Pattern Recognition 43 (2010) 3833–3845

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Viewpoint independent object recognition in cluttered scenes exploiting ray-triangle intersection and SIFT algorithms

Georgios Kordelas, Petros Daras *

Informatics & Telematics Institute, 1st km Thermi Panorama Road, 57001 Thermi, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 9 April 2009
Received in revised form
29 March 2010
Accepted 22 May 2010

Keywords:
3D object recognition
Distance maps
Ray-triangle intersection
Clutter
Occlusion

ABSTRACT

Viewpoint independent recognition of free-form objects and estimation of their exact position are a complex procedure with applications in robotics, artificial intelligence, computer vision and many other scientific fields. In this paper a novel approach is presented that addresses recognition of objects lying in highly cluttered and occluded scenes. The proposed procedure relies on distance maps, which are extracted and stored off-line for each of the 3D objects that might be contained in the scene. During the on-line recognition procedure distance maps are extracted from the scene. Greyscale images, derived from scene's distance maps, are matched with those of the object under recognition by applying similarity measures to the descriptors that are extracted from the images. The similarity is then estimated from image patches, which are defined using the SIFT descriptor in an appropriate way. After finding the best similarities the position of the object in the scene is estimated. This process is repeated until all objects are successfully recognized. Multiple experiments, which were performed on both 2.5D synthetic and real scenes, proved that the proposed method is robust and highly efficient to a satisfactory degree of occlusion and clutter.

1. Introduction

In the recent years, significant progress has been made towards the recognition of free-form objects. The immediate objective of object recognition systems is to correctly identify an object in a scene of objects, in the presence of clutter and occlusion and to estimate its position and orientation. Those systems can be exploited in robotic applications where robots are required to navigate in crowded environments and use their equipment (i.e. range scanners, arms) to recognize and manipulate objects. Robots with advanced capabilities could be used to service elderly/impaired people or for surveillance in sensitive environments. Object recognition can be performed using 2D images, which is an affordable solution due to the wide availability of low cost cameras. Approaches exploiting cameras are fast and low cost, yet they are also very sensitive to illuminations, shadows and occlusions and do not provide accurate estimation of object's pose. Thus, the focus of the relevant scientific communities is on the development of 3D object recognition algorithms that overcome the aforementioned limitations.

The idea of recognizing objects in range data has already been investigated in several scientific studies. Campbell's and Flynn's survey [1] provides an extended overview of 3D object recognition techniques. However, a short complement to this survey and a report to recent methods is presented here for the sake of completeness. COSMOS [2], one of the earliest algorithms, is based on the computation of principal curvatures of the surface. This method is limited to objects with smooth surfaces and is applicable to just unoccluded views of an object. Chua and Jarvis [3] propose a point signature (PS) for 3D object recognition where a sphere centered at a given point is intersected with the surface and creates a 3D space curve on which a plane is fitted. Point signature was proved to be sensitive to noise and surface sampling [14]. Hetzel et al. [13] combine pixel depth, surface normals and curvature in a multidimensional histogram in order to directly model the probability distribution of different feature combinations. Their experiments proved the efficiency of this method; however, the database used includes only non-cluttered, self-occluded range images of 30 free-form objects. Johnson and Hebert propose the spin image method [7], which is vulnerable to sampling and resolution (level-of-detail) of the models and has low discriminative power. Additionally, this method is applied to every vertex of the object or the scene, therefore the number of the descriptors increases as the number of vertices does. When the number of descriptors is compressed, using principal component analysis (PCA), the average recognition rate decreases significantly (almost 10%). Nevertheless, spin images have been used in many applications such as parts-based 3D object classification [4] and for recognizing members of classes of

* Corresponding author.
E-mail addresses: kordelas@iti.gr (G. Kordelas), daras@iti.gr (P. Daras).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.030

3D shapes [5]. In [9], an enhancement of the spin images algorithm is presented by using vertex interpolation. Although these changes resolved sensitiveness of spin images to variations in resolution, descriptor's discriminative power was not improved significantly. Spherical harmonics [11] and locality-sensitive hashing [12] are exploited in [10] to perform efficient retrieval of shapes; the work is tested on 3D shape information obtained from laser range scanners. However, the approximate location of the shape to be retrieved is already known, thus algorithm's task is limited to identify database's correct shape. Mian et al. [14], recently proposed a tensor-based surface representation defined on pairs of oriented points. Their descriptors are 3D tensors that measure the variation of surface position. Correspondence between 3D Tensors is established using a voting process to find pairs of tensors with high overlap ratio.

A more recent approach is presented in [17], where the similarities between input 3D images are computed by matching their descriptors with a pyramid kernel function. The similarity matrix of the images is used to train support vector machines-based (SVM) classification [19], and new images can be recognized by comparison with the training set. The experiments were performed on the same database as in [13], and thus robustness with respect to clutter was not examined. In [18], an initial implementation of the distance map descriptor was presented; however, this approach was viewpoint dependent. The algorithm presented in [16] (an extension of work in [15]) calculates the local surface properties of patches, which are defined on the extracted feature points. By comparing local surface patches for a model and a test image, and casting votes for the models containing similar surface descriptors, the potential corresponding local surface patches and candidate models are hypothesized. The evaluation experiments were simple, since at most two objects existed in the scene. In [20], the generalized Hough transform is extended to detect instances of an object in laser range data, independently to the scale and orientation of the object. However, this method is restricted to simple objects that can be represented with few parameters, such as planes, spheres and cylinders.

The plethora of the existing algorithms [6,17,5,4] use spin images [7]. These methods either modify the spin image or integrate it with other components, so as to improve its performance. Moreover, the majority of the methods was tested on self-occluded scenes without presence of clutter [2,4,3,13]. Thus, there is a need for the development of novel methods that address the object recognition problem in a more efficient way.

In this paper a novel approach for recognition of 3D objects in range scenes, is presented. The primary step of the proposed algorithm is to place the 3D object in a proper position and then to form a coordinate basis used to extract distance maps for this object. During the 3D object's recognition procedure, distance maps are extracted for the scene according to a coordinate system, which allows keeping their total number very low. Matching between scene's and object's distance maps is established using the SIFT algorithm on greyscale images that are generated from the distance maps. The whole procedure is novel and provides a different insight in the "treatment" of the object recognition problem. A major difference to previous methods (i.e. [7]), where descriptors are extracted on the vertices of the reconstructed point cloud, lies on the extraction of scene's descriptors, which is based on a coordinate system that is formed according to scanning parameters.

The advantages of the proposed method are the following: the approach used to extract distance maps, especially for the scene, allows keeping their number low since it is independent of 3D object's number of vertices. Added to this, the employment of a simple 1D hash table allows significant acceleration of the execution time. Another advantage is its robustness with respect to objects' level-of-detail since, unlike spin images, it is not required the library objects to have similar resolution.

The results on synthetic scenes proved that the proposed algorithm is robust to a high degree of clutter and occlusion and experimental comparison with the spin image approach on real scenes verified the superiority of the proposed algorithm.

The rest of this paper is organized as follows. In Section 2, the off-line extraction of 3D object's distance maps along with the automatic extraction of scene's distance maps are presented. Section 3 introduces a similarity measure based on the SIFT algorithm. In Section 4, the performed experiments both on synthetic and real data are given, while conclusions are drawn in Section 5.

2. Computation of distance maps

2.1. Model's initial distance maps

The goal of this procedure is twofold: firstly to place the 3D model in a proper initial position and secondly to create a coordinate basis around the object, which is used to define object's initial distance maps in such a way that largest portion of object's surface will be described with the minimum number of descriptors.

2.1.1. Initial position of 3D model

Each model's vertices are stored in the matrix $\boldsymbol{V}_{model}$ (where $\boldsymbol{V}_{model}$ is a $N\times 3$ matrix of 3D coordinates). The PCA [8] on $\boldsymbol{V}_{model}$ is computed and the three orthogonal principal components are derived. The object is rotated around its center of mass, so that the first principal component becomes parallel to z-axis and the second principal component becomes parallel to y-axis. After rotation, the object is denoted as $\boldsymbol{V}_{PCA}$. The object is then translated by $\boldsymbol{V}_{final}=\boldsymbol{V}_{PCA}-[x_m,y_m,z_a]$, where $\boldsymbol{C}_m=[x_m,y_m,z_m]^T$ is $\boldsymbol{V}_{PCA}$'s center of mass and $\boldsymbol{P}_a=[x_a,y_a,z_a]^T$ is $\boldsymbol{V}_{PCA}$'s point with minimum z-coordinate. This procedure intends to place the object in such a position that z-axis passes centrally through object's volume since the coordinate basis used to extract object's initial distance maps is constructed around z-axis. The points of intersection between $\boldsymbol{V}_{final}$ and z-axis with minimum z-coordinate and maximum z-coordinate are $\boldsymbol{P}_{mim}=[x_{min},y_{min},z_{min}]^T$ and $\boldsymbol{P}_{max}=[x_{max},y_{max},z_{max}]^T$, respectively (Fig. 1(a.2) and (b.2)). Fig. 1 depicts two objects after estimation of their initial position.

2.1.2. Extraction of 3D object's initial distance map

2.1.2.1. Circular sector formation. Before advancing to the extraction of initial distance maps, a circular sector S of N points, indexed by variable f ($f=0,1,\ldots,N$), with radius R (a global parameter used throughout this paper) and center $O=[0,0,0]^T$ is created on xy plane. Circular sector's points are sampled uniformly on the circular disc by creating a centroidal Voronoi tessellation (CVT) [21] of points within the sector region. Since points are sampled uniformly its rather impossible that a point coincides with O; however, the circular sector's point that has the minimum Euclidean distance from O is denoted as point K and it is assumed to coincide with O. The distribution of points over a specific circular area, using a polar coordinate system and CVT is depicted on Fig. 2(a) and (b), respectively. This figure proves the efficiency of CVT to generate uniform points. S is adapted around points of

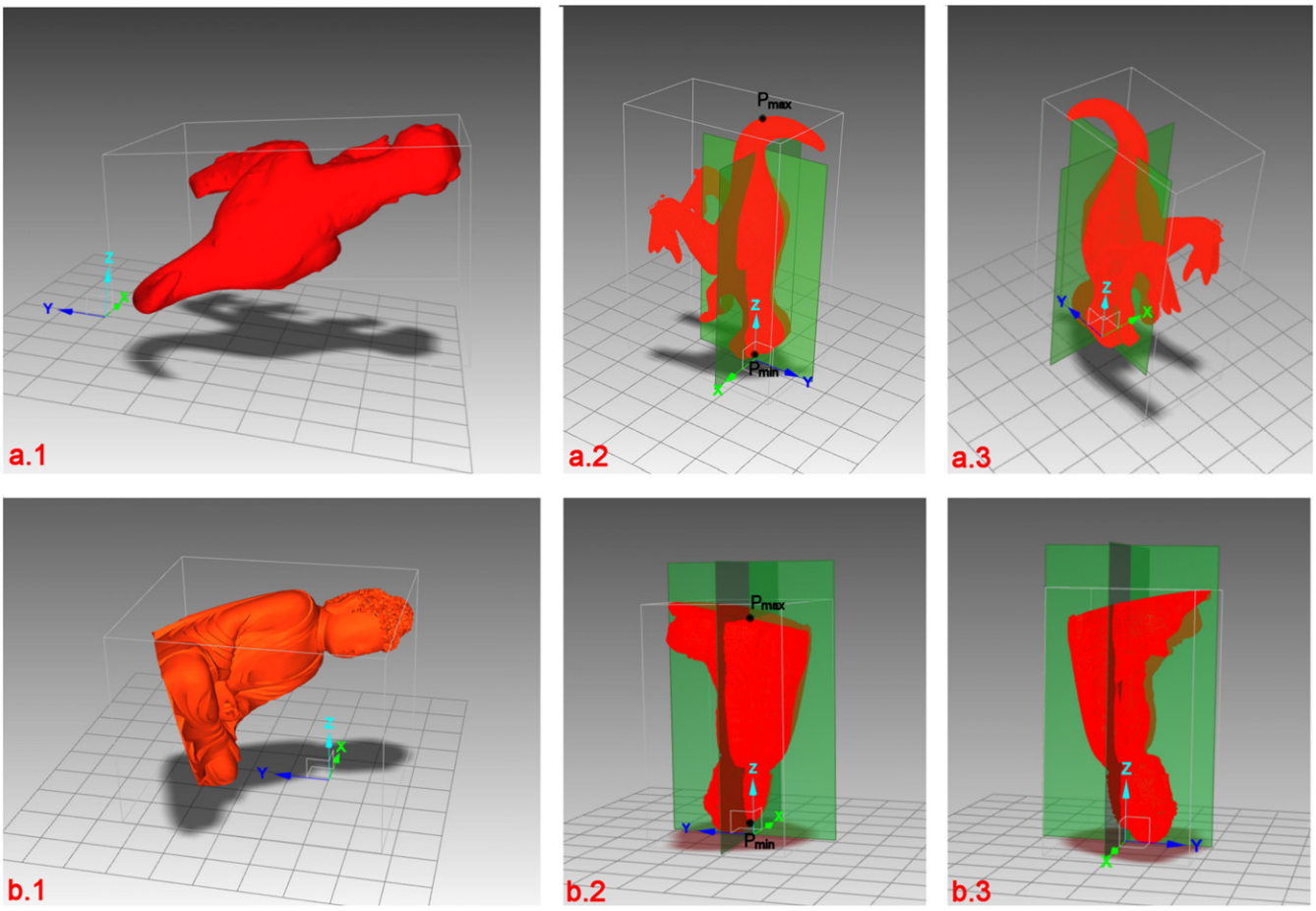

Fig. 1. (a.1) Random position of the "T-rex" model, (a.2) initial position of "T-rex", (a.3) initial position from a different viewpoint, (b.1) random position of the "budha" model, (b.2) initial position of "budha", (b.3) initial position from a different viewpoint.

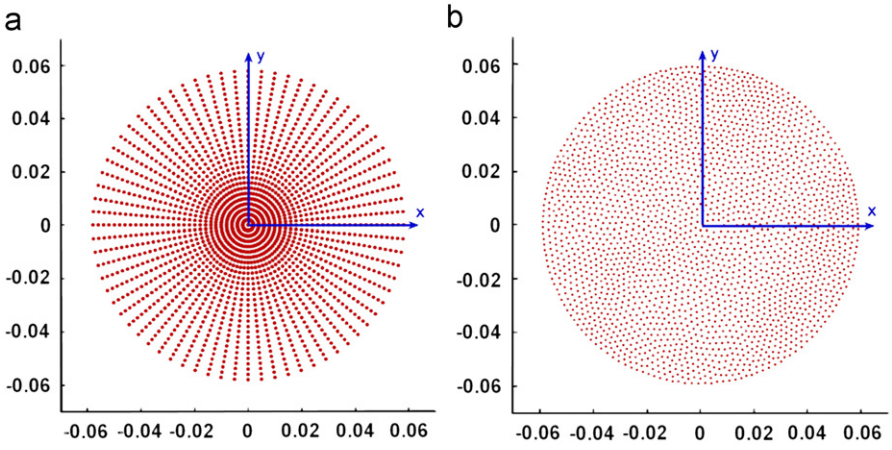

Fig. 2. Circular sector's sampled points using (a) polar coordinates and (b) centroidal Voronoi tessellation.

spherical sub-grids that form object's coordinate basis as described below.

2.1.2.2. Sub-grids formation. Initially a $\boldsymbol{G}=\{G_i, i=1,\ldots,N_G\}$ (Fig. 3(a)) grid of points, which are uniformly distributed on a sphere, is created. G points' spherical coordinates have been precomputed, so that for a G_i point the longitude and the latitude are $\theta_i \in [0,360^\circ)$ and $\phi_i \in [-90^\circ, +90^\circ]$, respectively. From $\boldsymbol{G}$ three sub-grids centered at $O=[0,0,0]^T$ are derived, by applying latitude thresholds. These are: $\boldsymbol{G}_a \subset G$ $(\forall \phi_i < +60^\circ)$ (Fig. 3(b)), $\boldsymbol{G}_b \subset G$ $(\forall \phi_i > -60^\circ)$ (Fig. 3(c)) and $\boldsymbol{G}_c = \boldsymbol{G}_a \cap \boldsymbol{G}_b$ (Fig. 3(d)).

Then, a z-axis parameter is defined as $h_j \in \{a \cdot j + z_{min} + R;\ j=0,1,2,\ldots,H/a\}$, where a is the z-axis variable and $H=|z_{max}-z_{min}|-2\cdot R$. $\boldsymbol{G}_a$ is adapted around point $g_1=[0,\ 0,\ R]^T$ to cap the bottom part of the object, $\boldsymbol{G}_b$ is adapted around point

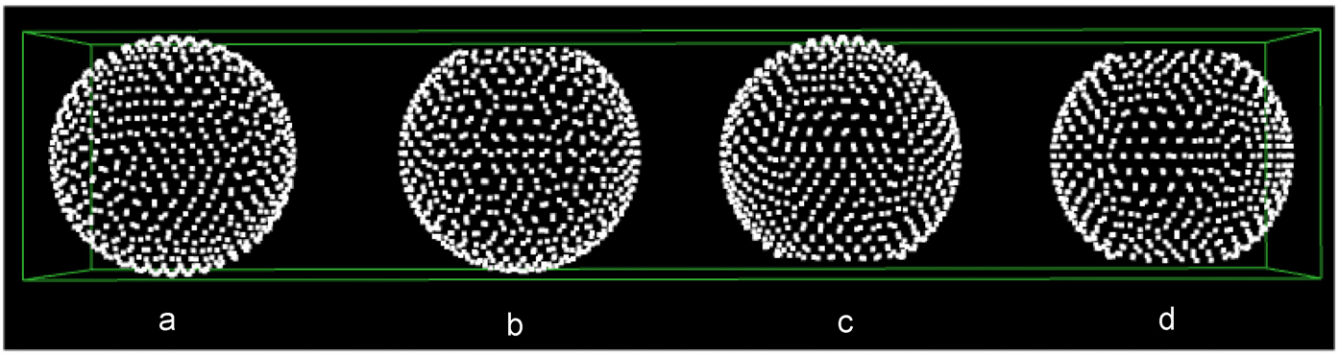

Fig. 3. Spherical grid (a) $\boldsymbol{G}$ and its sub-grids: (b) $\boldsymbol{G}_a$, (c) $\boldsymbol{G}_b$, (d) $\boldsymbol{G}_c$.

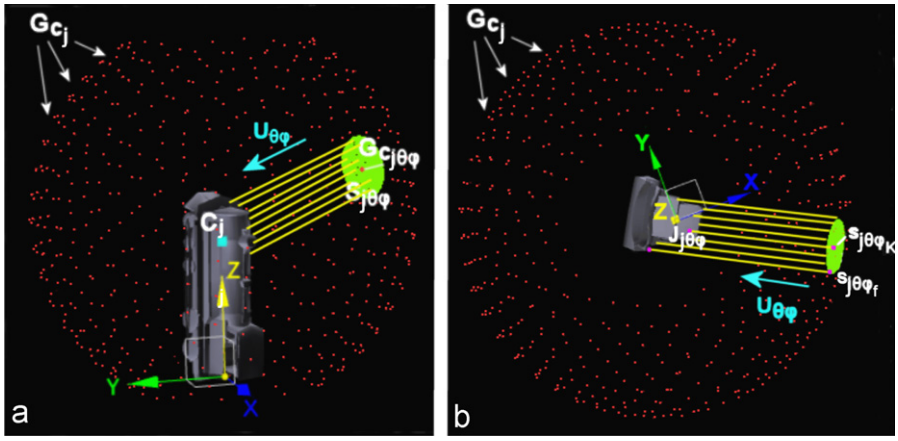

Fig. 4. Illustration of 3D object's initial distance map computation: (a) frontal view and (b) overview.

$g_2=[0,\ 0,\ H]^T$ to cap the upper part of the object, while $\boldsymbol{G}_c$ is adapted sequentially to points that correspond to all intermediate values of h.

Provided that the descriptors for h_j are computed, the following initial steps are required:

1. The sub-grid corresponding to h_j is formed as $G_{C_j}=G_\psi+C_j$, where $C_j=[0,0,h_j]^T$ (cases: *I.* $C_j\equiv g_1\Rightarrow\psi=a$, *II.* $C_j\equiv g_2\Rightarrow\psi=b$, *III.* $\psi=c$) (Fig. 4(a)).
2. Equation: $S_{j\theta\phi}=\boldsymbol{R}_z(-\theta)\boldsymbol{R}_y(-(\pi/2-\phi))S+G_{C_{j\theta\phi}}$ adapts circular disc $S_{j\theta\phi}$ around each point $G_{C_{j\theta\phi}}$ of G_{C_j} (Fig. 4(a) and (b)), where $\boldsymbol{R}_z$, $\boldsymbol{R}_y$ are the rotation matrices about the y and z-axes, respectively.

Each point $s_{j\theta\phi_f}\in S_{j\theta\phi}$ is the origin of a ray with direction $\overrightarrow{v}_{\theta\phi}=[-\cos(\theta)\cos(\phi),-\sin(\theta)\cos(\phi),-\sin(\phi)]^T$ (Fig. 4(b)).

Using the ray-triangle intersection algorithm, presented in [22], the distance $d_{j\theta\phi_f}$ between $s_{j\theta\phi_f}$ and the triangulated $\boldsymbol{V}_{model}$ is computed. The minimum distance per $S_{j\theta\phi}$ is $d_{j\theta\phi_0}=\min(d_{j\theta\phi_f})$. The point giving $d_{j\theta\phi_0}$ is denoted as $s_{j\theta\phi_0}$. In cases the ray intersects more than one triangles, the smallest distance is stored for a specific point. The intersection of the ray with origin $s_{j\theta\phi_K}$, with the model at point $\boldsymbol{J}_{j\theta\phi}$ (Fig. 4(b)) is given by the following equation and is called $S_{j\theta\phi}$'s central intersection.

$$\boldsymbol{J}_{j\theta\phi}=\overrightarrow{v}_{\theta\phi}\cdot d_{j\theta\phi_K}+s_{j\theta\phi_K} \tag{1}$$

The computed distances for all $s_{j\theta\phi_f}$ are used to extract the initial distance map $\boldsymbol{\Phi}_{j\theta\phi}$ per $S_{j\theta\phi}$. The cartesian coordinates of a point in $\boldsymbol{\Phi}_{j\theta\phi}$ are $\Phi_{j\theta\phi_f}=[x_f,y_f,d_{j\theta\phi_f}-d_{j\theta\phi_0}]^T$, where x_f, y_f are the abscissa and the ordinate of point f on S (Section 2.1.2.1). Points that have z-coordinate below $2R$ are stored in the initial distance map. Initial distance maps for all models are created and stored (off-line) in a model library. So far the methodology for the extraction of model initial distance maps is encapsulated in the following steps:

- G_{C_j} grids are centered at different C_j to form the coordinate basis for the extraction of model initial distance maps.
- A $S_{j\theta\phi}$ circular sector is adapted around each point of G_{C_j}.
- The initial distance map $\boldsymbol{\Phi}_{j\theta\phi}$ per $S_{j\theta\phi}$ is computed.

2.2. Scene's initial distance maps

Let us suppose that a 3D reconstruction computer vision system, which is placed in a predefined position in a room, creates a triangulated mesh of the observed scene. From the scanning parameters, the parallelogram volume that encloses the reconstructed objects can be derived. The variables that define this parallelogram are W (width), L (length) and D(depth). The viewpoint of observation (i.e. laser scanner center) is $\boldsymbol{D}=[x_d,y_d,z_d]^T$ and the depth scanning direction is $\overrightarrow{q}=[q_1,q_2,q_3]^T$ (Fig. 6(c)). Then a CVT of points, enclosed in an orthogonal region defined by the parameters W, L, is created. The distribution of points over a specific parallelogram area, using a cartesian coordinate system or CVT is depicted in Fig. 5(a) and (b), respectively. Although points in Fig. 5(a) are uniformly sampled over the plane, their coordinates are strictly defined $((i\cdot\alpha,j\cdot\alpha)$, where $\{i,j\}\in\mathbb{N})$, while in Fig. 5(b) points are uniformly sampled without any systematic way, thus CVT is selected for the generation of points.

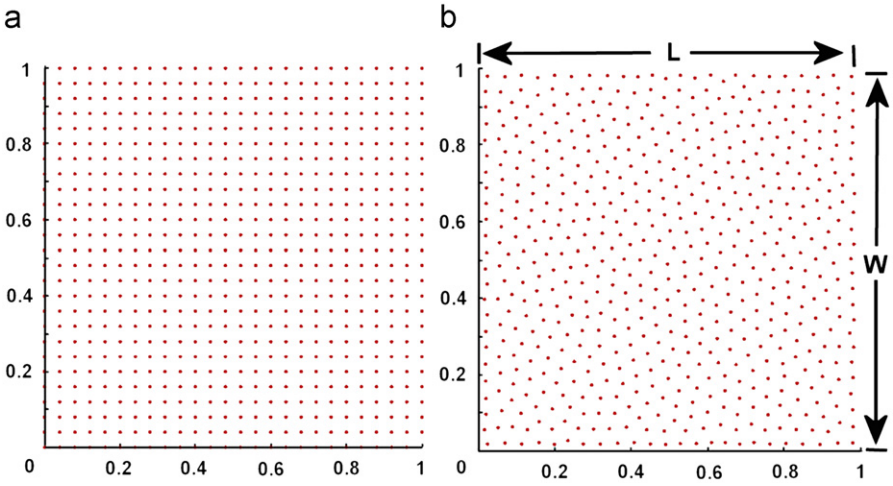

Fig. 5. Orthogonal region points in (a) method [1] and (b) in the proposed method.

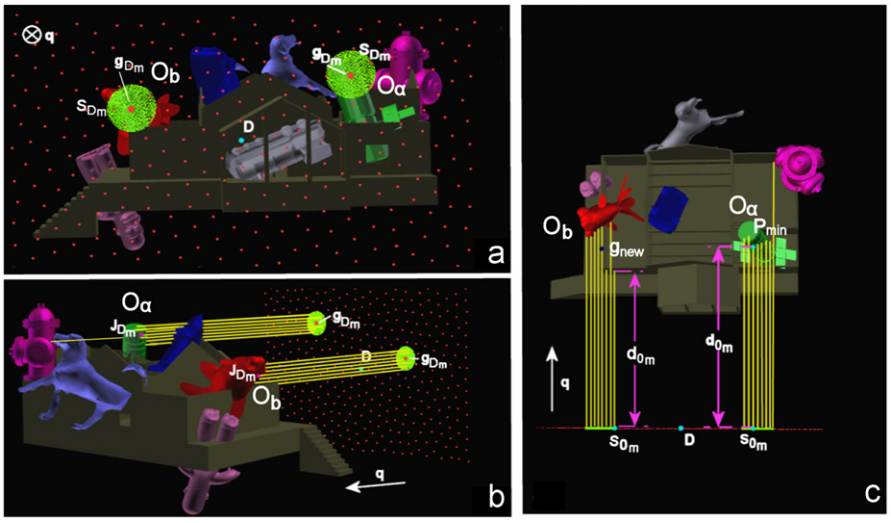

Fig. 6. Illustration of scene's initial distance map computation: (a) frontal view, (b) sidelong view and (c) overview.

A point on G_D is indexed by variable m ($m=0,1,\ldots,M$), where M is the total number of G_D points. The centered at $\boldsymbol{D}$, G_D grid is given by

$$G_D = \boldsymbol{R_z}(-\theta_1)\boldsymbol{R_y}(-(\pi/2-\phi_1))G+\boldsymbol{D} \tag{2}$$

where $\boldsymbol{R}_z$ and $\boldsymbol{R}_y$ are the rotation matrices about y and z axes, respectively, and $\overrightarrow{q}$'s spherical coordinates are θ_1(*longitude*) and ϕ_1(*latitude*). Each point $\boldsymbol{g}_{D_m} \in G_D$ is the center of a circular sector S_{D_m} (Fig. 6(a) and (b)) given by

$$S_{D_m} = \boldsymbol{R_z}(-\theta_1)\boldsymbol{R_y}(-(\pi/2-\phi_1))S+\boldsymbol{g}_{D_m} \tag{3}$$

Each point $\boldsymbol{s}_{D_{m_f}} \in S_{D_m}$ is the origin of an oriented ray with direction $\overrightarrow{q}$. The distance $d_{D_{m_f}}$ between $\boldsymbol{s}_{D_{m_f}}$ and the 2.5D triangulated scene is computed. The minimum distance per $\boldsymbol{s}_{D_{m_f}}$ is $d_{0_m} = \min(d_{D_{m_f}})$. The point giving d_{0_m} is denoted as s_{0_m}. The intersection of the ray, with origin $\boldsymbol{s}_{D_{m_K}}$, with the scene at point $\boldsymbol{J}_{D_m}$ (Fig. 6(b)) is given by the following equation and is called $\boldsymbol{S}_{D_m}$'s central intersection.

$$\boldsymbol{J}_{D_m} = \overrightarrow{q} \cdot d_{D_{m_K}} + \boldsymbol{s}_{D_{m_K}} \tag{4}$$

The initial distance map $\boldsymbol{\Phi}_m$ per $\boldsymbol{S}_{D_m}$ is defined by the points $\boldsymbol{\Phi}_{m_f} = [x_f, y_f, d_{D_{m_f}} - d_{0_m}]^T$.

In Fig. 6 for the $\boldsymbol{S}_{D_m}$ whose rays intersect O_a object, it can be noticed that $\boldsymbol{J}_{D_m}$ and the point P_{min} that had the minimum distance d_{0_m} from $\boldsymbol{S}_{D_m}$ are on the same object (first category initial distance map). However, when an object O_b (Fig. 6) is further from the reconstruction system than other scene's objects perhaps it is occluded by some of them. In this case, it is probable that even the vast majority of the rays of a circular sector S_{D_m} intersect O_b there will be some rays that intersect other objects that are nearer to the reconstruction system. Thus, the minimum distance d_{0_m} of S_{D_m} perhaps corresponds to a ray that intersects another object that is closer to the reconstruction system than O_b while central intersection $\boldsymbol{J}_{D_m}$ is on O_b's surface (second category initial distance map). The threshold that distinguishes between the two categories is ε. With $\boldsymbol{\Phi}_m(3)$ is denoted $\boldsymbol{\Phi}_m$'s third row while *median* denotes the median value (median is a robust estimate of the center of data since outliers have little effect on it).

If $\varepsilon > median(\boldsymbol{\Phi}_m(3))$, the $\boldsymbol{\Phi}_m$ falls into the first category. In this case $\boldsymbol{\Phi}_{m_2} \subseteq \boldsymbol{\Phi}_m$, where $\boldsymbol{\Phi}_m(3) < \varepsilon$ (this relation removes the initial distance map points which are probably computed from rays that intersect objects that are further from the reconstruction system than O_a).

If $\varepsilon \leq median(\boldsymbol{\Phi}_m(3))$, the $\boldsymbol{\Phi}_m$ falls into the second category and the following statistical analysis sequential steps have to be

executed in order to isolate the initial distance map points that come from rays that intersect O_b:

- The relation: $size(\mathbf{\Phi}_m(3) > [median(\mathbf{\Phi}_m(3)) - std(\mathbf{\Phi}_m(3))])/size(\Phi_m) > 0.75$ (where *size* gives the total number of points that satisfy the expression inside the parentheses) confirms that the majority of $\mathbf{\Phi}_m$'s points (about 75%) is derived from intersection with O_b's surface.
- Initial distance map points that represent O_b's surface are $\mathbf{\Phi}_{m_2} \subseteq \mathbf{\Phi}_m$, where $median(\mathbf{\Phi}_m(3)) - std(\mathbf{\Phi}_m(3)) \leq \mathbf{\Phi}_m(3) \leq median(\mathbf{\Phi}_m(3)) + std(\mathbf{\Phi}_m(3))$.

A viewpoint that lies between O_b and the objects that are nearer to the reconstruction system is $g_{new} = \boldsymbol{J}_{D_m} - \vec{q} \cdot std(\mathbf{\Phi}_m(3))$ (Fig. 6(c)). If a circular sector was adapted to g_{new}, its central intersection and the point that has the minimum distance d_{0_m} from it would be on O_b.

So far the methodology for the extraction of scene's initial distance maps is encapsulated in the following steps:

- The G_D plane forms the coordinate basis for the extraction of scene's initial distance maps.
- A S_{D_m} circular sector is adapted around each point of G_D.
- According to ε, per S_{D_m}, an initial distance map $\mathbf{\Phi}_{m_2}$ is extracted. However, the notation $\mathbf{\Phi}_m$ is used instead of $\mathbf{\Phi}_{m_2}$ through this paper.

2.3. Initial distance map correction

It is obvious that the extraction of the initial distance maps, described in Sections 2.1.2 and 2.2, respectively, for the model and the scene, depends on a coordinate system that is independent of the surface of the objects. Thus, the same surface observed from different viewpoints, would give different initial distance maps and therefore, they are viewpoint dependent. This algorithm proposes a simple, yet effective solution to extract, for each initial distance map, a final distance map dependent on the topology of the surface patch from where initial map was extracted. More specifically, the initial distance map is used to compute a novel viewpoint, which is aimed to be almost parallel to the normal of the surface patch described by this initial distance map, since a normal view of a surface region provides more explicit information about its topology rather than a slantwise view. Then, a circular sector S (Section 2.1.2.1) is adapted around the novel viewpoint and the final distance map is computed. In this way, the final distance map is interrelated to the normal of the surface patch it describes and as a sequence is viewpoint independent. This process can be explained using an example for a model initial distance map. Let us assume that the initial distance map $\mathbf{\Phi}_\lambda$ of the circular sector S_λ, which is adapted around G_{C_λ} sub-grid point, was computed (Fig. 7(a)). S_λ's central intersection is J_λ and its orientation is $\vec{u}_\lambda$. Using PCA a plane (green plane in Fig. 7(b)) fits to the point cloud of $\mathbf{\Phi}_\lambda$ (red point cloud in Fig. 7(b)). The coefficients for the first two principal components define vectors that form a basis for the plane. Moreover, the third principal component is orthogonal to the first two, and its coefficients define plane's normal vector $\vec{u}_{\lambda_{PCA}}$. Its spherical coordinates are $\theta_{PCA}$and ϕ_{PCA}. The orientation of the novel viewpoint is $\vec{u}_\mu = \boldsymbol{R}_z(-\theta_\lambda)\boldsymbol{R}_y(-(\pi/2-\phi_\lambda))\boldsymbol{R}_z(\theta_{PCA})\boldsymbol{R}_y(\pi/2-\phi_{PCA})[0\ 0\ 1]^T$ and its center is $\boldsymbol{C}_\mu = J_\lambda - \vec{u}_\mu \cdot \xi$. The new circular sector's S_μ computation steps are:

Step 1: $S_\mu = \boldsymbol{R}_z(-\theta_\lambda)\boldsymbol{R}_y(-(\pi/2-\phi_\lambda))\boldsymbol{R}_z(\theta_{PCA})\boldsymbol{R}_y(\pi/2-\phi_{PCA})S$ rotates the S sector (Section 2.1.2.1), so that its surface normal is $\vec{u}_\mu$.

Step 2: $S_\mu = S_\mu + \boldsymbol{C}_\mu$ translates the new sector around a new viewpoint.

Rays starting from points of S_μ with orientation $\vec{u}_\mu$ create the final distance map (blue point cloud in Fig. 7(b)) in a straightforward manner as in Section 2.1. It is clear that the final distance map is aligned to the surface it represents. The above procedure is utilized for the initial distance map if $a\cos(\vec{u}_{\lambda_{PCA}} \cdot [0\ 0\ 1]^T) \geq 15^\circ$. Otherwise, it is assumed that $\vec{u}_\lambda$ is almost parallel to the normal of the surface patch described by $\mathbf{\Phi}_\lambda$; therefore, the initial distance map coincides with the final distance map and no further computation is required. Only points with distance below $2R$ are stored in the final distance map. The maximum z-coordinate value (z_{max}) of each final distance map is used as index for a 1D hash table. z_{max} is quantized into bins of Δz_{max}.

Concluding, through the correction of initial distance maps, very similar final distance maps are computed. This fact is useful for compressing the number of the similar final distance maps.

3. Object recognition

In this section the procedure for finding correspondences between object's and scene's final distance maps is described.

3.1. Greyscale images and sift descriptors

The square that encloses the circular disc (Fig. 2(b)) is divided into equally spaced bins (the total number of bins is $n_b \times n_b$). For a final distance map, the value of each bin is defined as the mean value of the z-coordinates of its points, whose x,y-coordinates fall within this bin (Fig. 8a). Empty bins are those that there is no any point inside them. The resulting $n_b x n_b$ matrix is represented as a

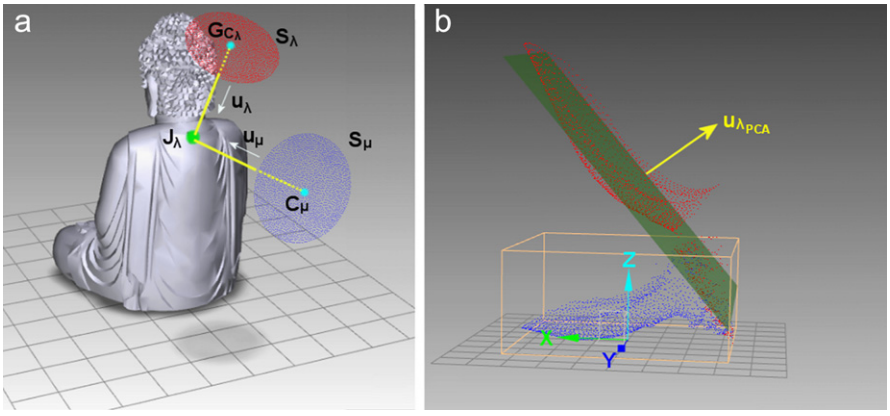

Fig. 7. (a) S_λ and S_μ circular sectors, (b) $\mathbf{\Phi}_\lambda$ and $\mathbf{\Phi}_\mu$ distance maps.

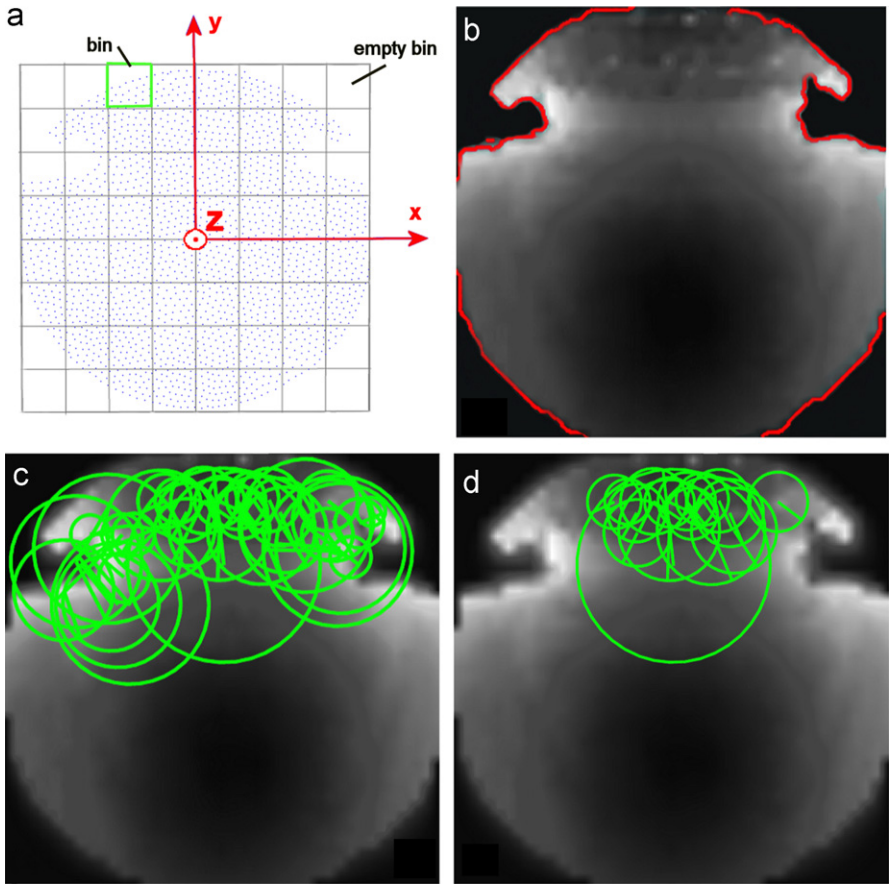

Fig. 8. (a) Final distance map points, (b) Canny edge boundary, (c) frames of SIFT descriptors and (d) frames of stored SIFT descriptors.

greyscale image with resolution of $n_b \times n_b$ pixels. Since, it was noticed that the number of SIFT frames was dependent on the image resolution, each image is resized to $5n_b \times 5n_b$ resolution by applying bilinear interpolation (Fig. 8b). Image areas that have zero intensity value (correspond to empty bins and do not actually represent real surface) are separated from non-zero areas using the Canny algorithm [24]. More specifically, Canny detects image edges and only edges (boundary edges) that separate regions with zero intensity from regions with non-zero intensity (Fig. 8b). Then, contrast stretching is used to emphasize intensity variations of the image, thus intensity values that are in the range [0, 199] are remapped to fill the entire intensity range [0, 255]. The SIFT algorithm [23] returns a $4 \times K$ matrix containing the total K frames (or keypoints) of the image and a $128 \times K$ matrix containing their descriptors. Each frame is defined by its center κ, its scale σ and its orientation ω and is denoted by a circle on the image with radius 6σ (Fig. 8c). SIFT frames, whose circles do not intersect with the boundary edges, are stored for each greyscale image (Fig. 8d). The rest frames take into account zero-intensity areas and thus they are excluded.

3.2. Greyscale image matching

Supposed that two final distance maps represent overlapping areas of a surface their corresponding greyscale images will have common patches according to the degree of overlapping. In this paragraph, a methodology is proposed that attempts to detect image pairs that have similar patches even if the first image is rotated or cropped when compared to the second one. Let us assume that a first image's I_1 random frame is F_α (Fig. 9(b.1)), which is defined by center $\kappa_\alpha = [x_{\kappa_\alpha}\ y_{\kappa_\alpha}]^T$, scale σ_α and orientation ω_α and a second image's I_2 random frame is F_β (Fig. 9(b.2)), which is defined by the center $\kappa_\beta = [x_{\kappa_\beta}\ y_{\kappa_\beta}]^T$, scale σ_β and orientation ω_β. In order to deduce that two SIFT frames are matched, the following relations must be verified:

- Distance ratio ≤ 0.8 (as defined in [23]).
- $|\sigma_\alpha - \sigma_\beta|/\sigma_\alpha \leq 0.2$ (since images have the same scale).

The matched frames for a pair of greyscale images are depicted in Fig. 9(a). Taking into account the centers and the orientation of the matched frames, the second greyscale image is rotated around its center in order to be aligned to the first one and then image patches are generated from the two images. The latter is accomplished through the following steps:

- The angle divergence is $\Delta\phi = \omega_\alpha - \omega_\beta$.
- $I_{2_{F_\beta}} = rotate(I_2, \Delta\phi)$ (Fig. 9(b.4)) (counter-clock rotation around center of I_2 by $\Delta\phi$).
- The center of F_β in $I_{2_{F_\beta}}$ is $\kappa_{\beta_{new}}$ and is given by
  - $x_{\kappa_\beta} = 5n_b/2 + \cos(-\Delta\phi)\cdot(y_{\kappa_\beta} - 5n_b/2) - \sin(-\Delta\phi)\cdot(x_{\kappa_\beta} - 5n_b/2)$.
  - $y_{\kappa_\beta} = 5n_b/2 + \sin(-\Delta\phi)\cdot(y_{\kappa_\beta} - 5n_b/2) + \cos(-\Delta\phi)\cdot(x_{\kappa_\beta} - 5n_b/2)$.
- Windows (of size $\eta \times \eta$) Π_α, $\Pi_{\beta_{new}}$ centered at κ_α and $\kappa_{\beta_{new}}$ (Fig. 9(b.3) and (b.4)) are defined for F_α, F_β, respectively (Fig. 9(c.1) and (c.2)).
- The degree of correspondence between Π_α and $\Pi_{\beta_{new}}$ is measured by a modified normalized cross-correlation [25]

$$mncc(\Pi_\alpha, \Pi_{\beta_{new}}) = \frac{2cov(\Pi_\alpha, \Pi_{\beta_{new}})}{var(\Pi_\alpha) + var(\Pi_{\beta_{new}})} \quad (5)$$

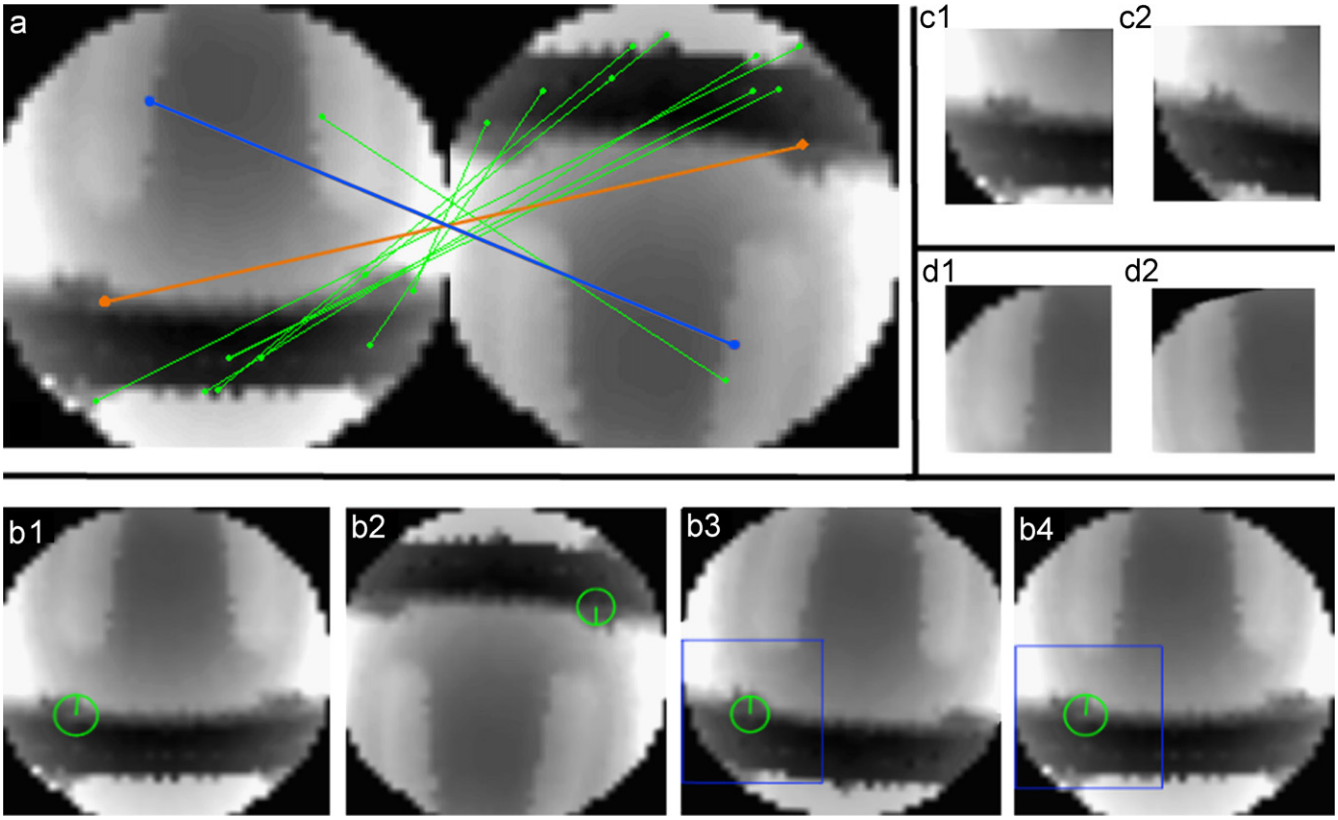

Fig. 9. (a) Two greyscale images and lines connecting the matched frames, (b) a matched pair of frames and the generation of image patches, (c) generated image patches for frames connected by orange line and (d) Generated image patches for frames connected by blue line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

By estimating the $mncc(\Pi_{\alpha},\Pi_{\beta_{new}})$ is verified whether frames actually match, since patches occupy a larger portion of the image than the area within circle that denotes each frame, as it is evident in Fig. 9(b.3) and (b.4). The absolute value of modified normalized cross-correlation lies between -1 and 1, and a value of 1 indicates perfect matching of the windows. The degree of correspondence is estimated for the patches of all matched frames (patches extracted for another pair of frames (Fig. 9(a)) are shown in Fig. 9(d.1) and (d.2)). The maximum degree of correspondence for a pair of images is $\Lambda_{1-2}=\max(mncc(\Pi_{1_{\upsilon}},\Pi_{2_{\upsilon}}))$, where $\upsilon=1,2,\ldots,\zeta$ (ζ is the total number of matched frames and $\Pi_{1_{\upsilon}},\Pi_{2_{\upsilon}}$ are the patches that correspond to each pair of matched frames on I_1, I_2 images).

So far, the process of matching a pair of greyscale images was described. In the following paragraph the methodology followed to match all model's greyscale images with all scene's greyscale images is presented.

3.3. Object recognition in a scene

For each library model its final distance maps, their corresponding greyscale images and their SIFT frames are extracted and stored off-line. During the on-line recognition procedure SIFT frames of scene's greyscale images are extracted. Then, scene's greyscale images are matched to model's greyscale images. In order to achieve time efficiency, a scene's I_s greyscale image can be matched to model's I_{m_k} (where $k=1,\ldots,F$) greyscale images whose final distance maps (F is their total number) have the same z_{max} index with I_s in the 1D hash map. Two matching criteria are established.

Regarding the first criterion, the matching results for I_s are ranked in descending order in terms of $\Lambda_{I_s-I_{m_k}}$. A specific I_{m_k}, when compared to different I_s with the same z_{max} index, is allowed to satisfy the $\Lambda_{I_s-I_{m_k}}>\tau_1$ condition for up to χ times. Then, for the top ranked pair of images per I_s, it is verified whether $\Lambda_{I_s-I_{m_k}}$ exceeds a predetermined threshold τ_2.

Apart from the above criterion that is based purely on image comparison, a more robust matching criterion is proposed below. This criterion is further applied for each I_s to its top ranked (I_s, I_{m_k}) image pair if and only if $\Lambda_{I_s-I_{m_k}}>\tau_3$. In brief, final distance maps with lower z-coordinate upper limit are created, then it is checked if the normal vector plane fitted to the points of final distance maps has limited angle divergence from the z-axis. Finally, the degree of correspondence of the images extracted from the final distance maps is computed.

Let us suppose that Φ_s and Φ_{m_k} are the final distance maps, from which I_s and I_{m_k} were generated. The following steps show how Φ_s and Φ_{m_k} are processed:

Step 1: $\Phi_{s_1}\subseteq\Phi_s$, where $\mathbf{\Phi}_s(3)<mean(\mathbf{\Phi}_s(3))+std(\mathbf{\Phi}_s(3))$,
$\Phi_{m_{k_1}}\subseteq\Phi_{m_k}$, where $\mathbf{\Phi}_{m_k}(3)<mean(\mathbf{\Phi}_{m_k}(3))+std(\mathbf{\Phi}_{m_k}(3))$.

Step 2: Using PCA [8] the third principal component $\vec{u}_{s_{PCA}}$ and $\vec{u}_{m_{k_{PCA}}}$ is computed for Φ_{s_1} and $\Phi_{m_{k_1}}$, respectively. Then is determined whether $a\cos(\vec{u}_{s_{PCA}}\cdot[0\ 0\ 1]^T)\leq 30^{\circ}$ and $a\cos(\vec{u}_{m_{k_{PCA}}}\cdot[0\ 0\ 1]^T)\leq 30^{\circ}$. These conditions verify that $\vec{u}_{s_{PCA}}$ and $\vec{u}_{m_{k_{PCA}}}$ have not great angle divergence from the z-axis.

Step 3: Images $I_{s_1}, I_{m_{k_1}}$ are generated for Φ_{s_1} and $\Phi_{m_{k_1}}$ using the same process described in Section 3.1 stopping right after image interpolation. The produced $I_{s_1}, I_{m_{k_1}}$ images depict more details about the represented surface than the images I_s, I_{m_k}.

Step 4: Using SIFT-frames previously computed for I_s and I_{m_k}, according to Section 3.2, $\Lambda_{I_{s_1}-I_{m_{k_1}}}$ is estimated for $I_{s_1}, I_{m_{k_1}}$. $\Lambda_{I_{s_1}-I_{m_{k_1}}}$ has to be over τ_4.

The above thresholds are defined explicitly in Section 4.1.

Let us assume that for a pair I_s,I_{m_k} one of the above criteria is satisfied. During the computation of the final distance maps, from which I_s,I_{m_k} were generated, their central intersections $\boldsymbol{J}_s$,$\boldsymbol{J}_{m_k}$ were also computed.

Therefore, the point $\boldsymbol{J}_{m_k}$ (which lies on M surface), corresponds to the point $\boldsymbol{J}_s$ (which lies on scene's surface). Thus, a point correspondence between scene and model is established. This procedure is repeated for all scene's images and the $\boldsymbol{J}_s$ correspondence points are localized in the scene separately for the points that satisfy each criterion. For some models whose surface has common geometric characteristics (i.e. nearly planar surface patches) probably some false correspondences might be found in the scene. By extending the ISODATA algorithm [26] for 3D data, the $\boldsymbol{J}_s$ correspondence points found for each criterion are separately classified into clusters based on their inter-distances. The largest cluster of each criterion is computed. Though the largest clusters for both criteria usually coincide, the largest cluster of the second criterion, at most times (85%) (when the occlusion of the model is not severe), contains the $\boldsymbol{J}_s$ points that are true correspondences of the model in the scene. In this case the object is positively identified in the scene. The largest cluster of the first criterion is used when there is no largest cluster (larger clusters have the same number of points) or there is no any cluster for the second criterion.

Simultaneous recognition of many object in a scene is not time consuming since it can be executed in parallel for these objects exploiting multi-core computer architecture.

4. Experimental results

This section includes experiments on both synthetic and real data. Prior to initiation of the experiments the parameters used in this method are clarified. Experiments on synthetic data were performed on a model library containing 20 models of varying surface structure (Fig. 10). Real data were used to compare our method against spin images.

4.1. Specification of parameters used

In Section 2.1.2, variable R that defines S's radius was mentioned. This variable is crucial for the method since some variables are defined according to it. It is desired that the rays starting from $S_{i\theta\phi}$ and S_{D_m} intersect the surface in a large enough area, so that the distance map will contain sufficient information for the topology of the surface in order to discriminate different surfaces. At the same time, if R's were set to be very large scene's distance maps would have more chances to store distances for rays that intersect surfaces from different objects due to occlusion

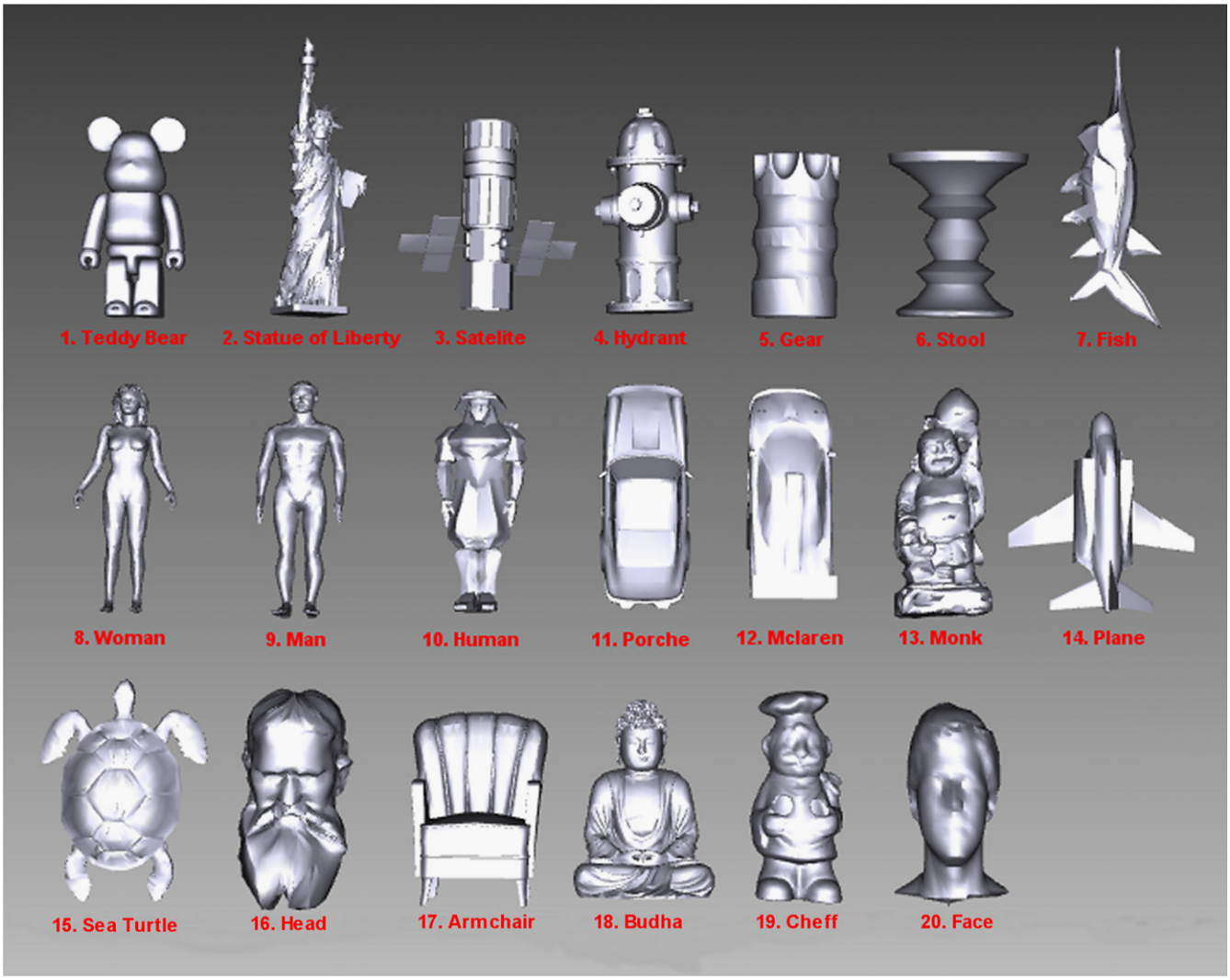

Fig. 10. Library models from 3DVIA [28] and Princeton [29] 3D object databases.

and clutter. For library objects that have comparable sizes R was experimentally set to be $mean(H_\gamma)/7 \leq R \leq mean(H_\gamma)/9$ (γ is the index for each library object). Whereas the total number of circular disc S points is $N=2000$, so that points of the extracted distance map are dense enough to describe efficiently the represented surface and to form the greyscale images. The N_G was set to 642 after experiments on the real data (Section 4.3) and was also used for the experiments on the synthetic data.

The sum of distance maps per model was controlled by assuming a maximum value for j in z-axis parameter h_j. The maximum j was set to 7 thus α parameter in h_j was $\alpha = H/7$ and as a consequence the total number of distance maps was limited to 4600 for objects with higher H. In Section 2.2, the density of G_D points is such that the distance between a G_D point and its direct neighbors is between $R/2$ and $2R/3$. ε used in this section is set to $2R$. In Section 2.3, for model and first category scene initial distance maps $\xi = 3 \cdot R$, while for the second category initial distance maps is $\xi = std(\mathbf{\Phi}_m(3))$ (so that $\boldsymbol{C}_\mu$ is computed in the same sense as g_{min} in Section 2.2). The z_{max} ranges between 0 and $2R$ while $\Delta z_{max} = 2R/8$. The square that enclosed the S circular disc in Section 3.1 was separated into 40×40 bins, so that their size allowed a sufficient number of distance map points to fall within each bin. Otherwise, if the initial number of bins was greater, then the number of bins would over exceed the total number of distance map points ($N=2000$), thus most of the bins would be empty.

The thresholds used for the matching criteria were defined after exhausting tests. A specific I_{m_k} when compared to different I_s shall not give large degrees of correspondence (over τ_1) for many of them (the total number where the degree of correspondence is over τ_1, is χ). Otherwise, it has to be discarded from the matching procedure since it has low discriminative power and may lead to erroneous matches. During the experimental trials it was observed that for $\tau_1 = 0.8$ and $\chi = 1$, I_{m_k} images that led to erroneous matches were successfully discarded. The purpose of the τ_2 is to separate image pairs that give true correspondences of the model in the scene from false ones. It was noticed that for τ_2 below 0.86 the number of false correspondences in the scene was increasing in fast rate as τ_2 reduced, while for τ_2 over 0.89 many true correspondences were discarded. Therefore, the ideal value for τ_2 was between 0.86 and 0.89. Experimentally, τ_2 was set to 0.88. For time efficiency, $\tau_3 = 0.7$ was used to reduce the number of the top ranked image pairs to be processed through the second criterion, since it was experimentally confirmed that pairs with lower τ_3 are rather impossible to satisfy this criterion. In the same way as τ_2, it was observed that τ_4 ranged from 0.66 to 0.72. Experimentally, τ_4 was set to 0.68.

For the ISODATA algorithm, the threshold for the minimum number of samples each cluster could have (used for discarding clusters) was set to 3, the threshold for the standard deviation (used for split operation) was set to $2R$ and the threshold for the pairwise distance (used for merge operation) was set to $R/2$ [26].

4.2. Synthetic scene experiments

To evaluate the performance of the recognition system a model library, which consisted of 20 synthetic models, was used (Fig. 10). These objects were chosen arbitrary, so that the geometric characteristics of their surface varied from object to object. In the experiments, the objects were placed randomly in scenes using a simulation program and the number of models per scene varied from 3 to 9. The total number that a library model was placed in the scenes was approximately the same for all models. In order to estimate the recognition rate at different clutter and occlusion rates additional objects were used. During the experimental procedure the performance of this approach was checked for cluttered and occluded scenes. Recognition success was verified by computing the rates of true positive (TP), true negative (TN) and false positive (FP) [7]. The searched model existed in the scene, thus true negative rate was not computed. The occlusion in the scene was defined as

$$occlusion = 1 - \frac{model\ surface\ patch\ data}{total\ model\ surface\ data} \tag{6}$$

The clutter was defined as

$$clutter = 1 - \frac{model\ surface\ patch\ data}{total\ scene\ surface\ data} \tag{7}$$

The library objects were manually segmented from the scene in order to compute their clutter and occlusion values. Totally 160 recognition experiments were performed on 35 synthetic scenes. The mean number of scene's descriptors was 650 and the mean time to recognize all models in a scene was about 80 min. From Fig. 11(a) it is concluded that the average recognition rate was 79% with 80% occlusion. When the occlusion percentage exceeds 85%, the rate decreases significantly. The recognition rate with respect to clutter was 83.3% at 90% clutter (Fig. 11(b)). It was shown experimentally that the rate is mainly affected by occlusion since the recognition rate is not reduced significantly as clutter increases. The average recognition rate was 78.13%, since 125 out of 160 recognition trials were successful. Here, it should be denoted that 118 recognition trials had over 75% occlusion. Fig. 12 depicts the experimental results for nine scenes. The correspondences of each

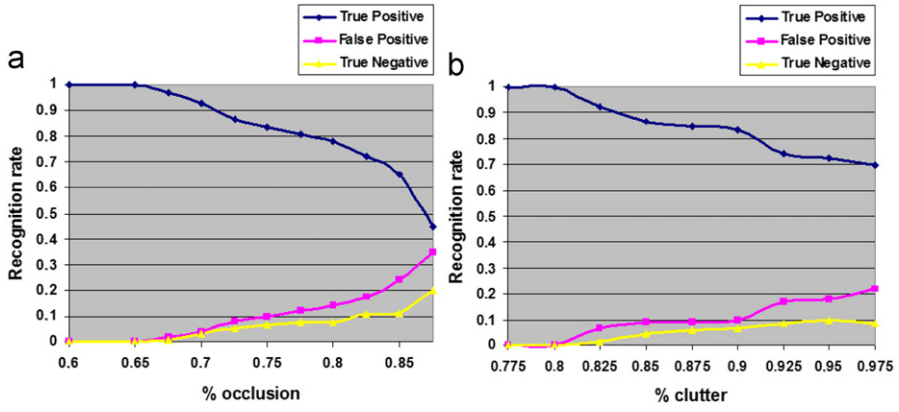

Fig. 11. Recognition rate against (a) occlusion and (b) clutter.

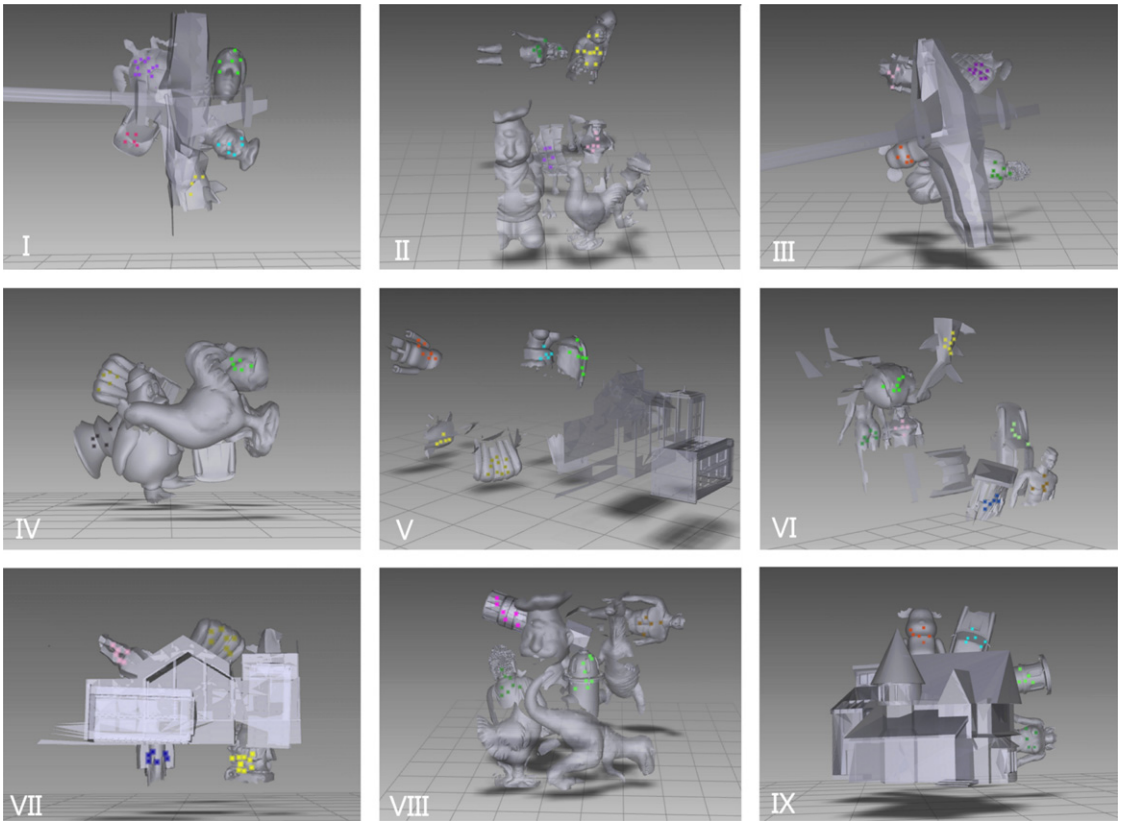

Fig. 12. Experimental results for synthetic scenes.

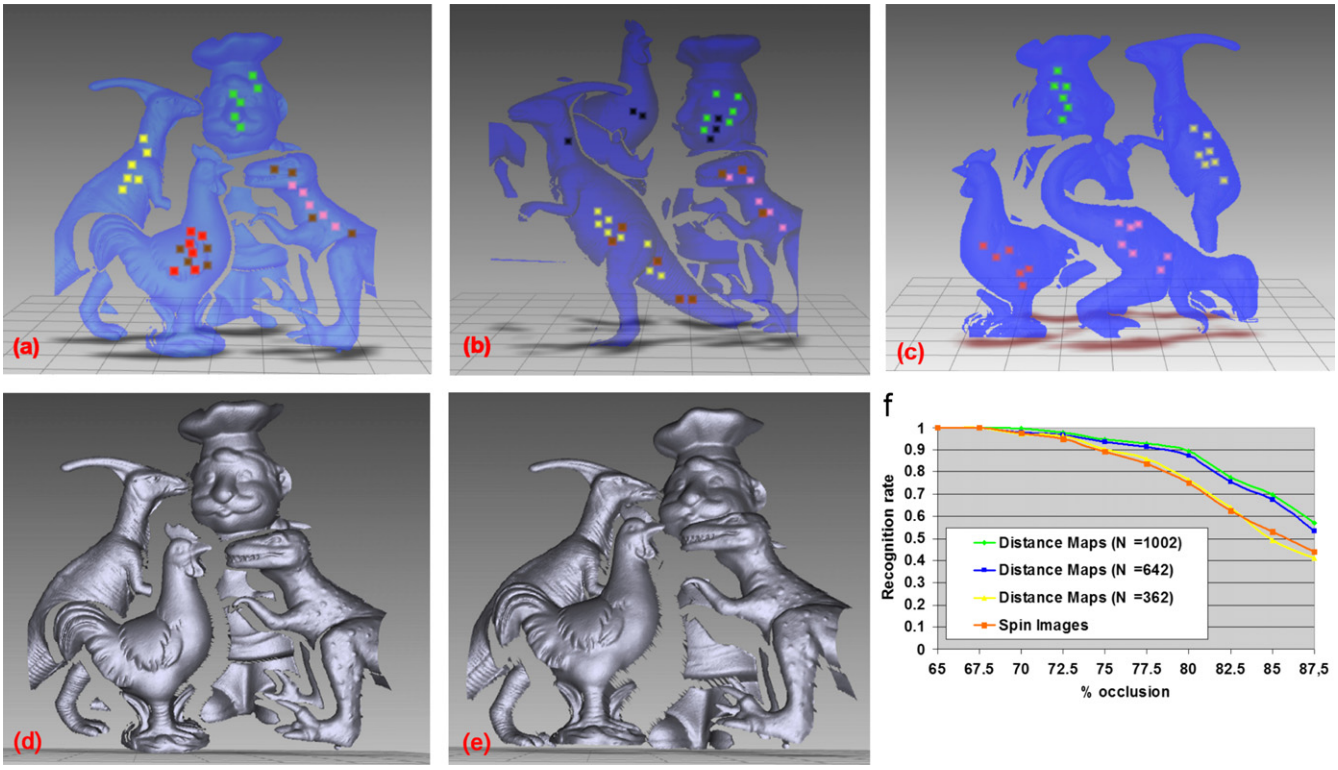

Fig. 13. (a–c) Experimental results, (d,e) visibility results and (f) recognition rate against occlusion.

model in the scene are displayed in different colors. It can be noticed that the vast majority of the models were successfully recognized in the exhibited scenes. Examples of unrecognized objects are exhibited in Fig. 12 III, IV, VI where "head", "Porche" and "Mclaren" objects were not recognized due to significant occlusion.

4.3. Experimental comparison with spin images

Before defining the N_G parameter the performance of the algorithm was tested for $N_G=1002$ (icosahedron subdivision level 10), $N_G=642$ (subdivision level 8) and $N_G=362$ (subdivision level 6) points. The results are depicted in Fig. 13(f). From this picture it is obvious that for $N_G=342$ the recognition rate was significantly decreased compared to the rate for $N_G=642$. Though the rate for $N_G=1002$ is slightly superior to the rate for $N_G=642$, the execution time was increased to about 25%. Consequently, spherical grid $\boldsymbol{G}$ is selected to have $N_G=642$ in order to provide an adequate number of viewpoints, while sub-grids $\boldsymbol{G}_a$, $\boldsymbol{G}_b$ and $\boldsymbol{G}_c$ have 605, 605 and 568 points, respectively. The results are discussed taking into consideration the distance maps recognition rate for $N_G=642$.

The proposed algorithm was compared towards uncompressed spin images using the testing data which is available in [27]. The available 50 real scenes were composed of five models. However, recognition of the "rhino" model was excluded from the final results since spin image had very low recognition rate (about 21%). The total recognition rate of the compared methods towards occlusion is indicated in Fig. 13(f). It is clear that the proposed method is superior to the spin image method. Normally, information about the viewpoint of observation is automatically provided since the coordinate system of a real scene is based on the pose of the scanner. Observing the real scenes from range scanner's viewpoint ensures maximum visibility, since reconstructed surfaces are fully visible (they do not occlude each other). The 2.5D real reconstructed scenes used for the real experiments in this paper were derived from the Internet [27] and information of scanner's viewpoint was not included in the provided experimental data. Therefore orientation and position were manually defined bearing in mind to obtain the best possible visibility of the scenes (in other words to ensure minimum occlusion), and since it was not the optimum the performance of our algorithm was probably negatively influenced. Fig. 13(d) and (e) shows the visibility of a scene from two manually selected viewpoints. The first one that gave the visibility depicted in Fig. 13(d) is closest to the range scanner's real viewpoint, than the second one that gave the visibility depicted in Fig. 13(e). This is evident due to the fact that in Fig. 13(e) foreground objects occlude more the background objects than in Fig. 13(d).

Another significant advantage of this approach is the number of the descriptors per model or scene. The mean number of scene's descriptors in our method was 340, while in spin image method was about 8500. This fact explains the observed time divergence, since our method requires about 65 min per scene while spin images about 320 min. Totally 168 recognition experiments were performed on 50 real scenes. From Fig. 13(f) it is concluded that the average recognition rate was 75.3% with up to 82.5% occlusion. When the occlusion percentage exceeds 85%, the rate decreases significantly. The average recognition rate was 86.9%, since 146 out of 168 recognition trials were successful, while for the spin image method this percentage was 78.6%. Fig. 13(a–c) depicts the results for experiments 5, 29, 46, respectively. The correspondences of each model in the scene are displayed with dots of different colors, i.e. red for the "Chicken", green for the "Chef", yellow for the "Parasaurolophus" and pink for the "T-rex" model. Models were successfully recognized in the scene, with the exception of chicken in Fig. 13(b) which was severely occluded. On the other hand, the spin image algorithm did not manage to recognize the "Chicken" and "T-rex" models in Fig. 13(b) and the "T-rex" model in Fig. 13(a), since most of their correspondences in the scene were on a false object. The correspondences of the unrecognized objects, for the spin image algorithm, are displayed with black dots for the "Chicken" and with brown dots for the "T-rex" model, respectively.

5. Conclusions

In this paper a novel algorithm for viewpoint independent recognition of 3D free-form objects was presented. The methodology for extracting scene and model distance maps allowed to keep their total number low. This fact, in conjunction with the simple 1D hash table, allowed for significant acceleration of the execution time. Additionally, the performance and the number of descriptors of this method is independent to the resolution of the models and the scenes.

Experiments conducted on synthetic scenes that contained objects with varying surface structure from a model library proved the robustness of this method to a satisfactory degree on clutter and occlusion. The efficiency of this algorithm was further verified by testing real scenes where noise was present. These tests indicated that this method is advantageous to the spin image algorithm in terms of occlusion and computational time. The average recognition rate for 318 experimental trials was 80.5%.

Acknowledgement

This work was supported by the 3DLife EU Network of Excellence (NoE) project.

References

[1] R. Campbell, P. Flynn, A survey of free-form object representation and recognition techniques, Computer Vision and Image Understanding 81 (2) (2001) 166–210.
[2] C. Dorai, A.K. Jain, A representation sheme for 3D free-form objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1997) 1115–1130.
[3] C.S. Chua, R. Jarvis, Point signatures: a new representation for 3D object recognition, International Journal of Computer Vision 25 (1) (1997) 63–85.
[4] D. Huber, A. Kapuria, R. Donamukkala, M. Hebert, Parts-based 3D object classification, Proceedings of IEEE International Conference on Computer Vision, vol. 2, 2004, pp. 82–89.
[5] S. Ruiz-Correa, L.G. Shapiro, M. Meila, A new paradigm for recognizing 3-D objects from range data, Proceedings of IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1126–1133.
[6] S. Ruiz-Correa, L.G. Shapiro, M. Meila, A new signature-based method for efficient 3-D object recognition, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 769–776.
[7] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 433–449.
[8] D.V. Vranic, D. Saupe, Tools for 3D-object retrieval: Karhunen–Loeve Transform and spherical harmonics, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, France, 2001, pp. 293–298.
[9] O. Carmichael, D. Huber, M. Hebert, Large data sets and confusing scenes in 3-D surface matching and recognition, in: Proceedings of International Conference on 3-D Digital Imaging and Modeling, 1999, pp. 358–367.
[10] A. Frome, D. Huber, R. Kolluri, T. Bulow, J. Malik, Recognizing objects in range data using regional point descriptors, Proceedings of European Conference on Computer Vision, vol. 3, 2004, pp. 224–237.
[11] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2003, pp. 156–164.
[12] P. Indyk, R. Motwani, Approximate nearest neighbor-towards removing the curse of dimensionality, in: Proceedings of Symposium on Theory of Computing, 1998, pp. 604–613.
[13] G. Hetzel, B. Leibe, P. Levi, B. Schiele, 3D object recognition from range images using local feature histograms, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 394–399.
[14] A.S. Mian, M. Bennamoun, R. Owens, Three-dimensional model-based object recognition and segmentation in cluttered scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1584–1601.
[15] H. Chen, B. Bhanu, 3D free-form object recognition in range images using local surface patches, in: Proceedings of International Conference on Pattern Recognition, 2004, pp. 136–139.
[16] H. Chen, B. Bhanu, 3D free-form object recognition in range images using local surface patches, Pattern Recognition Letters 28 (10) (2007) 1252–1262.

[17] X. Li, I. Guskov, 3D object recognition from range images using pyramid matching, in: Proceedings of International Conference on Computer Vision, 2007, pp. 1–6.
[18] G. Kordelas, P. Daras, Recognizing 3D objects using ray-triangle intersection distances, Proceedings of IEEE International Conference on Image Processing, vol. 6, 2007, pp. 173–176.
[19] C. Chang, C. Lin, LIBSVM: a library for support vector machines, 2001.
[20] K. Khoshelham, Extending generalized hough transform to detect 3D objects in laser range data, in: Proceedings of International Society for Photogrammetry and Remote Sensing Workshop, 2007, pp. 206–210.
[21] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, Society for Industrial and Applied Mathematics Review 41 (1999) 637–676.
[22] T. Moller, B. Trumbore, Fast, minimum storage ray-triangle intersection, Journal of Graphics Tools 2 (1) (1997) 21–28.
[23] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 2 (2004) 91–110.
[24] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679–714.
[25] H. Moravec, Robot rover visual navigation, Computer Science: Artificial Intelligence (1981) 105–108.
[26] N. Venkateswarlu, P. Raju, Fast ISODATA clustering algorithms, Pattern Recognition 25 (3) (1992) 335–342.
[27] 〈http://www.csse.uwa.edu.au/∼ajmal/recognition.html〉.
[28] 3DVIA database: 〈http://www.3dvia.com/〉.
[29] Princeton database: 〈http://shape.cs.princeton.edu/search.html〉.

Georgios Kordelas was born in Mytilini, Greece, in 1983. He received the Diploma degree in electrical and computer engineering in 2006 from Aristotle University of Thessaloniki, where he is currently pursuing the master's degree in Advanced Computing and Telecommunication Systems. He is a Research Assistant at the Informatics and Telematics Institute, Thessaloniki. His main research interest is computer vision. Mr. Kordelas is a member of the Technical Chamber of Greece.

Petros Daras was born in Athens, Greece, in 1974. He received the Diploma degree in electrical and computer engineering, the M.Sc. degree in medical informatics, and the Ph.D. degree in electrical and computer engineering, all from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1999, 2002, and 2005, respectively. He is a Senior Researcher at the Informatics and Telematics Institute, Thessaloniki. His main research interests include computer vision, search and retrieval of 3-D objects, and medical informatics. He has been involved in more than 15 European and national research projects. Dr. Daras is a member of the Technical Chamber of Greece.