# *Videogame Interaction through Vision-based Input*

Shahbano Imran

Advisor: Professor Hao Jiang

2009 Undergraduate Honors Thesis

Computer Science Department, Boston College

## Table of Contents

# Abstract

The purpose of this project was to build an interface allowing applications and videogames to use non-invasive vision based input as a replacement for conventional input device like a mouse or a joystick for enhanced HCI. The phases of the implementation involve template matching and tracking in real-time processing of video, and recognition of gestures through trajectory tracking. The limited pose estimation involves using the information about the location of the head and hands—regions of interest are first isolated by classifiers such as skin tone, then smaller parts of the frame are processed to achieve a real-time calculation to recognize and track the different parts of the body. Tracking requires segmenting the subject from the background, and matching these segments during consecutive frames. A robust implementation of pose estimation could be used to create interesting vision-based interfaces. Fundamental limitations to current algorithms in pose estimation include the compromise between accuracy and real-time processing costs. I'm focusing on restricting variable factors and using controlled conditions to get maximum accuracy for specific parts of the body, such as hands.

# Introduction

Interfaces today are not as deep or emotionally engaging as they could be. The standard general-purpose model of computer peripherals gives way to un-engaging, rigid user interaction. Instead of a standard point and click model, a more engaging variation would be one directly involving the user, where the user's gestures and movements control the application, rather than a mouse or keyboard. This provides for more intuitive interaction. Devices with user centered design (UCD), where users take center stage in the application, are beginning to emerge particularly in the realm of gaming. These devices function under the feedback principle of Human-Computer Interaction. A prime example of this is the recent shift towards active videogames. Games such as Rock Band (Harmonix Systems, EA Games), where the peripheral input devices include game guitars, bases, drums, and microphones, are becoming more popular. The hugely successful Wii (Nintendo) uses the wiimote as its controller (a input device with an infrared camera and an accelerometer that sends data to the Wii regarding distance measurements, and pitch, yaw, and roll movements-- these gestures are then used to control actions within the game). This creates games that are more emotionally engaging since there is a physical effort on the part of the user. As the shift from standard videogames to active gaming shows, the next logical step in gaming is to eliminate input devices all together, allowing players to control games only with their bodies and using gestures and movements as input to the application.

# Software

Non-invasive Vision-based input for video games creates new, emotionally engaging interactions between the computer and player. Pressing a key on the keyboard or moving a mouse is now replaced with hand movements and gestures. For instance, a user can rotate an image or zoom in out using his/her hands rather than a mouse or a player can actually runs to make the character in the game run. Points of interest can be isolated from an image and gestures can be extrapolated from this data over time.  This information is used to create an interface for applications using non-invasive vision based input as a replacement for conventional input devices.

## Problem

Human body pose estimation is the process of identifying how a human body and individual limbs and segments of a body are arranged in a given image. This project involves limited pose estimation-- detecting specific parts of the body in an image and recognizing dynamic visual processes based on spatial and temporal characteristics to produce gestures and actions which can be mapped to videogames.

The realtime realistic reconstruction of human motion gives rise to certain problems:

- *Complex Background*
    - Noise from a complicated background image with many objects can confuse recognition and decrease accuracy.
- *Lighting Conditions*
    - Lighting changes from one scene to the next, which affects results of the classifiers used to identity parts in an image.
- *Arbitrary Clothing*
    - Clothing length and color can affect skin classifiers
- *Unintentional Movements*
    - Support for unintentional movements must be added in order to detect continuous gestures.
- *Fast response time*
    - There is a compromise between processing speed and accuracy. In order to create an intuitive interface, response time of the application must be minimal. There cannot be a significant delay between the user's gesture and the response from the application.
- *Reliability/Robustness with little cooperation by the user*
    - The system must work for different people (different physical characteristics) and backgrounds. Body parts have similar color

and texture, and this must be taken into account in the classification.

- *Natural and intuitive gestures and actions*
    - Creating actions within the limits of the assumptions made about scenery can be challenging.

# Solution

A set of actions compiled by analyzing each image in a video feed using a set of assumptions can be used to interact with applications and videogames.

## Assumptions

### *Body Characteristics*

- Skin Color within expected range
- Symmetry in Body
- Clothing- Assuming half-sleeved or full-sleeved shirt
- Known start pose
- Scene limited to one person

### *Environment/Scenery*

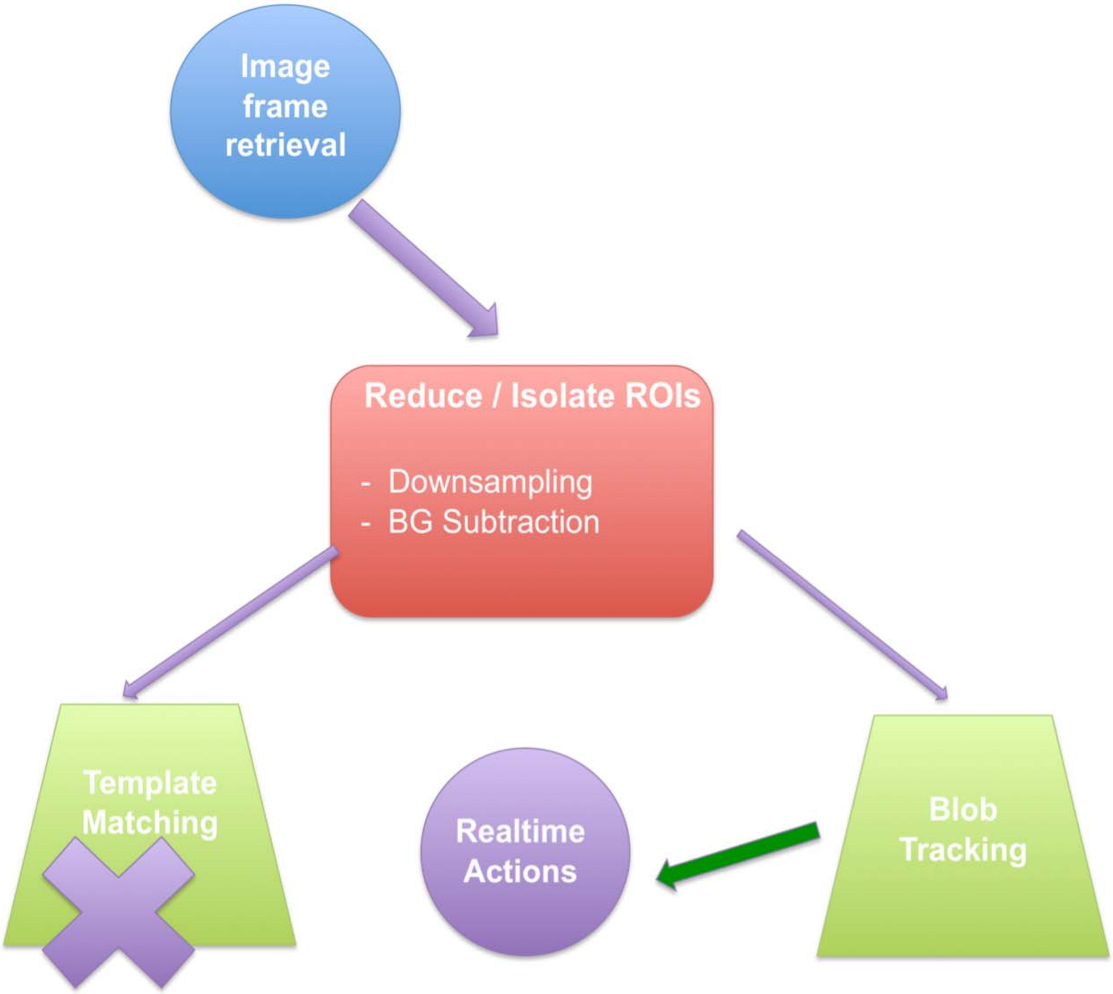
- Static Background
- Relatively Simple, Uniform Background
- Known Camera Parameters
- Steady Lighting

### *Motion*

- Space Limitations- Player movement is limited to a certain space

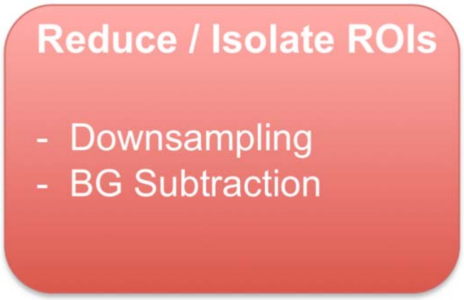
- No Camera Motion- frames are static
- No occlusion- nothing between the player and the camera
- Player Generally Camera facing
- Distance from the Camera is fixed

# General Strategy Diagram



The general methods used involve a few phases: retrieving an image from the webcam, isolating the regions of interest in that image, then finding information on select parts of the image to transform into a series of actions.

# Background Subtraction



Reduce / Isolate ROIs

- Downsampling
- BG Subtraction

Downsampling the image (decreasing its resolution by taking every other pixel or some other factor) reduces computation time while preserving the required information, and reducing noise in the image.

The first frame retrieved is labeled as the background and each subsequent frame is subtracted using the simple frame difference method:

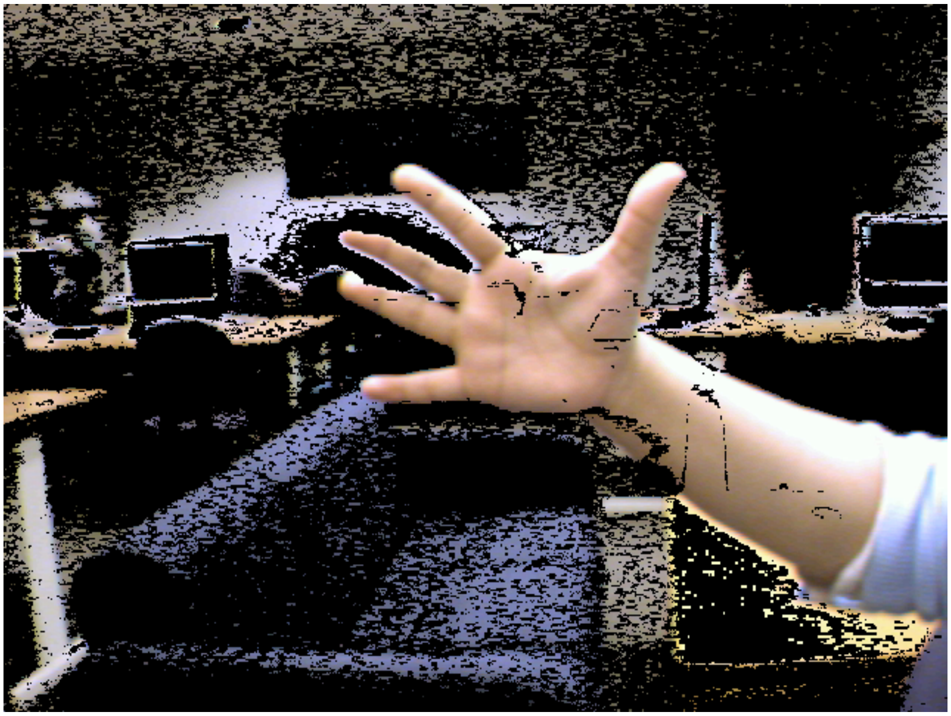
*Frame = Current Frame – Background,*

where rgb intensity pixel values are subtracted.

A few problems arise from this method:

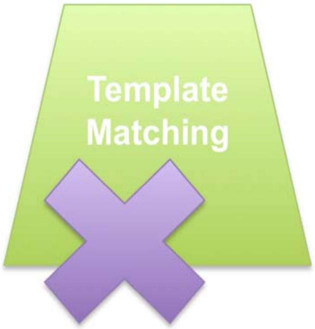
- *Initiation*- there is a fixed model for choosing the background- a user can only enter the frame after the background is stored.
- *Motion*- motion decreases the robustness of the background subtraction and part of the background is included in the search (a).

The image is then reduced to only the player/user, rather than the entire complicated background.



(a) Background Subtracted image with movement

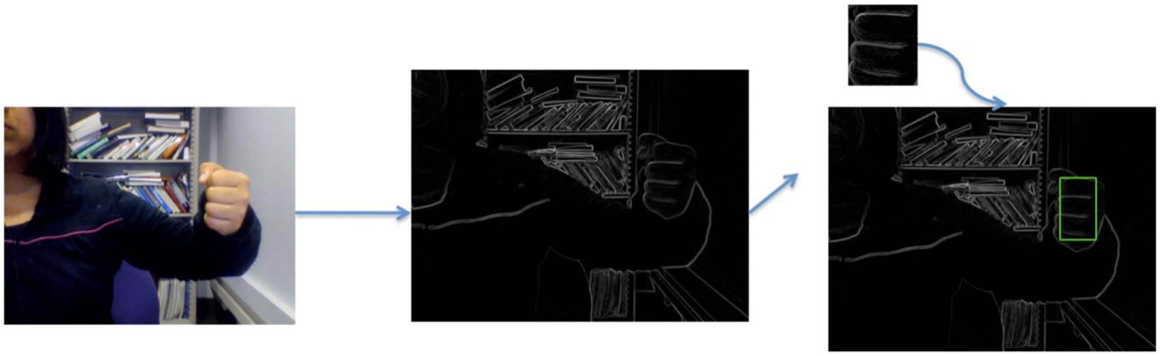
# Template Matching



Template matching simply quantifies the similarity of two images: in this case, the pattern of a person’s fist in an image. It follows a sliding Window approach- the template is moved over the entire image and each pixel stores the difference value. The minimum of this value is taken as the best match. The image is corrected for illumination (either by keeping the lighting constant or through Sobel edge detection—taking the derivative of an image to isolate edges) and the difference approximation is applied (b).

$$S = |I_{src} - I_{templ}|$$
$$\min(S)$$

*Matching by Correlation*



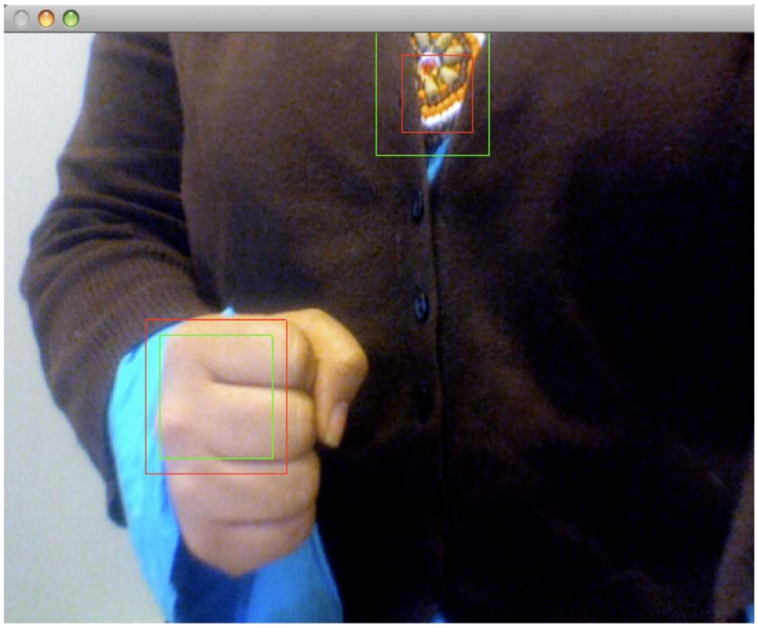
(b) Template Matching by Correlation

This technique is not completely sufficient for finding body parts in an image since adjusting for scale and rotation are computationally expensive and not robust in realtime.

Issues also arise when tracking multiple parts in an image—distance between parts must be kept at all times.  This limits the range of motion a user has, as well as movement in the scene. There can also be incorrect matches as well as partial matches.

*Problems*

- Runtime is proportional to |Template| * |Area|
- Not Realtime with acceptable accuracy
- Changes in illumination
- Adjusting for scale
- Low Tolerance to Deformation
- Partial Matches

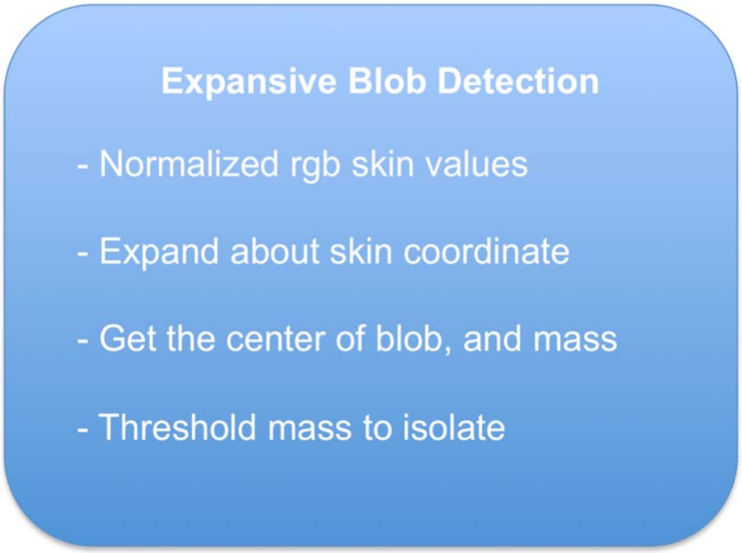


(c) Incorrect match

# Blob Detection using skin Classifier

Blob Tracking

Going on the aforementioned assumptions about skin color, symmetry in the human body, clothing (assume generally half/full sleeves, not wearing skin colored clothing, etc), a method for expansive blob detection based on skin classifiers can be used.

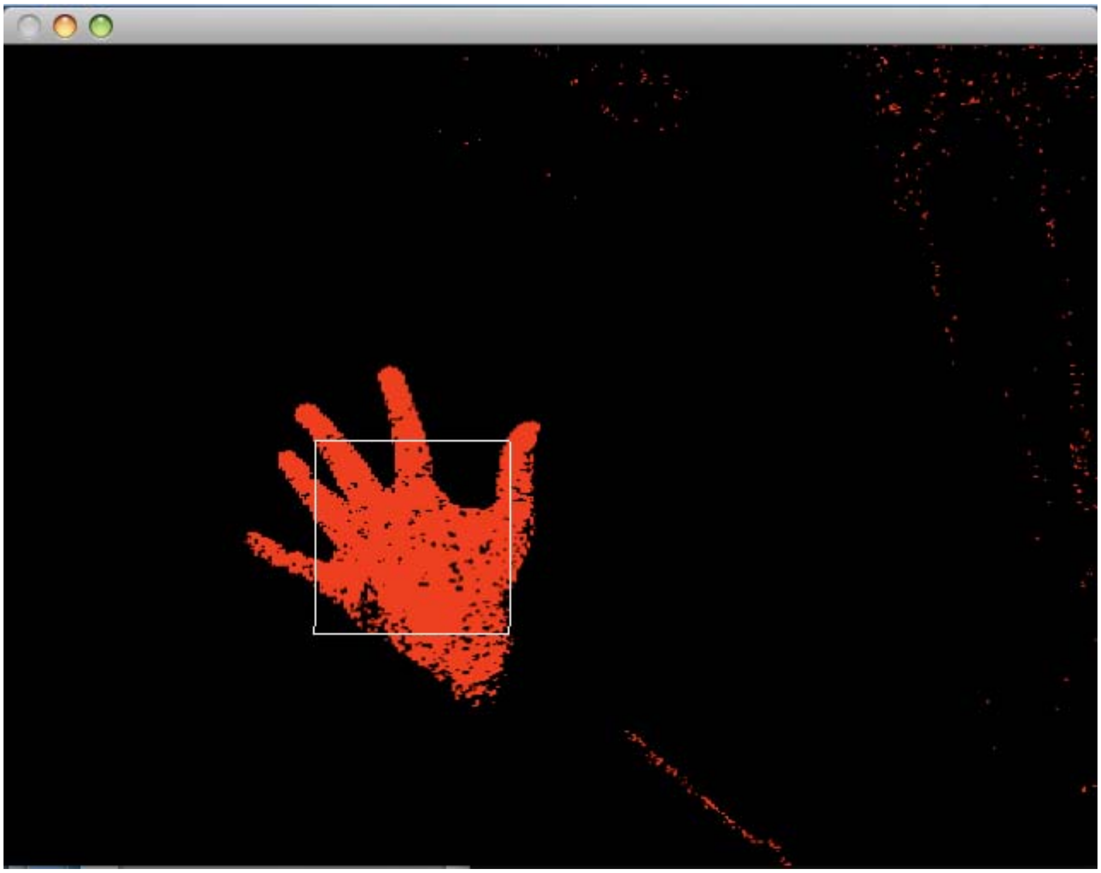


Expansive Blob Detection

- Normalized rgb skin values
- Expand about skin coordinate
- Get the center of blob, and mass
- Threshold mass to isolate

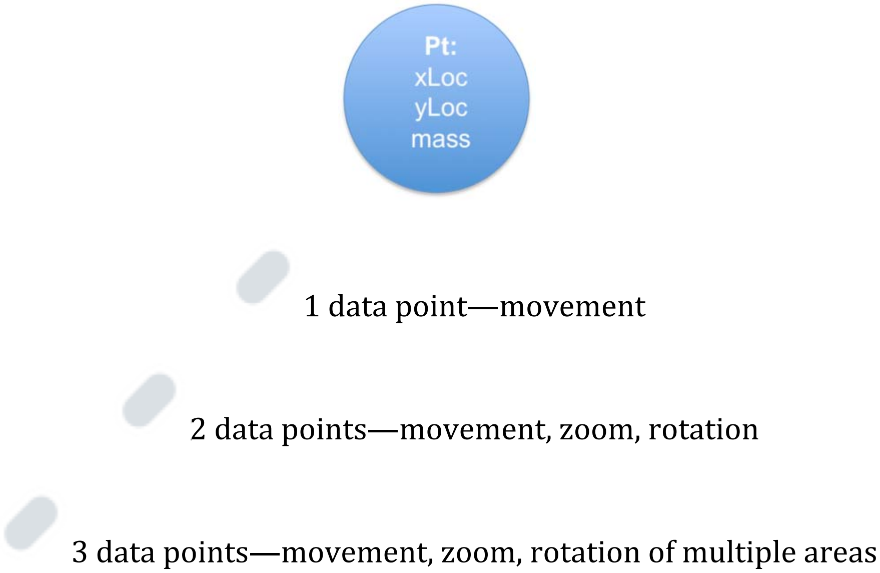
RGB pixel values are normalized, and skin pixels are found through a search. Expansion is performed about the skin coordinate to assess its size and center (d). This data is returned and the blobs are thresholded by mass to return the appropriate points.

This creates a method of recognizing and tracking the hands and head with little influence by lighting and environment conditions and little variation from person to person.



(d) Expansive Blob Detection

# Realtime Actions



**Pt:**
xLoc
yLoc
mass

1 data point—movement

2 data points—movement, zoom, rotation

3 data points—movement, zoom, rotation of multiple areas

Each data point returned from the image has a center and a mass. This information can be used to control actions through location and tracking. For instance, one data point can control movement on the screen (ie. a user can move a cursor around the screen with their hand). Two data points add more functionality, like zoom and rotation (ie. 2 hands can be used to zoom in and out of Google Earth). Three data points can be used for limited pose estimation, since we know the location of both hands and the head. These points are stored in separate lists of tracks and track data over time is used to recognize simple gestures like waving or up and down movements.

**Trajectories**

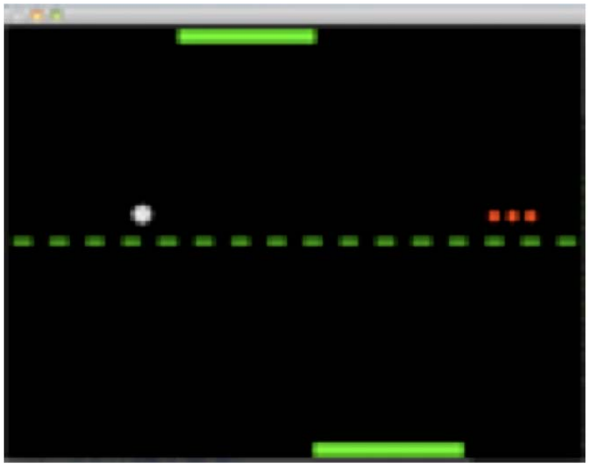
estimated through analysis of preceding frames

# Applications

There are many possible applications that can be created to use this or existing applications that can adapt to it. This is based on the number of data points available (ie. 1 hand = 1 data point).

*2D Pong*- 1 data point

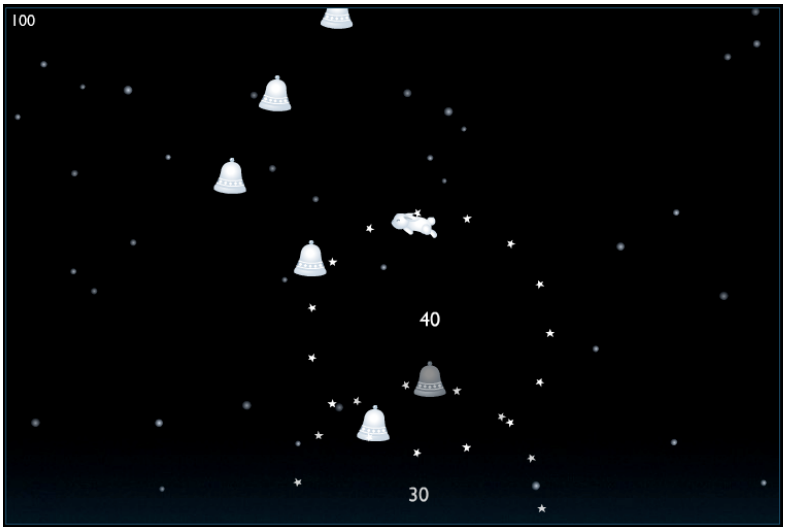
A player controls the pong paddle with their hand.


[1]

(1) Pong

*Interface for Flash Game- Winterbells*- 1 data point

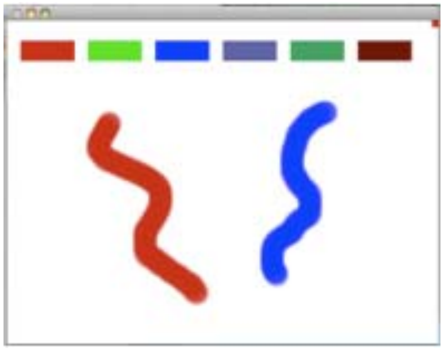
A player controls the cursor with their hand.



(2) Winterbells

*Paint Application*- 2 data points

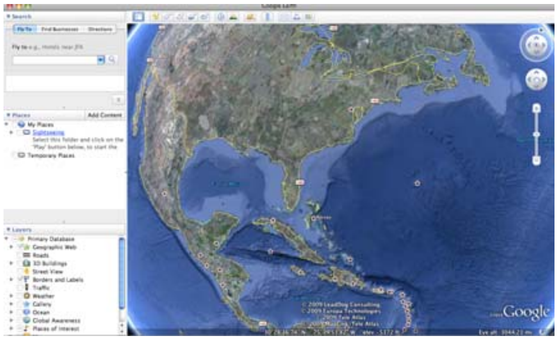
Paint Application where user can use both hands to paint.



(3) Paint

*Interface for Google Earth*- 2 data points

User can zoom in and out and navigate Google Earth using 2 hands.

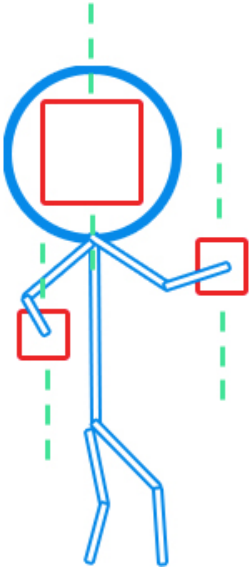


(4) Google Earth

*Interface for Flash Game Super Mario*- 3 data points

The two hands and head are located and tracked for this game. The player makes an actual running gesture (arms move up and down) to make Mario run and a real jump to make Mario jump.

(5) Super Mario Flash



(e) Super Mario Model

Three blobs are isolated and each corresponds to its own track list. This list of locations is analyzed over several frames to obtain an underlying movement which is them applied to the game—the up and down movement of the hands controls running and the movement of the head controls jumping.

# Implications for H-CI

There are interesting implications for user interaction using this model. For instance, although games can be more engaging this way, they can also be more emotionally stressful. For example, 4 subjects were asked to play Super Mario flash, first using a mouse and keyboard, and afterwards, using this interface. All users reported feeling more anxiety when Mario fell into a ditch (for a game over) when they were actually running and jumping then when they were pressing keys on a keyboard. Even in flash games, there is a deeper connection when actual full physical interaction by the user is involved.

# Future

Next steps for this project include expanding the list of body parts tracking robustly. This means adding joints like elbows to the recognition so that complete arm position can be determined. Arm configuration can lead to a series of new actions for games and applications . Also, adding lower body movements would create applications where a user used their entire body to interact with the application. Adding more data points using different techniques for recognition and tracking can be used to create a model for realtime pose estimation.

# Related Work

Many researchers are working on the problem for realtime pose estimation since its applications are varied and useful. There is need for these projects in the realm of assistive devices as well as for tools for learning.

Researchers are also working on practical implementations of existing methods of pose estimation of the human body and/or of just other objects in an image. For instance, Armon Miller at the Kindergarten group, MIT Media Lab created an interface that lets users interact with videogames and applications not with their body parts, but with any item in the room, like scissors or a notebook (referred to as “Hook-ups”).  His goal was to get people to move their bodies and stay away from the computer. These create more engaging interfaces and that can be used to create learning applications for kids.

# References

Hookups

http://llk.media.mit.edu/projects.php?id=1647

Maximum Likelihood Template Matching

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00854735

Pose Estimation Definition

http://en.wikipedia.org/wiki/Pose_(computer_vision)

Recognition Based Gesture Splotting in videogames

http://portal.acm.org/citation.cfm?id=1045943

Body Part Detection for Human Pose Estimation and Tracking

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04118819