3D Low-profile Evaluation System (LES)
An Unobtrusive Measurement Tool for HRI

Mahisorn Wongphati, Hirotaka Osawa and Michita Imai

Abstract—Low-profile Evaluation System (LES) is an implementation of a portable and low-cost system for evaluation of Human-Robot Interaction (HRI) by measuring objects inside 3D scene. The implementation is based on low-cost sensors and freely available open-source software resources. LES is designed to operate on a single computer for ease of setup, portability and opportunity to conduct the experiment outside laboratory environment. LES utilizes Kinect [1] and USB camera as its main sensors for recording all interaction between human and robot during the experiment. Kinect provides full unobtrusive 3D measurement and high-speed video for scene analysis and object identification. USB cameras provide additional views for the scene analysis process. LES has 2D and 3D scene display for monitoring the experiment from all sensors in real-time. In 3D scene display, LES also provides a set of measurement tools for scene and interaction analysis. All sensors can be simultaneously recorded with bookmark feature for marking important events and situations in the experiment. The recorded data can be used for off-line playback, analysis and evaluation with 3D measurement tools as same as real-time data from all sensors.

In this paper, we explain the implementation of LES and demonstrate the functionality with Robovie humanoid robot to show the usage in real-world setup.

I. INTRODUCTION

In HRI research, it is necessary to evaluate interaction between human and robot during and after an experiment. Data in the experiment scene, especially visual data, need to be recorded for further analysis and reference after the experiment. In this paper, we are interested in the analysis that involves with scene and object measurement. We separate the measurement into two types, static and dynamic measurement. In this context, static measurement means measurement that can be measured in a single frame such as position and distance of the objects. On the other hand, dynamic measurement requires two or more frames of data such as measurement of velocity and angular velocity. Both types of measurement are important for the evaluation of close proximity interaction between human and robot [2].

Many types of sensors can be used for recording the interaction during the experiment. The conventional methods such as video recording and motion capture are widely used in HRI research. Video recording is a primary choice for analysis of the experiment. Two or more cameras are usually used to provide overall scene and scenes from specific points of view in the experiment, especially when behavior monitoring is required [3], [4]. In addition, motion capture is also an essential tool for the experiment that needs detailed measurement of interaction in the experiment [5].

Video recording and motion capture have their own strong and weak points. Video recording is able to provide high-resolution data for scene analysis and object identification. However, it cannot provide 3D measurement information without additional effort on 3D image reconstruction (Ch. 12 in [6]). On the other hand, motion capture system such as OptiTrack [7] can provide high-accuracy objects’ positions in 3D space. However, motion capture is primarily designed for capturing high-speed motion in 3D space and it needs additional markers to be attached on the preselected objects before the experiment. The additional markers also introduce an obtrusive measurement problem into the experiment [8].

Other problems aside from the mentioned problems are system cost and setup overhead. Commercial motion capture and video recording system are usually expensive and fragile. Both systems require a number of accessories such as cables and mounting equipment for setup and operation. System cost and setup overhead also eliminate an opportunity to perform the experiment outside laboratory, for instance, a public indoor environment where the experiment can be done with number of participants in real situation [9] such as cafeteria or department store.

In this paper, we purpose a tool for HRI experiment called Low-profile Evaluation System (LES). This research offers three contributions to HRI community. First, LES offers an integrated software that can monitor and record HRI experiment in 3D in real-time. Second, within the same software, LES offers an extensible unobtrusive measurement tool for measuring all interaction in HRI scene using simple and intuitive GUI. Third, LES is a free software that utilizes low-cost sensors and a single computer for its functionality, therefore it enables the opportunity to conduct the experiment outside laboratory by reducing the setup overhead and system cost.

The rest of the paper is organized as follows. We begin with LES’s design, sensor calibration and software components. We then show the use of LES with the real robot called Robovie [10] along with setup and measurement processes in LES. In the later sections, we conclude with current capabilities of LES and related works.
II. LES DESIGN

A. Software Design

LES is designed as a tool for monitoring, recording, and evaluating HRI experiment. It has 3D scene recording function with bookmark feature and a set of 3D measurement tools. The recording function in LES performs a synchronized recording from attached deceives. During the recording, the bookmark feature provides a time stamp mechanism that allows user to mark any interesting events and situations while focusing on the experiment scene using 2D scene display (Fig. 1a) or 3D scene display (Fig. 1b). The measurement tools are available in both real-time display and playback modes.

LES is written in C++ using Qt as a framework for GUI and operating system functions. It allows unlimited number of devices to be connected. However, total number of devices is limited by number of USB root hubs and processing power of the computer. LES can display and manipulate data from all devices in 3D scene display in both real-time and playback modes.

The rest of the section explains sensor calibration process, measurement tools, and recording with bookmark function respectively.

B. Calibration

LES 3D scene display renders all data from Kinects along with positions of both Kinect and camera (see Fig. 6c). To fulfill this requirement, we need to calibrate both devices to obtain their intrinsic and extrinsic parameters. Additional information about camera model, intrinsic and extrinsic parameters can be found in the book [6] about OpenCV [11].

1) Camera Calibration: We use camera calibration function in OpenCV to estimate intrinsic parameters and lens distortion coefficients of both Kinect and camera. With these data, we can mathematically correct errors those might be caused by both manufacturing process and lens quality. An example of lens distortion is shown in Fig. 2.

The calibration data will allow the use of functions in OpenCV those are based on pinhole camera model with both Kinect and camera.

2) Kinect Calibration: Kinect is a 3D RGBD (red-green-blue-depth) camera that combines information from 2D color camera and 2D depth camera (infrared camera). The depth camera recognizes an infrared pattern projected from an integrated infrared projector and registers depth information with 2D color camera using in-camera process for full 3D RGBD output [12].

Unfortunately, Kinect is mainly designed for Microsoft’s Xbox game console [13] and does not have any official supported development tool. Nevertheless, with its price, performance and measurement accuracy, there are many projects those provide communication API and usage examples for Kinect on many computer platforms [14], [15], [16].

In LES, we use libfreenect [16] for Kinect communication and manually calibrate each Kinect by following the method described in ROS’s Kinect node [17]. The rest of this subsection shows a step-by-step Kinect calibration process.

The first step performs a calibration on both color and depth cameras to obtain their intrinsic parameters and distortion coefficients using chessboard pattern (Fig. 3).

With depth camera’s calibration data and additional data from ROS’s Kinect node, we can compute a depth of each pixel in meter unit using

\[ d = \frac{b \cdot f}{0.125 \cdot (d_{off} - kd)} \]  

where:
- \( b \) is the horizontal baseline distance between depth camera and infrared projector in meter unit.
- \( d_{off} \) is the disparity offset that has value 1090 in 1:0.125 pixel unit.
- \( kd \) is raw depth measurement information from depth camera.

With depth information of each pixel (\( d \)), we can create the projection matrix (\( Q \)) for mapping projective coordinate...
of depth camera to 3D world coordinate $X = [x\ y\ z]^T$ in meter unit with

$$X_{\text{depth}} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & f \\ 0 & 0 & 1/b & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix}$$ (2)

c$_x$ and c$_y$ are coordinates of camera’s optical axis in pixel unit. f is a focal length of the camera in pixel unit. u and v are pixel’s position in depth camera.

At this step, we perform a stereo calibration between color and depth cameras to obtain rotation ($R$) and translation ($t$) matrices. Both matrices are combined into a transformation matrix $[R|t]$ for mapping the coordinate of the measurement from depth camera coordinate to color camera coordinate

$$X_{\text{color}} = [R|t]X_{\text{depth}}$$ (3)

By combining projection matrix Q from (2) and $[R|t]$ matrix with intrinsic matrix of color camera ($P$), we can register color information with depth information using

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}_{\text{color}} = P[R|t]Q \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix}_{\text{depth}}$$ (4)

3) Sensor Position Calibration: In order to correctly display Kinect data and sensor’s position in 3D scene display, we need to know the position of each sensor in 3D world coordinate. LES obtains positions of Kinects and cameras by finding their extrinsic parameters using a chessboard pattern and SolvePnP function in OpenCV. The process begins by placing the pattern where all sensors can detect it (Fig. 6b) and computes each sensor extrinsic parameters in chessboard coordinate system. The extrinsic parameters of each sensor is used as transformation matrix in 3D scene display rendering.

C. Measurement Tools

1) Static Measurement Tools: All objects in 3D scene display are rendered at 1:1 scale in meter unit. This property allows direct measurement of position and distance inside the 3D scene display. Currently, LES provides four tools for static measurement in 3D scene display. The first one is a tool for measuring the position of an object by clicking at interesting point (Fig. 4a). Second tool provides a distance measurement between two arbitrary points in the scene (Fig. 4b). The third tool is an angle measurement tool that computes the angle from three selected points in the scene (Fig. 4c). The last tool is for measuring a length of arbitrary path inside 3D scene display (Fig. 4d).

2) Dynamic Measurement Tools: Dynamic measurement tool uses static measurement across multi-frames for computing dynamic information of objects in 3D scene display. Currently, velocity and angular velocity measurement are supported. Velocity measurement requires user to select the same point in an object from two arbitrary frames for velocity computation (Fig. 9). Angular velocity measurement tool computes rotation speed of 3D vector that is selected by user from two arbitrary frames (Fig. 10). More details of the measurement steps will be described in subsection III-B

D. Recording and Playback with Bookmark

In normal video recording or motion capture, examiner usually starts recording the experiment, focuses on the experiment and stops the recording when the experiment has finished. When evaluation begins, examiner has to manually and repetitively search forth and back in the recorded data for some specific points in the experiment for the evaluation.

In order to eliminate the repetitive work and efficiently evaluate the experiment scene after data are recorded, LES provides a recording function with bookmark feature that can be added during the recording by clicking on a software button (Fig. 5) or pressing an assigned key. In playback mode, the bookmark will be loaded and used as a navigation key to each marked event or situation for quick accessing in both 2D & 3D scene display.

III. LES Demonstration

A. Setup

This section shows the setup of LES system when Kinects and USB cameras are already pre-calibrated. In this setup, we attach two Kinects and three USB cameras (Fig. 6a) to a computer and adjust the position of each device until all
devices can detect the calibration pattern (Fig. 6b). After positions of all devices are calibrated, 3D scene display that renders data from all Kinects with all devices’ positions will be available for monitoring and evaluating the experiment in real-time (Fig. 6c). This setup requires less than 10 minutes after positioning all sensors in the experiment scene.

On a 64-bit Windows 7 computer with Core i7 960 CPU and 6 GB of RAM, LES utilizes less than 15% of CPU load and less than 150 MB of memory for real-time data displaying from all devices at 30 fps.

The demonstration starts with static measurement tools when a human is trying to grab Robovie’s hand. In this demo, the human moves toward Robovie and stops at a certain distance in front of the robot. At this point, we can measure the position of the human’s hand (Fig. 8a) and the eye contact distance between human and Robovie (Fig. 8b). When the human is grasping Robovie’s hand, we can measure the human’s body angle (Fig. 8c) and show the path between the human’s arm and Robovie’s arm when the human body has returned to normal standing posture using path measurement tool (Fig. 8d).

Next demonstration is the use of dynamic measurement tools in LES. When the human is trying to grab Robovie’s hand in the previous demonstration, we can measure the hand approaching speed with velocity measurement tool by selecting the start point in Fig. 9a and fast forwarding to select the end point in Fig. 9d. LES will compute the velocity using frame count between the start and end points.

After human has released Robovie’s hand, we can measure the rotation speed of Robovie’s upper arm with angular velocity measurement tool. The measurement starts by the selection of rotation point and center of rotation (Fig. 10a) and then fast forwards to select the end point of rotation in Fig. 10d. LES will automatically compute angular velocity and report in program’s status bar.

IV. CONCLUSION

We show that LES is suitable for HRI experiment, especially when 3D unobtrusive measurement of the scene and interaction are required. LES can be set up and calibrated in an indoor environment using simple calibration method and pattern. Recording and playback functions with bookmark feature in LES also provide a convenient way to mark important situations in the experiment for the evaluation. The recorded data can be used for evaluating with provided static and dynamic measurement tools.

V. FUTURE WORKS

LES is still in developing state but proved useful in the HRI experiment when unobtrusive 3D recording and measurement are required. The current version of LES and manual are available at www.ayu.ics.keio.ac.jp/~mahisorn/les. The planned future works are sound recording, calibration...
Fig. 8: Static measurement when human trying to grab Robovie’s hand

Fig. 9: Hand approaching speed measured by LES at 0.493 m/s
without chessboard pattern, and a plug-in system for measurement tools, posture & gesture recognition, and object tracking. After additional features are added into LES, it will be available at the same address as an open source project.

VI. RELATED WORKS

LES is the 3D measurement system that shares many concepts with 3D CAD/CAM software but designed specifically for the HRI experiment. With 3D CAD/CAM as its basic concept, there are some projects those can be configured or hacked for the same purpose such as Vrui VR Toolkit [18] and ROS’s rviz [19]. However, most projects have not been designed for the HRI experiment.

REFERENCES