Vision Assisted Safety Enhanced Shooting Range Simulator

Shamsuddin Ladha & Sharat Chandran
IITB-Monash Research Academy & IIT Bombay
ViGIL, Department of Computer Science & Engineering
Indian Institute Of Technology Bombay
{sladha|sharat} @cse.iitb.ac.in

Kate Smith-Miles
IITB-Monash Research Academy & Monash University
kate.smith-miles @sci.monash.edu.au

Abstract

Traditional shooting range simulators involve large investment in real estate, and other infrastructural resources. This paper describes a novel application of projector-camera systems to create a virtual shooting range.

Projector-based display systems offer an attractive combination of dense pixels over large regions. When coupled with a camera, the feedback system enables the display to be adjusted to environments enabling a variety of applications. We analyse two different implementation approaches for the simulator. As compared to a traditional shooting range, our simulator is inexpensive, safe and eco-friendly.

1. Introduction

Projection technologies using commercial off-the-shelf projectors is a viable way of creating life size displays. When coupled with traditional computer vision techniques using commodity camera, the duo form powerful closed loop systems called Projector-Camera systems (or PROCAMS). PROCAMS are powerful because they are not expensive, can create large displays with high resolution and brightness, and can use the feedback provided by camera to improve the performance and quality of the system. Some of the applications of PROCAMS include creation of life size displays that are seamless and visually correct [1], interactive presentation systems [3], shadow removal in front projection system [2] and artistic creations [4]. PROCAMS can be used in many different configurations — single or multiple cameras or projectors, moving or stationary cameras, etc. Despite their powerful nature, PROCAMS have NOT been widely used.

A traditional shooting range is a specialized facility designed for shooting practice using firearms. Shooting ranges can be indoor or outdoor. Fig. 1 shows an indoor shooting range. It is very expensive to setup, build and maintain such a facility. It is also important to observe gun safety rules at all times to avoid mishap and injury. Further, the toxic waste produced needs proper treatment and disposal. Despite these problems, shooting ranges are indispensable for instance, to train the police or the defense services. The standard simulation equipment, like a Head Mounted Display (HMD), is not cost effective. By simulating a shooting range using PROCAMS all the above drawbacks are eliminated and the system can be easily tuned to simulate complex scenarios.

Figure 1. An Indoor Shooting Range [Wikipedia]

1.1 Contributions:

In this paper we present an application of PROCAMS, namely, the creation of a *shooting range simulator*. The key creative novelty we introduce in this paper is the concept of the camera on weapon, an idea that, to our knowledge, has not been seen elsewhere. We analyse two different implementations that impact the real-time nature, and the resources needed for practical use. The technical and engineering contributions in this work include a real-time implementation with inexpensive off-the-shelf components, and analysis under various illumination conditions to ensure that the system works as advertised – which we believe is important in computer vision applications.

The remainder of this paper is organized as follows. We first describe our approach to creating a PROCAMS based shooting range simulator. Next, we look at two different implementation approaches, one using a single stationary camera and another with multiple moving cameras. We look at the advantages and disadvantages of each approach. In Section 3 we provide details of our implementation, and finally we summarize this work, and future work in the last section.

2. Shooting Range Simulation

In the simulation, the computer generates a virtual shooting scene with targets. Targets can be, for instance, soldiers or tanks. The virtual scene is projected onto a planar display surface as a First Person View (FPV). Trainees then shoot at the displayed scene, e.g. using a laser pointer. The entire process is observed by one or more cameras. The captured images are processed to compute the actual hit location on the virtual scene.

More specifically, the virtual scene is rendered in the FPV, and what one might call the Projector Co-ordinate System (PCS). Once shots are "fired", using image processing techniques we can identify the shot (hit point) in the captured image, *i.e.,* in the camera coordinate system (CCS). Our goal is to find the hit point in PCS. Fig. 2 illustrates this concept.

For training purposes, it is often sufficient to assume that the depth in the virtual scene is limited. Therefore we can measure the (two-dimensional) accuracy by simply mapping points from CCS to PCS using the 3×3 homography matrix.

We have implemented our system using two different configurations and techniques:

- A single stationary camera with an off-the-shelf laser pointer mounted on a weapon for firing shots. The camera observes various trainees, much like an instructor would. See Section 2.2 for details.
- The camera is mounted on a weapon (multiple moving cameras – one per shooter), and there is no laser pointer. See Section 2.3 for details, but note that here the camera will move with respect to time.

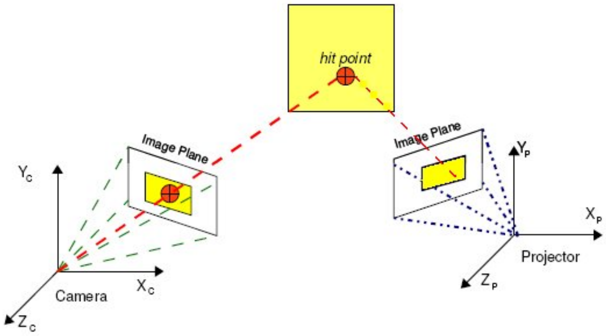


Figure 2. Hit point computation. The hit point as seen by a trainee is also observed by a camera (left). This point must be mapped to the projector coordinate system (right) to verify the accuracy.

There are two important aspects which dictate the overall performance and thus the viability of the two systems.

2.1 Quantitative Performance Indicators

- **Detection of hit point in the camera image:** Irrespective of whether the camera is stationary or moving, the hit point has to be detected in the image captured by camera whenever a shot is fired. Performance of the system is directly linked to the efficiency of the camera hardware and image processing algorithm used to detect this hit point.
- **Computation of hit point in the virtual scene** If the camera is stationary then the homography matrix doesn't change. But if the camera is moving then with every shot fired we need to recompute the homography matrix. This recomputation can have some bearing on system's performance.

2.2 Single Stationary Camera

In this approach, there is only one (observer) camera that remains stationary. A homography between the camera and the projector can thus be computed as a preprocessing step with respect to the planar display surface. Each shooter has a laser pointer that is used to fire shots. When a shot is fired, a colored dot is created on the display surface, and that is captured by the camera. The position of the laser dot is compared against the target positions to determine shot accuracy.

Figure 3. Multiple laser pointers captured by an inexpensive camera. Notice how a red laser pointer appears white due to saturation.

Hit point detection in CCS: We know that the laser dot is brighter than the projected image in the background. We also use the knowledge of pointer size limits and aspect ratio to discard background objects. Furthermore, the intensity of a laser pointer is observed to decrease in a Gaussian manner as we move away from its center towards the periphery of blob as shown in Fig. 3. We use these features to confidently detect a laser dot in the presence of objects having similar color and size as the pointer itself.

Hit point computation in PCS: Once the hit point in CCS is detected the hit point in PCS is trivially available using the previously computed homography for the stationary camera.

This system is very simple and efficient but it has several drawbacks:

- The laser dot on the display surface provides a visual feedback to the trainee for the next shot. This kind of visual cue is generally not available either in a traditional shooting range simulator, or in real life combat situations. This drawback can be eliminated by using Infrared (IR) lasers to fire shots and IR camera to capture images. However, IR lasers have related health hazards.
- More importantly, when multiple trainees are present, the laser dots can come too close to each other or overlap thereby confusing the system. This can easily happen when more than one trainee is aiming at the same target.

2.3 Multiple Moving Cameras

The laser based approach discussed above has the serious limitation of system getting confused when multiple trainees shoot at a single target. One of the goals of a shooting range is for trainees to collaborate as a team. Many of them shooting at a single target cannot be avoided. For this we use an entirely new approach for firing shots. We call this approach as "Camera on Weapon" (COW). A camera is mounted on each weapon, *i.e.,* we will have as many cameras as the number of trainees. Fig. 4 shows the schematics of the COW approach. With the COW approach, the viewing direction of a shooter and the corresponding mounted camera is aligned, thereby enabling accurate hit point computations.

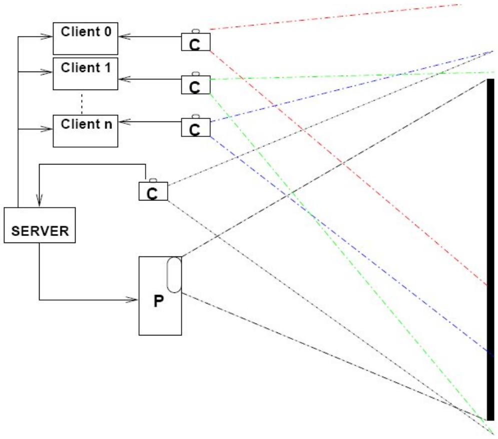


Figure 4. Camera on Weapon based approach (C - Camera, P - Projector).

Hit point detection in CCS: Each camera mounted on weapon captures images of the projected scene on the display surface. *Without loss of generality, the center of each camera frame is considered as the hit point in the CCS*. For example, if the camera frame resolution is 640×480 then the hit point in CCS is $(320, 240)$. Thus hit point detection in CCS is trivial and requires no additional processing.

Hit point computation in PCS: The camera mounted on a weapon is NOT stationary. It moves as the shooter and weapon moves, thereby invalidating the homography matrix. As a result of which the homography matrix has to be recomputed every time a shot is fired.

It is well known that only four points (no three of which are collinear) are required to reliably compute homography. Therefore, as a proof of concept, we project the following four marker pattern (Fig. 5) with the virtual scene to efficiently compute homography matrix. In our implementation, we use a wall as the projection screen; in actual practice, we expect that these markers are attached to the projector screen, or suitable markings are made on the display surface to aid homography computation.

The task now is to detect this marker pattern in camera frames. This is similar to laser shot detection and takes about the same amount of time. Once the markers are detected in CCS, the homography matrix is computed using their known positions in PCS. Shot location in PCS is then computed using this homography matrix and shot location in CCS (center of camera frame).

Figure 5. Sample four marker pattern to aid homography.

There are several advantages to this scheme

- Theoretically there is no limit on the number of trainees that can be supported
- There is no degradation in system performance with increase in number of trainees

However, the system is more expensive as compared to the laser based system and we have to detect markers in every single shot that is fired.

3. COW Implementation

Implementation of the COW approach is described in the following paragraphs.

System Design: Our program uses a simple client server design and is written in VC++ using OpenCV and OpenGL. The two programs (client and server) run on two different machines. The server is responsible for creating a virtual shooting scene with 4 marker pattern and multiple targets (colored rectangular blocks). This virtual scene is then rendered by a projector onto the display surface (projection area). The client is in control of the camera which sits atop a weapon. Each shooter's camera is controlled by a separate instance of the client.

When the client runs it detects the camera attached to the system and pops a GUI box to allow changes in the camera settings (so that a good quality image can be captured by the camera). The client registers with the server and receives a unique ID. Now, as the shooter moves the weapon (keeping all the 4 markers in the camera's field of view) the mounted camera also moves and captures the image of the projected scene. The position of 4 markers is detected in the captured image and using the known positions of these markers in PCS the homography matrix is computed. Next, the hit point in PCS is computed using the homography matrix and the center of camera image as hit point in CCS. The computed hit point is then reported to the server alongwith client's ID. The server then checks the hit point location against all target boundaries. If any target is hit the position of that target is changed and the simulation continues.

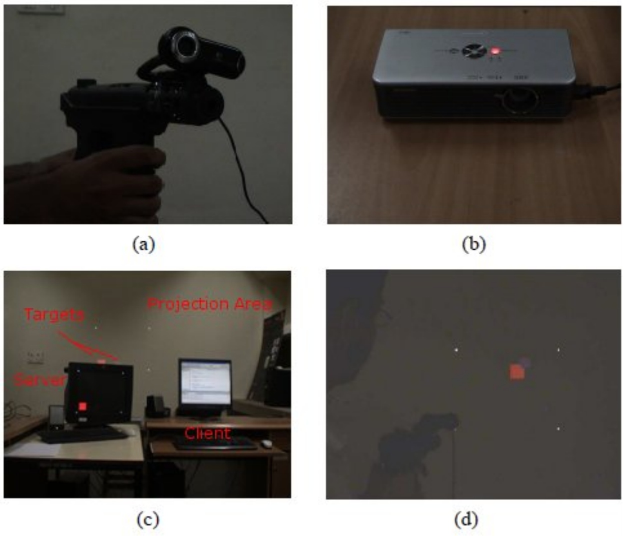
Observations: We have made use of off-the-shelf easily available components. The performance of the system is intimately connected to the quality of components used. Depending on the hardware configuration one or more trainees can be connected to a single PC. Theoretically we can have any number of trainees in the system without compromising system performance by adding additional hardware. **Approximate time taken to detect markers is about 22ms (in an unoptimized fashion), whereas homography computation virtually takes no time (0.15ms). Thus the system can provide up to 42 (shots) per second.**

One of the drawbacks of this implementation is that all four markers have to be in the camera's field of view for homography computation. To circumvent this problem we propose to use following (Fig. 6) sixteen marker scheme.

With the above scheme, all sixteen markers need not be in camera's frame of view to compute homography matrix. If four markers in any one corner of the display screen are visible to the camera then homography ma-

Figure 6. Sixteen marker scheme

trix can be computed reliably. Consider four markers in a corner, one marker is larger than the remaining three and horizontal and veritcal distance between markers is different. These extra features aid in identification of the correct screen corner that is visible in the camera frame irrespective of the location, and field of view of the camera.



(a) (b) (c) (d)

Figure 7. Experimental setup and snapshot of COW run

3.1 Hardware Configuration

Both the experiments were run on same hardware. Following hardware components were used. Fig. 7 depicts the components and the experimental setup.

- A web-cam that connects to a computer via USB port and supports 640 x 480 resolution (Creative Web Cam/Logitech Quick Cam)
- A projector of native resolution 800 x 600 (Sharp XR-1X Multimedia Projector, brightness 1100 ANSI Lumens)
- Two x86 machines (2.66GHz), running Windows XP Service Pack 2 and 512MB RAM

4. Conclusion and future work

PROCAMS are powerful but have not been widely used. In this paper we have come up with one of the vision based applications of PROCAMS namely, a shooting range simulator. The simulator is more safe and eco-friendly as compared to the traditional shooting range, and runs in real time. We have provided two different implementations of this simulator. One of this is relatively inexpensive, but limits the number of multiple users. The alternative, which we recommend, is reasonable in cost, and works in real time.

Our simulation is far from the actual shooting range. In future we plan to make our simulator more realistic by,

- Creating 3D virtual scenes with moving targets and various levels of difficulty
- Implementing effect of projectile motion of the bullets depending on the distance between targets and trainees

Our simulator is a classic example of confluence of three areas of computer science viz. computer graphics (virtual scene generation), computer vision (homography computation) and image processing (marker identification).

5. Acknowledgement

We acknowledge insightful inputs from Rohit Jhunjhuwala, Shyam Sunder Reddy, and Aman Parnami, and the support of ATB.

References

[1] M. Ashdown, M. Flagg, R. Sukthankar, and J. Rehg. A flexible projector-camera system for multi-planar displays. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–172, 2003.

[2] S. Audet and J. Cooperstock. Shadow removal in front projection environments using object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[3] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *International Conference on Computer Vision*, volume 1, pages 247–253, 2001.

[4] A. Toshiyuki and K. Hirokazu. Appearance enhancement using a projector-camera feedback system. In *IEEE Conference on Pattern Recognition*, pages 1–4, 2008.