15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Skeletonization and 3D Graph Approach for Thin Objects Recognition in Pick and Place Tasks

Pierre WILLAUME
Icube laboratory, Hager Group
pierre.willaume@ecam-strasbourg.eu

Pierre PARREND
Icube laboratory
Ecam Strasbourg-Europe

Etienne GANCEL
Hager Group

Aline DERUYVER
Icube laboratory
Unistra

Abstract

Objects retrieval is an important task in pick and place processes, especially when the objects are provided in a disorganized set. However, the methods proposed in the literature are not suitable for thin objects because only key points or edges of the shapes are studied, whereas in real cases, the objects can be thin and without special features. In this paper, a method based on graph comparison between stereoscopic images and a model is proposed to retrieve thin featureless objects and without having a long pre-processing time. Firstly, a thinning algorithm is applied to extract the skeleton of both shapes. Secondly, the skeleton is translated into a graph to extract the significant information of each part of the skeleton. Finally, the distances between the nodes are computed and the topography of the graph is analyzed to retrieve the desired objects. A comparison has been realized with a key-points extraction algorithm.

1 Instructions

Pick and place is a process that aims to bring an object to a selected location. This process is constantly improved with the desire of companies to use automated systems adaptable to the work environment [Berger, 2014]. When the starting position is not fixed, a method consisting in using a robot with one or several cameras is usually applied. The cameras detect objects and provide their positions to the robot to pick them up. This method is fast because the objects have not to be pre-placed by another system. However, these systems require image processing for detecting and locating the objects. Object detection difficulties mainly depend on the aspect and on the orientation of the object [Ikeuchi, 1987]. Currently, the proposed solutions require a large database because they are sensitive to the deformations and the method is not adapted for thin objects. The aim of this paper is to propose a method to randomly detect disposed thin objects in 3 dimensions without having a long pre-processing time. We propose a method based on graph matching. The first step is the application of a filter to reduce the noise in the images, which emphasizes the recovered objects. To distinguish the objects among the others and to bring out the thinnest ones, we apply a thinning algorithm to the images. This process allows to retrieve the skeleton of the set of objects but it does not allow to find the match between the representation of the left image and the right image. A solution is to translate the shapes into a graph for performing the localisation of the different objects. This step permits to find the parts of the image belonging to the same skeleton. For each node of the graph, we match the extrema and the junction nodes. A set of rules based on depth, distance and angle between nodes is established to make the graph matching. Object recognition is performed according to the topology of the graph. The method stands out from the others by the non-necessity to have a CAD file and by its ability to manage objects without using key points features. This paper is organised as follows: The second section describes the state of the art. The third section details the proposed method. The fourth section presents the experimentation. The fifth section discusses the advantages and the limitations of our method. The last section states the conclusion.

2 Related Work

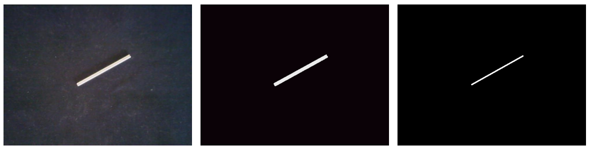
The field of bin-picking studied for decades [Buchholz, 2016, Ikeuchi, 1987, Ulrich et al., 2012, Willaume et al., 2016] still arouses interest mainly to recover disorganized objects. The objects in the bin must be compared with a model to recognize its shape in order to be able to take it. Algorithm performance is calculated according to three different criteria: 1) the computation time of image processing to detect the object is one of the most important criterion, which determines the computation time required to give the position of the object to the robot; 2) the kind of object to be processed (size, shape...); 3) the time to set up and build the model of the system. Some authors [Ikeuchi, 1987] proposed a method to generate 3D algorithms for recognizing objects with a geometric model for pick and place tasks. Apparent shape of the model is generated in various directions of the object. They are classified according to their position and based on a set of characteristics such as the dominant visible faces. An interpretation tree is generated according to the shape of the objects. This process has been improved [Ulrich et al., 2012] to recognize the orientation and position of a 3D part using a single camera and a CAD file. Only the geometric information from the CAD model is generated as a hierarchical model. Information of texture and reflectance of the surface of the object are not taken into account. The large number of applications using these methods in industry proves their efficiency. Indeed, the setup is easy because it only requires a calibration and a CAD file of the object that we look for. However, as the pre-processing step of the CAD method is greedy in time, the system has to save the set of all the possible positions. Piccinini [Piccinini et al., 2012] proposed to use SIFT algorithm [Lowe, 1999] for the detection and the location of objects. The computation of key points and their description is made during the image acquisition step. The different models are saved in a database. The key points and the descriptions are compared with the models. A clustering is performed in order to detect if the key points are part of the objects. This method was applied to find parts with graphic features that stand out such as a marking. Its application works fine with hidden objects but does not fit for thin parts where the key points do not stand out. Detection of the object to look for depends on its characteristics (shapes, pattern...) and of the number of template images saved in the database. Kumar [Kumar et al., 2014] proposed an algorithm to recognize different objects. The algorithm starts by converting the image into greyscale in order to convert the image in a binary image. Edge detection is performed in order to retrieve the edge of the objects. A double thresholding is performed in order to preserve high value pixels. An edge tracking by hysteresis is performed followed by an image filling of the image dilatation. The algorithm ends by the BLOB analysis. This method allows to retrieve objects with high contrasts. However, the algorithm is not suitable for thin objects where the edge and the threshold would not put forward the object to take. In this paper, we propose a method to retrieve objects where the key points of the images are not present and where the CAD file is not provided due to the long pre-processing time.

3 The Proposed method

In this paper, we propose an approach based on the 3D graph matching. The graphs represent the skeletons of objects in the stereo images in order to recover the position and orientation of thin objects. The aim is to retrieve the location information of the different objects placed in the cameras field of view.

3.1 Binarization and Skeletonization

Algorithms of Binarization (bi-level or two-level) as represented on the image 1b allow to convert an image (1a) with different values of pixels into a binary image. It aims at removing the background elements and to highlight the target objects. The choice of the threshold is performed according to the shape of the histogram (number of pixels by color). The threshold is chosen according to the color of the object to retrieve and the color of the elements to remove like for example the background. Then, a Thinning or Skeletonization algorithm is applied in order to represent the structural shape of a plane region. It reduces the shape to a set of curves called skeleton which are centred to the original shape. Several propositions have been proposed in 3D matching [Cornea et al., 2005]. Topology properties are kept from the original shape. As shown on figure 1c, after having applied the thinning algorithm to the left image 1a, the thin objects stand out clearly on the right image. The fast method used [Zhang and Suen, 1984] remove all the points of the image edge with the exception of the point belonging to the Skeleton. To preserve the connectivity of the skeleton, the iterations are divided in 2 subiterations. In the subiterations, the edge point is deleted according to a set of conditions (numbers of nonzero neighbours, number of pattern...).



(a) Model Image (b) Binarization (c) Skeletonization

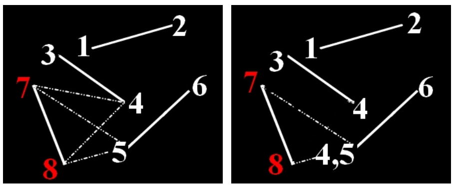
Figure 1: Image simplifications (threshold: [100,255])

3.2 Image Matching by using Graph approach

Graphs are representation tools and are widely used in the field of pattern recognition [Conte et al., 2004, Biasotti et al., 2008] and several methods have been proposed to relate them [Bunke and Shearer, 1998, Champin and Solnon, 2003]. However, in this method we do not attempt to calculate a similarity criterion because these graphs are similar, as it happens with every pair of stereoscopic images in the same plane. The approach has to focus on the best match by using the structural features of the graphs. We define a graph by a two-tuple G=(V,E) where V is the finite set of vertices V={v1,v2,v3...vn} with n=Card(V) (called also nodes) and E is a set of Edges E={e1,e2, e3...em} with m=Card(E). In our case, the nodes represent the endpoints and the connection points of the skeleton image. The edges are defined by the link between the nodes. Firstly, and as shown in the algorithm 1, an image is defined as a reference and each node is labelled. For each node of the labelled picture, the depth and their position and those of the other image is processed. If the depth and the position is included inside the scale defined by the user, then the label of the first node is added to the second one. This selection allows to have a first pre-selection of the graph matching. However, this first selection is performed according to the accuracy defined by the user and a node can have several candidates. To remove the nodes with several candidates, two matched nodes are selected as referent. For each node with several candidates, a distance and an angle is computed between both selected points and the candidates of the nodes. In order not to reduce the method for planar configurations, a threshold can be added to the angle to permit the projective invariant. As shown in figure 2a and 2b, the node 5 in the left image 2a has two candidates in the right image 2b with the possibilities of 4 and 5. Two referents points are selected (7 and 8), in the right picture 2b, angle and distance node is determined by the points 7,8,"4,5" points, and on the left figure 2a both 4 and 5 nodes are processed. The skeleton of the picture is not perfect and can provide some useless node. To avoid this situation, a selection of the node is done by selecting them by the size of their segment. A smooth function can be added to pull out the element already used. This function is used when a node is matched. This function browses the element of matched nodes and removes the taken nodes from the set of nodes. However, if a node is badly matched, then the graph matching will fail.

Algorithm 1: Graph matching

```
input  : (V(G1),V(G2)): G1 and G2 graphs
         nodes set (D_Min,D_Max): Depth interval
         (P_Min,P_Max): Position interval
output: (v.ID,w.ID): Graphs list labels
Integer label = 0;
foreach node v ∈ V(G1) do
    label++;
    v.ID.Add(label);
    foreach node w ∈ V(G2) do
        D ← D_calc(v, w); P ← P_calc(v, w);
        if D_Min<D<D_Max & P_Min<P<P_Max then
            w.ID.Add(label);
        end
    end
end
```



(a) Left Image (b) Right Image

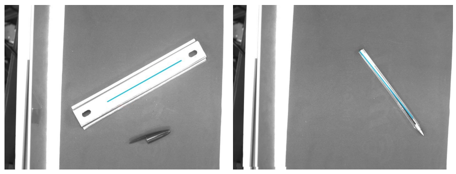
Figure 2: Graph matching between two pictures

3.3 Object retrieval

The link between right and left image allows to correctly calculate the coordinates of the 3D points to give to the robot so he can pick the object. The object extraction step is applied by retrieving known characteristics of the object. Firstly, we choose couples of linked nodes belonging to a same object. Secondly, the object can be found by processing the distance of the desired part. For each node on the right image and their match on the left image, a distance is calculated with the other nodes. We note G1 and G2 the graphs representing respectively the left and the right image. V(G1) and V(G2) are the sets of nodes of the graphs G1 and G2. The equation is $3D = \sum_{i=1}^{k} \sum_{j=1}^{l} \sqrt{(Xi - Xj)^2 + (Yi - Yj)^2 + (Zi - Zj)^2}$ where k=Card(V(G1)) and l=Card(V(G2)) and X Y Z are the 3D coordinates of each node. For each node v∈V(G2), a distance is calculated between v and each other node of the graph. This distance can be calculated in 2D to avoid the system to detect an object with an orientation difficult to catch by the robot. To distinguish objects with several nodes, the characteristics like the number of neighbours with the distance and the angle between them can be processed to highlight specific nodes of the desired object.

4 Experimentation

In this paper, experimentation has been performed on several kinds of objects and with different orientations. Experimental part has been realised with 2 webcams (640*480) placed at 400mm and 2 industrial cameras Cognex IS-5603-10 (800*600) with 16mm 1:1.4 lens placed at 700mm of the scene. The computer has an Intel(R) Core(TM) i7-5700HQ CPU @ 2.70GHz, 2701 MHz. The library OpenCV has been used to calibrate the cameras ([Zhang, 2000]). Zhang algorithm has been used for the skeletonization of the images. Images have been obtained in standard luminosity in order to prove the efficiency of the algorithm. Firstly, the objects have been placed side by side in different angles in order to show that the algorithm finds the object in an easy case. Secondly, the program has been tested with overlapped objects. This situation allows to show that the algorithm can dissociate each object. This situation is the best way to represent a pick and place task because the same object is available in several exemplars in a bin. Thirdly, the algorithm has been tested on an image with overlaps and a combination of different kinds of objects (figure 4). Objects were selected for their colours and shape. The algorithm correctly retrieves the components, placed at different position with overlaps. In the figure 4, it can be seen that the algorithm is able to bypass the problem of tangled objects. The algorithm also shows an efficiency to find non-stick objects (figure 3a) and with transparency (figure 3b). The processing time varies between 600 and 1800 milliseconds for images between 10 and 70 nodes.



(a) Large and long (b) Transparency

Figure 3: Looking for different objects

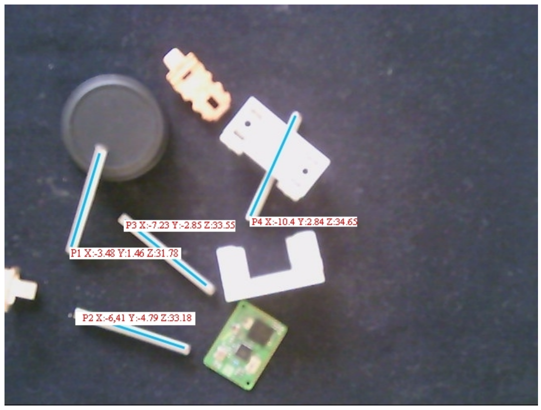


Figure 4: Looking for thin sticks placed randomly (Line: Segment of the objects, Text: Objects position)

5 Discussion

The experimental part allowed to prove that our method can find thin objects in random positions. The use of a thinning algorithm allowed to highlight the objects. Graph matching between the nodes of the right

Image	A	B	C	D	E
Image					
Objects number	4	5	1	2	1
Candidates number	4	5	1	1	1
Nodes number	8	11	313	283	85
Object					
Thin	Y	Y	Y	N	N
Random	Y	Y	Y	Y	Y
Cluttering	N	Y	N	N	N
Transparency	N	N	Y	N	N
Binarization Threshold					
Min	186	186	170	185	170
Max	255	255	255	255	255
Score					
SIFT (%)	50	54	60	92	96
SURF (%)	49	56	74	59	71
Graph (%)	100	100	100	100	100

Table 1: Algorithms performances. Y = Yes, N = No

image and the left image allows to perform the object retrieval step based on the data returned by the user as the distance of the segments and the links between them. The results have demonstrated their efficiencies from several points of views. First, all the objects have been identified under different conditions of orientation, tilt, tangle and colors. The second advantage of the method is to have no pre-treatment time because the algorithm directly retrieves the element part of the object to take. Lastly, the method permits to identify thin objects. A comparison with SIFT and SURF algorithm has been done on the images. Processing time in SIFT/SURF is under 300ms for each image, which is better than our method. However, it is negligible compared to the time taken in the pick and place task. Table 1 shows the percent of correct number of segments belonging to the sticks according to the total number of segments belonging to the sticks. Some features have been found, however some matches are wrong because looking for key points is difficult for this kind of image where the sticks are uniform and featureless.

6 Conclusion and Perspectives

This paper proposes an approach to recognize thin objects in stereovision for pick and place tasks. A skeleton has been built for each object and the mapping problem between the junction points and the extrema was solved by graph matching by removing points where position and depth cannot match. The other points are mapped thanks to the matching based on the angle and the distance between the points of the right image and the left image. The process has been tested with objects placed close to each other, with overlapping and cluttering. All objects have been found. The algorithm can work with different kinds of noises like luminosity and brightness. The advantage of our method is to be quickly initialized because the parameters (shape characteristics, dimensions) are known by the operator. There is also no problem of distortion because the extraction is made directly on the parts of the object of interest. The perspective of our work is to enhance the detection in real time of the object by using for example an evolutionary algorithm.

References

[Berger, 2014] Berger, R. (2014). Industrie 4.0: The new industrial revolution how europe will succeed. *Think Act*, page 8.

[Biasotti et al., 2008] Biasotti, S., Giorgi, D., Spagnuolo, M., and Falcidieno, B. (2008). Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1):5–22.

[Buchholz, 2016] Buchholz, D. (2016). *Bin-Picking: New Approaches for a Classical Problem*, chapter Bin-Picking—5 Decades of Research, pages 3–12. Springer International Publishing, Cham.

[Bunke and Shearer, 1998] Bunke, H. and Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, 19(3):255–259.

[Champin and Solnon, 2003] Champin, P.-A. and Solnon, C. (2003). Measuring the similarity of labeled graphs. In *International Conference on Case-Based Reasoning*, pages 80–95. Springer.

[Conte et al., 2004] Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298.

[Cornea et al., 2005] Cornea, N. D., Demirci, M. F., Silver, D., Dickinson, S., Kantor, P., et al. (2005). 3d object retrieval using many-to-many matching of curve skeletons. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*, pages 366–371. IEEE.

[Ikeuchi, 1987] Ikeuchi, K. (1987). Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. *International Journal of Computer Vision*, 1(2):145–165.

[Kumar et al., 2014] Kumar, R., Kumar, S., Lal, S., and Chand, P. (2014). Object detection and recognition for a pick and place robot. In *Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on*, pages 1–7. IEEE.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

[Piccinini et al., 2012] Piccinini, P., Prati, A., and Cucchiara, R. (2012). Real-time object detection and localization with sift-based clustering. *Image and Vision Computing*, 30(8):573–587.

[Ulrich et al., 2012] Ulrich, M., Wiedemann, C., and Steger, C. (2012). Combining scale-space and similarity-based aspect graphs for fast 3d object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1902–1914.

[Willaume et al., 2016] Willaume, P., Parrend, P., Gancel, E., and Deruyver, A. (2016). The graph matching optimization methodology for thin object recognition in pick and place tasks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8.

[Zhang and Suen, 1984] Zhang, T. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334.