major algorithms, *MV&A*, pp. 272–290, 1997

Gordon I., and Lowe, D.G., What and Where: 3D Object Recognition with Accurate Pose, *Toward Category-Level Object Recognition*, pp. 67-82, 2006

Hartley, R., and Zisserman, A., *Multiple View Geometry, Second Edition*, Cambridge University Press, Cambridge, UK, 2003

Lepetit, V., and Fua, P., Monocular Model-Based 3D Tracking of Rigid Object: A Survey, *FTCGV*, pp. 1-98, 2005

Lowe, D.G., Distinctive image features from scale-invariant keypoints, *IJCV*, pp. 91-110, 2004

Lowe, D.G., Local Feature View Clustering for 3D Object Recognition, *CVPR*, pp. 682-688, 2001

Masuda, T., Sakaue, K., and Yokoya, N., Registration and Integration of Multiple Range Images for 3-D Model Construction, *ICPR*, 1996

Muja, M., and Lowe, D.G., Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, *VISAPP'09*, 2009

O. Faugeras and Q-T Luong. *The geometry of multiple images*. MIT Press, 2001.

O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

P.A. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. *In CVPR (1)*, pages 511 – 518, 2001.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J., 3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-view Spatial Constraints, *CVPR*, pp. 272-277, 2003

Schaffalitzky, F., and Zisserman, A., Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?", *ECCV*, 2002

Vezhnevets, V., Velizhev, A., Chetverikov, N., and Yakubenko A., "GML C++ Camera Calibration Toolbox", 2011, available at: http://graphics.cs.msu.ru/en/science/research/calibration/cpp