Two Applications of RGB-D Descriptors in Computer Vision

Mariano Bianchi[1(✉)], Nadia Heredia[1], Francisco Gómez-Fernández[1], Alvaro Pardo[2], and Marta Mejail[1]

[1] University of Buenos Aires, Buenos Aires, Argentina
{mabianchi,nheredia,fgomez,marta}@dc.uba.ar
[2] Universidad Católica Del Uruguay, Montevideo, Uruguay
apardo@ucu.edu.uy

Abstract. In this paper an evaluation of RGB-D descriptors in the context of Object Recognition and Object Tracking is presented. Spin-images, CSHOT and ECV context descriptors were used for detecting objects in point clouds. Empirical evaluation over a dataset with ground truth shows that shape is the most important cue for RGB-D descriptors. However, texture helps discrimination when objects are large or have little structure.

Keywords: RGB-D · Object recognition · Tracking · Point clouds

1 Introduction

The study of feature descriptors has been one of the main areas of research in image processing and computer vision for many years. Nowadays, the introduction of new low-cost sensors capable of capturing 3D scene information, has generated attention to descriptors that combine color and depth.

Traditionally, 2D descriptors capture, for example, color, texture or shape information from the image. RGB and depth sensors (RGB-D), such as the Kinect, allow the extraction of color and depth information of the scene. With this data, 3D descriptors can be used to model the geometry and color appearance of objects. The combination of color and depth allows for more discriminative power, making this an interesting area of research.

In this work, 3D point clouds are used instead of RGB and depth images independently. Working directly with 3D point clouds is a recent trend in computer vision; traditionally applications build polygonal meshes and then discard point cloud data. Here we evaluate different RGB-D descriptors in terms of their performance in two important applications: object recognition and tracking.

There are many works about 3D and RGB-D descriptors [7,11,3,4,9] (to name a few) but generally they do not assess their performance in particular applications. The analysis of their impact on the results of two applications, such as Object Recognition and Tracking, is very important and provides new cues and methodologies to understad which descriptors perform better in real applications.

A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 236–244, 2015.
DOI: 10.1007/978-3-319-25751-8_29

In order to quantitatively analyze object recognition and tracking we make use of a dataset with ground truth [8]. This is a large dataset, thus, first we analyze object recognition with three different descriptors and then the best descriptor is used in the tracking application.

This work is organized as follows. In Section 2 we describe RGB-D sensors and point clouds, then in Section 2.1 we talk about 3D descriptors and the datasets used. After that, in Section 4 and 5 discuss object recognition and tracking contributions respectively. Finally, in Section 6 we present a discussion of the presented work along with the conclusions.

2 RGB-D Sensors and Point Clouds

RGB-D sensors are composed of an RGB camera that captures color information of the scene, and a depth sensor which provides information about the distance from the camera to the objects in the scene.

There are different implementations for this kind of sensors to choose from, each one having its own unique sets of characteristics: image quality, frame rate, depth range. One that has gained popularity in recent years is the Kinect by Microsoft which provides RGB-D images with a default resolution of 640x480 pixels at rate of 30 fps.

Depth information used in conjunction with color information (RGB) allows 3D scene reconstruction. The calibration parameters for the RGB-D sensor can be used to estimate each pixel's position in the 3D scene, forming what is called a point cloud, where each point contains depth and color information.

2.1 Evaluated RGB-D Descriptors

In this work, we use local descriptors for object recognition because they are more robust to noise, clutter and occlusion, typical problems that arise when working with real RGB-D data. We use three histogram-based descriptors: CSHOT [11], Spin-Images [7] and ECV context descriptors [4], each one capturing information about the underlying surface in a different way. The selection of the previous descriptors takes into account the results presented in [1], where the author concludes that CSHOT and histogram-based descriptors are well suited for object recognition.

All descriptors are built for a set of selected input points but they use the entire point cloud to create them.

CSHOT (Color Signatures of Histograms of Orientations) [11]. The SHOT descriptor, first defines a robust local reference frame that is unique and unambiguous, built upon a local sign disambiguation method for the problem of normal estimation. At each input point a 3D grid is superimposed defining 3D volumes where local histograms are computed. Each local histogram accumulates into bins the angles (cosines) between the reference frame and the normal at each point. The Color SHOT (CSHOT) descriptor is an enhanced version of SHOT

that also accumulates in a histogram the absolute color difference between points and then is chained together to the SHOT descriptor itself.

Spin-Images [7]. A Spin-Image (SI) is a cumulative 2D histogram which is computed on each oriented point (3D position plus surface normal) over a given set of points over the model's surface mesh or point cloud. At each input point a 2D histogram is built binning neighboring points that are projected on a plane (image) passing through the point's normal. Each bin is defined as a ring product of spinning that plane.

ECV Context Descriptors [4]. Early Cognitive Vision (ECV) systems were first developed with the aim to understand how human visual and cognitive systems work. Image pixels are classified as belonging to either an edge or a texture region and accompanied with an appearance and geometry-based description of their spatial neighborhood. At each input point all point-pair relations are considered, estimating geometric (cosine between orientations) and appearance (color gradients) relations and then binning them into histograms. The geometry part, the 3D position and orientation, is used, where the orientation are the direction along the edge or the normal vector to the local surface.

3 Dataset

Along this work we used sequences of RGB-D images with *ground truth* information with the goal of testing the studied methods and creating a reference to compare and assess the performance of the RGB-D descriptors under evaluation. We took scenes of a database from Kevin Lai et al. [8] where objects and scenes databases were created. Since our work is focused on real world scenarios, we took 3 annotated scenes from the dataset containing 10 different objects. Figure 1 shows the sequences employed for the results. For the objects we construct 3D models from their partial views acquired in a turning table, see Figure 2 for some of them. Note that we discard the flashlight due the impossibility of building the 3D model from its views.

4 Object Recognition

An object recognition system has the purpose of finding objects in images or point clouds. From now on, we will call *model* to the point cloud which has the object to be searched, and *scene* to the point cloud in which we wish to find it.

We divide our object recognition pipeline in the following steps: 1. Keypoint Extraction, 2. Descriptor Generation and 3. Matching. An overview of the steps of the employed procedure to recognize objects is shown in Figure 3a.

1. **Keypoint Extraction:** a subset of points from the original set is extracted. Keypoints are interesting points which are repeatable, there is a high chance that the same point is found in the same scene with different data-acquiring methods. Also, since they are distinctive, they are very useful for achieving

Fig. 1. Example frames taken from the sequences of the dataset.

Fig. 2. Object models built from partial views.

an effective recognition. This point selection can be done using a keypoint detector, or by simply subsampling the points and using the resulting subsampled set as keypoints. In this work, we use subsampling every 1 cm for the model and the scene.

2. **Descriptor Generation:** for each keypoint in the model and the scene, RGB-D descriptors are computed, representing object's information in a distinctive and meaningful way. In our work, we use SI, CSHOT and ECV, described in Section 2.1.
3. **Matching:** descriptors computed for the model and the scene are compared in order to establish associations between them. A match between a pair of descriptors can translate into a match between the corresponding points. Point matches can then be used to determine the roto-translation needed

to align the model with its instance in the scene. To find a match, vector distances are used, such as $\mathcal{L}_1$ and $\mathcal{L}_2$ norms.

For the matching stage, we compare all pair of descriptors using $\mathcal{L}_1$ norm and then we impose a threshold over the ratio between the first and the second nearest match.

5 Tracking

Several ways of tracking objects in 3D images are known. In [10] the well known Iterative Closest Point (ICP) is used for this goal, originally proposed in [12,2]. Then, they improve the obtained results using edge information taken during the training phase. In [5] a frame by frame object tracking is proposed based on edge detection.

A video tracking system can be divided into three main different stages (see Figure 3b):

1. **Training:** consists in obtaining a model or representation of the object to follow.
2. **Object Detection:** using the object model obtained in the previous stage, the main goal is to detect the object in a video frame and report its position in a previously defined coordinate system. It is used as a starting point in the video tracking system and when frame by frame tracking fails.
3. **Tracking:** using the object location in the previous frame, the goal is to track the object frame to frame. This is the most important stage because it is used in most of the video frames. The efficiency of this method will determine the system efficiency.

We use this pipeline because it allows us to test different methods for each stage without much effort and is a modular way of treating the tracking problem.

We choose different methods for each of the proposed stages. First, we reconstruct a complete 3D model of the object to track merging several views of it taken from different angles and all around of it, using an ICP based method. Reconstructions obtained using this method are then improved manually. Figure 2 shows an example of some of the constructed models in the dataset.

Once the model is built, we use the method explained in Section 4 to detect it in the first frame of the scene. Once a matching between the model and the scene is found, we use a KD-Tree to filter the nearest object points from the scene's point cloud. The points obtained by this filter are then used as the model to track in the next stage. This is done to increase robustness in tracking.

The most important features of the tracking stage are efficiency and robustness. The selected method needs to be computationally efficient because the system needs to be fast. Thus, it has to be robust because otherwise we will need to use the object detection method and that would be time consuming, losing the overall efficiency.

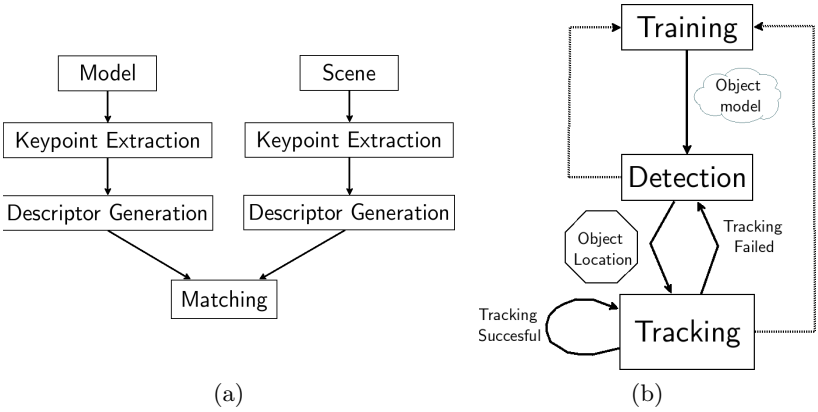

Fig. 3. Diagrams showing: (a) object recognition method, (b) tracking method.

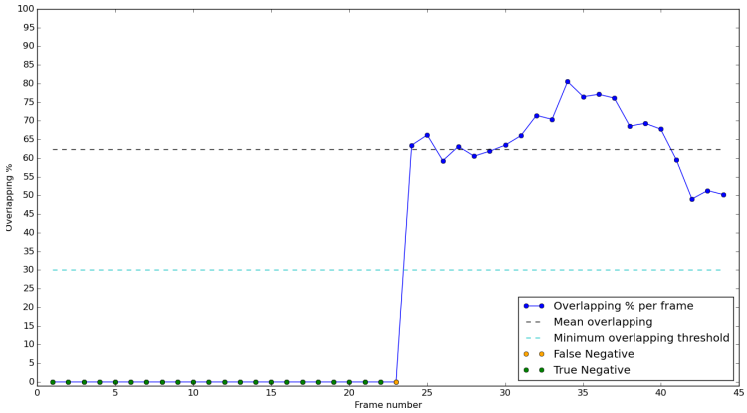

Fig. 4. Frame by frame overlapping analysis. Object: coffee_mug_5. Scene: desk_1. The object to track shows up at frame 24.

In order to have an efficient method to track objects frame by frame it is natural to think of an ICP based method. To apply this method, we take the result from the object recognition stage and take a bounding box twice as big as the detected one. Then apply ICP between this two point clouds. If there is no match, we go back to the previous stage and apply object detection. Otherwise, if a match is found, we filter scene points using a KD-Tree as mentioned before and continue tracking.

In Figure 4, we show an analysis frame by frame of tracked object. We extract the object's bounding box and then compute the overlap between the ground truth bounding box. In order to measure the quality of the computed object area of our method we use a formula taken from [6]. This measure punishes reported areas smaller than or bigger than the ground truth area. We show how our method obtains a mean percentage overlapping of 62.34% in this example.

6 Discussion

Object recognition and tracking are two applications that heavily depend on finding good representations for the object models to recognize and track. In [1], the author presents a study of several RGB-D descriptors over the RGB-D object dataset, finding that CSHOT is the best option for object recognition. In this work we show that SI outperforms CSHOT, which was excluded from their analysis. As we can see in Table 1, SI outperforms CSHOT and ECV in 10 out of 12 cases.

Results show that SI describe the shape of objects in a better way than CSHOT and ECV in spite of having only shape information, mainly due to the characteristic of the objects present in the dataset. Mostly they are small and have low texture, i.e. just one or two colors.

We also carried out several tests varying parameters and methods and show that subsampling is faster than using Harris detector over point clouds. Also, reducing descriptor support and normal radius decreases computational time but increases the time spent on matching correspondences.

Note that the ground truth provides only the bounding box information, so the results are not extremely precise.

In order to achieve a successful tracking over frames, an initial good pose of the object model is required.

Our tracking application achieves low false positives values (see Table 1) resulting in a very robust method. Depth information complements RGB and improves the overall tracking system accuracy. This shows the potential of using RGB-D descriptors in computer vision applications.

Object size influences both recognition and tracking. On one hand, when the object to be tracked is near the sensor the tracking accuracy increases.

Table 1. Columns three to five show the object recognition accuracy for three different descriptors SI (Spin-Images), CSHOT and ECV. Last columns show the tracking results (Tracking Accuracy) and false positive rate (% FP).

		Recognition			Tracking	
Object	Scene	SI	CSHOT	ECV	Accuracy	% FP
cap_4	desk_1	**93.90**	81.71	80.89	97.73	0.00
coffee_mug_5	desk_1	**93.18**	**93.18**	87.12	19.51	40.24
soda_can_6	desk_1	**91.67**	78.33	50.00	58.33	0.00
bowl_3	desk_2	**73.13**	**73.13**	59.38	66.88	3.13
soda_can_4	desk_2	**66.03**	53.85	27.88	19.87	30.13
bowl_2	table_1	**89.91**	86.24	65.14	36.90	0.00
cap_1	table_1	**95.26**	59.48	50.86	55.83	3.07
cap_4	table_1	**70.62**	68.04	47.42	76.61	0.00
cereal_box_4	table_1	81.09	**91.18**	54.58	70.93	2.33
coffee_mug_1	table_1	**44.05**	**44.05**	54.76	48.97	5.15
coffee_mug_4	table_1	**68.10**	65.03	43.56	53.36	0.00
soda_can_4	table_1	46.15	**49.45**	34.07	32.97	9.89

On the other, when the object appears small in the captured image it contains less points to be used for recognition and tracking; less points translates into less discriminative power.

Besides the analysis of the computational performance was not the main focus of this work we found that working with point clouds acquired from commercial RGB-D sensors is mandatory to tune methods and theirs parameters to deal with the noise present in RGB and specially in depth data.

7 Conclusions

Based on our empirical evaluation we showed that shape it is the most important cue for recognizing and tracking 3D objects. Texture plays an important role when objects are large or have simple structure.

In particular, objects from the RGB-D database [8] tend to have very low texture, just one or two colors, making it necessary to have a descriptor that focuses more on the shape of the objects. Our results showed that SI perform better than CSHOT and ECV in spite of having only shape information.

Also, we observed that RGB-D descriptors that work well with object recognition not necessary perform in the same way in a tracking application.

In order to achieve robust tracking we saw that a good initial pose is required and a good object model for the detection phase.

References

1. Alexandre, L.A.: 3D descriptors for object and category recognition: a comparative evaluation. In: IEEE/SRJ Int. Conf. on Intelligent Robots and Systems (IROS), Workshop on Color-Depth Camera Fusion (2012)
2. Besl, P., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence **14**(2), 239–256 (1992)
3. Bo, L., Ren, X., Fox, D.: Depth kernel descriptors for object recognition. In: Intelligent Robots and Systems (IROS), Int. Conf. on. pp. 821–826 (2011)
4. Buch, A.G., Kraft, D., Kamarainen, J.K., Petersen, H.G., Kruger, N.: Pose estimation using local structure-specific shape and appearance context. In: Int. Conf. Robotics and Automation (ICRA), pp. 2080–2087 (2013)
5. Drummond, T., Cipolla, R.: Real-time tracking of complex structures with on-line camera calibration. In: BMVC, pp. 1–10 (1999)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. Jour. of Computer Vision **88**(2), 303–338 (2010)
7. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 433–449 (1999)
8. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1817–1824 (2011)

9. Nascimento, E.R., Oliveira, G.L., Campos, M.F.M., Vieira, A.W., Schwartz, W.R.: Brand: A robust appearance and depth descriptor for rgb-d images. In: Int. Conf. on. Intelligent Robots and Systems (IROS), pp. 1720–1726 (2012)
10. Park, Y., Lepetit, V., Woo, W.: Texture-less object tracking with online training using an rgb-d camera. In: Int. Symp. Mixed and Augmented Reality (ISMAR), pp. 121–126 (2011)
11. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010)
12. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision **13**(2), 119–152 (1994)