Analysis of Shapes to Measure Surfaces
An Approach for Detection of Deformations

C. M. Mateo[1], P. Gil[2], D. Mira[1] and F. Torres[2]

[1]*Institute for Computer Research, University of Alicante, San Vicente del Raspeig, Spain*
[2]*Physics, Systems Engineering and Signal Theory Department, University of Alicante, San Vicente del Raspeig, Spain*

Keywords: 3D Shape, Surface, Curvature, Surface Normal, Geometric Modelling, Deformations Visual Perception.

Abstract: This paper presents a method to analyse 3D planar surfaces and to measure variations on it. The method is oriented to the detection of deformations on the elastic object surfaces formed by flat faces. These deformations are usually caused when two bodies, a solid and another elastic object, come in contact and there are contact pressures among their faces. Our method describes a strategy to model the shape of deformation using a mathematical approach based on two concepts: Histogram and Map of curvature. In particular, we describe the algorithm for deformations in order to use it in visual control and inspection tasks for manipulation processes with robot hands. Several experiments and their results are shown to evaluate the validity and robustness of the method to detect and measure deformations in grasping tasks. To do it, some virtual scenarios were created to simulate contacts with fingers of a hand robot.

1 INTRODUCTION

Generally, researchers about robotic manipulation have been focused to recognize rigid objects such as solids (Saxena et al., 2008). But, in recent years, the manipulation process has changed to recognize articulated objects (Katz, 2008), deformable objects (Hirai et al., 2001) such as soft objects (Berenson, 2013), semi-solids such as organic material (Gemici et al., 2014) or tissue (Ramisa et al., 2013). Therefore, three basic types of rigid objects can be considered solids, elastics and deformable objects.

The object rigidity can be mathematically measured with three different interpretations: stiffness which is dependent on the force and the size of area where it is applied, hardness which defines the forces required to penetrate the material and toughness which is the amount of energy that a material can tolerate before it can be fractured.

Using tactile and/or force control is usually to manage the grasping processes (Khalil et al., 2010a). Thus, a robot hand able to do task manipulation of objects applying forces that is not usually enough large to break the surface structure or to drill it. In a rule, the forces never exceed a value which can cause a rupture or penetration. Notwithstanding, this is a complex task in which just the tactile control does not allow us to avoid deformations. More data are required to control the manipulation process if the deformations wish be controlled and measured (Khalil et al., 2010b). Sometimes, the tactile and force information is poor, inconsistent or ambiguous to detect and analyse deformations in an object which is being manipulated (Pomares et al., 2008). In contrast to (Boonvisut et al., 2014), the goal of this paper is to model and identify deformation in semi-solids with elastic properties from contactless sensors. Thus, visual sensors can assist to other sensors (Smith, 1996; Saxena et al., 2008; Mkhitaryan et al., 2012). Firstly, they built a geometrically modeled object and secondly, they identify the object deformation for comparison between points of surface with and without deformation.

Consequently, the elastic object can be construed just as a stiff object because there is not surface penetration or rupture of its structure. The elastic modulus has often been used to measure the stiffness properties in the study of materials when they are known. The elastic modulus measures the applied force per unit area to deform an object surface. However, the elastic modulus cannot be used when the objects were made with an unknown material. In this paper, the stiffness can be measured by the curvature in surface points from the object geometry. The curvature features are computed from mesh points which models the object surface.
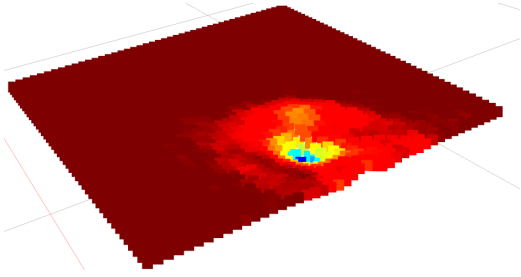
Figure 1: Level curves set $\mathbf{S_P}$ computed from a surface $\mathbf{P}$ with a deformation.

The paper is structured as follows: the concept of curvature to measure curves by means differential geometry is shown in Section 2. In Section 3, we present the method based on surface variation to model surfaces, detect and measure deformation in those. Experimental results of the deformations caused in virtual robot manipulation tasks are shown in Section 4. Finally, section 5, contains the conclusions.

2 MATHEMATICAL APPROACH TO COMPUTE CURVATURE OF SURFACES

In this paper, the differential geometry of curves is used as a tool to propose a method which allows us to analyse 3D surfaces. Generally, in computer vision, the 3D object surface consists of an unstructured point cloud represented as $\mathbf{P} = \{\mathbf{p}_i \in \Re^3\}$. If a plane object in the Euclidean space is deformed by pressing on its outside surface, the geometric properties change and some smooth curves could appear. Here, the curves are understood as a variation of surface, and they can be computed by measuring of the orientation change of normal vector to the surface.

Generally, each point $\mathbf{p}_i$ satisfies a set of axioms in relation with its neighbourhood environment. Thus, the computation of the geometric properties of the curves depends on the k-nearest neighbour points to each $\mathbf{p}_{ij}$. Therefore, $\mathbf{P}$ can be sampled as a set of patch $\mathbf{N}_j$ and each one is a subset of points $\mathbf{p}_{ij}$. Both, the size (radius) and number of points of a patch (dense) influence the accuracy to compute the geometric properties of the curves. Few points cause inaccuracy. In contrast, many points distort the values hindering the detection of orientation changes in the surface. In this last case, the detection is smoothed.

An analysis of the eigenvalues computed from covariance matrix of points in a neighbourhood environment according to (1) can be used to estimate the local geometric properties of a patch of the surface (Shaffer et al., 2001). To define the covariance matrix is applied PCA (Principle Component Analysis) as follows:

$$\mathbf{C_P} = \mathbf{PP}^T = \begin{bmatrix} \mathbf{p}_{i1} - \overline{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{ik} - \overline{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i1} - \overline{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{ik} - \overline{\mathbf{p}} \end{bmatrix}^T \qquad (1)$$

where each $\mathbf{p}_{ij}$ is a point of the neighbourhood environment $\mathbf{N}_j$ and $\overline{\mathbf{p}}$ is the centroid of the patch. And k defines the number of points $\mathbf{N}_j$.

Equation (1) is solved by Turk and Pentland method (Turk et al., 1991) that allow us the computation of eigenvalues and eigenvectors of $\mathbf{C_P}$ with low computational cost by building a matrix $\mathbf{A} = \mathbf{PP}^T$ and by applying singular value decomposition (SVD) to $\mathbf{A}$ as follows:

$$\mathbf{A} \cdot \mathbf{v}_j^{'} = \lambda_j^{'} \cdot \mathbf{v}_j^{'} \rightarrow \mathbf{P}^T \mathbf{P} \cdot \mathbf{v}_j^{'} = \lambda_j^{'} \cdot \mathbf{v}_j^{'} \qquad (2)$$

where $\mathbf{v}_j^{'}$ are the eigenvectors and $\lambda_j^{'}$ are the singular values of the matrix $\mathbf{A}$. Multiplying by $\mathbf{P}$ is obtained:

$$\mathbf{PP}^T \mathbf{P} \cdot \mathbf{v}_j^{'} = \lambda_j^{'} \cdot \mathbf{P} \mathbf{v}_j^{'} \rightarrow \mathbf{C_P} \mathbf{P} \cdot \mathbf{v}_j^{'} = \lambda_j^{'} \cdot \mathbf{P} \mathbf{v}_j^{'} \qquad (3)$$

Then, the eigenvalues and associated eigenvectors of $\mathbf{C_P}$ can be obtained as:

$$\lambda_j = \lambda_j^{'} \quad \text{and} \quad \mathbf{v}_j = \frac{\mathbf{p} \cdot \mathbf{v}_j^{'}}{\sqrt{\lambda_j^{'}}} \qquad (4)$$

The eigenvalues sum provides information about the surface variation between each point of the patch $\mathbf{N}_j$ and its centroid. In addition, the smallest eigenvalue provides a measure of the variation along the normal vector to the surface. Consequently, the eigenvalues can help to classify the concavity of the surface at each point, and the set of all defines the curvature parameter. Thus, the local maximum curvature (Pauly et al., 2002) of $\mathbf{p}_{ij}$ within the patch $\mathbf{N}_j$ can be computed by:

$$c_{ij} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad (5)$$

where $\lambda_0 \le \lambda_1 \le \lambda_2$ are eigenvalues of $\mathbf{C_P}$. The associated eigenvector $\mathbf{v}_0$ is the normal vector to the tangent plane of the patch surface $\mathbf{P}$ that define $\mathbf{v}_1$
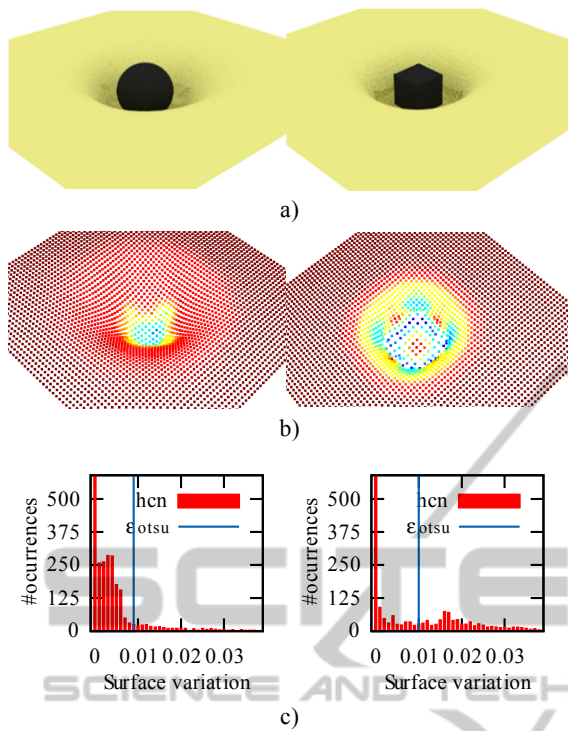
Figure 2: a) CAD Model of a deformation caused in a planar surface by contacting of two different geometric objects: Sphere and Cube. b) Level curves set computed from the deformation and represented with a different colour l. c) Histogram.

and $\mathbf{v}_2$. The set of level curves is defined as a function $\mathbf{S_P} : \Re^3 \rightarrow \Re$ as follows:

$$\mathbf{S_P}(\Phi) = \left\{ (x, y, z) \in \Re^3 : \Phi(x, y, z) = l \right\} \qquad (6)$$

where l is a constant value and it represents a level curve in the surface. Then, every level curve is computed like a cluster with the same colour that represents the points of $\mathbf{P}$ with a similar value of curvature (Fig. 1).

3 OUR METHOD TO FIND SURFACE VARIATIONS

In this work, the surfaces are represented as a point clouds. It is worth nothing that the data of $\mathbf{P}$ can be obtained from a structured or unstructured way. $\mathbf{P}$ is structured whether it can be stored in a matrix $\mathbf{P}_{X \times Y}$ where X and Y are the number of rows and columns, respectively. This occurs whether $\mathbf{P}$ is acquired from a sensor like a RGBD or ToF, then each point corresponds with a position of the sensor. In contrast, $\mathbf{P}$ is unstructured whether it can only be stored as a

vector matrix $\mathbf{P}_{X \times 1}$. In this last case, $\mathbf{P}$ has no direct relationship with the data of range image. $\mathbf{P}$ is usually unstructured when it is built from a virtual CAD model (Fig. 2a). Consequently, in order to find the neighbor of each point of $\mathbf{P}$ is typically used either kdtree or octree data structure.

The proposed algorithm has two phases: Initialization and Extract-Curvatures. The first step is only computed when $\mathbf{P}$ is unstructured because it it needs to be structured to get sorted points. The closest points in the Euclidean space must be stored as neighbors within $\mathbf{P}$. Thereby, it is possible to search points with the same geometrical properties and located in the same neighbourhood environment. This is essential to calculate the level curves from the curvature parameter and to detect surface variations given by the transversal paths to the level curves.

The proposed method detects deformations of any surface represented as $\mathbf{P}$. It is based on finding the critical points of $\mathbf{P}$ in which there are surface variations. In this work, the critical points are points belonging to different level curves Φ of the surface but also, they lie in the transversal path that goes across the level curves by fitting of singular points. The singular points are defined as the points with maximum curvature values, and whose curvature value is estimated like a curvature threshold computed from a Histogram of curvature.

3.1 Histogram of Curvatures

Once the curvature parameter, according to (5), is calculated for all points within $\mathbf{P}$ (Fig. 2b), a curvature histogram is built (Fig. 2c.). It represents the distribution of the surface variation and it can be computed as follows:

$$H_\mathbf{P} = \frac{\text{Number}(c_{ij})}{\text{size}(\mathbf{P})} \qquad (7)$$

where c_{ij} is the curvature value, and $\text{size}(\mathbf{P})$ represents the density or number of points used to sample the surface. $H_\mathbf{P}$ changes depending on the size of the neighbourhood environment $\mathbf{N}_j$ used to compute each c_{ij}. The user must choose the radio according to the accuracy for the detection of abrupt changes in the surface (Fig. 4).

The histogram $H_\mathbf{P}$ is useful to find the singular points, that is to say both the points with $\max(c_{ij})$ and the boundary points which define the border where there is no curvature variation in the surface. The boundary points split the curvature region and the non-curvature region. They are computed from $H_\mathbf{P}$

using the technique presented in (Otsu, 1979). The histogram allows us to find the curvature threshold ε_{otsu} by minimizing the standard deviation of the Gaussian distributions that represent the two zones (Fig. 2c).

3.2 Path of Critical Points to Measure the Surface Variations

We assume that the surface is always differentiable, then the surface gradient at a point is either zero or perpendicular to the level curve which represents the surface variation at that point. We inspect the critical points of the surface function along its level curves. Our critical points are the curve points which follow the direction of gradient.

The proposed algorithm is used to find the critical and boundary points and the transversal path which define the deformation on the surface. The algorithm uses an octree O like (Papon et al., 2013) in order to contain all points p of **P** as a sorted structured. Later, we create a priority queue Q of pairs $\{p, N\}$, where p is each point of O and N is its adjacency list. N represents the 26-conneted neighbour points. Q puts on the head, the points p with the greater curvature values. Later, Q is filtered to only obtain the values of Q which are greater than ε_{otsu} then they are stored in R. Subsequently, R is crossed according to algorithm from line 18 until 24 in order to get a list with the deformations from the clusters D. Finally, we use Dijkstra's algorithm $\text{FindMinPath}(v_1, v_2)$ to find the minimum paths. Let v_1 be the point with maximum curvature (the head into each D') and let v_2 be the target point (points into the boundaries).

Algorithm: Finding curvatures.

```
Begin
// Initialization:
O ← P
For each leaf of O with points do:
  For i = leaf_x −1 : leaf_x +1 do:
5       For j = leaf_y −1 : leaf_y +1 do:
          For k = leaf_z −1 : leaf_z +1 do:
            N ←—add— p_ijk
          End For
        End For
10    End For
      Q ←—add— {p, N}
  End For
  // Extract-Curvatures:
  R ←—curvature(p)≥ε_otsu— Q
15 While R ≠ { } do:
```

$C \xleftarrow{\text{add}} \text{head}(R)$ // Candidate list of point for deformation D'
$R \xleftarrow{\text{remove}} \text{head}(R)$
While $C \neq \{ \}$ **do:**
  If $\text{curvature}(\text{head}(D')) \leq \varepsilon_{\text{otsu}}$ **then:**
20     $D' \xleftarrow{\text{add}} \text{head}(D')$
       $C \xleftarrow{\text{addAdjacentPoints}} \text{head}(C)$
  End If
  $C \xleftarrow{\text{remove}} \text{head}(C)$
End while
25 $D \xleftarrow{\text{add}} D'$
While $D' \neq \{ \}$ **do:**
  If $\text{curvature}(\text{head}(D')) \equiv \varepsilon_{\text{otsu}}$ **then:**
    $\text{FindMinPath}(\text{head}(\text{tail}(D)), \text{head}(D'))$
  End If
30 $D' \xleftarrow{\text{remove}} \text{head}(D')$
End while
End while

4 EXPERIMENTS IN GRASPING TASKS

Several experiments and tests have been done. We have modelled the movements of a real robot hand considering both the kinematics and the virtual model (shape and structure) of their fingers and palm without physical constraints and without considering singularities, by using free software Blender. In particular, the model of robot hand corresponds to a Shadow Hand Robot available in our research laboratory.

Our experiments simulate several deformations which are caused by contact between fingers of the virtual robot hand and a planar object. The deformations depend on the orientation of the fingers, the trajectory used in the contact process, the shape of contact occurs (Fig.3).

Each experiment consists of a movement that shows an image sequences. Each sequence is done from 60 images for testing. The algorithm computes the level curves set from both the curvature parameter and the histogram of curvatures. They allow us to obtain the evolution of the curvatures during a time sequence through the formation process of the deformation (Figs. 3 and 4). Afterwards, we estimate the path of critical points to measure the surface variation and the topographic profile of the deformation (Figs. 3 and 5).

The algorithm is implemented with the C++ language and using the open source libraries PCL, Boost and Eigen. It runs over a computer with a Core i7-4770k processor, equipped with 8GB of system memory and an nVidia GeForce 760GTX.
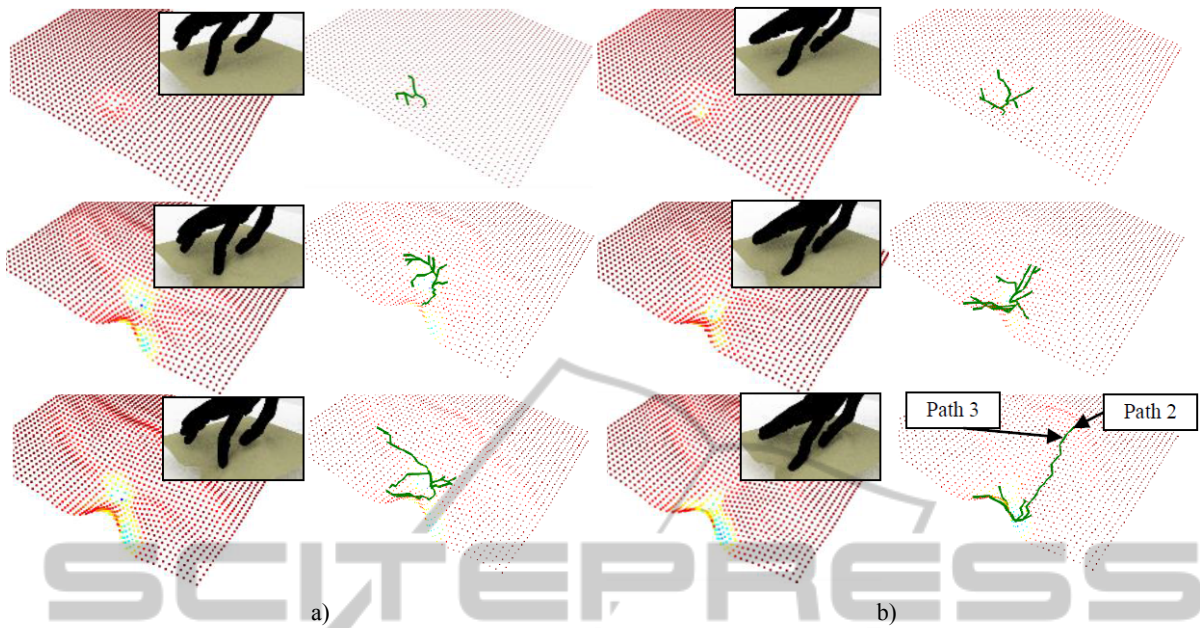
Figure 3: Temporal evolution of the deformation of a planar surface caused by contact of a robot finger from two movement sequence of the robot hand. a) Test 1. b) Test 2.

Fig. 4 shows the dependence of the curvature measures when the radius of the neighborhood is chosen in a proper way or not. If the radius is enough large then the smallest gaps on the surface are not detected by the algorithm. Also, Fig 4 shows the evolution of the boundary points and how they grow quickly in the first iterations. This fact indicates that the contact between robot hand and elastic surface is occurring in that moment. Later, the growth is smoother and more progressive. The curve slope indicates the deformation level generated during the grasping over time. Fig. 5 shows us how the minimum transversal paths retrieve us information to determine the deformation in an instant time. In the test 2, there are two dominant paths 2 and 3 because they are far
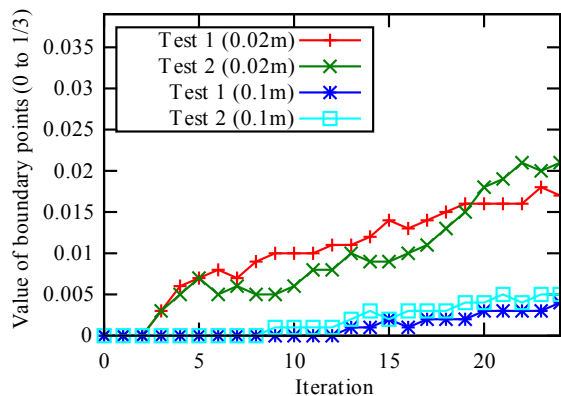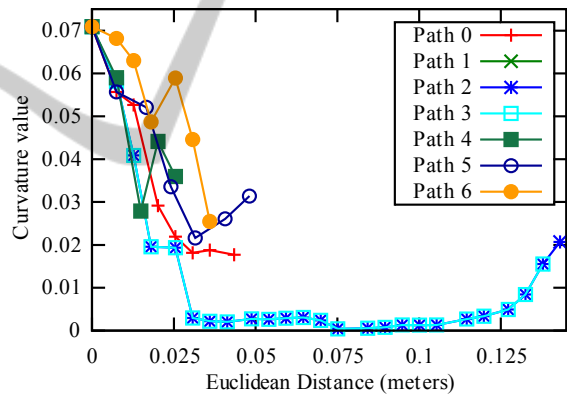

Figure 5: Characterization of the transversal path variations computed from critical points for the Test2.

from the rest of paths (those other are close and have similar length). The paths 2 and 3 determine the shape of the deformation caused on the surface. But also, the paths 0,1,4,5 and 6 show a sharp slope which represents an abrupt deformation. This is much variation among level curves located near on the surface.

5 CONCLUSIONS

This paper has described a novel approach to analyse the deformation of flat surfaces. The proposed method builds curvature variation maps to measure

Figure 4: Evolution of the curvature mean value through the movement sequences shown in Fig. 3.

the local unevenness in the surface by comparison among the points that lie on the same surface. Also, the method uses a novel algorithm to characterize and to describe the curvature variation caused by deformation. To do it, our algorithm identifies the level curves and later, it finds the critical points which define transversal paths among those level curves. Thus, we generate topographic profiles in the local maximum gradient directions. The gap, between the curvature value and the Euclidean distance of the critical points, determines whether or not deformation and the degree of slope that defines if the deformation is smooth or abrupt.

The results reveal that this surface analysis provides an empirical investigation on the effect of the deformations about the surface of elastic objects when several grasping tasks with robot hand are performed. And most importantly, an algorithm to measure these deformations has been implemented and tested successfully. It is programed in C++, and it can be integrated into robotic platforms.

ACKNOWLEDGEMENTS

REFERENCES

Berenson, D. 2013. Manipulation of deformable objects without modeling and simulating deformation. In *Proc. Intelligent Robots and Systems (IROS)*. Pp. 4525-4532. Doi: 10.1109/IROS.2013.6697007.

Boonvisut, P., Cenk-Cavusoglu, M. 2014, Identification and active exploration of deformable object boundary constraints through robotic manipulation. In *The International Journal of Robotics Research,* vol. 33(11), Pp. 1445-1461. Doi: 10.1177/0278364914536939.

Gemici, M.C. and Saxena, A. 2014. Learning haptic representation for manipulating deformable food objects. In, *Intelligent Robots and Systems (IROS)*. pp.638-645. Doi: 10.1109/IROS.2014.6942626.

Hirai, S., Tsuboi, T., Wada, 2001. T. Robust grasping manipulation of deformable objects. In *Proc. IEEE Symposium on Assembly and task planning.* Pp. 411-416.

Katz, D. 2008. Manipulating articulated objects with interactive perception. In *Proc. IEEE Robotics and Automation (ICRA)*. Pp. 272-277. Doi: 10.1109/ROBOT.2008.4543220.

Khalil, F.F. and Payeur, P. 2010a. Dexterous robotic manipulation of deformable objects with muti-sensory feedback- a review. *Robot manipulators trends and development. Chapter 28.* Doi: 10.5772/9183.

Khalil, F.F. Curtis, P., Payeur, P. 2010b. Visual monitoring of surface deformations on objects manipulated with a robotic hand. In *Proc IEEE Int. Workshop on Robotic and Sensors Environments (ROSE).* Pp. 1-6, Doi: 10.1109/ROSE.2010.5675327.

Mkhitaryan, A., Burschka, D. 2012. Vision based haptic multisensory for manipulation of soft, fragile objects. In *Proc. IEEE Sensors,* Pp. 1–4.

Otsu N., 1979. A Threshold Selection Method from Gray-Level Histograms. *In: IEEE Transactions on systems, Man, and cybernetics*, vol. 9 (1), pp 62-66. Doi: 10.1109/TSMC.1979.4310076.

Pauly, M., Gross, M., Kobbelt, L.P. 2002. Efficient Simplification of Point-Sampled Surfaces. In *Proc IEEE Conference on Visualization*. Pp. 163-170.

Papon, J., Abramov, A., Scholer, M., Woergoetter, F. 2013. Voxel cloud connectivity segmentation-supervoxels form pointclouds. In *Proc IEEE Computer Vision and Pattern Recognition (CVPR)*. Pp. 2027-2034. Doi: 10.1109/CVPR.2013.264.

Pomares, J., Gil, P., García, G.J., Sebastián, J.M., Torres, F. 2008. Improving detection of surface discontinuities in visual–force control Systems. *Image and Vision Computing,* vol. 26(10), pp. 1435-1447.

Ramisa, A., Alenya, G. Moreno-Noguer, F., Torras, C. 2013. FINDDD: A fast 3D descriptor to characterize textiles for robot manipulation. In *Proc. IEEE/RSJ Ingelligent Robots and Systems (IROS)*. Pp. 824-830.

Saxena, A., Driemeyer J., Ng, A.Y. 2008. Robotic grasping of novel objects using vision. In *The International Journal of Robotics Research, vol. 27(2),* Pp. 157-173. Doi: 10.1177/0278364907087172.

Shaffer, E., Garland, M. 2001. Efficient Adaptive Simplification of Massive Meshes. In *Proc. IEEE Conference on Visualization*. Pp. 127-134.Doi: 10.1109/VISUAL.2001.964503.

Smith., P.W. 1996. Vision based manipulation of non-rigid objects. In *Proc. IEEE Robotics and Automation (ICRA)*. Pp. 3191-3196. Doi: 10.1109/ROBOT.1996.509198.

Turk, M., Pentland, A.P. 1991. Face recognition using eigenfaces. In *Proc IEEE Computer Vision and Pattern Recognition (CVPR)*. Pp. 586-591. Doi: 10.1109/CVPR.1991.139758.