Computation of Curvature Skeleton to Measure Deformations in Surfaces

Carlos M. Mateo, Pablo Gil and Fernando Torres

Abstract This work presents a method to analyse 3D flat objects and to measure variations of its surface. The method represents of the objects using processsing techniques based on point cloud. The proposed method is focused on the deformation detection of elastic objects which are 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 an algorithm to estimate the deformation shape. It is done by calculating its skeleton. Particularly, the algorithm calculates the curvature values of the surface points on the object using an analysis of eigenvectors and eigenvalues. Afterwards, the points of the similar curvature are grouped in level curvatures. Finally, a set of the minimum path among level curvatures are calculated to obtain the skeleton. The paper shows a set of experiments which simulate the deformations caused by a robot hand in manipulation tasks of flat objects.

Keywords 3D shape · Surface · Curvature · Surface normal · Geometric modelling · Deformations visual perception

C.M. Mateo (✉)
Institute for Computer Research, University of Alicante,
San Vicente del Raspeig, Alicante, Spain
e-mail: cm.mateo@ua.es

P. Gil · F. Torres
Physics, Systems Engineering and Signal Theory Department,
University of Alicante, San Vicente del Raspeig, Alicante, Spain
e-mail: pablo.gil@ua.es

F. Torres
e-mail: fernando.torres@ua.es

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*,
Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1_11

1 Introduction

Generally, researchers about robotic manipulation have been focused to recognize rigid objects such as solids [1, 2]. But, in recent years, the manipulation process has changed to recognize articulated objects [3], deformable objects [4] such as soft objects [5] semi-solids such as organic material [6] or tissue [7]. 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 [8]. Thus, a robot hand is 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 [9]. Sometimes, the tactile and force information is poor, inconsistent or ambiguous to detect and analyse deformations in an object which is being manipulated [10]. In contrast to [11], 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 [1, 12, 13]. 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.

The paper is structured as follows: the concept of curvature to measure curves by means differential geometry is shown in Sect. 2. In Sect. 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 Sect. 4. Finally, Sect. 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 [14]. To define the covariance matrix is applied PCA (Principle Component Analysis) as follows:

$$\mathbf{C_P} = \mathbf{P}\mathbf{P}^T = \begin{bmatrix} \mathbf{p}_{i1} - \overline{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{ik} - \overline{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i1} - \overline{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{ik} - \overline{\mathbf{p}} \end{bmatrix}^T \tag{1}$$

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 [15] that allow us the computation of eigenvalues and eigenvectors of $\mathbf{C_P}$ with low computational cost by building a matrix $\mathbf{A} = \mathbf{P}\mathbf{P}^T$ and by applying singular value decomposition (SVD) to $\mathbf{A}$ as follows:

$$\mathbf{A} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{v}'_j \rightarrow \mathbf{P}^T\mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{v}'_j \tag{2}$$

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

$$\mathbf{P}\mathbf{P}^T\mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{P}\mathbf{v}'_j \rightarrow \mathbf{C_P}\mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{P}\mathbf{v}'_j \tag{3}$$

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

$$\lambda_j = \lambda'_j \text{ and } \mathbf{v}_j = \frac{\mathbf{p} \cdot \mathbf{v}'_j}{\sqrt{\lambda'_j}} \tag{4}$$

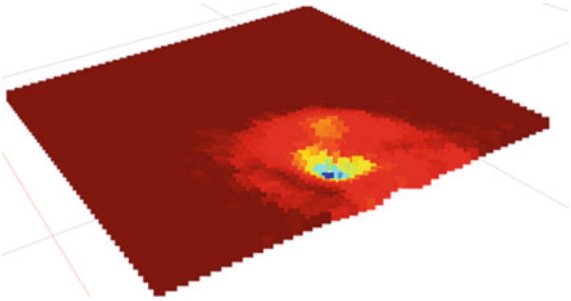


Fig. 1 Level curves set $\mathbf{S_P}$ computed from a surface $\mathbf{P}$ with a deformation

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 [16] of $\mathbf{p}_{ij}$ within the patch $\mathbf{N}_j$ can be computed by:

$$c_{ij} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{5}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$ 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 $\mathbf{v}_1$ and $\mathbf{v}_2$. The set of level curves is defined as a function $\mathbf{S_P} : \Re^3 \rightarrow \Re$ as follows:

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

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.

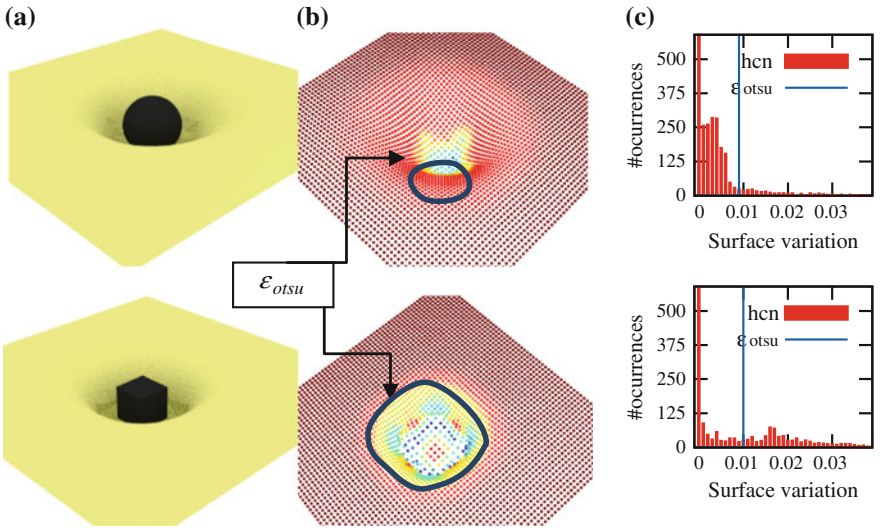


Fig. 2 **a** CAD Model of a deformation caused in a planar surface by contacting of two different geometric objects: *Sphere* and *Cube*. **b** Level curves set computed from the deformation and represented with a different colour *l*. **c** Histogram

The proposed algorithm has two phases: Initialization and Extract-Curvatures. The first step is only computed when **P** is unstructured because it needs to be structured to get sorted points. The closest points in the Euclidean space must be stored as neighbors within **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 **P**. It is based on finding the critical points of **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 **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:

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

where c_{ij} is the curvature value, and 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. 5).

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 [17]. 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).

```
Algorithm. Finding curvatures
Begin
// Initialization:
O ⟵ P
For each leaf of O with points do:
   For i = leaf_x − 1 : leaf_x + 1 do:
05        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 ⟵(add) head(R)  // Candidate list of point for def. D′
       R ⟵(remove) head(R)
    While C ≠ {} do:
20         D′ ⟵(add) head(D′)
       If curvature(head(D′)) ≤ ε_otsu then:
        C ⟵(addAdjacent Points) head(C)
       End If
       C ⟵(remove) head(C)
       End while
25     D ⟵(add) D′
       While D′ ≠ {} do:
          If curvature(head(D′)) ≡ ε_otsu then:
               FindMinPath(head(tail(D)), head(D′))
          End If
30        D′ ⟵(remove) head(D′)
```

```
    End while
End while
```

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 [18] in order to contain all points p of $\mathbf{P}$ 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-connected 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 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).

4 Experiments

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.

4.1 Experiments to Compute Curvatures Skeleton

The deformations are described by curvature skeletons. A curvature skeleton is a set of minimum transversal paths. These path goes across the level curvatures, thus they start in the same critical point (maximum curvature) and end over the boundary points. An example of the skeleton is represented in the curvature map of the Fig. 3. The chart of that figure shows the minimum transversal paths which define the behaviour of the deformation. Thereby, if all paths have similar distances then

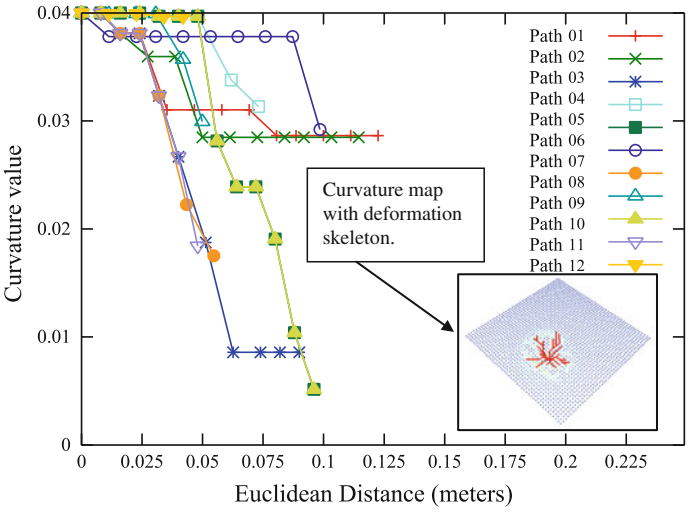


Fig. 3 Characterization of the transversal path variations computed from critical points

the deformations are homogeneus (in this case, the values are all close to 0.1 m and then the deformation shape could be understand like a circle). In opposition, if the paths represent different distances then the deformation is heterogeneous (in this case, other irregular deformation shapes can be read). Furthermore, the chart shows the deformation as a set of branches which represents several minimum transversal paths. The number of branches and the value of slope of them determine how is the deformation: steep, smooth, flat, etc. In Fig. 3, there are two groups of path (group 1: path 1, 2, 4, 6, 8, 11; group 2: path 3, 5, 7, 9, 10, 12) and each one tends to curvature values around to 0.03 (group 1) and 0.005 (group 2).

4.2 Experiments Applied to Grasping Tasks

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 the finger and the pressure produced by it when the contact occurs (Fig. 4).

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. 4 and 6). Afterwards, we estimate the path of critical points to measure the surface variation and the topographic profile of the deformation (Figs. 5 and 6).

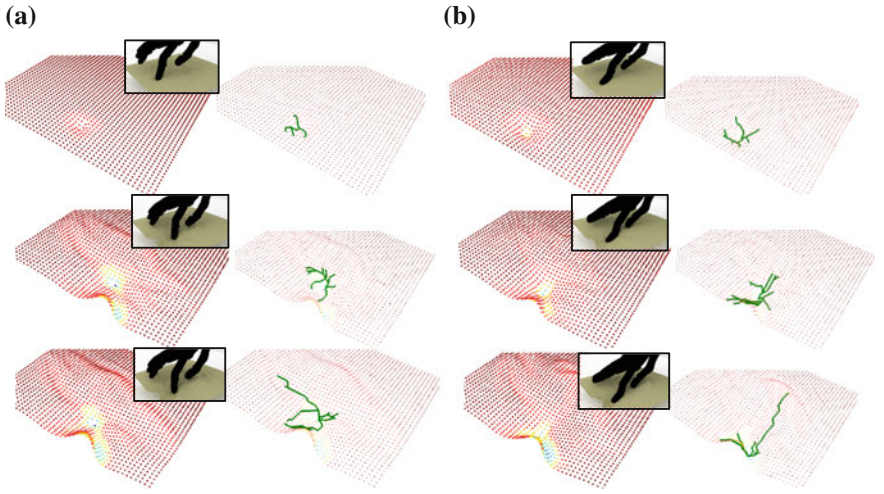


Fig. 4 Temporal evolution of the deformation of a planar surface caused by contact of a robot finger from two movement sequence of the robot hand. **a** Test 1. **b** Test 2

Fig. 5 Evolution of the curvature mean value through the movement sequences shown in Fig. 4

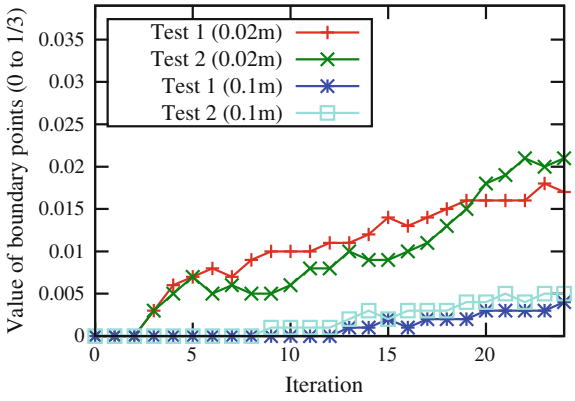
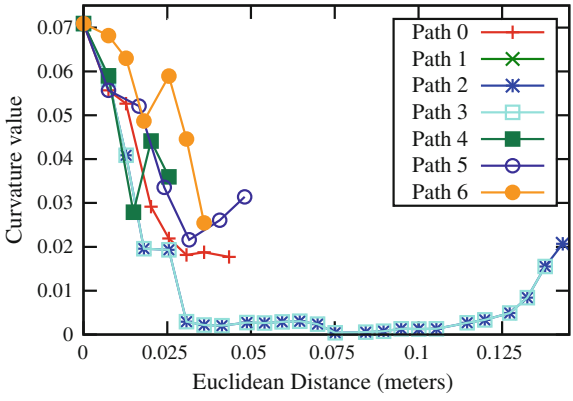


Fig. 6 Characterization of the transversal path variations computed from critical points for the last frame of the Test 2



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 5 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 not enough large then the smallest gaps on the surface are not detected by the algorithm. Also, Fig. 5 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. Figure 6 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 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 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.

Acknowledgments The research leading to these results has received funding from the Spanish Government and European FEDER funds (DPI2012-32390 and DPI2015-68087R) and the Valencia Regional Government (PROMETEO/2013/085).

References

1. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. Int. J. Robot. Res. **27**(2), 157–173 (2008). doi:10.1177/0278364907087172
2. Mateo, C.M., Gil, P., Torres, F.: Visual perception for the 3D recognition of geometric pieces in robotic manipulation. Int. J. Adv. Manuf. Technol. (IJAMT) (Springer, London) (2015). doi:10.1007/s00170-015-7708-8
3. Katz, D.: Manipulating articulated objects with interactive perception. In: Proceedings of the IEEE Robotics and Automation (ICRA), pp. 272–277 (2008). doi:10.1109/ROBOT.2008.4543220
4. Hirai, S., Tsuboi, T., Wada, T.: Robust grasping manipulation of deformable objects. In: Proceedings of the IEEE Symposium on Assembly and Task Planning, pp. 411–416 (2001)
5. Berenson, D.: Manipulation of deformable objects without modeling and simulating deformation. In: Proceedings of the Intelligent Robots and Systems (IROS), pp. 4525–4532 (2013). doi:10.1109/IROS.2013.6697007
6. Gemici, M.C., Saxena, A.: Learning haptic representation for manipulating deformable food objects. In: Intelligent Robots and Systems (IROS), pp. 638–645 (2014). doi:10.1109/IROS.2014.6942626
7. Ramisa, A., Alenya, G. Moreno-Noguer, F., Torras, C.: FINDDD: a fast 3D descriptor to characterize textiles for robot manipulation. In: Proceedings of the IEEE/RSJ Ingelligent Robots and Systems (IROS), pp. 824–830 (2013)
8. Khalil, F.F., Payeur, P.: Dexterous robotic manipulation of deformable objects with mutisensory feedback—a review. In: Robot Manipulators Trends and Development, Chap. 28 (2010). doi:10.5772/9183
9. Khalil, F.F., Curtis, P., Payeur, P.: Visual monitoring of surface deformations on objects manipulated with a robotic hand. In: Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE), pp. 1–6 (2010). doi:10.1109/ROSE.2010.5675327
10. Pomares, J., Gil, P., García, G.J., Sebastían, J.M., Torres, F.: Improving detection of surface discontinuities in visual-force control Systems. Image Vis. Comput. **26**(10), 1435–1447 (2008)
11. Boonvisut, P., Cenk-Cavusoglu, M.: Identification and active exploration of deformable object boundary constraints through robotic manipulation. Int. J. Robot. Res. **33**(11), 1445–1461 (2014). doi:10.1177/0278364914536939
12. Smith, P.W.: Vision based manipulation of non-rigid objects. In: Proceedings of the IEEE Robotics and Automation (ICRA), pp. 3191–3196 (1996). doi:10.1109/ROBOT.1996.509198
13. Mkhitaryan, A., Burschka, D.: Vision based haptic multisensory for manipulation of soft, fragile objects. In: Proceedings of the IEEE Sensors, pp. 1–4 (2012)
14. Shaffer, E., Garland, M.: Efficient adaptive simplification of massive meshes. In: Proceedings of the IEEE Conference on Visualization, pp. 127–134 (2001). doi:10.1109/VISUAL.2001.964503
15. Turk, M., Pentland, A.P.: Face recognition using eigenfaces. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), pp. 586–591 (1991). doi:10.1109/CVPR.1991.139758
16. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: Proceedings of the IEEE Conference on Visualization, pp. 163–170 (2002)
17. Otsu N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979). doi:10.1109/TSMC.1979.4310076
18. Papon, J., Abramov, A., Scholer, M., Woergoetter, F.: Voxel cloud connectivity segmentation-supervoxels form pointclouds. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), pp. 2027–2034 (2013). doi:10.1109/CVPR.2013.264