International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes

A. Mian · M. Bennamoun · R. Owens

Received: date / Accepted: date

Abstract 3D object recognition from local features is robust to occlusions and clutter. However, local features must be extracted from a small set of feature rich keypoints to avoid computational complexity and ambiguous features. We present an algorithm for the detection of such keypoints on 3D models and partial views of objects. The keypoints are highly repeatable between partial views of an object and its complete 3D model. We also propose a quality measure to rank the keypoints and select the best ones for extracting local features. Keypoints are identified at locations where a unique local 3D coordinate basis can be derived from the underlying surface in order to extract invariant features. We also propose an automatic scale selection technique for extracting multi-scale and scale invariant features to match objects at different unknown scales. Features are projected to a PCA subspace and matched to find correspondences between a database and query object. Each pair of matching features gives a transformation that aligns the query and database object. These transformations are clustered and the biggest cluster is used to identify the query object. Experiments on a public database revealed that the proposed quality measure relates correctly to the repeatability of keypoints and the multi-scale features have a recognition rate of over 95% for up to 80% occluded objects.

Keywords Keypoint detection · keypoint quality and repeatability · local features · 3D object retrieval

1 Introduction

Historically, 3D models were used in computer graphics and computer aided-design (CAD) applications. These days, we can find their applications in the film industry, robotics, archaeological site preservation, forensics, surgery, biometrics, scientific research and education as well. Initial 3D models were synthetically generated from geometric primitives. However, with the availability of low cost scanners, detailed and accurate 3D models of free-form objects are becoming more popular. This has also increased the demand for efficient and accurate representation and retrieval techniques for 3D free-form objects.

Campbell and Flynn [5] give a survey of 3D free-form object recognition techniques and classify them into appearance based recognition, recognition from 2D silhouettes and free-form object recognition in range data. Appearance based techniques are similar to techniques applied to intensity images. These techniques treat 3D data as range images where the pixels represent the surface depth rather than its texture. A number of range images of the 3D object from different viewpoints are used to train the system. Examples of appearance based techniques include Murase and Nayyar [25] and the local feature histogram technique [14]. Ashbrook et al. [2] used pairwise geometric histograms for 3D object recognition. Appearance based recognition has also been used in the case of 3D face recognition [4]. Recognition from 2D silhouettes are the earliest 3D object recognition techniques because they require intensity images to extract the silhouettes. Ponce and Kriegman [30] used silhouette contours for 3D object recognition.

The third class of techniques i.e. free-form object recognition in range data uses pointcloud (see Fig. 3)

Supported by ARC Grants DP0881813, DP0664228 and LE0775672.

A. Mian · M. Bennamoun · R. Owens
School of Computer Science and Software Engineering
The University of Western Australia Tel.: +61-8-64882702
Fax: +61-8-64881089
E-mail: ajmal@csse.uwa.edu.au

or mesh representation (e.g. [22]). Campbell and Flynn [5] argue that the computational complexity of these techniques can become one of the prohibitive factors if brute force matching is used. For example the classic Iterative Closest Point (ICP) algorithm [3] has been used for surface matching but it is computationally expensive and requires an initial estimate of registration. Other techniques have resorted to various heuristics to reduce the amount of computation required.

Many techniques have been proposed in the literature to avoid the computational complexity of point-cloud and mesh representations. The traditional way was to perform a sparse sampling of the surface and reduce computational cost at the expense of accuracy. Later techniques used mesh decimation to reduce the density of the mesh. Recent technique use local features to represent the meshes. Our survey (Section 2) shows that local features are reliable for 3D object recognition or retrieval from cluttered scenes. However, extracting local features at every point is computationally inefficient and arbitrary sparse sampling of the object for local feature extraction can result in sub-optimal recognition performance. Both these approaches will also result in features which are not sufficiently descriptive to uniquely characterize the object. Detecting a subset of keypoints over the 3D model or partial surface can significantly reduce the computational burden of the subsequent feature extraction and matching phases. Ideally such keypoints must be repeatedly detectable in different partial views of the object and its 3D model. Moreover, the keypoints must be detected at locations where invariant and descriptive features can be extracted.

This paper extends our prior work [20] and proposes a multi-scale keypoint detection algorithm for extracting scale invariant local features. The following are the major differences from [20]. This paper presents a novel quality measure to rank the keypoints unlike [20] where a binary decision is taken to select or reject a keypoint. The surfaces are sparsely sampled and a random subset of keypoints are picked for feature extraction in [20] whereas in this paper the keypoints are ranked according to their quality and the best ones are selected for feature extraction. This paper also performs extensive keypoint repeatability experiments with respect to various criteria and demonstrates that the proposed quality measure is correlated with the repeatability of the keypoints. In our prior work [20], keypoints and features were extracted at a fixed scale. However, this paper proposes a technique for automatically selecting the appropriate scale at each keypoint and extracts multi-scale and scale invariant features. Finally, this paper presents results for 3D object retrieval from complex scenes containing clutter and occlusions and compares the results to two other techniques using the same public database [22].

The rest of this paper is organized as follows. Section 2 provides a literature survey of free-form object recognition and keypoint identification techniques. Section 3 describes the keypoint detection algorithm and presents a novel keypoint quality measure. Section 4 gives results of our keypoint repeatability experiments. Section 5 describes our automatic scale selection technique and compares the repeatability of keypoints detected using fixed scale and automatic scale selection. Section 6 describes the proposed local feature extraction and matching algorithms. Experimental results for 3D object retrieval from cluttered scenes with comparison to other techniques are given in Section 7. Section 8 concludes the paper.

2 Related Work

Dorai and Jain [9] proposed a shape index using the principal curvatures of the surface for 3D free-form object recognition. Their method requires a view of the 3D object without occlusion. Spherical Attribute Images (SAI) were developed by Hebert et al. [13] to represent and recognize 3D objects that are topologically equivalent to a sphere. This representation is also scale invariant. Joshi et al. [16] proposed HOT curves which relies on the localization of inflection points.

Chua and Jarvis [7] introduced point signatures for 3D object recognition. A point signature is a one dimensional signature that describes the surface surrounding a point. A limitation of the technique is that the starting point of the signature may not be unique in which case multiple signatures could be obtained from the same point. Sumi et al. [33] used segment based stereovision and local features for 3D object recognition in cluttered environments.

Johnson and Hebert [15] proposed spin images for local 3D shape representation and demonstrated its performance for 3D object recognition in the presence of clutter and occlusions. A spin image is formed by spinning an image plane about the normal of a vertex and summing vertices of the mesh into the bins of the image plane. A prerequisite is that the mesh must have uniform resolution i.e. all edges have approximately equal length. Mian et al. [22] proposed the tensor representation for describing partial surfaces and 3D models. Tensors are calculated by defining three dimensional grids over a surface and calculating the surface area that passes through each bin. Tensors are three dimensional compared to spin images which are 2D histograms of the cylindrical coordinates of the mesh vertices. Therefore, tensors have more discriminating capability than spin images [21] and also outperform spin images during object recognition in cluttered scenes [22].

Most of the earlier work in the literature has focused on feature extraction, invariance of the features to geometric transformations and their performance in 3D object retrieval. However, there has been a considerable research interest in the identification of keypoints for local feature extraction. Castellani et al. [6] detected and matched salient points for establishing correspondences between partially overlapping 2.5D views of an object. They applied a series of Gaussian filters on the mesh and selected points that are maximally displaced along the normal direction as salient points. Gelfand et al. [11] compute an integral volume descriptor for each surface point and pick a sparse number of points according to the uniqueness of their descriptors. Some work on mesh saliency from human perception point of view also exists [17] however, we are interested in points that are significant from pattern recognition point of view and can be repeatably identified by computer vision algorithms.

Apart from identifying key or salient points, the scale and distinctiveness of the local descriptors has also been investigated. An interesting approach is taken by Wessel et al. [36] who transform different local descriptors into a meta Class Distribution Descriptor (CDD). During a supervised learning phase, different types of local descriptors, depending upon their distinctiveness for a particular class of objects, are transformed to form a CDD which is used for object retrieval. Unnikrishnan et al. [34] estimate the scale, for a point sampled curve, as a neighbourhood for which the principal eigenvector of the points is best aligned with the tangent of the curve in an iterative procedure. This idea has been extended to point clouds for identifying multi-scale interest regions [35] where the mean curvature is used to identify scale-space extrema. However, this definition becomes ambiguous when the surface does not have a constant mean curvature. Novatnack and Nishino [26] convert the mesh model to a dense 2D normal map and convolve it with Gaussian kernels of increasing standard deviations. The Gaussian kernel is modified to account for the surface distortion which occurs while parameterizing the mesh on a 2D plane. This scale-space representation and analysis is used to detect corners and edges at different scales on the mesh model. The scale of each feature is automatically selected using a method analogous to the 2D scale-space theory i.e. the scale at which the response of the feature detector is maximized. The outcome of this approach is a set of scale-dependent geometric features. Novatnack and Nishino [27] later extended their technique to derive scale-invariant features by normalizing the support size of each feature to a constant radius. These features were used to match rigid objects at different scales.

In existing literature, the detection of keypoints has mainly been treated as a binary process and little importance is given to explicitly rank the keypoints according to their quality. We believe it is important to rank keypoints using a measure that is directly related to their repeatability and the distinctiveness of the underlying local descriptor. Moreover, the performance of keypoints and subsequent descriptors has not been tested for object retrieval from complex scenes containing clutter and occlusions. In this paper, we cover this gap and propose a generic keypoint detection algorithm for the task of 3D object retrieval or recognition. We also propose a quality measure for ranking keypoints on 3D models and partial surfaces and demonstrate that the quality measure is directly related to the repeatability of the keypoints. The identification of repeatable keypoints increases the efficiency and accuracy of the subsequent recognition stage. Finally, this paper proposes a matching algorithm for 3D object retrieval from cluttered scenes which is a challenging case because objects of interest could be occluded.

3 Keypoint Detection

We define keypoints (on a 3D model or 2.5D surface) as the points which satisfy three constraints. (1) The detected keypoints must have high repeatability between different 2.5D views and 3D model of the same object. (2) A unique 3D coordinate basis can be defined from the neighbourhood surface to extract invariant local features. (3) The neighbourhood surface of the keypoint must contain sufficient descriptive information which uniquely characterizes that point. This ensures that the local feature extracted at the keypoint is unique to facilitate accurate recognition.

It is still possible that for certain 3D models, a very large number of points may satisfy these three constraints. This undermines the first purpose of keypoint detection i.e. improved efficiency by matching only a limited number of keypoints. In such cases, the models are sparsely sampled or a random subset of keypoints are picked for feature extraction [20]. However, sparse sampling or random selection of keypoints will not result in the best keypoints in terms of their repeatability and descriptiveness of the local surface. To overcome this problem, we propose a keypoint quality measure based on the principal curvatures of the local surface. This quality measure is used to rank the keypoints and select the best N number of keypoints for subsequent matching.

3.1 Keypoint Detection in Full 3D Models

Let $\mathbf{M} = \{\mathbf{v}, \mathbf{f}\}$ be the mesh data structure of a 3D model comprising vertices $\mathbf{v}$ and polygons $\mathbf{f}$. Vertices (or points) are the 3D coordinates of each point and polygons are index numbers of the points which make up individual polygons. Similar to computer graphics, all polygons are essentially simple i.e. convex and planar. The normal of the plane in which the polygon lies is the normal of the polygon and can easily be calculated from the equation of the plane

$$ax + by + cz + d = 0, \tag{1}$$

where the normal of the plane is $[a\ b\ c]^T$. Since three or more points define a polygon, Eqn. 1 can easily be solved. The normal of a point is determined by averaging the normals of all polygons that share that point (vertex). At each sample point $\mathbf{v} = [x \quad y \quad z]^T$, a local surface is cropped from the 3D model using a sphere of radius r_1 centered at $\mathbf{v}$. The value of r_1 can be chosen depending upon how local we want the feature to be. It is a trade off between the feature's descriptiveness and sensitivity to occlusions. Small values of r_1 will be less sensitive to occlusions but will contain less information about the underlying surface and hence the feature will be less descriptive. The value of r_1 is also related to the scale of the 3D models. For small size objects, r_1 should be small and for large objects it should be large. In Section 5 we describe a technique for automatic scale selection.

Local surface is cropped to extracted a local coordinate frame that is robust to occlusions and other global perturbations. The idea of local surface (with or without cropping) has been widely used for the extraction of local reference frame and subsequent invariant features. The reader is referred to the survey of Petitjean [29] for details. Many existing methods fit a local surface to the data points and derive the reference frame from the fitted surface. However, we extract the reference frame directly from the data points for computational efficiency and perform surface fitting later for feature extraction only if the point qualifies as a keypoint.

The cropped surface is rotated so that the normal of the point is aligned with the positive z-axis. Since we are dealing with complete 3D models, the cropped surface may also contain polygons that are occluded. These polygons are removed using hidden surface removal [1]. Points that are shared only by occluded polygons are removed. Let $\mathbf{L}_j = [x_j\ y_j\ z_j]^T$ (where $j = 1, 2, \ldots n_l$) be the remaining points. The mean of the points is given by

$$\mathbf{m} = \frac{1}{n_l} \sum_{j=1}^{n_l} \mathbf{L}_j \ , \tag{2}$$

and the covariance matrix of $\mathbf{L}$ is given by

$$\mathbf{C} = \frac{1}{n_l} \sum_{j=1}^{n_l} \mathbf{L}_j \mathbf{L}_j^T - \mathbf{m}\mathbf{m}^T \tag{3}$$

where $\mathbf{L}_j$ is the jth column of $\mathbf{L}$. Performing Principal Component Analysis on the covariance matrix $\mathbf{C}$ gives us

$$\mathbf{CV} = \mathbf{DV} \ , \tag{4}$$

where $\mathbf{V}$ is the matrix of eigenvectors, $\mathbf{D}$ is the diagonal matrix of the eigenvalues of $\mathbf{C}$. The matrix $\mathbf{L}$ can be aligned with its principal axes using Eqn. 5, known as the Hotelling transform [12]:

$$\mathbf{L}'_j = \mathbf{V}(\mathbf{L}_j - \mathbf{m}) \quad \text{for} \quad j = 1 \ldots n_l. \tag{5}$$

Let $\mathbf{X}$ and $\mathbf{Y}$ represent the x and y components of the point cloud $\mathbf{L}'$ i.e.

$$\mathbf{X} = \{x_1\ x_2\ \ldots x_{n_l}\} \quad \text{and} \quad \mathbf{Y} = \{y_1\ y_2\ \ldots y_{n_l}\} \ . \tag{6}$$

$$\delta = \frac{\max(\mathbf{X}) - \min(\mathbf{X})}{\max(\mathbf{Y}) - \min(\mathbf{Y})} \ , \tag{7}$$

where δ is the ratio between the first two principal axes of the local surface. For symmetrical surfaces, the value of δ will be equal to 1 and for unsymmetrical surfaces, δ will be greater than 1. To derive a unique 3D coordinate basis from a surface the value of δ must be greater than one. However, to cater for noise we chose a threshold $t_1 > 1$ and proceeded with the keypoint detection if $\delta > t_1$. Small values of t_1 will qualify every point as a potential keypoint at this stage and result in computational complexity. Large values, on the other hand, may eliminate good keypoints i.e. high quality value to be calculated at a later stage. We chose $t_1 = 1.06$ in our experiments i.e. 6% longer first principal axis compared to the second one.

Gelfand and Ikemoto [10] also used the covariance matrix for selecting stable points for the quick and correct convergence of the ICP algorithm [3]. Our approach differs from [10] because we use a 3x3 covariance matrix as opposed to a 6x6 covariance matrix. We are only interested in defining an unambiguous 3x3 coordinate axes at the keypoints and there is no ambiguity of translation in our case because the location of the keypoint is taken as the origin of the coordinate basis. In the case of [10], there may be ambiguity in 3 rotations and 3 translations and hence a 6x6 covariance matrix is used. Since translational ambiguity is not an issue in our case, points on many surfaces that are considered unstable by [10] can qualify as keypoints as per our criteria. Finally, [10] requires that $\frac{\lambda_1}{\lambda_6}$ (the ratio between the most significant and least significant eigenvalues) be as close to one as possible whereas we reject points where the ratio between the first two principle axes is close to one. In automatic scale selection (see Section 5), we chose a scale where this ratio has a local maximum.

3.2 Keypoint Detection in Partial Views

Two changes are made in the case of partial 2.5D views. First, hidden surface removal is not performed as the partial view only contains surface that is visible to the scanner. Two, a boundary condition is added to avoid keypoints along the boundary of the view. Local surface cropped near the boundary will have very high value of δ even if the surface is planar. This is because part of the surface is missing due to self occlusion. Two types of boundary conditions are used depending upon the format of the surface data. If the surface data is present in the form of arranged X, Y and Z grids (e.g. a range image) then the boundary condition considers points that are r_1 distance away from the boundary. Internal holes in the range image (due to sensor errors) are interpolated so that the partial view has a continuous external boundary. This method perfectly removes all boundary points from being considered as keypoints.

In case the partial view data is in the form of an unorganized pointcloud, the expected number of points inside the radius r_1 are estimated from the point cloud density. During keypoint detection, if the local cropped surface has substantially lower number of points than expected, that point is dropped. Curved surfaces are usually non-uniformly sampled by scanners and the density of points varies over the surface. Therefore, a soft threshold is used to eliminate possible boundary points. Any boundary points that are not detected at this stage will get a lower quality value in the next stage and be eliminated during the top quality feature selection.

It is interesting to note that our selection criteria does not detect keypoints at the center of symmetric surfaces. Examples of such surfaces are peaks, spheres, parabolic and conic surfaces. We argue that such features are not totally neglected by our detection criteria. Rather than detecting a single keypoint at the center of a symmetric surface, our algorithm detects multiple keypoints around it and once features are extracted from the neighbourhoods of the keypoints, they will enclose the symmetric surface as well. An example of keypoint detection on a 2D Gaussian surface is shown in Fig. 1.

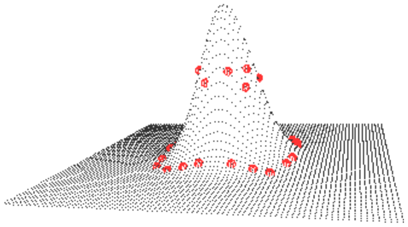


Fig. 1 The proposed algorithm detects many keypoints (red dots) around a symmetric surface (like the 2D Gaussian) but none at the center where the principal axes of a local surface are ambiguous.

3.3 Keypoint Quality Calculation

The proposed keypoint quality measure is motivated by the work of Shi and Tomasi [31] who defined (for the purpose of tracking in 2D image sequences) good feature points as exactly the ones which can be tracked reliably. In other words, good feature points are those which offer good measurements to solve their tracking equation

$$\mathbf{Zd} = \mathbf{e} \ , \tag{8}$$

where $\mathbf{e}$ is the difference between two frames of a video sequence, $\mathbf{d}$ is the displacement which needs to be calculated and $\mathbf{Z}$ is given by,

$$\mathbf{Z} = \int_W \begin{bmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y)}{\partial x \partial y} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{bmatrix} dA \ . \tag{9}$$

Integration is performed over a small feature window W of the image $I(x,y)$. Notice that the matrix in Eqn. 9 is the Hessian matrix of second order partial derivatives of image $I(x,y)$. Shi and Tomasi [31] argued that good features are those which can lead to a reliable solution of Eqn. 8. This requires that both the eigenvalues of the matrix $\mathbf{Z}$ in Eqn. 9 are large (for good signal to noise ratio) and do not differ by several orders of magnitude.

We take a similar approach in the 3D domain and define a keypoint quality measure based on the principal curvatures of the local surface within a neighbourhood of the keypoint. Since principal curvatures also correspond to the second order derivatives of a surface, we do not calculate them from the raw data. Instead, we fit a surface [8] to the raw local surface data $\mathbf{L}'$. Recall that $\mathbf{L}'$ has been aligned along its principal axes. Surface fitting is performed using a smoothing (or stiffness) factor that does not allow the surface to bend abruptly thereby alleviating the effects of noise. A uniform $n \times n$ lattice (grid) is used to sample the surface $\mathbf{S}$ which is fitted [8] to the points in $\mathbf{L}'$ (where $n = 20$) . Principal curvatures are then calculated at each point of the surface $\mathbf{S}$ and the keypoint quality Q_k is calculated as

$$K = \kappa_1 \kappa_2 \tag{10}$$

$$\begin{aligned} Q_k = \frac{1000}{n^2} \sum |K| + \max(100K) + |\min(100K)| \\ + \max(10\kappa_1) + |\min(10\kappa_2)| \end{aligned} \tag{11}$$

In the above equations, κ_1, κ_2 are the principle curvatures and K is the Gaussian curvature. Summation,

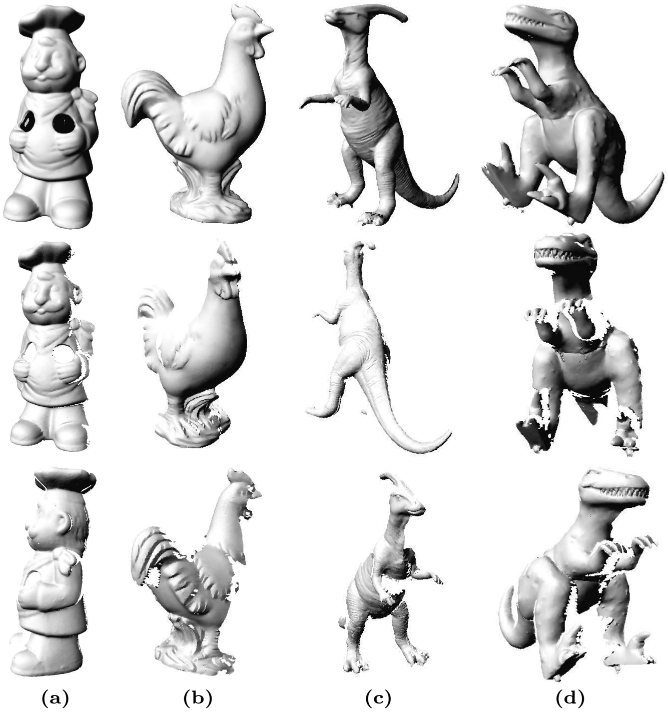


Fig. 2 Complete 3D models of (a) chef, (b) chicken, (c) parasaurolophus and (d) t-rex and their sample 2.5D views.

maximum and minimum values are calculated over the $n \times n$ surface. Absolute values are taken so that positive and negative curvatures do not cancel each other; negative and positive values of curvature are equally descriptive. The constant multiplicative terms are empirically chosen and used to give appropriate weights to each term. The value of principle curvatures is mostly less than one for smooth surfaces and since Gaussian curvature is the product of the principle curvatures, it is of the order to 10 smaller than the principle curvatures. This relationship was experimentally confirmed using training data. Therefore, the multiplication terms of K were chosen 10 times greater than that of the principle curvatures. The training data also revealed that the first term of Eqn. 11 was of the order of 100 smaller than the principle curvatures. A possible explanation for this relationship is that at most points on a smooth surface the Gaussian curvature has a small value which when averaged over the surface results in a value which is of the order to 10 smaller than the maximum and absolute minimum values of the Gaussian curvature. Finally, all terms in Eqn. 11 were multiplied by 10 to get a quality measure Q_k which is always greater than one. Recall that a smooth surfaces was fitted [8] to the data points before the calculation of the principle curvatures in order to avoid sensitivity to noise. Our results show that this quality measure relates correctly to the repeatability of the keypoints (see Fig. 4-c) and results in distinctive features capable of retrieving objects from cluttered scenes (see Section 7).

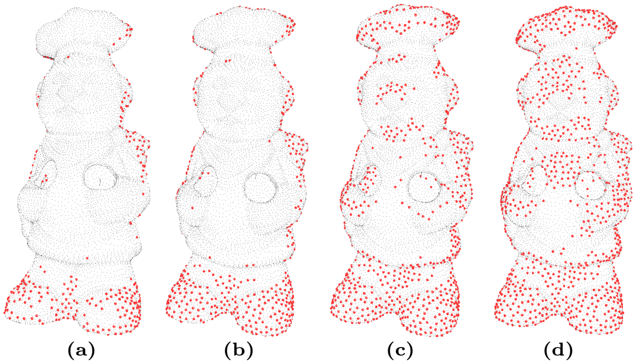


Fig. 3 (a) Pointcloud of the chef. Keypoints with $Q_k > 30$ are shown as dark (red) dots. (b) Keypoints with $Q_k > 20$. (c) Keypoints with $Q_k > 10$. (d) Keypoints with $Q_k > 5$.

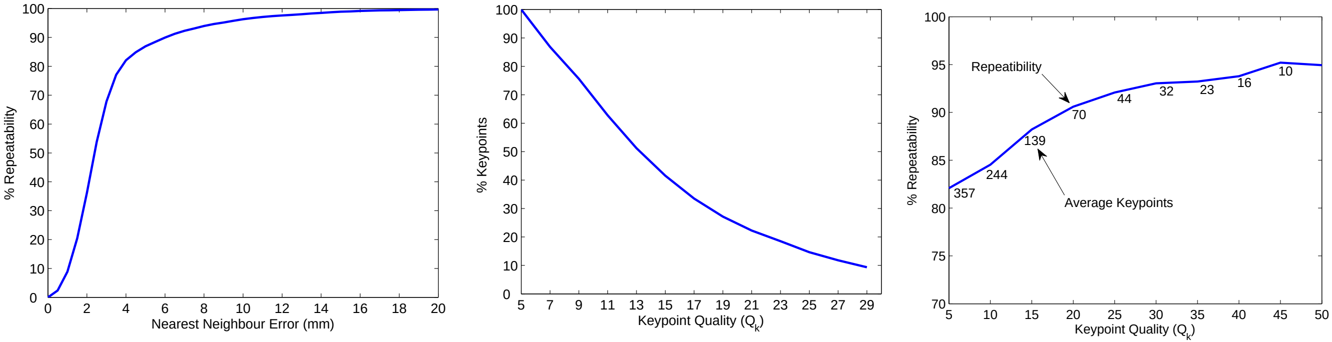
Closely located keypoints are essentially similar as they describe more or less the same local surface. Therefore, a threshold d_1 is used to keep minimum distance between keypoints. In our experiments we chose $d_1 = 4mm$ i.e. twice the sampling rate of the surface. The detected keypoints are sorted according to their quality values and starting from the highest quality keypoint, all keypoints within its radius of d_1 are removed. This ensures that, for a given neighbourhood, only the point with the highest quality is taken as a keypoint. Notice that this is a better way of limiting the number of keypoints as opposed to sparse sampling or random selection of keypoints [20].

4 Keypoint Repeatability Experiments

We performed our experiments on the publicly available 3D model database used in our prior work [22]. For the repeatability experiments, we used four 3D models of real world objects and 10 different views of each model. Fig. 2 shows the complete 3D models and two sample views of each model. This data was acquired with the Minolta Vivid 910 scanner which has a spatial resolution of 640 × 480.

The keypoint quality measure provides a means of selecting the best required number of keypoints. Alternatively, a cutoff threshold can be used to keep the keypoints with Q_k greater than the threshold. Fig. 3 shows keypoints detected on the 3D model of chef at different cutoff thresholds of the quality Q_k. Notice that as the threshold is decreased, more and more keypoints appear at less curved parts of the 3D model.

In a model based object recognition (or retrieval) scenario, full models are matched with their partial views (or vice versa). Therefore, keypoints were detected on each 3D model and their 10 different views to



(a) **(b)** **(c)**

Fig. 4 (a) Keypoint identification repeatability at $Q_k > 10$ between full 3D models and their partial views. (b) Percentage of keypoints as a function of quality Q_k for $d_1 = 4mm$. (c) Keypoint identification repeatability as a function of quality Q_k for $d_1 = 4mm$. The Average number of detected keypoints for different values of Q_k are shown below the curve.

calculate the repeatability between the model keypoints and the partial view keypoints of the same object. The partial views were registered to their full models using ground truth registration calculated from coarse alignment [21] and refinement with the ICP algorithm [3]. Next the distance of every keypoint in the partial view was calculated to the nearest neighbour keypoint in the full model. Fig. 4-a shows the plot of keypoint repeatability between the four 3D models and their respective 10 partial views each i.e. a total of 40 experiments. Keypoints with $Q_k > 10$ were considered, resulting in 240 keypoints per view on the average. The y-axis shows the percentage keypoints of the partial views which could find a corresponding keypoint in the 3D model within the distance shown on the x-axis. A similar criteria was used by Lowe [24] for reporting the repeatability of Scale Invariant Feature Transform (SIFT) however, the experiments were performed on synthetic data. Fig. 4-a shows that at a nearest neighbour error of $4mm$, the repeatability of keypoints is 84.2%. Note that $4mm$ is the minimum allowed distance d_1 between two keypoints and is therefore selected as a benchmark error for calculating the repeatability in the remaining experiments in Fig. 4-b and Fig. 4-c.

Fig. 4-b shows the number of keypoints detected above a given keypoint quality Q_k for all the models and partial views used in our experiments. Taking the number of keypoints detected at $Q_k > 5$ as a benchmark (i.e. 100%), the percentage of keypoints drops almost linearly when the threshold for Q_k is increased. This makes perfect sense however, it is interesting to note that the repeatability of keypoints increases when the threshold for Q_k is increased. In other words, the repeatability increases when the number of keypoints decrease. Fig. 4-c shows the repeatability plot for a nearest neighbour error of $4mm$ as a function of Q_k threshold. The repeatability increases from 82.1% for $Q_k > 5$ to 95.2% for $Q_k > 45$. The increase in the repeatability of keypoints with increasing Q_k indicates that the measure Q_k correctly reflects the quality of keypoints i.e. the higher the quality, the higher the repeatability.

5 Automatic Scale Selection

So far we have detected keypoints at a fixed scale. However, at a fixed scale, it is possible that keypoints may never appear on the less curved or featured parts of certain 3D models. For example, the belly of the chicken in Fig. 5-a does not contain any keypoint when detection is performed at $r_1 = 20mm$ (sampling rate of $2mm$). This could be a problem when recognition is performed in real scenes containing clutter and occlusions and the only visible part of the chicken is its belly. Note, that this is a scenario where humans can also find difficulty in recognizing the object in the absence of any texture. To overcome this problem, we propose keypoint detection and feature extraction at multiple scales. For example, after lowering the scale by four times, keypoints are detected at the less featured belly of the chicken as well in Fig. 5-b.

Selecting the right scale for keypoints and subsequent features is an important problem. Detecting features at different pre-selected scales would be an easy alternative however, a more elegant solution is to automatically determine the right scale for a keypoint and the subsequent feature extraction. We propose an automatic scale selection approach whereby the right scale for a keypoint is determined as the neighbourhood size (as cropped by a sphere of radius r_1) for which the ratio between the principle axes of the neighbourhood reaches a local maximum. Fig. 5-c shows plots of the principle axes ratio for three different points at varying scale. Two of the points reach local maxima and are therefore selected as keypoints at the respective scale which means that the subsequent features (see Section 6) will be extracted at the respective scales. The scale

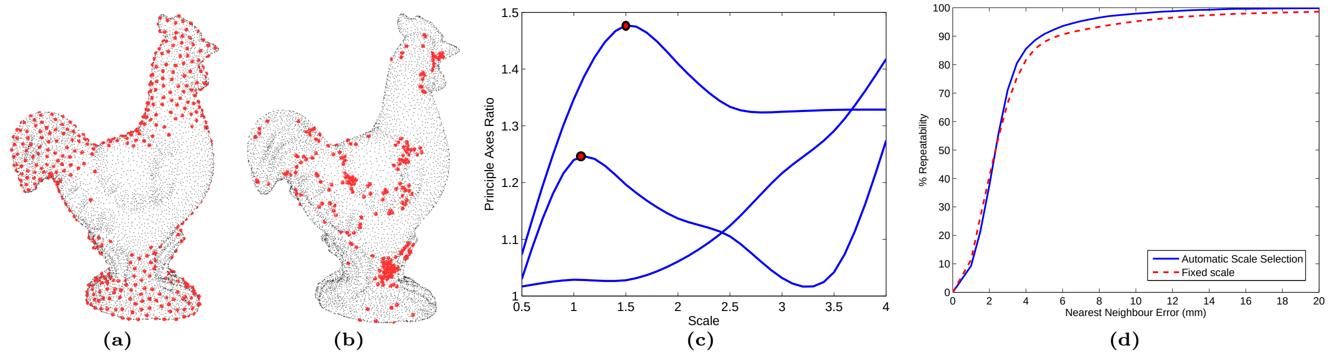


Fig. 5 (a) Keypoints (with $Q_k > 10$) shown on the pointcloud of the chicken. There are no keypoints on the belly of the chicken even though some small scale features (feathers) exist there. These features were not picked as keypoints because they were insignificant at the selected scale. (b) Lowering the scale by a factor of 4 detects keypoints at these locations. Only keypoints with Q_k between 35 and 45 are shown for brevity. (c) Principle axes ratio with respect to scale for three points. Two of the points have a local maxima at different scales. (d) Repeatability of keypoints between partially overlapping views using fixed scale and automatic scale selection.

on the x-axis of Fig. 5-c represents the sampling rate in mm. A $1mm$ sampling rate corresponds to $r_1 = 10mm$ and $1.5mm$ sampling rate corresponds to $r_1 = 15mm$. One of the points however, does not have a local maximum and is therefore not considered as a keypoint. It is worth mentioning that in automatic scale selection, the threshold t_1 is not required to be specified and points close to the boundary are unlikely to qualify as keypoints because their principle axes ratio does not exhibit a local maximum.

We compared the repeatability of keypoints using automatic scale selection and fixed scale. For this experiment, we calculate the repeatability between the 10 partially overlapping views each of the four objects in Fig. 2 as opposed to the repeatability between partial views and full 3D models as shown in Fig. 4-a. This setup is chosen to demonstrate the effectiveness of the proposed algorithm for partial views. Since the partial views do not have 100% overlap, repeatability is calculated for points that lie inside the overlapping regions of the partial views. Fig. 5-d shows the repeatability of the keypoints at fixed scale and when automatic scale selection is used. From Fig. 5-d, it is clear that our keypoint identification algorithm works equally good for partially overlapping views. However, the repeatability of keypoints is slightly higher when automatic scale selection is used.

6 Feature Extraction and Matching

We have validated the repeatability of the keypoint identification algorithm in the previous sections. However, the ultimate objective of the keypoint detection is to facilitate the subsequent 3D object retrieval task. This relates to our last two constraints defined in Section 2 namely invariance and descriptiveness of the features extracted at the keypoints. In this section, we will describe invariant feature extraction and in the next section we validate the performance of this feature for 3D object retrieval in the presence of clutter and occlusions.

Recall that the surface **S** fitted to the neighbourhood of a keypoint is aligned with its principal axes. However, without any prior knowledge or assumption of the shape, size and pose of the object, there could be two possible alignments i.e. the surface will still be aligned with its principal axes if it is rotated 180 degrees along the z-axis. To overcome this problem, both possibilities are considered. For the second possibility, **S** is flipped horizontally which amounts to 180^o rotation about the z-axis. While matching two surfaces, both possibilities must be considered only for one of the surfaces.

Since, the surface **S** (and its flipped version) is defined in a local coordinate basis derived from the principal axes of the surface itself, it is rotation invariant. This means that any feature extracted from this surface will also be rotation invariant. At the least, the depth values of the surface could be used as a feature vector of dimension 20 × 20 and projected to the PCA subspace for dimensionality reduction [20]. Our prior work [20] showed that these features give very high recognition rates when applied to 3D face recognition under neutral expressions. Note that the proposed technique applies to rigid objects only and is in contrast to the non-rigid 3D object retrieval work of Passalis et al. [28].

In this paper, we take the depth values of the **S** to form a feature vector and use a matching algorithm that clusters possible transformations between the query and database models for 3D object retrieval from cluttered scenes. Note that **S** has already been calculated at the keypoint detection stage and it is computationally efficient to use it as a feature. Another important thing to note is that in automatic scale selection, **S** will be calculated at different scales (for each keypoint) where the ratio between the principle axes of the neighbourhood exhibit a local maximum. Therefore, when automatic scale selection is used, **S** is vectorized and normalized to unity which means that the feature vector is effectively scale invariant as well.

A dimension of 400 is quite high for the representation of a local surface. The local features are projected to a PCA subspace. The PCA subspace can be learned from the local features extracted from the complete 3D models, their partial views or a combination of the two. In either case, the local features are similar and hence lead to the same PCA subspace. In our experiments, we used local features of all the 3D models in the database for training. Moreover, the same subspace was used to project the local features of the partial views when object retrieval was performed using partial views. Let **F** be the matrix of feature vectors where each column represents a different feature from the database 3D models. The covariance matrix of **F** is

$$\mathbf{C} = (\mathbf{F} - \mathbf{m})(\mathbf{F} - \mathbf{m})^{\mathbf{T}} \ , \tag{12}$$

where **m** is the mean feature vector. Using Singular Value Decomposition

$$\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{C} \ , \tag{13}$$

where **U** is the matrix of eigenvectors such that the eigenvector corresponding to the highest eigenvalue is in the first column. **S** is a sorted diagonal matrix of the eigenvalues. A PCA subspace is defined from the eigenvectors corresponding to the highest k eigenvalues. The value of k is chosen such that 95% fidelity is preserved in the data. In our experiments, $k = 10$ gave 95% fidelity and therefore, the subspace feature vectors were

$$\mathbf{F}' = \mathbf{U}_k^T \mathbf{F} \ , \tag{14}$$

where $\mathbf{U}_k^T$ is the transpose of the first k columns of matrix **U** in Eqn. 13. Each feature vector in $\mathbf{F}'$ has a dimension of 10 and is normalized to a unit vector.

During 3D object retrieval, local features from the query model are matched with the local features of each database model. Note that they could both be complete 3D models or partial surfaces or one of them could be a partial view and the other a complete model. Let $\mathbf{f}'_i$ be a local feature from the query model (or partial surface) and $\mathbf{f}'_j$ be the local feature from the database model. These error metric between the two features is given by

$$e = \cos^{-1}(\mathbf{f}'_i(\mathbf{f}'_j)^T) \ , \tag{15}$$

where the term between the brackets is the dot product of the two vectors and e represents their mutual angle. The value of e will vary from zero (perfectly similar feature vectors) to $\pi/2$ (complete dissimilar or orthogonal vectors). Local features from a query and database model that give the minimum error e are considered a match. Moreover, only one-to-one matches are considered i.e. a feature from a database model can be a match for only one query feature. If a database model feature turns out to be a match to more than one query features, only the one with the minimum value of e is considered.

The remaining list of matches (or correspondences) between the query and database model are clustered based on their relative transformation. Recall that each keypoint was extracted after aligning the local surface **L** with its principal axes (Eqn. 5). **V** together with the location of the keypoint **v** defines a local coordinate basis of the surface $\mathbf{L}'$. More precisely, **V** contains three orthogonal vectors pointing in the x, y and z-axes and **v** contains the origin of the new local 3D coordinate basis. The rigid transformation required to align a query feature to a database feature is given by

$$\mathbf{R} = \mathbf{V}_q^T \mathbf{V}_d \ , \tag{16}$$

$$\mathbf{t} = \mathbf{v}_q - \mathbf{v}_d \mathbf{R} \ , \tag{17}$$

where $\mathbf{V}_q$ and $\mathbf{V}_d$ are the orthogonal vectors (see Eqn. 4) of the query and database feature respectively and $\mathbf{v}_q$ and $\mathbf{v}_d$ are the location of the keypoints. **R** is the rotation matrix and **t** is the translation vector that will align the two features and hence their underlying 3D models. The rotation matrix **R** is converted to Euler angle rotations and concatenated with the translation vector to form a six dimensional vector. This vector represents the six degrees of freedom i.e. three rotations and three translations along the x, y and z-axis. This vector is calculated for all matching pairs of features between the query and database model and used to cluster the matches. Clustering can also be performed using the motion matrix itself over the manifold space [32] however, in our case, the local features are robust and give good results using simple methods such as the k-means and hierarchical clustering. We tested both k-means and hierarchical clustering methods in our experiments and obtained similar results. Matching pairs of features that agree on the same rotation and translation will fall into the same cluster. The cluster that contains the maximum number of matches defines the correct transformation between the query and database model. The average error e and the size of the largest cluster also give the similarity between the query and database model.

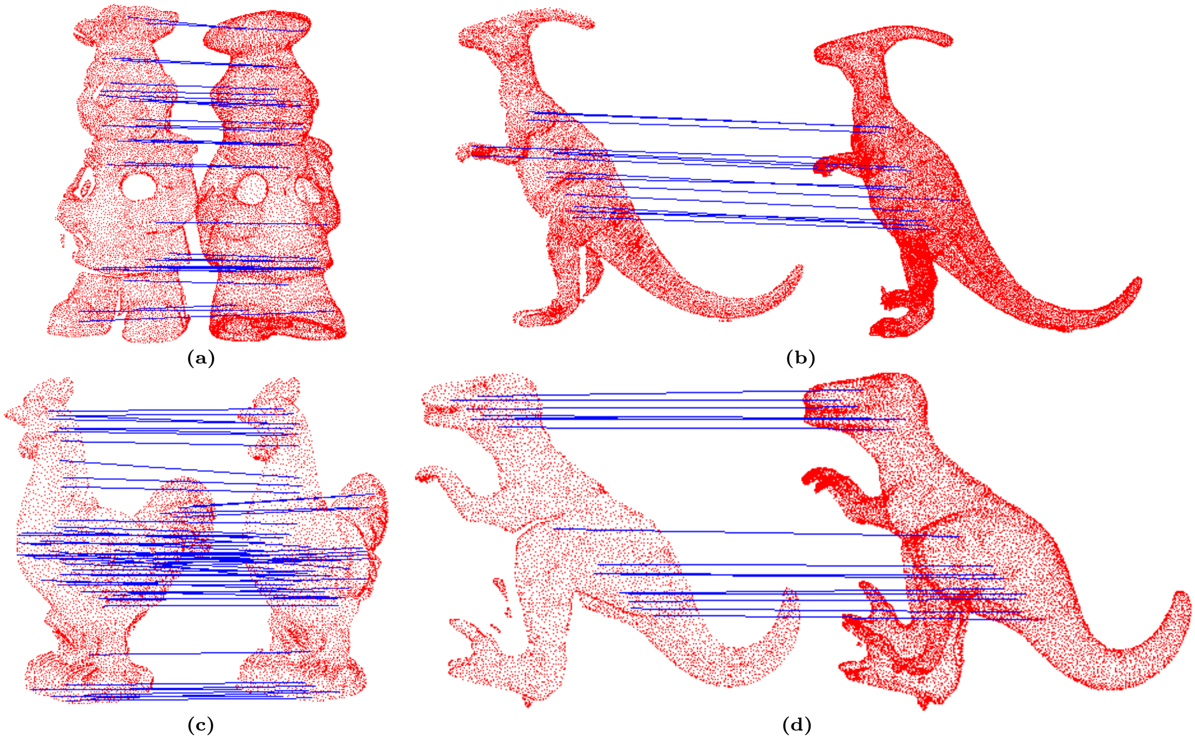

(a) (b) (c) (d)

Fig. 6 Illustration of matching partial views (left) to full models (right). Only half the matches in the largest cluster are shown as blue lines.

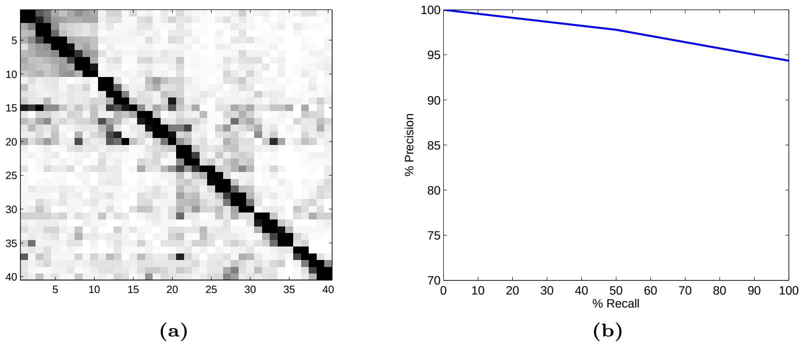


Fig. 7 (a) Confusion matrix of matching 10 partially overlapping views each of the four objects. The dark diagonal indicates high similarity between neighbouring (overlapping views). Dark dots away from the diagonal indicate false positives. (b) Precision recall curve calculated from the matrix.

7 3D Object Retrieval Experiments

We performed 3D object retrieval experiments using the publicly available database [22]. This section presents both qualitative and quantitative results for object retrieval using fixed scale and automatic scale selection.

7.1 Fixed scale Features

Fig. 6 shows qualitative results for matching partial views of the chef, parasaurolophus, chicken and t-rex with their complete 3D models. The matching pairs of keypoints are indicated by straight lines. Only half the number of matches in the largest cluster are shown. All matching experiments were performed at a single scale i.e. $r_1 = 20mm$. One of the side effects of a single fixed scale is that regions which have a small surface area e.g. the head of the parasaurolophus, do not contribute towards the final cluster of matches. This is mainly because they either do not qualify as keypoints due to the boundary condition (see Section 3.2) or cause large perturbations in the local coordinate basis or the extracted feature since the 20×20 lattice used to fit a local surface $\mathbf{S}$ is large compared to available surface data points i.e. $\mathbf{L}'$. This problem is overcome in the next Section by using automatic scale selection which extracts multiscale local features at keypoints which increases the number of correct matches and hence increase the chances of correct 3D object retrieval.

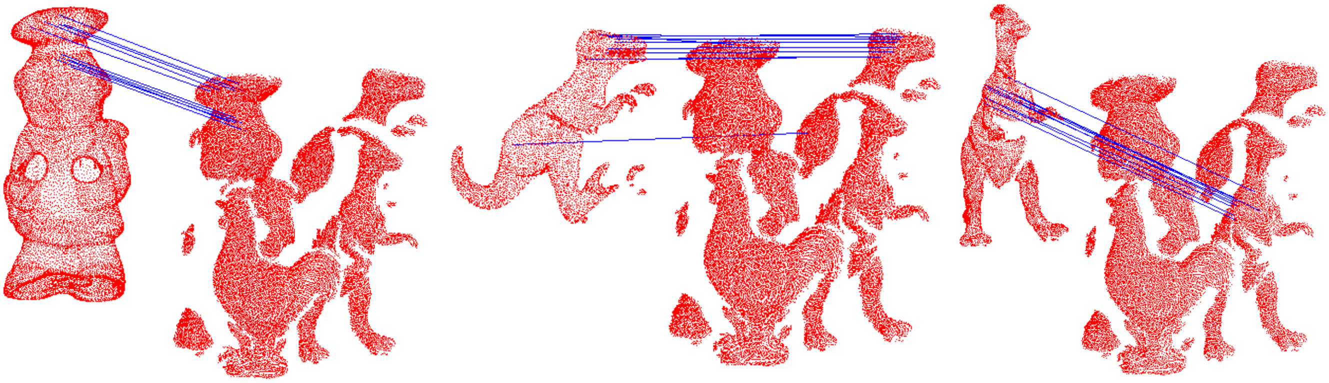


Fig. 8 Illustration of 3D model and view based object retrieval from a real wold scene containing clutter and occlusions. A full model is used to retrieve the chef in (a) whereas partial views are used to retrieve the T-rex in (b) and parasaurolophus in (c).

In the next experiment, we made a database of 40 partial views of the four objects. The first 10 partial views of each object were considered where each view had a partial overlap with the next view in the sequence. Each view was then used turn by turn as a query to retrieve its partially overlapping view (or views) from the remaining database of 39 views. Each query view was expected to retrieve the next and previous view in the sequence except for the first (i.e. view 1, 11, 21 and 31) and last views (i.e. 10, 20, 30, 40) in the sequence which were expected to retrieve only the next or previews views respectively. Fig. 7-a shows the confusion matrix of matching all the 40 views of the database with each other. The similarity score was calculated as the weighted sum of the normalized number of matches in the largest cluster and the normalized average error between the matching pairs of features in the largest cluster. The thick dark diagonal of the confusion matrix indicates the high similarity between the neighbouring overlapping views of the same object. Some dark patches away from the diagonal indicate false positives. Fig. 7-b shows the precision recall curve of the experiment indicating a 94.5% precision at 100% recall. It is important to emphasize that we calculate our results purely on the basis of feature matching unlike Novatnack and Nishino [27] who also consider the area of overlap between the two views after registering them. Feature matching can be performed efficiently using indexing whereas calculating the overlap between two views is a costly process.

Fig. 8 shows qualitative results for 3D object retrieval from a cluttered scene. Note that 3D object retrieval or recognition in a real scene containing clutter and occlusion is a very challenging scenario. In Fig. 8-a, a complete 3D model of the chef is used to retrieve its partial (and occluded) view from the cluttered scene. However, in Fig. 8-b and Fig. 8-c partial views of the t-rex and parasaurolophus are used for 3D object retrieval from the cluttered scene. Partial views were used to illustrate that the retrieval can also be done with partial views as long as some overlap exists between the query view and the view of the object in the scene.

7.2 Multi-scale Features

In automatic scale selection, features are extracted at multiple scales and normalized with respect to the scale. Therefore, these features are scale invariant and can be used for object retrieval at different unknown scales. Matching between multi-scale features at unknown scales of the objects is performed in a similar way except that only the three rotations and the scale ratio between the matching pairs of features are used to cluster the matches. Translation is not used because correct matching pairs will not have the same translation if the query and target object have different scales. Fig. 9 shows two partially overlapping views of the parasauralophus matched at different scales. Note that unlike the fixed scale features which matched only at the large regions of the parasauralophus (see Fig. 6-b), multi-scale features are also matched from the smaller regions e.g. the head, neck, tail and foot. This is because features are extracted at smaller scales from these regions. Fig. 10 shows object retrieval from a cluttered scene at unknown scale. The proposed algorithm was able to find a large number of correct correspondences in this complex case. A few incorrect matches do end up in the biggest cluster because clustering is performed using rotations from individual pairs of matches. These matches are easily eliminated as outliers when determining the overall transformation between the query and target object.

In our final experiment, we use the multi-scale features for object retrieval from all the 50 cluttered scenes [22] and compare the results to tensor matching [22] and the spin image matching [15]. The tensors and spin

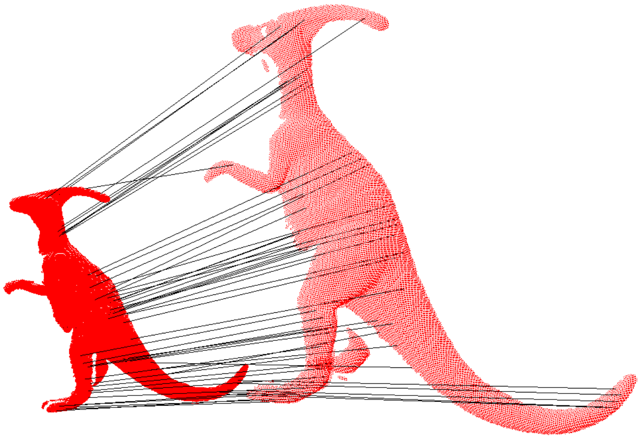


Fig. 9 Scale invariant matching between two partially overlapping views at different unknown scales.

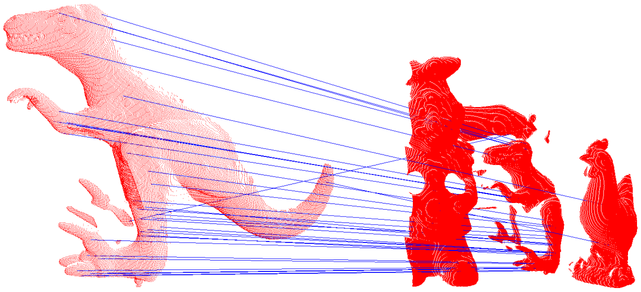


Fig. 10 Scale invariant object retrieval from a cluttered scene.

images were both calculated using fixed neighbourhood regions. For a fare comparison, we used the spin image recognition implementation available at [19] and used recognition without spin image compression. Four to five objects were placed randomly and scanned from a single viewpoint to construct 50 complex scenes containing clutter and occlusions [22]. Matching was performed using multi-scale features (i.e. automatic scale selection) but for comparison reasons the query object and scene were both at the same scale. The correspondences in the largest cluster were used to align the query object with the scene. If the query object was aligned correctly to its view in the scene, the recognition was considered a success. In the event of alignment to a different object or incorrect alignment, the recognition was considered a failure. The object recognition results are presented with respect to occlusion for easy comparison with the other two techniques. Occlusion is defined according to [15] as

$$\text{occlusion} = 1 - \frac{\text{object surface area visible in scene}}{\text{total surface area of object}} \quad (18)$$

Ground truth occlusions were automatically calculated for correctly recognized objects i.e. by registering the object to the scene, and manually calculated for objects that could not be recognized due to excessive occlusion. The ground truth occlusions, locations and orientations of objects in the scenes are available at [23] along with the mesh data. Fig. 11 shows the results of our experiment and compares the performance of the proposed algorithm with tensor matching [22] and spin image matching [15]. The proposed keypoint matching algorithm has 100% recognition rate up to 70% occlusion. The recognition rate is above 95% up to 80% occlusion. This performance is better than the spin image matching and almost equal to the tensor matching. However, the proposed algorithm's performance degrades more gracefully with increasing occlusion. Moreover, the recognition rate of our proposed algorithm in Fig. 11 is calculated purely on the basis of matching features and as opposed to [22] no costly verification steps were used.

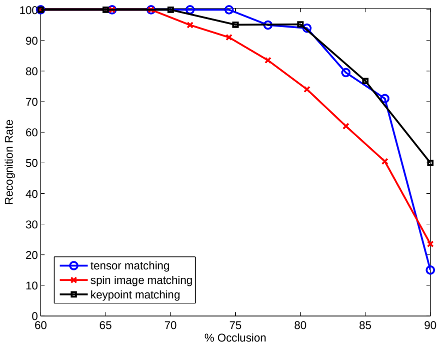


Fig. 11 Recognition rate of the proposed keypoint features compared to the tensor matching and spin images.

The time required to calculate the scale invariant features depends upon the density of the pointcloud, the number of scales over which the search is performed and the shape of the object. Objects with rich features tend to take more time compared to objects with planar surfaces because more points qualify as keypoints in the initial stages. Using pointclouds with $2mm$ resolution and searching over 6 scales, our Matlab implementation of the algorithm can detect 50 keypoints/sec and extract local features from them at the same time on a 2.3 GHz machine with 4GB RAM. Likewise, the matching time also depends upon the number of local features. Our Matlab implementation takes 0.12 seconds to match 500 local features of a query object to 1000 local features of a complex scene and cluster the matches.

8 Conclusion and Discussion

We presented a keypoint detection algorithm along with an automatic scale selection technique for subsequent feature extraction. The automatic scale selection technique facilitates multi-scale features which are scale invariant and can be used for matching objects at different and unknown scales. We also proposed a keypoint quality measure which is used to rank keypoints and select the best keypoints for further processing. The detected keypoints are highly repeatable between 3D models and its partial 2.5D views and between partially overlapping 2.5D views. Our results show that the proposed keypoint quality measure is in agreement with the repeatability of the keypoints as higher quality keypoints have higher repeatability rates. This is of great significance as only a few keypoints that are highly repeatable can be used to extract features and reduce computational complexity.

We performed extensive experiments for object retrieval from complex scenes containing clutter and occlusions. Our results show that the proposed features are highly distinctive and achieve over 95% recognition rate up to 80% occlusion in complex scenes. It is important to emphasize that the results presented in this paper are purely on the basis of feature matching and costly verification steps have been avoided. Such verification steps include the calculation of the area of overlap between the query and target object after registering them using the corresponding pairs of features. A more accurate overlap is some times calculated after iteratively refining the registration with the ICP algorithm [3]. Another common verification step in 3D object retrieval is active space violation which determines if any part of the query object after registration to the scene comes in the way of the sensor. Although, such verification steps can significantly improve the results, they are computationally costly and overshadow the performance of the 3D descriptors. Reporting combined results using feature matching followed by one or more of the verification procedures makes it hard to separate the performance of the feature matching from the overall framework.

Another important aspect of the proposed features is that each feature has an underlying local 3D reference frame and a single pair of correctly matching features is sufficient to align the query model (or partial view) with the database model (or partial view). This means that clustering can be performed using individual pairs of matches. In the absence of these 3D reference frames, an C_3^N search would need to be performed to cluster the matches (where N is the total number of matches and 3 is the minimum number of correspondences required to register two surfaces).

Acknowledgements We thank the anonymous reviewers for their detailed comments and suggestions which resulted in the improvement of this paper. We also acknowledge J. D'Erico for the surface fitting code [8]. This research is sponsored by ARC Grants DP0881813, DP0664228 and LE0775672.

References

1. E. Angel, "Interactive Computer Graphics, A Top-down Approach Using OpenGL", *Addison Wesley*, 5th Edition, 2009.
2. A. Ashbrook, R. Fisher, C. Robertson and N. Werghi, "Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, pp. 674–686, 1998.
3. P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 14(2), pp. 239–256, 1992.
4. K.W. Bowyer, K. Chang and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition", *Computer Vision and Image Understanding*, vol. 101(1), pp. 1–15, 2006.
5. R.J. Campbell and P.J. Flynn, "A Survey of Free-Form Object Representation and Recognition Techniques," *Computer Vision and Image Understanding*, vol. 81(2), pp. 166-210, 2001.
6. U. Castellani, M. Cristani, S. Fantoni and V. Murino, "Sparse points matching by combining 3D mesh saliency with statistical descriptors", *Eurographics*, vol. 27(2), pp. 643-652, 2008.
7. C.S. Chua and R. Jarvis, "Point Signatures: A New Representation for 3D Object Recognition," *International Journal of Computer Vision*, vol. 25(1), pp. 63-85, 1997.
8. J. D'Erico, "Surface Fitting using Gridfit", *MATLAB Central File Exchange*, 2008.
9. C. Dorai and A.K. Jain, "COSMOS: A Representation Scheme for 3D Free-Form Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(10), pp. 1115-1130, 1997.
10. N. Gelfand and L. Ikemoto, "Geometrically Stable Sampling for the ICP Algorithm", *3D Digital Imaging and Modeling*, 2003.
11. N. Gelfand, N. Mitra, L. Guibas, and H. Pottmann, "Robust Global Registration", *Eurographics Symposium on Geometry processing*, 2005.
12. R.C. Gonzalez and R. E. Woods, "Digital Image Processing", *Addison Wesley*, 3rd Edition, 2008.
13. M. Hebert, K. Ikeuchi, and H. Delingette, "A Spherical Representation for Recognition of Free-Form Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17(7), pp. 681-690, 1995.
14. G. Hetzel, B Leibe, P. Levi and B. Schiele, "3D object recognition from range images using local feature histograms", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 394–399, 2001.
15. A.E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21(5), pp. 674-686, 1999.
16. T. Joshi, J. Ponce, B. Vijayakumar, and D. Kriegman, "Hot Curves for Modeling and Recognition of Smooth Curved 3D Objects," IEEE Conference on Computer Vision and Pattern Recognition, pp. 876-880, 1994.
17. C.H. Lee, A. Varshney and D.W. Jacobs, "Mesh Saliency", *SIGGRAPH*, pp. 659–666, 2005.
18. G. Mamic and M. Bennamoun, "Representation and Recognition of Free-form Objects", *Digital Signal Processing*, vol. 12, pp. 47–76, 2002.
19. "Mesh ToolBox", Vision and Mobile Robotics Laboratory, Carnegie Mellon University, http://www.cs.cmu.edu/∼vmr/software/meshtoolbox/executables.html, 2009.

20. A. Mian, M. Bennamoun and R. Owens, “Keypoint Detection and Local Feature Matching for Textured 3D Face Recognition”, *International Journal of Computer Vision*, vol. 79(1), pp. 1–12, 2008.
21. A. Mian, M. Bennamoun and R. Owens, “A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images”, *International Journal of Computer Vision*, vol. 66, pp. 19–40, 2006.
22. A. Mian, M. Bennamoun and R. Owens, “3D Model-based Object Recognition and Segmentation in Cluttered Scenes”, *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 28(10), pp. 1584–1601, 2006.
23. A. Mian, Home Page, 3D modeling and 3D object recognition data, http://www.csse.uwa.edu.au/∼ajmal/, 2009.
24. D. Lowe, “Distinctive Image Features from Scale-invariant Key Points,” *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.
25. H. Murase and S. Nayar, “Visual Learning and Recognition of 3D Objects from Appearance”, *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.
26. J. Novatnack and K. Nishino, “Scale-Dependent 3D Geometric Features”, *International Conference on Computer Vision*, pp. 1–8, 2007.
27. J. Novatnack and K. Nishino, “Scale-Dependent/Invariant Local 3D Shape Descriptors for Fully Automatic Registration of Multiple Sets of Range Images”, *ECCV*, pp. 440–453, 2008.
28. G. Passalis, I. Kakadiaris, T. Theoharis, “Intraclass Retrieval of Nonrigid 3D Objects: Application to Face Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29(2), pp, 218-229, 2007.
29. S. Petitjean, “A survey of methods for recovering quadrics in triangle meshes”, *ACM Computing Surveys*, vol. 34(2), pp. 211–262, 2002.
30. J. Ponce and D.J. Kriegman, “On Recognizing and Positioning Curved 3D Objects from Image Contours,” *Image Understanding Workshop*, pp. 461–470, 1989.
31. J. Shi and C. Tomasi, “Good Features to Track”, *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.
32. R. Subbarao and P. Meer, “Nonlinear Mean Shift for Clustering over Analytic Manifolds”, *CVPR*, pp. 1168–1175, 2006.
33. Y. Sumi, Y. Kawai, T. Yoshimi and F. Tomita, “3D Object Recognition in Cluttered Environments by Segment-Based Stereo Vision”, *International Journal of Computer Vision*, 46(1), pp. 5-23, 2002.
34. R. Unnikrishnan, J. Lalonde, N. Vandapel and M. Hebert, “Scale Selection for the Analysis of Point-Sampled Curves”, *3DPVT*, pp. 1026–1033, 2006.
35. R. Unnikrishan and M. Hebert, “Multi-scale Interest Regions from Unorganized Point Clouds”, *CVPR Workshop*, 2008.
36. R. Wessel, R. Baranowski and R. Klein, “Learning Distinctive Local Object Characteristics for 3D Shape Retrieval”, *Vision, Modeling, and Visualization*, pp. 167–178, 2008.