AN INTEGRATED SYSTEM FOR OBJECT TRACKING, DETECTION, AND ONLINE LEARNING WITH REAL-TIME RGB-D VIDEO

I-Kuei Chen[⋆] *Chung-Yu Chi*[⋆] *Szu-Lu Hsu*[⋆] *Liang-Gee Chen*[†]

DSP/IC Design Lab, National Taiwan University, Taiwan

[⋆]{eugenegx, craig, bismarck}@video.ee.ntu.edu.tw [†]lgchen@video.ee.ntu.edu.tw

ABSTRACT

This paper introduces a highly integrated system providing very accurate object detection with RGB-D sensor. To solve the problem that there are always insufficient training sets for object detection in real world, we present an online learning architecture to learn templates and to detect objects real-time. The proposed novel concept skips the training phase required in previous recognition works, and it comprises independent tracking and detection function, which collaborates with each other to make the detection more precise. We furthermore illustrate four strategies for online learning and compare the efficiency. With depth information, the experiment results perform remarkable in challenging scenarios.

Index Terms— Online learning, tracking, detection, real-time, RGB-D

1. INTRODUCTION

In the field of computer vision, object detection, tracking, and recognition have become challenging and important tasks for a long time. State-of-the-art algorithms perform good results for above applications in 2D computer vision. However, those methods still suffer from limited variation and cluttered backgrounds. Since the prevalence of RGB-D sensor, many algorithms with depth have been developed to overcome the disadvantages of 2D computer vision.

Since there are millions of different objects in the real world, it is very hard to get the training data for detection and recognition. Even with the help of the Internet, it is still nearly impossible to get the complete data needed for a specific instance object. Furthermore, the huge variations in illumination, size, and rotation of the object also makes the training data insufficient when dealing with the real world object detection problems. Therefore, the proposed online learning function can use the object tracking information introduced as reliable positive labels, and the positive templates of the object can be learned online in real-time to generate a reliable, robust, and adapting object detector(Fig. 1). The users can manipulate the camera to interact with the object to learn the appearance of the object in different situations, and the object can thus be tracked and detected in real-time.

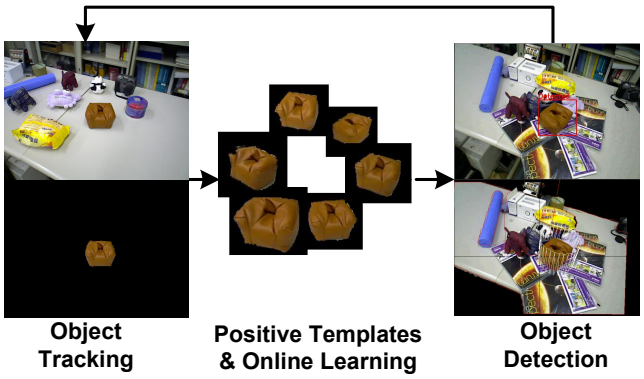


Fig. 1. The concept of our highly integrated system comprising tracking, online learning, and detection.

The rest of the paper is organized as follows: Section 2 shows the similar previous works. Section 3 describes the details of proposed system. In Section 4 we illustrate four strategies for online learning, and do experiments to compare the results in Section 5. Finally we conclude this work in Section 6.

2. PREVIOUS WORKS

Previous 2D detection and recognition methods fail in real world situations, as they cannot deal with illuminance change or printed version of targets very well [1]. Because they hugely rely on 2D visual image appearance and the lighting and rotation influences the detection results heavily, it produces false positives on printed version of the objects, However, using depth information may solve some of the problems since 3D data provides real world physical characteristics, so combining depth data with RGB for detection brings more robust results.

Another severe drawback of previous 2D methods is that they cannot detect textureless objects. All feature points methods like SIFT [2] or SURF [3] are only suitable for complex patterns or complex textured targets, which have to be visually complex in order to generate feature points on them [4]. However real world detection tasks are required for textureless objects, so depth data with surface normals provides

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

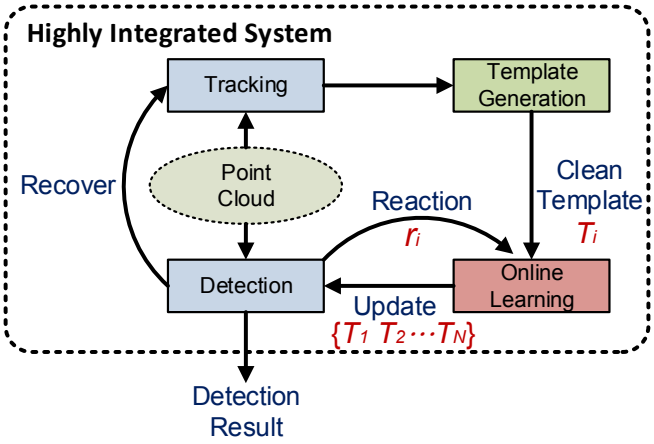


Fig. 2. Details of the proposed highly integrated system and the dataflow.

more potential in dealing with those issues.

Furthermore, the patterns in current 2D object recognition databases are sometimes too simple and lack of variations in different conditions [5, 6], and thus the detection in real world may fail due to huge visual difference. Even provided with database in the wild [7], current visual learning algorithms have difficulties in separating and locating targets to learn. Actually, current 2D object recognition methods are successful only in certain clean database without complex backgrounds, and in many real world situations they fail frequently. Thus, those 2D object detection methods become impractical in real world object recognition applications.

One solution to the problems about insufficient datasets is a work called TLD(Tracking, Learning, Detection) [8]. It combines tracking, learning, detection in 2D video, and it uses the tracking results as training data to online train the detector. [9, 10] basically use a complete offline object dataset for detection, and the dataset contains segmented appearances of objects in different angles. However, the proposed online learning can freely train object detectors on arbitrary objects whenever needed, and the operating environments is more casual in terms of object position, camera angles, and plane property, compared to the rigid setting of RGB-D datasets.

3. PROPOSED SYSTEM

With the help of PCL [11], we construct highly integrated system as Fig. 2. The point cloud in the current frame is delivered to both tracking and detection functions to perform accurate detection. Tracking produces templates, while detection can work robustly even in cluttered scenes, and thus it can be used to restart tracking function if needed. The ensemble position information from detection and tracking also makes the object recognition more robust. The trained object detector and positive template datasets can be stored and transferred to many different object recognition applications.

3.1. Template Generation by Object Visual Tracking

When the object tracking function is in action, the object is tracked and segmented out from the background to generate clean templates without background noises. Based on [12], the conditions for successful tracking is that the objects are physically separated and on the main plane, and the main plane needs to be large enough in the current frame to be detected. The tracked object segmentation is then used as the positive templates for the object detector to learn in an online fashion, and the detector can detect the known pattern of the object. For unknown appearances of the object, it relies on tracking to label it and then learn it. The trained final object detector can work alone in a more severe cluttered environments, but tracking cannot be used in those harsh conditions.

3.2. Template Matching by Similarity Measure

Color gradients and surface normals are used for template matching. The online learning function extracts these two features from each positive learned templates. And the RGB-D data of the current frame is also extracted with the two features for similarity measurement. The template matching is done in each window of the frame, as proposed in [13, 14, 15]. Currently, the assumption is that only one object is for detection in the scene, and thus the window with highest similarity is considered detected. If no window has a similarity score higher than the threshold, then no object is detected in the current frame.

$$I_G(p) = \underset{C \in \{R,G,B\}}{\arg\max} \left\| \frac{\partial C}{\partial p} \right\| \tag{1}$$

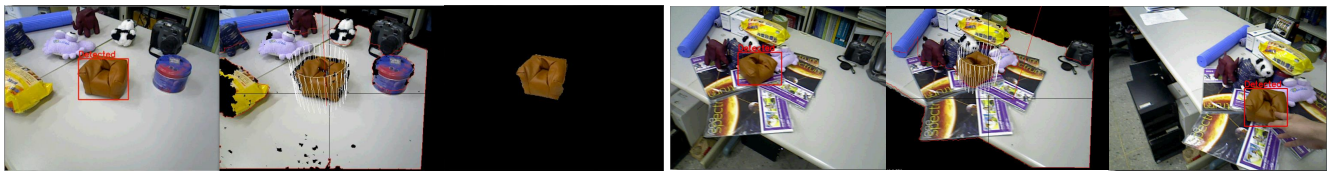
$$Sim_G(T_G(r), I_G(p)) = \left| T_G(r)^T I_G(p) \right| \tag{2}$$

$$Sim_N(T_N(r), I_N(p)) = \left| T_N(r)^T I_N(p) \right| \tag{3}$$

Color gradients I_G is extracted from color images I for gradients in each RGB channel, and the highest value in three channels R, G, B is considered as representative of the gradient, as Eq. (1). p is a position in color image, which can correspond to a point in point clouds. The similarity measure for color gradients Sim_G and surface normals Sim_N are shown in Eq. (2) and Eq. (3). It's basically the absolute inner product between two gradients or normals, and the pattern matched will show high similarity. For example, $T_G(r)$ is the color gradient from the template T at position r, and $I_G(p)$ is the color gradient from the color image I at position p.

3.3. Efficient Object Detection

No plane detection or physical separation between objects is needed in object detection since it is only a pattern matching with learned templates. The tracking results of the specific object are sent into the template generation function as positive training data to perform online learning for the target



(a) (b)

Fig. 3. (a)Tracking and detection co-recognition for the targeted object. (b)The detection works in the cluttered background, and it can tolerate human interaction.

object, and the boundary of the object can be used on color data and depth data to generate features needed for the template. LINEMOD [13, 14] feature is fast and efficient enough for the object detection as long as proper positive templates can be provided by tracking.

4. STRATEGIES FOR LEARNING

The online learning block is the most important function of our system. With tracking, the temporal continuity information can be used to make sure that these clusters are the same object [8], though their visual appearance may vary dramatically due to sizes, viewing angles, and illumination changes. The template of the objects contains the segmented color image, the segmented depth, the segmented point cloud, and the corresponding surface normals. As long as the tracking is reliable and robust, each tracking result can be used for learning. However, different sampling methods can be used for choosing the specific templates to learn from all tracking results.

With the tracking function consecutively generating clean templates of the target objects, the learning mechanism decide whether new templates should be learned in concern with the number of templates and detection accuracy. Inspired by [8, 16], we observe two factors that impact learning most: whether the detection function react, and how similar between new templates and learned templates. Referring to Fig. 2, we define four strategies for learning as follows:

1. **Initial**: Learn few templates from initially tracked object. T_i represents the i-th successfully segmented template. As Eq. (4), if $Learn(T_i)$ is true, then we keep T_i and update the number of learned templates N. Otherwise we discard T_i. This method saves the initial states of the object, so $N_{\max}$ is set to be less than 5.

2. **Learn All(LA)**: Learn all templates from successfully segmented results as Eq. (4). To avoid missing, learn every aspect of templates from tracking regardless of what have already learned. $N_{\max}$ is set to maximum under memory limit.

$$Learn(T_i) = \begin{cases} \text{true}, & \text{if } N < N_{\max} \\ \text{false}, & \text{otherwise} \end{cases} \tag{4}$$

3. **Learn No Detect(LND)**: Learn templates only if the detection does not react simultaneously(Eq. (5)). If the object is not detected, the segmented result may have large difference with learned templates. Thus we append new template to our model for updating the detection function.

$$Learn(T_i) = \begin{cases} \text{true}, & \text{if } r_i = \text{false} \\ \text{false}, & \text{otherwise} \end{cases} \tag{5}$$

4. **Learn No Detect or Different(LNDD)**: Learn templates if detection does not react or the segmented result exceeds the threshold δ of similarity between all learned templates and it (Eq. (6)). In case that the object has been detected but the tracked template is far from all learned template, we learn new template to perform more robust result.

$$Learn(T_i) = \begin{cases} \text{true}, & \text{if } (r_i = \text{false}) \quad or \\ & (\forall n \in N, Sim(T_i, T_n) > \delta) \\ \text{false}, & \text{otherwise} \end{cases} \tag{6}$$

5. EXPERIMENT RESULTS

The proposed system runs on a 3.40GHz CPU. Fig. 3(a) shows the snapshots of tracking and detection co-recognition for the targeted object. After the online training is completed to a certain level, the object detection can be used in a highly cluttered background, and this kind of situation is very difficult for 2D computer vision methods. As shown in Fig. 3(b), the hand interaction with the object can also be included since the detection can tolerate partial occlusions. The detection works in challenging scenarios like moving camera, object disappearance, and object scale change. Almost all hand-held objects with proper can be trained online by using the proposed online training methods, and typically only few templates are needed for a certain angles of the target.

We test our system with 7 sequences, generated by an RGB-D sensor with 640×480 resolution (Fig. 4), to prove the robustness and accuracy. We also compare the memory consumed and efficiency for four learning strategies proposed in Section 4. Each sequence is about 20 seconds, containing 9 different objects in several challenge scenarios. It consumes totally about 0.16s per frame, so it's feasible for real-time RGB-D applications.

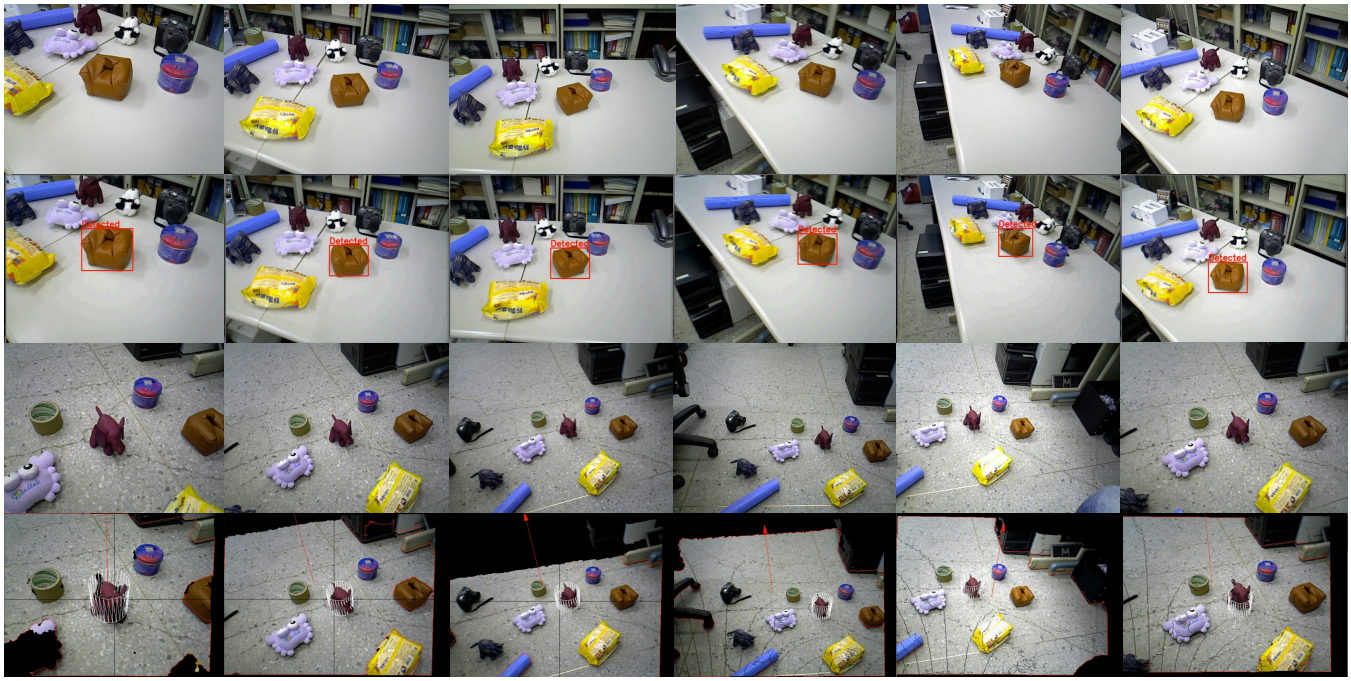
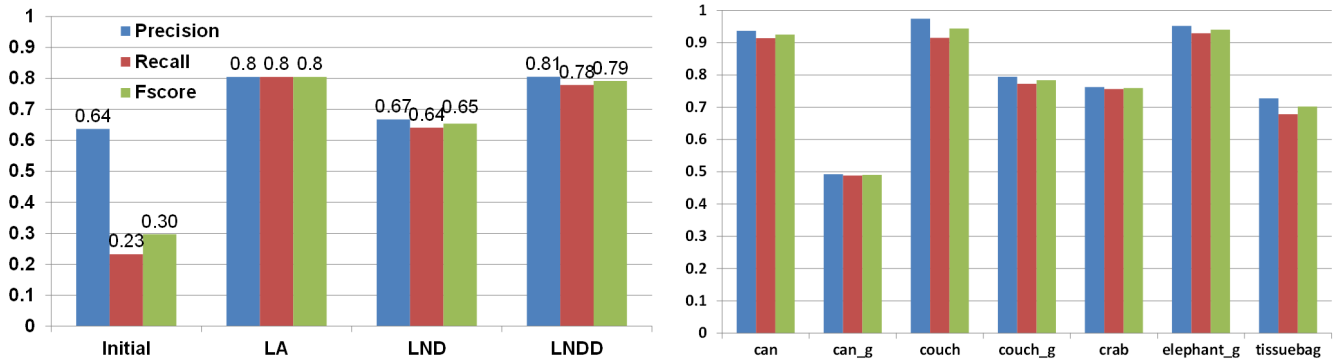


Fig. 4. Sample frames and detection results of sequences *couch* and *elephant*$_g$(#1, #16, #31, #46, #61, #76).



(a) (b)

Fig. 5. (a)Accuracy comparison of 4 learning strategies. (b)The detection results for 7 sequences with LNDD learning strategy.

Table 1. The averaged number of templates saved in memory with 4 proposed learning strategies.

Templates	Initial	LA	LND	LNDD
Learned	3.0	87.6	3.4	15.1

Table 1 and Fig. 5(a) are the average result of detection in overall 7 sequences. It indicates that LNDD strategy leads to the same performance as LA with 82.8% less templates required. We get the centric point $P_d{=}(x_d, y_d)$ of each detection box and locate the golden 2D center position $P_g{=}(x_g, y_g)$ of each object by human labeling. By measuring the precision (proportion of true existence in detected results), recall (proportion of frames where object is detected in overall frames where object exists), and F-score$=\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$, it can be proved that LNDD has great performance in 3D object detection (Fig. 5(b)).

6. CONCLUSION

We develop a highly integrated system for object detection, tracking and online learning with RGB-D information, and can be easily operated with low cost compared with the techniques requiring strictly controlled environment in laboratory. It uses successfully tracked templates as positive training data to perform online learning for the targeted object. With the help of depth information, features like color gradient and surface normals are used to precisely detect the desired target object in challenging indoor scenarios. We propose four strategies for learning and conclude that LNDD is best in consideration of both memory and accuracy. Most importantly, the trained object detector can be exploited in many real-time applications in real world.

7. REFERENCES

[1] Radu Bogdan Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[2] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417. Springer, 2006.

[4] Qualcomm, "Image targets — vuforia developer portal," `https://developer.vuforia.com/resources/dev-guide/image-targets`, 2013, [Online; accessed 04-November-2013].

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[6] Gregory Griffin, Alex Holub, and Pietro Perona, "Caltech-256 object category dataset," 2007.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[9] K. Lai, Liefeng Bo, Xiaofeng Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1817–1824.

[10] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox, "Sparse distance learning for object recognition combining rgb and depth information," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4007–4013.

[11] R.B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.

[12] I-Kuei Chen, Szu-Lu Hsu, Chung-Yu Chi, and Liang-Gee Chen, "Automatic video segmentation and object tracking with real-time rgb-d data," in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, 2014.

[13] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 858–865.

[14] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 876–888, 2012.

[15] I-Kuei Chen, Chung-Yu Chi, Szu-Lu Hsu, and Liang-Gee Chen, "A real-time system for object detection and location reminding with rgb-d camera," in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, 2014.

[16] V. Nair and J.J. Clark, "An unsupervised, online learning framework for moving object detection," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 2, pp. II–317–II–324 Vol.2.