Enabling User Interactions with Video Contents

Khalad Hasan†, Yang Wang†, Wing Kwong‡ and Pourang Irani†
†Department of Computer Science ‡C3A
University of Manitoba, Winnipeg, MB Winnipeg, MB
{khalad, ywang, irani}@cs.umanitoba.ca wkwong@c3a.ca

***Abstract*—Many people spend countless hours watching videos online or on TV. Current smart TV systems provide some basic two-way communications where users can interact with some features (e.g. browsing web, accessing social media, etc) provided by service providers. We would like to move beyond such primitive interactions and explore the possibility of allowing users to interact with *video contents*. For example, users can select objects shown in videos and place further queries on them. We start with exploring different state-of-the-art object detection and tracking techniques to obtain an object's location in the video. Using the best performing tracking technique, we extract an object's location in each frame and allow users to interact with the object using Microsoft Kinect. Finally, we have developed and compared a set of selection techniques that assist users to select moving objects in video. We conclude with guidelines for designing such interaction systems.**

***Keywords*-object tracking; video contents; video-based human computer interaction**

I. Introduction

Imagine that you are watching TV in your living room. Some item (e.g. a dress, perfume, appliance, etc) on the screen catches your eyes, and you want to find out where you can buy it. Wouldn't it be wonderful if you could simply aim the remote at the object, press a button to select it while the TV program is still progressing? Then the TV will automatically send the information to a search engine and tell you where to purchase this item. In this paper, we develop a system representing our first step toward realizing this dream.

Videos are becoming one of the most popular and powerful communications tools nowadays. With the rapid growth of the Internet, people rely on videos for a wide variety of daily activities such as education, entertainment, advertising or even business. According to Youtube [30], over four billion hours of videos are watched and over 800 million unique users visit the site every month. In addition to online videos, television is another popular media that delivers video contents to users. Over the last few years, smart TVs are gaining significant attentions as they provide services with more interactive viewing experiences compared with traditional TVs. For instance, it provides two-way communications where users can directly interact with different features provided by the service providers, such as accessing the web, social media and online applications.

However, the interactions provided by current smart TV systems are still quite limited. In particular, users do not have the option to interact with the *video contents*. For example, users cannot identify an object appearing on the screen and place further queries on the object while the video is still progressing in the background. This form of new interactions can potentially revolutionize how companies market, advertise, and sell their products.

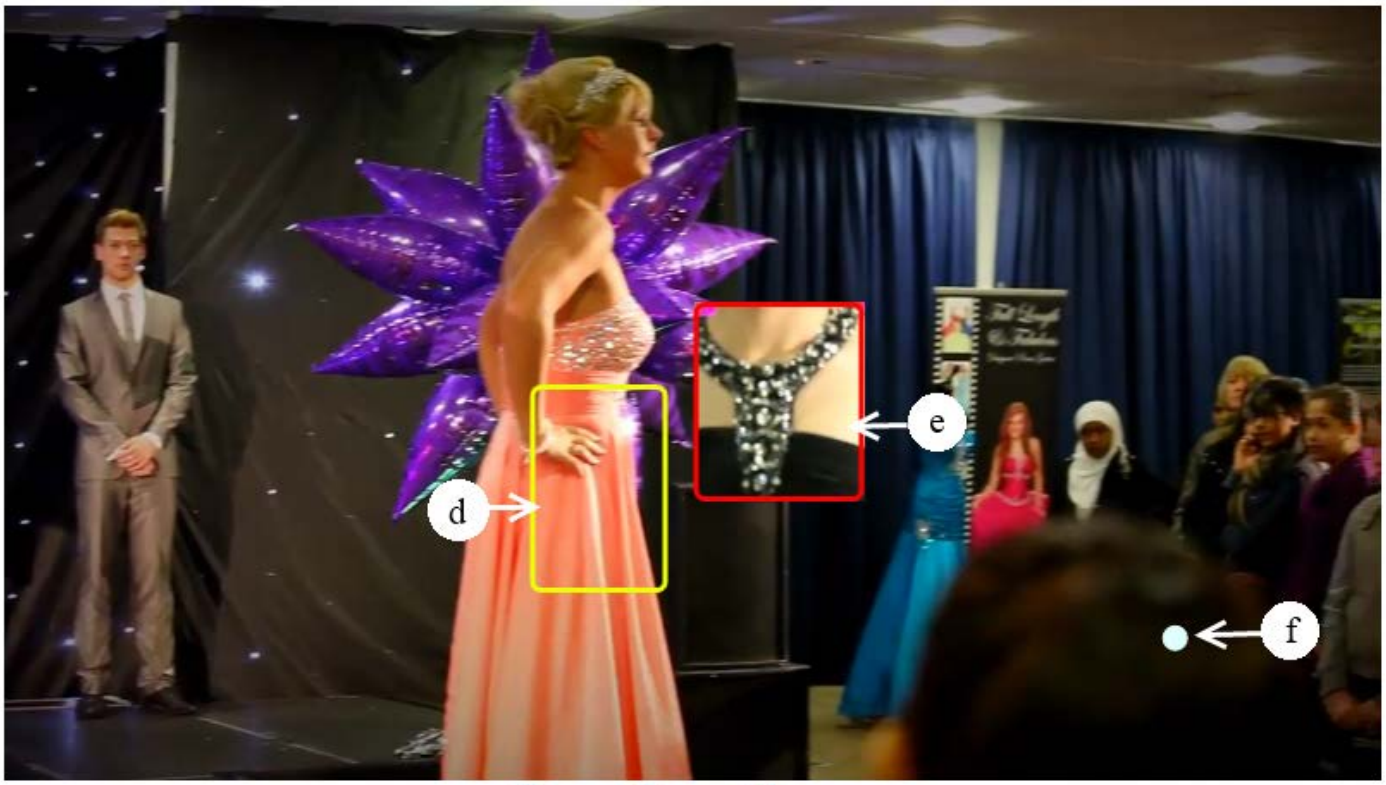
In this paper, we develop an interactive system that enables users to directly select objects from the video using their hand gestures. In our system, users' hand movements are mapped to the cursor control and used to select the object. Once an object in a video scene is selected, users can perform basic queries (e.g., product name, price, store information) on the object.

To achieve our goal, we first investigate and compare different object tracking and detection methods that provide object positions in the video frames. We then apply the best found method to track objects and use Microsoft Kinect to allow users to interact with those moving objects. Selecting a moving object in a video involves continually tracking the object and simultaneously planning to move the cursor over it. We develop a set of selection techniques that use the hand gestures and depth information from Kinect sensors for object selection. We have compared these techniques and found that object selection is best done when both hands are used (right hand for cursor control and left hand to confirm the selection). In addition, our results have revealed that selecting items from static proxies is faster compared with selecting moving objects.

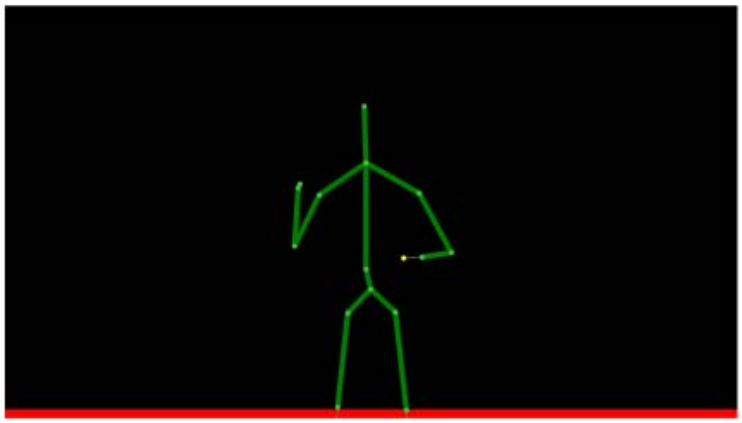
The contribution of this work include: 1) an evaluation of different object detection and tracking methods for video content interaction; 2) object position extraction from videos by applying the best tracking method; 3) design of a system that allow users to interact with video contents; 4) a set of techniques that assist in moving object selection in videos.

II. Related Work

In this section, we review previous work in several areas related to our work, including object tracking (Section II-A) and kinect sensor (Section II-B) in computer vision, and moving object selection in HCI (Section II-C).



(a) Video with tracked object



(b) User Skeleton

(c) Selected item

Figure 1. An illustration of our video content interaction system. (a) presents the original video where a user (b) selected an item (c). Current tracked object is shown in (d) and the static proxy of the selected item is shown in (e) with the cursor (f). **A video demonstration of the system is available online [32]**.

A. Object Tracking

Object tracking is one of the most extensively studied areas in computer vision. Interested readers are referred to [28] for an extensive review. Here we briefly mention some of the work most relevant to ours.

Tracking-by-detection has gained significant attentions in recent years [1]. The basic idea of this approach is to treat tracking as a sequence of detection problems and repeatly apply an object detector on each individual frame. One of the limitations of this approach is that the detector needs to be trained beforehand.

Adaptive tracking-by-detection is another popular method used in [1], [7] where classifier is trained online. During the initialization, a initial classifier is used to search for the object position in the frame using a sliding window approach. This generates a new set of training samples which are used to update the classifier. Examples include the semi-supervised boosting method in Grabner et al. [10], and the multiple-instnace learning method in Babenko et al. [2]. Kalal et al. [21] propose a robust method called Tracking-Learning-Detection (TLD) that combines tracker, learner and detector to discover different appearances of the object and to detect it in a frame. They proposed a new learning paradigm called P-N learning [20] that generates positive and negative constrains in runtime which enforces the labeling of the unlabeled frame.

B. Kinect

Kinect is the first motion capture device on the consumer market and was originally developed as a game controller. It has now been used in a wide variety of applications ranging from art and advertisement to healthcare and business [29]. In the research community, this device has been used for 3D reconstruction, augmented reality, image processing, interaction and visual recognition [8].

3D reconstruction using Kinect has been a very popular research topic [3], [17], [23]–[25] in the last few years since Kinect is a low cost and handheld device. It is capable of capturing objects' geometry and colors of a scene in real time. This 3D reconstruction can be applied in several applications in including object synthesis, augmented reality, robotic navigation, image processing, etc [8]. Kinect cameras have also been used extensively in various application in computer vision, such as unsupervised feature learning [4], people detection [27], object detection [22], human pose estimation [26], etc.

C. Moving Object Selection

According to Fitts' law [9], the time it takes to select an object depends on (i) the effective width of the target and (ii) the distance between the cursor and the target. In recent years, researchers have studied object selection techniques by modifying these two properties to improve the target selection performance [11], [13]–[15], [18]. Area cursor [18] and comet [13] improve the targeting performance by increasing the effective target width. Instead of using a small activation area (i.e., point cursor), area cursor uses large width thus provides a larger activation area. Comet also works based on the principle of increased activation area where a tail (that is seen with comet in the sky) is added to every object. One major limitation of these approaches is that the enlarged area yields more overlaps or causes visual distraction. To reduce this problem, bubble cursor [11] used a dynamically resized activation area that changes its width based on the location of surrounding targets to the position of the cursor. This cursor technique is known to be well performed under different circumstances.

In general, positions of a moving object in the subsequent frames are uncertain. To reduce the uncertainty, Ilich [16] proposed a technique called click-to-pause where entire scene is paused with a mouse click and thus allow users to select a static object. It is not a suitable solution for real-time applications as pausing the entire scene would hide frames that are running in background, thus viewers might miss other important information. Target ghost [13] overcomes this problem by creating static proxies of all moving objects on their position at time of invocation. In a user evaluation, authors showed that the ghost technique results in lower selection times and less error for a task involve object selection.

III. Comparison of Tracking Methods

Our system requires tracking objects in a video as the first step. In our work, we consider two state-of-the-art tracking methods: the tracking-learning-detection (TLD) method in [21] and the structured output tracking with kernels (Struck) in [12]. Both methods have been shown to be effective. But there is no direct comparison between these two methods. In this section, we first briefly summarize these two tracking methods, then perform a quantitative comparison of their performances.

A. TLD

TLD has three main components: (i) Tracker: a tracker follows an object from one to other frame by assuming limited frame-to-frame motion. TLD uses a median-flow tracker [19] with failure detection scheme; (ii) Detector: the detector localizes all appearances that have been observed and corrects the tracker when necessary. The detector applies scanning-window method to an input image and for each patch it estimates the presence or absence of the object. (iii) Learner: a learner checks the performance of the tracker and detection and estimates the detector error and updates it to avoid further error [21]. One of the major contributions of this work is a novel learning method called P-N learning that helps to estimate the error by recognizing the missing detection (generated by P-expert) and false alarm (generated by N-expert). The learner initializes the object at the first

Figure 2. TLD with positive (green border) and negative samples (red border). This figure is best viewed in color.

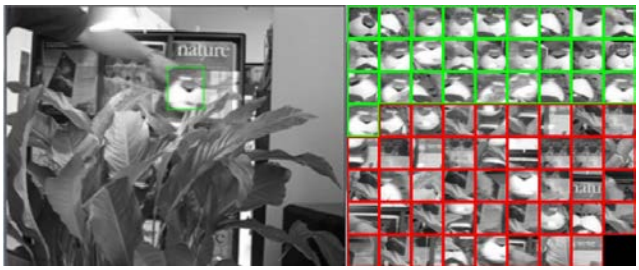


Figure 3. Struck with positive and negative support vectors bordered with green and red color. This figure is best viewed in color.

frame and updates itself on runtime with the help of P-expert and the N-expert (Figure 2).

B. Struck

Current adaptive tracking-by-detection methods consider the tracking process as classification tasks and use online learning methods to predict the object location during runtime. However, intermediate processing steps (e.g., sampler, labeler) could be eliminated by directly predicting the changed object location in frames. Struck [12] use this approach by merging the learning and tracking state, thus avoiding the intermediate classification phase (Figure 3). They used online structured output SVM learning method described in [5], [6] and adopted it to tracking problem [12]. To limit the number of support vectors that increases in training, they applied a fixed budget (i.e., specific limit) mechanism to control its growth and showed that the budget scheme helped the technique to be computationally faster.

C. Quantitative Evaluation

This section reports on quantitative experiments comparing TLD and Struck. We use the implementations provided by their respective authors. We initailize both trackers with a bounding box in the first frame. With this initialization, both algorithms start tracking the object of interest in the subsequent frames. We used the dataset in [12] to evaluate both tracking methods. We conducted the experiment on an Intel Pentium dual-core processor running at 2.0 GHz with 3GB RAM. During the experiment, we logged the tracked object position with other necessary information so that we could use that information later for object selection and interaction.

The performances of two methods are evaluated using two criteria: (i) *precision* measured by the ratio between the number of correct detections and the total number of detections; (ii) *speed* measured by the average number of frames a technique can process in one second.

Table I presents a comparison of the execution time and speed for TLD and Struck. For all the video sequences, the average execution time for Struck (345 sec) is significantly higher than TLD (63.75 sec). Therefore, TLD is more than five times faster than Struck. We also found that TLD tracked the object in less time than struck in every sample video. Table II shows the comparison of precision for these two methods results. Again, we observe that TLD achieves higher accuracies than Struck. Since TLD is faster than more accurate than Struck according to our evaluation, we choose to use TLD as our tracking method in our system.

Table I
SPEED COMPARISON OF TLD AND STRUCK WITH SAMPLE DATASET1. FIRST TWO COLUMNS ARE THE TOTAL EXECUTION TIME FOR TWO TECHNIQUES AND LAST TWO COLUMNS PRESENT NUMBER OF FRAMES PROCESSED IN EVERY SECOND.

	Struck	TLD	Struck	TLD
	in sec	in sec	frame/sec	frame/sec
Coke	251	47	1.16	6.21
Girl	401	93	1.32	5.69
Tiger1	394	53	0.9	6.68
Tiger2	334	62	1.09	5.89
Average	345	63.75	1.12	6.12

Table II
PRECISION FOR TLD AND STRUCK. RESULTS SHOW THAT TLD ACHIEVES HIGHER PRECISIONS COMPARED WITH STRUCK ON ALL THE VIDEOS.

	Struck	TLD
Coke	0.69	0.94
Girl	0.80	0.93
Tiger1	0.77	0.86
Tiger2	0.63	0.79
Average	0.72	0.88

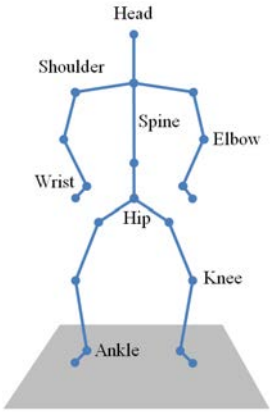


Figure 4. Body parts extracted from Microsoft Kinect.

IV. Kinect as Input Device

Our final goal is to provide interaction capabilities to streaming videos on TVs or computers. Popular input devices for them include TV remote, Wii remote and Microsoft Kinect. In our work, we choose to use Microsoft Kinect as it provides interesting features such as depth information, user body Skeleton in real time.

Kinect has become an importent motion sensing input device for many researchers to create new forms of interactions with computers. Kinect is capable of tracking user skeletons and body joints from the depth sensor. In skeleton tracking, a human body is represented by a number of joints corresponding to body parts such as head, neck, shoulders, and arms (Figure 4). Each joint is represented by its 3D coordinate (x,y,z) in meters. The Kinect API provides different functionalities such as tracking moving people with skeletons, determining the distance between an object and the kinect camera, etc. We take advantage of the Kinect API to track the user's skeleton in real-time.

The Kinect sensor doesn't have enough resolutions to ensure consistancy accross different frames. So the tracked skeleton has jitters over time. To get smoothed skeleton tracking, we apply smoothing and filtering operations provided by the Kinect SDK to reduce jitters. The smoothing filter is based on the Holt Double Exponential Smoothing method that provides smoothing with less latency than other smoothing filter algorithms [31].

We are interested in capturing hand movements so that we can map the movements to the cursor control and the object selection. We first extract the positions of hand joints (i.e., shoulder, elbow, wrist), then map the middle of spine as the center of screen. Displacement of right wrist from the middle of spine is considered as the cursor displacement from the center of the screen (i.e., moving right wrist to the left moves the cursor to the left). We also record hand bending gesture and hand extend gesture to assist object selection. We discuss details of selection techniques in the following section.

V. Object Selection

For a given video, we apply TLD to extract interesting objects (e.g., dress, necklace, shoe) from it. TLD provides us the locations of different objects in each frame. We use that information to draw a bounding box around each object of interest (Figure 5). At this point, we have a video with different objects labelled in it. Placing queries to those objects requires the development of some form of interaction capabilities with input devices.

Previous research on selection tasks has demonstrated that selecting moving objects is considerably harder and more error prone than static ones. Any form of additional assistance to selection techniques leads to significant performance benefits [13]. Therefore to assist users selecting an object from video, we develop a set of selection techniques using the depth and body skeleton information from Kinect.

A. Selection Techniques

We develop three depth-based selection techniques and three left-hand based selection techniques using the depth and body skeleton information extracted from the Kinect. First of all, we map the right hand movement to the cursor control (see the first row in Fig. 6). When a user moves his/her left hand to the left of the body, the cursor will move to the left on the screen. For the depth-based selection technqiues, we calculate the distance between the wrist and the body using their depth information (see the left image of the second row in Fig. 6). For the left-hand based selection techniques, we track the left hand movement and use bending gesture (see the right image of the second row in Fig. 6) to invoke the selection.

In the following, we describe in details the three left-hand selection techniques.

Left-hand with Basic Cursor: In this technique, a user controls the cursor with the right hand and the selection is invoked with the bending gesture of the left hand. In other words, when the cursor is moved over the target, the user needs to bend the left hand to confirm selection.

Left-hand with Ghost: This technique is similar to the target ghost technique proposed in [13] where a user's certain interaction creates static proxies of moving objects on the position at the time of interaction. We modify the technique for both hand interaction. When a user bends the left hand, our system creates static images of all moving objects (Figure 1e) and s/he selects the static target object by moving the cursor over it.

Left-hand with Crossing: This novel technique is inspired by the memory buffering method. Here we considered a virtual bin and all the objects that are overlapped with the cursor are stored in that bin. Now users can access the last stored object by bending his/her left hand. The main

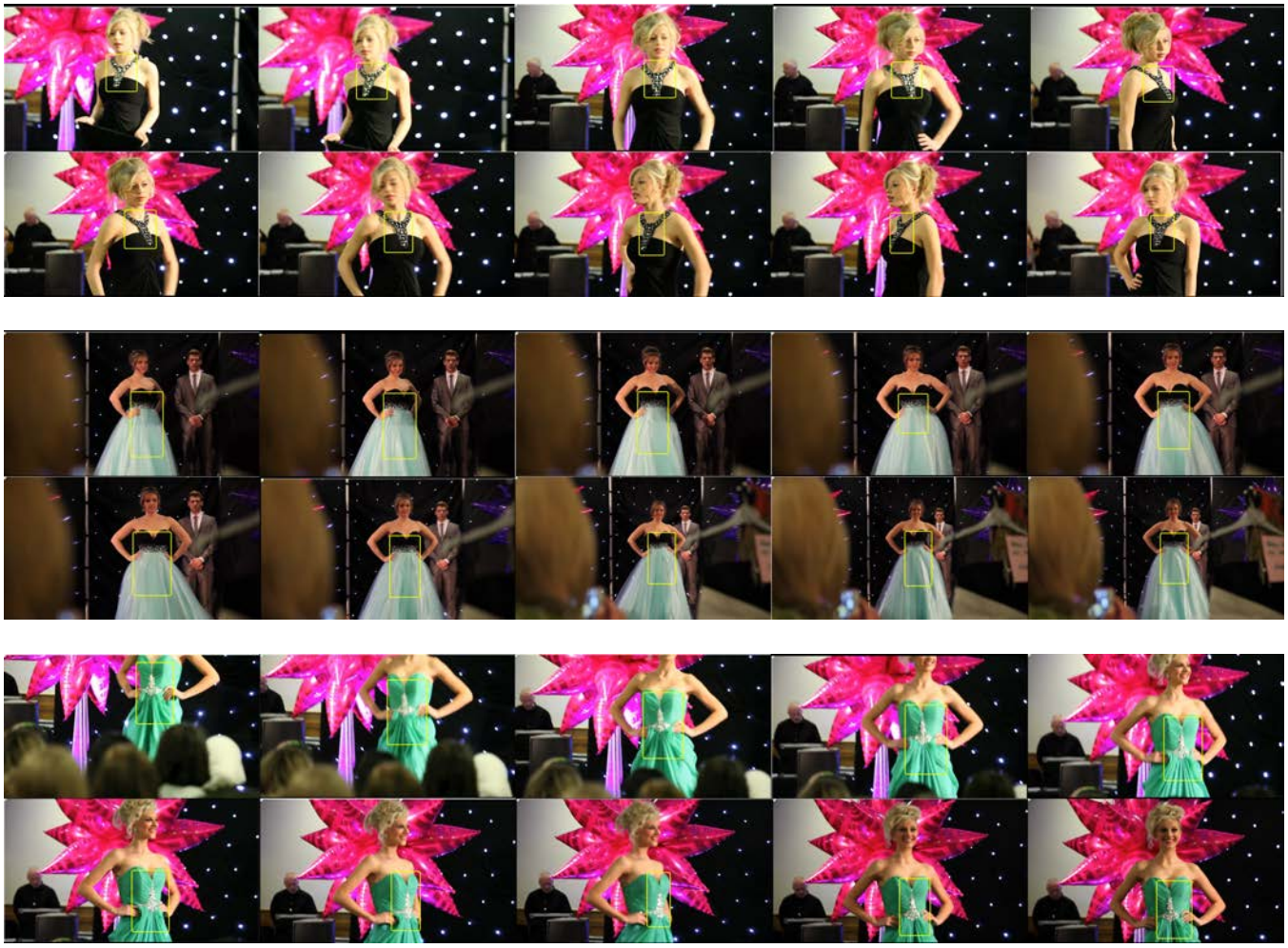


Figure 5. TLD is applied on a sample video. The yellow bounding box indicates the tracked object. This figure is best viewed in color with magnification.

advantage of this technique over left-hand with basis is that a user doesn't need to simultaneously follow and plan the selection. Thus it provides more flexibility on object selection.

Similarly,we define three depth-based selection techniques: *Depth with Basic Cursor*, *Depth with Ghost* and *Depth with Crossing*. All of these techniques have similar functionalities that described above. The only difference is that the selection is done by placing the right hand closer to the screen. For instance, in *Depth with Basic Cursor* technique, a user needs to move the cursor into the bounding box and at the same time s/he needs to extend the wrist away from body to a certain distance to confirm the selection.

B. Quantitative Evaluation

We conducted a user study to compare the performance of all those techniques in an object selection tasks. In the experiment, at least one object was always moving on the screen and we asked participants to select that object with all six techniques. Three right-handed participants with ages range from 25 to 30 participated in this experiment. The experiment ran on a Windows 7 PC connected to a Microsoft Kinect. In this experiment, we measure the task completion time, i.e. the time from when a user starts the trial to when s/he successfully selects the target. We also logged the number of attempts a user took to select the target object.

The results of our study show that participants were faster with all left-hand selection techniques than all depth-based selection (Figure 7). Among the left-hand techniques, participants were fastest with *Ghost* (2,824ms), followed by *Crossing* (4,325ms) and then with the *Basic cursor* (5,183ms). We find similar trends for depth-based techniques where *Ghost* technique (3,543ms) took less time than *Crossing* (4,779ms). *Depth with Basic* cursor took the longest task completion time (6,491ms) among all the techniques.

We observe similar results for the number of attempts. All left-hand selection techniques took less attempts (Figure 8)

Figure 6. Illustration of different selection techniques. First row: control the cursor with right hand movement. Second row (left): left-hand selection by bending the left arm. Second row (right): depth-based selection by placing the right hand closer to the screen.

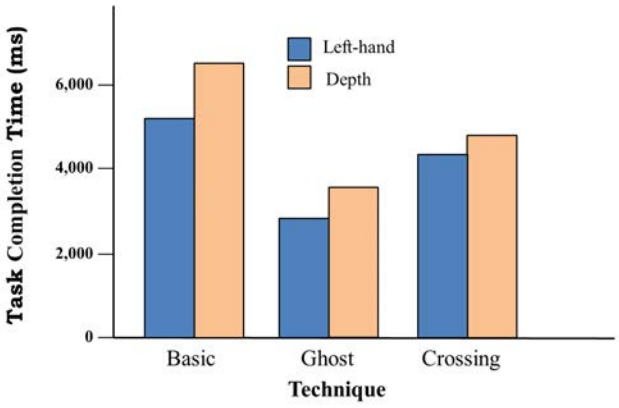


Figure 7. Task completion time across different techniques with left-hand and depth.

than depth-based techniques. Users took only one attempt to successfully select an item with *Left-hand with Ghost* and *Left-hand with Crossing* techniques, whereas they took 1.60 attempts for *left-hand with basic* on average. *Depth with Ghost* took the minimum number of attempts (1.07) among all depth based techniques, followed by *Depth with basic* (1.67). Surprisingly *Depth with Crossing* technique took the maximum number of attempts (1.80) to select an object.

Overall, our study shows that participants were faster and more accurate with left-hand techniques. In all cases, basic cursor takes the longest completion time and requires more

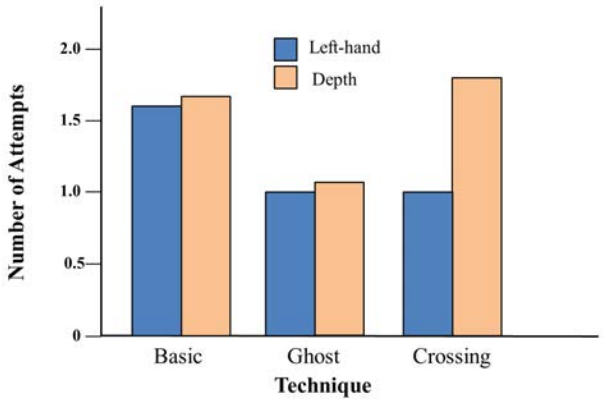


Figure 8. Average number of attempts across different techniques.

attempts for selection. The same trend has been reported in the literature [13], [16]. As crossing techniques involve two sequential operations (i.e., cross an object and then bend the arm for selection), overall it shows slower performance than ghost techniques. All the ghost techniques show significant improvement of results in comparison to other techniques. With the ghost techniques, since participants can select the proxies of moving objects (i.e., motionless target), they have more control on the selection and require less attempts. Overall, our results show that for moving object selection task, creating and allowing users to interact with a static version of moving objects provides better results.

From the results, we propose the following guidelines for developing application involving user interactions with video contents:

- For object tracking, the TLD method is the prefered method since it is faster and provides more accurate results compared with Struck.
- For Kinect based interaction, left-hand selection is prefered over depth-based selection.
- For a task involving moving object selection, better results can be achieved by using a static proxy of the moving object.

Interested readers are referred to the video demonstration of our system [32] for a more intuitive understanding of these approaches.

VI. Conclusion and Future Work

In this work, we developed a novel application that allow users to interact with video contents using Kinect. Our system uses the state-of-the-art tracking method to obtain object positions in a video. We then proposed several different object selection techniques that allow users to select targets using various gestures.

This work can be extended further in several directions in the fields of computer vision and human-computer interaction. For instance, the current version of TLD only tracks

one object at a time. This can be extended to track multiple objects simultaneously. Furthermore, in our implementation, we first extracted objects location off-line and then apply that information to video frames. The whole process can be done online and in a single step. Finally, we only explored a small set of selection techniques for Kinect based system. As future work, we plan to explore a richer set of interaction techniques for this application.

REFERENCES

[1] S. Avidan. Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(8):10641072, 2004.

[2] B. Babenko, M. H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[3] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. Conference on Human Factors in Computing Systems, 2012.

[4] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. International Symposium on Experimental Robotics (ISER), 2012.

[5] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with LaRank. Interantional Conference on Machine Learning (ICML), 2007.

[6] A. Bordes, N. Usunier, and L. Bottou. Sequence Labelling SVMs Trained in One Pass. In Proc. ECML-PKDD, 2008.

[7] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. IEEE Transactions on PAMI, 27(10):16311643, 2005.

[8] L. Cruz and D. Lucio and L. Velho. Kinect and RGBD Images: Challenges and Applications. 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), pp. 36 -49, 2012

[9] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, (47), 1954, 381-391

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. European Conference on Computer Vision (ECCV), 2008.

[11] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. Proc. of CHI '05, 281-290.

[12] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 263270

[13] K. Hasan, T. Grossman, and P. Irani. Comet and target ghost: techniques for selecting moving targets. In Proc. of CHI '11. ACM, NY, USA, 839-848.

[14] Z. Hossain, K. Hasan, H. Liang and P. Irani. EdgeSplit: Facilitating the Selection of Off-Screen Objects. In Proc. of MobileHCI'12. San Francisco, CA. ACM.

[15] T. J. Gunn, P. Irani and J. Anderson.An Evaluation Of Techniques For Selecting Moving Targets.In Proc. of the CHI EA '09, 3329-3334.

[16] M.V.Ilich. Moving Target Selection in Interactive Video, M.Sc. Thesis, University of British Columbia, pp. 110. (2010)

[17] S. Izadi, R. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time dynamic 3d surface reconstruction and interaction. ACM SIGGRAPH, 2011.

[18] P. Kabbash and W. Buxton. The "Prince" technique: Fitts' law and selection using area cursors. Proc. of CHI'95, p. 273-279. 1995

[19] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-Backward Error: Automatic Detection of Tracking Failures, Proc. IAPR International Conference on Pattern Recognition, pp. 23-26, 2010.

[20] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. IEEE Confer. on Computer Vision and Pattern Recognition, 2010.

[21] Z. Kalal, K. Mikolajczyk, J. Matas, J. Tracking-Learning-Detection. IEEE Transactions on PAMI, vol.34, no.7, pp.1409-1422, July 2012

[22] K. Lai, L. Bo, X. Ren, , and D. Fox, Detection-based object labeling in 3d scenes. Proceedings of the International Conference on Robotics and Automation, 2012.

[23] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon., Kinectfusion: Real-time dense surface mapping and tracking. IEEE International Symposium on Mixed and Augmented Reality, 2011.

[24] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, Tracking a depth camera: Parameter exploration for fast icp. IEEE Intelligent Robots and Systems, 2011.

[25] S. Rusinkiewicz and M. Levoy, Efcient variants of the icp algorithm. IEEE 3-D Digital Imaging and Modeling, 2001.

[26] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from a Single Depth Image. IEEE Conference on Computer Vision and Pattern Recognitio, 2011.

[27] L. Spinello and K. Arras, People detection in rgb-d data. International Confer. on Intelligent Robots and Systems, 2011.

[28] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. ACM Computing Surveys, 2006.

[29] http://www.xbox.com/en-GB/kinect/kinect-effect

[30] http://www.youtube.com/t/ press_statistics

[31] http://msdn.microsoft.com/en-us/library/hh973078.aspx

[32] Video demo. http://www.cs.umanitoba.ca/∼ywang/crv13_supp/