Building 3D Object Representation Using SURF Local Features

Marek Jaszuk
University of Information Technology and Management
ul. Sucharskiego 2
Poland, 35-225, Rzeszów
marek.jaszuk@gmail.com

Grażyna Szostek
University of Information Technology and Management
ul. Sucharskiego 2
Poland, 35-225, Rzeszów
grazyna.szostek@gmail.com

Janusz A. Starzyk
School of Electrical Engineering and Computer Science
Ohio University
USA, OH 45701, Athens
starzykj@ohio.edu

ABSTRACT

The paper discusses an approach to create 3D representation of physical objects. The aim is creating a visual representation of an object, which allows for robust recognition, irrespectively of the distance and the direction of observation. The approach uses a set of rotational views of an object, which are transformed into a set of keypoints using the SURF visual feature detector. The key points are then collected to build a 3D model of the object. Such representation allows both for recognizing the objects based on local characteristics, and distinguishing different global geometry transformations that are needed to recognize the object in its 3D environment.

Keywords
visual reconstruction, visual memory, SURF descriptors

1 INTRODUCTION

Creating virtual models of physical objects using photographs, video records, or 3D scanning became very popular in computer vision. There is a growing number of techniques and devices serving this purpose. Most of the work done in this field focuses on reconstructing global geometry of objects. Such approaches usually lead to obtaining a set of characteristic points located on the surface of a reconstructed object. The points are then transformed into a polygon mesh, to use in a variety of applications, like 3D visualization. Although the precise geometry is highly desired and useful, such representation is not convenient for recognizing an object, because many objects may change their geometry, or part of the geometry might be invisible.

Object recognition typically uses 2D feature detection. Depending on the method used, the visual features can either be edges, corners, regions of interest, interesting points or ridges detected within an image. Detection of features starts from the pixel level, and transforms a local image contents into a set of low-level parametric objects. The collection of visual features related to a particular object should allow for identifying the object within an image in which the object is visible. Local features are robust with respect to occlusions and changes in global geometry of the observed object. There is a number of approaches developed so far, designed for building recognition systems based on local features like [Low01, Rot04, Jun05]. However, the weakness of systems based exclusively on local features is their inability to distinguish changes in global geometry.

Various problems are faced, when one tries to build visual 3D recognition system. Such system should be able to recognize an object irrespectively of its translation, rotation, and scale in addition to changes in lighting conditions, shading, partial occlusions, and local deformations. The scale invariance is important, because it allows for recognizing objects irrespectively of the distance. An important achievement to overcome these difficulties, was introduction of the Speeded Up Robust Features (SURF) algorithm [Bay08]. This method first identifies a set of key points within an image, together with vectors of descriptors for each of the points. In this way, the local image contents around each of the key points is characterized using either 64 or 128 dimensional floating point vectors.

The problem of object recognition is, however, more complex when we consider 3D objects. In this case two possible approaches can be distinguished. Either the objects will still be described by 2D features, but transformed into 3D model, or we can develop 3D feature descriptors. The first approach is relatively simple to apply, because we can use the same well known feature descriptors that are used for recognizing objects in flat images. Moreover this approach requires nothing more than ordinary camera images. In the second approach the 3D object representation is assumed. This is more complicated, because we have to make 3D scans of the objects that we want to work with. There is a number of technologies available on the market designed for 3D scanning. They use laser rays, structured light, multiple camera views, or time-of-flight cameras [Pfe15, Tanb08] to obtain 3D geometry of an object. The literature is dominated with approaches describing such geometry in terms of global shape features, but there are also approaches to treat the problem with the local features, which are more appropriate for recognition purposes. An example of such an approach is the 3D extension of the SURF algorithm [Kno10].

The goal of this paper is to demonstrate an approach based on using SURF local feature descriptors, and sequences rotational views of objects to reconstruct the objects in 3D space. Using rotational views allows for recognizing the object irrespectively of the direction of observation. The SURF descriptors are scale invariant, which allows for recognizing the objects irrespectively of the distance. However, the scale invariance demonstrated by SURF is not sufficient for wide range of distances. Thus we have to extend the object representation to contain interest points identified in sequences of rotational views recorded from different distances. This allows to capture more effectively the features, which are not visible from large distance, or might be omitted, when seen from close distance. The same refers to the height from which the object is observed. To make the object representation complete we add additional sequences of views recorded from different vertical locations with respect to the object.

The described object representation is desired in many applications. An example could be a mobile robot memory, which would allow for recognizing objects and navigating 3D environment irrespectively of the temporary position of the robot. The additional advantage of placing the key points in 3D space is the possibility of identifying changes in geometry of objects. In this way objects can be analyzed in two stages. First is identification of the object as a loose collection of characteristic points found within an image, and then verification of the respective distances between the points. In this way we can find out, if there are any changes in the spatial configuration of the object, or identify missing or invisible parts of the object.

The paper is organized as follows. In Sec. 2 we describe the turntable approach to reconstructing objects from a sequence of images. In Sec. 3 the way we use SURF local features to characterize objects is discussed. Sec. 4 describes how the points from 2D images are converted into 3D coordinates. Next, in Sec. 5 we describe the way of extending the basic approach in order to efficiently capture the features from different distances, and the height of the observation point. Finally in Sec. 6 we describe selected experimental results.

2 THE TURNTABLE APPROACH TO 3D OBJECT RECONSTRUCTION

Creating an object representation in our approach is based on registering sequences of rotational views of an object. We assume, that the object rotates on a turning table around a vertical axis within the field of view of a fixed camera. This solution has already been applied for creating 3D object reconstruction [Fre04, Zha09]. The method works as a camera based 3D scanners, which leads to creating a 3D mesh representing the object's surface. While useful in many applications, a mesh is of little use for recognition purposes. Our work is aimed at creating an object representation, which could be considered a visual object memory. We need to create this representation in a way, which allows for easy recognition of the object, irrespectively of the direction, and the distance of observation. Such a representation is useful in many applications, like a mobile robot memory system, which allows for object recognition, and environment navigation.

The turntable approach presented here originates from the method presented in [Fre04]. However, the mentioned work was focused on reconstructing global geometry of an object, while the goal of our work is building the recognition system. The method deals with two 3D coordinate systems - one associated with the object and the other with the camera (Fig. 1). Moreover, the environment view is registered on a 2D image plane, with its own 2D coordinates. What we know is the position of particular scene elements $\mathbf{V^I} = (\mathbf{u}, \mathbf{v}, \mathbf{1})$ in the 2D image plane (for convenience expressed in the homogeneous coordinates). This position results from location of the scene element with coordinates $\mathbf{V^O} = \left(\mathbf{X^O}, \mathbf{Y^O}, \mathbf{Z^O}, \mathbf{1}\right)$ in the 3D coordinate system associated with the object, as well as from the projection matrix $\mathbf{P}$, which transforms position of each point from the object coordinates $\mathbf{V^O}$ into the image coordinates $\mathbf{V^I}$. This matrix results from the camera position and orientation with respect to the object coordinate system, and the camera focal length. We do not know the $\mathbf{P}$ matrix *a priori*, but it can be found using a calibration procedure, like the one described in [Fre02], or any other suitable calibration method. So in further considerations we assume that the projection matrix is known.

The next point is the formula used to obtain the screen coordinates from the original object coordinates. In other words, this is the transformation from the 3D $\mathbf{V^O}$ coordinates into the 2D image coordinates:

$$\lambda \mathbf{V^I} = \mathbf{P} \cdot \mathbf{V^O}, \qquad (1)$$

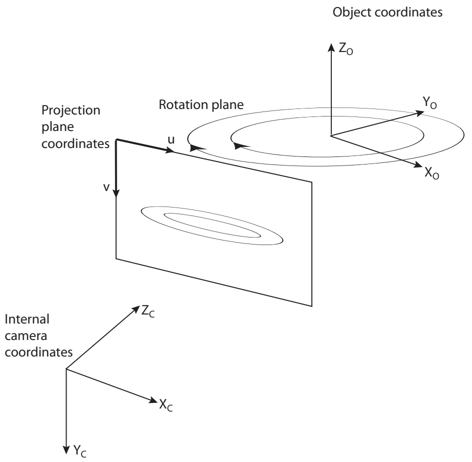

Figure 1: The coordinate systems considered in the turntable approach

where λ is a positive scalar value. Explicitly the above formula can be written as follows:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \end{bmatrix} \cdot \begin{bmatrix} X^O \\ Y^O \\ Z^O \\ 1 \end{bmatrix}. \quad (2)$$

From the above we determine the λ constant:

$$\lambda = P_{3,1}X^O + P_{3,2}Y^O + P_{3,3}Z^O + P_{3,4}, \quad (3)$$

and the u, v image coordinates:

$$u = \frac{P_{1,1}X^O + P_{1,2}Y^O + P_{1,3}Z^O + P_{1,4}}{P_{3,1}X^O + P_{3,2}Y^O + P_{3,3}Z^O + P_{3,4}}, \quad (4)$$

$$v = \frac{P_{2,1}X^O + P_{2,2}Y^O + P_{2,3}Z^O + P_{2,4}}{P_{3,1}X^O + P_{3,2}Y^O + P_{3,3}Z^O + P_{3,4}}. \quad (5)$$

Now we have to note, that the Z^O axis is the rotation axis of the object. Thus the vertical position of every element of the object is constant, and will be denoted as h (height from the rotation plane 0 level). The coordinates that vary are X^O and Y^O. The trajectory of every element of the rotating object is a horizontal circle with center in the $0, 0, h$ point. The equation of the circle can be written as follows:

$$O\left(X^O, Y^O, Z^O = h\right) = \left(X^O\right)^2 + \left(Y^O\right)^2 - R^2 = 0, \quad (6)$$

where R is the radius of the circle obtained separately for every interest point. Now we transform eq. (4) in order to extract X^O:

$$X^O = \frac{(P_{1,2} - P_{3,2}u)Y^O + (P_{1,3} - P_{3,3}u)h + (P_{1,4} - P_{3,4}u)}{P_{3,1}u - P_{1,1}}. \quad (7)$$

After rearrangement we get:

$$X^O = \frac{(P_{1,2} - P_{3,2}u)}{P_{3,1}u - P_{1,1}}Y^O + \frac{(P_{1,3} - P_{3,3}u)}{P_{3,1}u - P_{1,1}}h + \frac{(P_{1,4} - P_{3,4}u)}{P_{3,1}u - P_{1,1}}, \quad (8)$$

or in simpler form:

$$X^O = A_1^1(u,v)Y^O + B_1^1(u,v)h + C_1^1(u,v). \quad (9)$$

This after substituting to eq. (5) gives:

$$v = \frac{P_{2,1}(A_1^1 Y^O + B_1^1 h + C_1^1) + P_{2,2}Y^O + P_{2,3}Z^O + P_{2,4}}{P_{3,1}(A_1^1 Y^O + B_1^1 h + C_1^1) + P_{3,2}Y^O + P_{3,3}Z^O + P_{3,4}}. \quad (10)$$

The formula can be rearranged to get Y^O in the following way:

$$Y^O = \frac{(P_{2,1} - P_{3,1}v)B_1^1 + P_{2,3} - P_{3,3}v}{(P_{3,1}v - P_{2,1})A_1^1 + P_{3,2}v - P_{2,2}}h + \frac{(P_{2,1} - P_{3,1}v)C_1^1 + P_{2,4} - P_{3,4}v}{(P_{3,1}v - P_{2,1})A_1^1 + P_{3,2}v - P_{2,2}}, \quad (11)$$

or shorter:

$$Y^O = C_1(u,v)h + D_1(u,v). \quad (12)$$

The above can be substituted to eq. (9):

$$X^O = A_1^1(u,v)C_1(u,v)h + A_1^1(u,v)D_1^1(u,v) + B_1^1(u,v)h + C_1^1(u,v), \quad (13)$$

or shorter:

$$X^O = A_1(u,v)h + B_1(u,v). \quad (14)$$

X^O and Y^O written in this way can be substituted to the equation for the circle in 3D object coordinates (eq. (6)). As a result we get the equation specified by the radius and the height:

$$O(u,v) = (A_1(u,v)h + B_1(u,v))^2 + (C_1(u,v)h + D_1(u,v))^2 - R^2 = 0. \quad (15)$$

After reorganizing and grouping with respect to h we get the circle equation in the following form:

$$O(u,v) = A(u,v)h^2 + B(u,v)h + C(u,v) - R^2 = 0. \quad (16)$$

In the above equation we have two parameters h and R, which are unknown. To find them we have to fit the circle equation to at least 2 image points (u, v) representing the same interest point of the object seen at different angles of rotation. To make the fitting more reliable, it is desirable to collect larger number of the same point views, by following the point in images representing subsequent rotations.

3 CHARACTERIZING OBJECTS WITH SURF DESCRIPTORS

To characterize an object we use the SURF detector [Bay08]. It identifies a set of key points within an image, and then computes vectors of descriptors for each

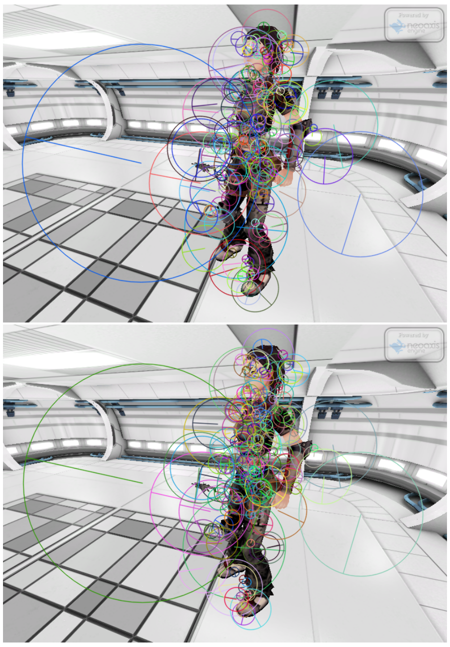

Figure 2: A pair of images with matched SURF key points between different object rotations

of the key points. The vectors are floating point number vectors of size 64 or 128 depending on the algorithm setting. Matching elements in two images is based on matching vectors of descriptors of particular key points To build the object representation we need to extract the key points belonging to the object from all the key points identified in an image. Assuming that the object of interest is the only moving element in the scene, and the camera is static, we can easily distinguish the points belonging to the object from these of the background. It is enough to match the points between two images presenting different rotations of the object, then identify the points that changed their positions, and remove all the static background points. Fig. 2) shows an example of artificially rendered scene using the NeoAxis 3D rendering engine [Neo15]. The Girl object[1] is rotated by 10^o between the left and right image. The operation repeated for full set of rotational views, delivers the collection of points representing the 360^o panorama of the object. This is the basis for building the 3D object representation.

The number of matched points between images depends on the angle of the object rotation. It cannot be too large, because the number of matched points drops rapidly with the increase in the rotation angle value. It cannot be too small either, for efficiency reasons. In our experiments, we assumed, the fixed value of 10^o angle of object rotation between subsequent images. It delivers sufficiently large number of matched points, and generates a complete rotational view of the object in 36 images.

All the subsequent images deliver new points, which are grouped into sequences matched between a number of subsequent views. We treat this sequence as the same key point observed in the subsequent images. Most of the sequences are limited to just two instances of a key point observed in two subsequent images. There is also a large number of longer sequences consisting of points identified in up to 7 subsequent images. This corresponds to 60^o rotation angle between the first, and the last point in the sequence. The larger number of points in a sequence, the more precisely the circular trajectory of the key point can be determined.

4 PLACING CHARACTERISTIC POINTS IN 3D MODEL

The sequences of characteristic points will be used to locate the point in the 3D model. To do that we start from fitting every sequence to the circle equation (16). The center of the circle is known, because this is the (0,0) point in the (X^O, Y^O) coordinates. To position the key point in the 3D space, we need to find the h and R parameters. h is equivalent to the Z^O coordinate, and R is the radius of rotation of the key point around the Z^O axis. Finding a circle, that best matches N instances (in N subsequent images) of a particular key point is a minimization problem, with objective function written in the following form:

$$J(R,h) = \sum_{i=1}^{N}[A(u_i,v_i)h^2 + B(u_i,v_i)h + C(u_i,v_i) - R^2]^2 \to 0. \tag{17}$$

The minimum of that function is obtained by calculating its partial derivatives with respect to R^2 and h and finding their zeros. To reduce the size of expressions we will use a simplified notation for sums, i.e. $\sum A = \sum_{i=1}^{N}[A(u_i,v_i)]$, etc. The equation for the partial derivative of J wrt. R^2 is the following:

$$\frac{\partial J(R,h)}{\partial R^2} = 2\sum[Ah^2 + Bh + C - R^2] \tag{18}$$

To simplify further transformations we can omit the constant before the sum, because it is irrelevant for the task of finding minimum of J. The resulting formula can be written as:

$$\frac{\partial J(R,h)}{\partial R^2} = -NR^2 + h^2\sum A + h\sum B + \sum C = 0 \tag{19}$$

[1] One of the objects available by default in the NeoAxis Game Engine.

Similarly we get the derivative of J wrt. h:

$$\frac{\partial J(R,h)}{\partial h} = 2\sum[(Ah^2+Bh+C-R^2)(2Ah+B)] = 2\sum(2A^2h^3+3ABh^2+B^2h+2ACh+BC- R^2 2Ah - R^2B) = 2(2h^3\sum A^2+3h^2\sum AB+ h\sum B^2+2h\sum AC+\sum BC - R^2(2h\sum A+\sum B)) \tag{20}$$

After omitting the constant before the bracket, and small reorganization, we get

$$\frac{\partial J(R,h)}{\partial h} = 2h^3\sum A^2+3h^2\sum AB+ h(\sum B^2+2\sum AC)+\sum BC-R^2(\sum B+2h\sum A)=0. \tag{21}$$

We can easily isolate R^2 from eq. (19):

$$R^2 = 1/N(h^2\sum A+h\sum B+\sum C) \tag{22}$$

After injecting it to eq. (21) and reorganizing the equation, the following expression of degree 3 with respect to h is obtained:

$$ah^3+bh^2+ch+d=0, \tag{23}$$

where

$$\begin{cases} a & = 2N\sum A^2 - 2(\sum A)^2 \\ b & = 3(N\sum AB - \sum A\cdot\sum B) \\ c & = N(\sum B^2+2\sum AC) - 2\sum A\cdot\sum C - (\sum B)^2 \\ d & = N\sum BC - \sum B\cdot\sum C \end{cases} \tag{24}$$

When $4p^3+27q^2<0$ with $p=c/a-b^2/(3a^2)$ and $q=b/27a(2b^2/a^2-9c/a)+d/a$ the eq. (23) gives three real solutions. To find them the Cardano formula can is used. Then the value of the radius R can be found using eq. (19), for each of the found values of h.

Of course, not every solution, which minimizes the sum (17) is the desired solution. We have no clue, other than verifying the correctness of the solution. Thus we have to take the first of the found values for h. Compute for it the value of R, and the rotation angle β, reconstruct the X^O and Y^O coordinates, and check if the computed 3D coordinates of the key point, multiplied by the transformation matrix (2), reproduce the u,v image coordinates. If yes, the key point has been placed correctly. If not, we have to repeat the checking procedure for the remaining solutions.

We already know how to compute the radius and height of the 3D circle. Reconstruction of the temporary location of the considered key point can be obtained in the following way:

$$\begin{cases} X & = R\cos(\beta \pm i\Delta\theta) \\ Y & = R\sin(\beta \pm i\Delta\theta) \\ Z & = h \end{cases} \tag{25}$$

where $\Delta\theta$ is the object rotation step between subsequent images, i is the image number in the sequence of all images, the $+$ sign is for counterclockwise rotations, the $-$ sign is for clockwise rotations, β is the unknown angle of rotation of the key point in the object coordinates. We assume that the object's coordinate system is stationary (not rotating with the object). As a consequence the key point changes its temporary angle of rotation along with rotations of the object, and β is the rotation angle in reference to the initial rotation of the object ($i=0$).

The method to identify β is based on comparing the u,v key point coordinates from the image with coordinates obtained using eq. (25) multiplied by the transformation matrix P. The estimated image coordinates computed on the basis of reconstructed 3D coordinates (eqs. (4) and (5)):

$$\begin{cases} \hat{u}_i = \frac{P_{1,1}X^O+P_{1,2}Y^O+P_{1,3}Z^O+P_{1,4}}{P_{3,1}X^O+P_{3,2}Y^O+P_{3,3}Z^O+P_{3,4}} \\ \hat{v}_i = \frac{P_{2,1}X^O+P_{2,2}Y^O+P_{2,3}Z^O+P_{2,4}}{P_{3,1}X^O+p_3 2Y^O+P_{3,3}Z^O+P_{3,4}} \end{cases} \tag{26}$$

should fit the original image coordinates u_i, v_i.

To solve eqs. (26) we start from computing the X^O and Y^O coordinates by using the previously derived formulas (see eq. (14) and (12)). Then we divide the second by the first equation from eqs. (25), which gives:

$$\frac{Y^O}{X^O} = \tan(\beta + i\Delta\theta) \tag{27}$$

from the above, we get the formula for the β angle:

$$\beta = \begin{cases} (\arctan(\frac{Y}{X}) - i\Delta\theta) \mod \pi - \mathit{counter\text{-}clockwise\ rotations} \\ (\arctan(\frac{Y}{X}) + i\Delta\theta) \mod \pi - \mathit{clockwise\ rotations} \end{cases} \tag{28}$$

This formula is, however, limited to the range of $(0,\pi)$ (if we add π for negative angles). For full reconstruction the whole range of 2π has to be considered. If β was from the range $(0,\pi)$ then the angle given by eq. (28) is correct. Otherwise (the angle from the range $(\pi,2\pi)$) we need to add additional π. But the problem is that we do not know what the actual range of the angle is. We can find this by verifying the results produced by eqs. (26), with X^O and Y^O computed using (25) and the identified angle. If the results reproduce original u_i, v_i, then the angle is correct, otherwise we should add π to the angle. In this way we get the β angle in the range $[0,2\pi]$, which allows for complete reconstruction of the position of the key point.

An additional issue, that has to be mentioned here, is how does a single key point is recorded in the object model. When the object is rotated, the key points are matched, between subsequent images, but this does not mean that their respective vectors of descriptors are identical. In fact, the vectors gradually change, along with the changing look of the key point neighborhood. In consequence, the memorized key point, is represented by a sequence of vectors of descriptors. And this sequence has to be memorized in order to increase the efficiency of object recognition. Choosing just one of the vectors, would reduce the ability of recognizing the point, when the angle of object rotation would significantly differ, from the angle for which the vector of descriptors has been recorded.

5 EXTENDING THE MODEL FOR LARGER ROBUSTNESS WITH RESPECT TO DISTANCE AND DIRECTION OF OBSERVATION

The presented approach assumed creating an object representation as seen by a camera from a fixed position (the same distance from the object and the same angle of observation). This allows for recognizing the object when the distance and angle of observation does not differ significantly from the one that was used while making photos. It is true that SURF descriptors are robust with respect to the scale transformation, which accompanies changes of the distance between the camera and the object. However, this robustness has its limits. When the difference of distances between the memory representation and the actual object view is too large, the number of matched key points drops too much. Thus we have to extend the model to allow it for incorporating the features visible from both large distance, and from close neighborhood.

This also refers to different directions of observation. We collect all rotational views of of an object, but with the camera located at a fixed height. However, when we look at the same object from a point located higher or lower, the number of matched key points will be reduced. Thus for having a complete object representation, we also have to consider this aspect in the memory model.

6 EXPERIMENTAL RESULTS

The Sec. 4 described all the computational steps needed to build a 3D model of key points. As already mentioned, we identify the key points using the SURF algorithm implementation available in the OpenCV library [Ope15]. The experiments presented in this work are based on images artificially generated using the NeoAxis game engine [Neo15]. Using artificial data allows for easy elimination of all kinds of noises persistent in photos of real objects, and focusing on the results generated by the algorithm itself. Moreover it is easy to set up any kind of experiments with different object and camera settings. The already presented Girl character (see Fig. 2) is transformed into the model, which is visualized in Fig. 3. It is easy to note, that the model does not reproduce the object's geometry precisely. Instead it can be considered a cloud of points around the object of interest. This is also visible in images presented in Fig. 2. However, in our case this is not a problem, because the model is designed for recognition purposes, not for precise shape reproduction. The model from Fig. 3 consist of about 1300 key points. This is more than enough, when we only expect to spot the object within the field of view. However, for more precise analysis of the details, and possible deformation of the object geometry, this might be accurate required number. The number of points in the model results from the particular choice of parameters of the SURF feature detector, and the assumed precision of point fitting. It is easy to regulate the model complexity by choosing appropriate values of the parameters.

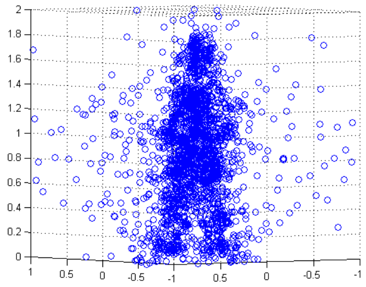

Figure 3: 3D reconstruction of the Girl object

We tested the ability of perceiving objects from different distances using the extended representation as described in Sec. 5. The object representation was extended, by making two sequences of rotational views from two different distances. The first sequence was taken from a large distance, where the object features are barely visible. The second sequence was taken from a small distance, where the object size is comparable to the size of the field of view, and a large number of object details can be perceived. The Fig. 4 demonstrates, how the memorized object is perceived in the environment. The number of identified key points differs significantly between the two images. When the object is seen from close distance, the number of key points typically exceeds 100. This number drops to no more than a few key points, when the object is seen from large distance. The demonstrated range in which the object can be spotted, would not be possible, without using representations coming from different distances.

7 CONCLUSIONS

The presented method allows for creating 3D representations of objects based on local SURF features. The advantage of this approach is the ability to reproduce

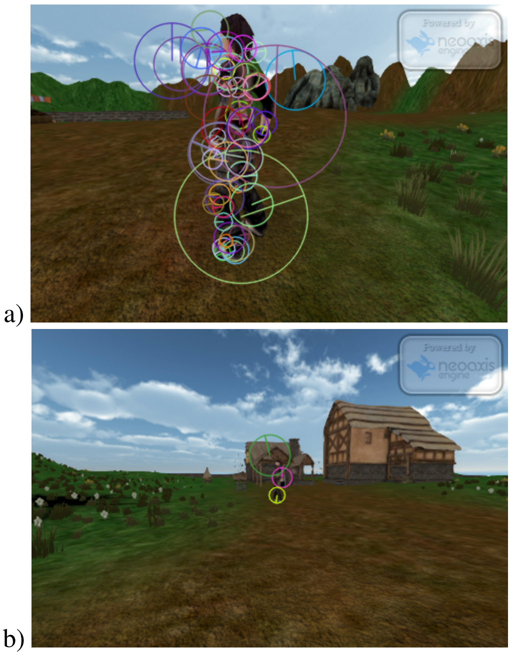

Figure 4: Key points identified within an object, (a) the Girl object seen from close distance, (b) the girl object seen from large distance

local object characteristics, which are robust with respect to occlusions, changing distance and direction of observation, as well as object geometry deformations. The method is extended, to allow for efficient recognition of objects from distant and close locations. We tested this approach in a virtual environment, where we create representations of selected objects, stored in the memory of a virtual agent. In this way, the agent is able to recognize the objects, irrespectively of the mutual location of the agent and the objects.

In addition to local features, the global object geometry is recorded in the 3D point locations. As a consequence the two aspects - local and global - can be analyzed separately. In this way the visual memory based on such a representation has a potential to maintain high recognition ability even if the object undergoes global geometry transformations, like changing body positions of a character. This gives the potential ability to distinguish different parts of an object, on the basis of their movements with respect to the remaining parts of the object. Further this can be used for creating internal characteristics of the objects. This issue will be investigated in the future. Our ultimate goal is creating the visual memory, which will be able to recognize different objects, create representation of the perceived scene on the basis of recognized objects‘ locations, and recognize different, states of the objects, on the basis of their geometry transformations.

8 ACKNOWLEDGMENTS

The research is supported by The Polish National Science Centre, grant No. 2011/03/B/ST7/02518.

9 REFERENCES

[Bay08] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. Speeded Up Robust Features (SURF), Computer vision and image understanding, Vol. 110(3), pp. 346-359, 2008.

[Fre02] Fremont, V., and Chellali, R. Direct Camera Calibration using Two Concentric Circles from a Single View, in Conf. Proc. ICART‘02, Tokyo, pp. 93-98, 2002.

[Fre04] Fremont, V., Chellali, R.: Turntable-based 3D object reconstruction, Conf. on Cybernetics and Intelligent Systems, 2004 IEEE, pp. 1277 - 1282

[Jun05] Yokono, J.J., Poggio, T. Boosting a Biologically Inspired Local Descriptor for Geometry-free Face and Full Multi-view 3D Object Recognition, DSpace@MIT: Massachusetts Institute of Technology, [Online] `http://hdl.handle.net/1721.1/30557`, 2005.

[Kno10] Knopp, J., and Prasad, M., and Willems, G., and Timofte, R., and Van Gool, L. Hough Transform and 3D SURF for robust three dimensional classification, Lecture Notes in Computer Science, Vol. 6316, pp 589-602, 2010.

[Low01] Lowe, D.G. Local feature view clustering for 3D object recognition, IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, pp. 682-688, 2001.

[Neo15] The NeoAxis Game Engine, `http://www.neoaxis.com/`, 2015.

[Ope15] The OpenCV library `http://opencv.org/`, 2015.

[Pfe15] Pfeifer, N., Briese, C., Laser Scanning - Principles and Applications, [Online] `http://publik.tuwien.ac.at/files/pub-geo_1951.pdf`, 2015.

[Rot04] Rothganger, F., Lazebnik, S., Schmid, C., and Ponce J. 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints, Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol. 2, pp. 272-277, 2003.

[Tanb08] Tangelder, J.W.H., Remco, C., Veltkamp, R.C. A survey of content based 3D shape retrieval methods, Multimedia Tools and Applications, Vol. 39(3), pp 441-471, 2008.

[Zha09] Zhang, J., Mai, F., Hung, Y.S., Chesi, G. 3D Model Reconstruction from Turntable Sequence with Multiple View Triangulation, Advances in Visual Computing, Vol. 5876, pp 470-479, 2009.