Construction of a Database of Annotated Natural Human Motion and its Application to Benchmarking

Adam Chmielewski

Boston College

Computer Science Department

Senior Honors Thesis 2004

Advisor: David Martin

0. Abstract

Humans possess an uncanny ability to perceive the motion of other human beings. The performance of current algorithms on this task falls far short of human performance. In aiming to close this gap, we have collected video clips of natural human motion (e.g. walking, shopping, sports, etc.) and annotated figure tracks and head locations. We then use the annotated clips as a baseline to evaluate the quality of identification and tracking algorithms such as face detection and figure tracking. To date, we have developed annotation tools, collected and annotated 93 5-30 second clips, and evaluated two current face detection algorithms.

1. Introduction

One of the fundamental problems in computer vision and computer science in general is how to measure the effectiveness of algorithms. How can one objectively rate the performance of an algorithm? There are two approaches, one based on running time, the other, on how well the algorithm does what it is supposed to. This second approach is what this project is attempting to facilitate. Especially in computer vision, accuracy is an important method of evaluating an algorithm's effectiveness. Many different vision algorithms can be evaluated with respect to accuracy: Face Detection, Figure Tracking, Pose Estimation, and Activity Recognition are all important problems in vision that could make use of accuracy data. To measure this however, we need a ground truth, that is, we need human annotated data, as there is no current computer-based way of annotating the data that is anywhere near as effective or accurate as a human operator. The dataset we created in this project represents this useful ground truth to measure an algorithm's accuracy against. This gives us a means to compare the effectiveness of different algorithms. It also helps in the development of algorithms, as different versions of the same algorithm can be compared against each other to gauge effectiveness from revision to revision. For this project, we started collecting our dataset, built the necessary tools to annotate the data, and then put the data to use in comparing the effectiveness of two detection algorithms.

2. Video Data Collection

The video clip dataset is a major part of our efforts. The broad goal is to eventually create a database that contains examples of a multitude of human motions and actions, in as many different conditions as possible. Our specific goal at this time was to collect a decent number of interesting clips, with enough variety among them to do a fair benchmarking of an algorithm, and to have example clips for a lot of these variations. We looked for the following variations in both human activity and recording conditions and techniques:

Human-based variations:

- Number: This is simply a variation in the number of subjects in the frame for the duration of the clip, anywhere from one to thirty people is feasible.
- Age: We wanted to collect a full range of people, from very young children to the elderly.

- Height: The height of the subjects is an important variation, as tall people will perform the same actions in a different manner than shorter people.
- Shape: Similarly to height, we wanted a good variation in the size of our subjects, from thin to overweight.
- Skin tone: Especially important for evaluating face detection algorithms, we looked for a range of skin tones from very pale to very dark.
- Race: We worked to get a good variation in the race of our subjects, which was helped by the fact we did our filming in a large urban setting (Boston).
- Clothing: A person's garments make a significant difference in how successful an automatic tracking can be, and consequently we want different styles and amounts of clothing to be represented.
- Orientation: The direction a person is facing is important. For face detection, finding a profile face is similarly a different operation (depending on the algorithm) than finding a frontal face.
- Motion: Which way a person is moving (if they are moving) is another essential variation. For example, it is a different operation to track someone walking across the frame than someone walking from the foreground into the background. This was one area that we were especially successful in gathering, with many different vectors of motion represented in the dataset.
- Actions: There are a vast number of possible human actions, from the simplest (and more common) such as walking, to extremely complicated (and rarer) ones, such as juggling or tumbling. Even among the simplest actions there is large sub-variation; a walk could be anything from fast-paced purposeful stride to a

leisurely stroll. We set out to capture a number of the more common actions, and when the opportunity arose, to capture any rare or interesting actions that we happened across. Examples of the actions that we captured are listed in the descriptions of the clips later in this paper.

Condition based variations:

- Lighting: As with any visual medium, lighting is an essential part of the video-recording process. As such, we wanted good variation in the lighting conditions of our scenes: indoor vs. outdoor lighting, sunny vs. overcast outdoor scenes, day vs. night scenes, frontal lighting vs. backlighting, fully illuminated vs. dimly lit, and so on.
- Shadow: A subset of lighting, shadows are an extremely important detail in computer vision, providing depth and size information, and clips with shadows are an important part of our dataset.
- Weather: While the actual weather itself was not a variation we were looking at for this first dataset, it does effect many of the other variations, such as clothing or the lighting conditions. Rain or snow will introduce different effects on the frame.
- Occlusion: We collected a number of clips where the entirety of our subjects was not visible, such as subjects walking behind other subjects or behind a fence. This is an important variation to consider as these conditions often arise in applications.

Filming-based variations:

- Camera Motion: While some variation in camera motion is desired, we quickly determined that rapid or jerky camera motion produced uninteresting video frames with grotesque motion blur. With that in mind, several kinds of slow camera motion were used in a number of the clips. They include:
  - Panning: horizontal or vertical movement of the camera in the scene, without change in the viewing angle.
  - Tracking: movement of the camera forward or backward in the scene, for example, following someone from behind.
  - Rotation around: A small number of the clips involved moving the camera around the subject.
  - Stationary rotation: rotating the camera on the y-axis. This is similar to panning but different in that the camera's location is not changed.
- Zoom: similar to tracking, zoom was used in several clips to either change the size of the subject, or to keep the size of the subject the same as they moved in the scene.
- Camera Placement: This was used to create variation in the viewing angle. The camera was placed in many different ways: looking up at people, looking down on people, at eye level, on the ground but level, and so on.

Tools and Techniques:

One important aspect of our efforts was that we did not want our subjects to know they were being filmed, thus making the motions as natural as possible. This required that the filmer did their best to appear to not be filming. This was accomplished in a few different ways. The camera was often held at waist level, facing a different direction than the filmer was. This naturally resulted in a sometimes less than accurate aim of the camera, but for the most part none of the subjects noticed they were being filmed. Another technique was to leave the camera on a surface, with the filmer leaving it running and not touching it. People don't seem to consider the camera to be filming when there is no one operating it, and therefore it went largely unnoticed.

Video was collected using a fairly high-end handheld digital camcorder, a Sony DCR-TRV950, recording at 720x480 resolution onto mini-DV tape. The video from the tape was imported to our workstation using Apple's iMovie. The interesting and useful parts of the video were then identified, and separated into clips ranging from five to thirty seconds in length. They were then exported as numbered still frames in the jpeg format. At thirty frames per second, the clips ranged from 60 to 900 frames in length.

One unfortunate aspect of the filming process was that our video could not be collected in a progressive scan manner. That is, the video we collected was interlaced, with only half of the scan lines of a particular frame being present at a time. This presented a problem when exporting still frames, as they contained comb artifacts wherever there was motion. This was caused by the efforts of the computer to create a whole frame from two separate frames from different times by splicing them together, using the present scan lines from one to fill in the missing scan lines in the other. Obviously, this jaggedness was undesired, so we needed to deinterlace the video. After looking into various deinterlacing algorithms and tools, we determined that best approach would be to use a simple Matlab function. Our other options had deficiencies that made them unfeasible for our purposes, particularly that they often introduced artifacts into their output. This obviously was undesired, as we did not want anything that would hurt the purity of our dataset. Our deinterlacer avoids this problem by simply removing half of the scan lines by removing every other scan line. This leaves a vertically half-size, horizontally full-size frame, but one where all of the data is from the same moment in time. We then resize the frame via bicubic sampling back to its normal size. Since we are sampling from data already in the frame, this does not introduce any new artifacts to the data set, and does a reasonably good job of removing the effects of interlacing without degrading the image quality too much. The dataset consists of both the original interlaced frames (so that reconstruction of full video clip can be accomplished) and the converted, deinterlaced frames.

The Dataset:

At this time, we have collected 92 different clips of varying length that are representative of the types of variations we are collecting. Because several of the clips are of similar scenes, we have grouped them together as one family of clips for the purposes of characterizing and describing them.

<u>Description of each clip, with an example frame from each following the description</u>:

- Attitash1-2: these two clips follow a single woman as she walks away from the camera, outdoors, on an overcast day. There is camera motion in the second clip, consisting of panning and zooming to keep track of the subject as she gets farther away. Some occlusion.

- Attitash3: follows a woman carrying a large load of equipment, including skis and poles. Same weather/lighting conditions as the first two attitash clips. Does not zoom, but pans and follows woman as she moves in frame. Some occlusion.

- Attitash4: two figures, one removing objects from a car trunk, the other walks by carrying a pair of skis by his side. Some occlusion occurs as cars drive by in front of them, and one figure passes in front of the other. One wears dark clothing, the other bright clothing.

- Attitash5: five figures, all in ski clothes. Age variation, 3 children and 2 adults. Children sit on sidewalk, one throws stones. Follows one child as they walk and then sit down. Some camera movement and zoom-out halfway through the scene.

- Attitash6, Attitash 8: two similar clips, 2 figures, one reclines against a waist level fence, the other cleans and places his skis in his car. Some occlusion occurs as he walks behind the front of the car, and he can only be seen from the waist up. Many different poses for the male figure. Slight camera motion.

- Attitash 7: Similar to 6 and 8, also includes 2 children who run up and jump up onto the fence and walk along it. Slight camera motion.

- Attitash 9: Main figure is a child who is digging in the ground. Also contains a woman who walks across the frame, and a man who is bending over to put things into his car trunk and straightening up. Slight camera motion.

- Bicycle: Short clip of a man riding a bicycle. Camera pans and rotates viewing angle to track him. No occlusion. Bright sunny day, but cold weather clothes (hat, scarf, etc).

- Bu1-5: these clips are distinguished by the level of zoom and the individual figures in each one. Camera motion includes panning, rotation, and zooming.

They all contain a number of people walking along the street in one of two directions. Bright sunny outdoor conditions. Winter clothes. Some waist level occlusion. Some different actions while walking; talking to each other, using a cell phone, reading, etc. Great variation in age, height, shape, race, and skin tone.

- Bu6 -7: Similar weather conditions to Bu1-5. Lighting is not as bright as there are shadows from trees in the frame. These clips contain fewer figures, but at different depths in the field, and different vectors of motion. A few interesting actions occur, such as one figure stopping to light a cigarette. Slight camera motion, no zooming.

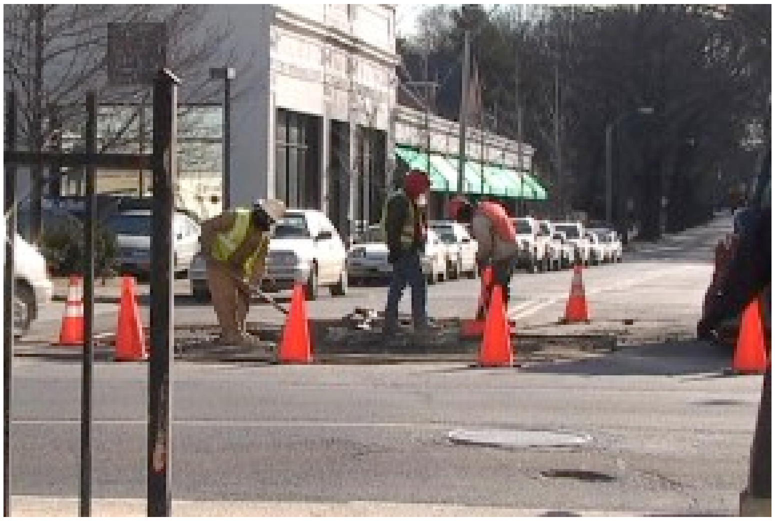
- Bustairs1-2: Same weather/lighting as Bu1-5. These sequences track a group of people as they climb a set of stairs (first one) and a single person as they climb the same stairs (second clip). Tracks from behind.

- Butrees1-5: similar to Bu1-7, with much darker shadows from the trees, and most of the lighting occurring from the rear or side of the scene. Many smaller figures in the background that would present a challenge for any tracking algorithm. Some camera motion.

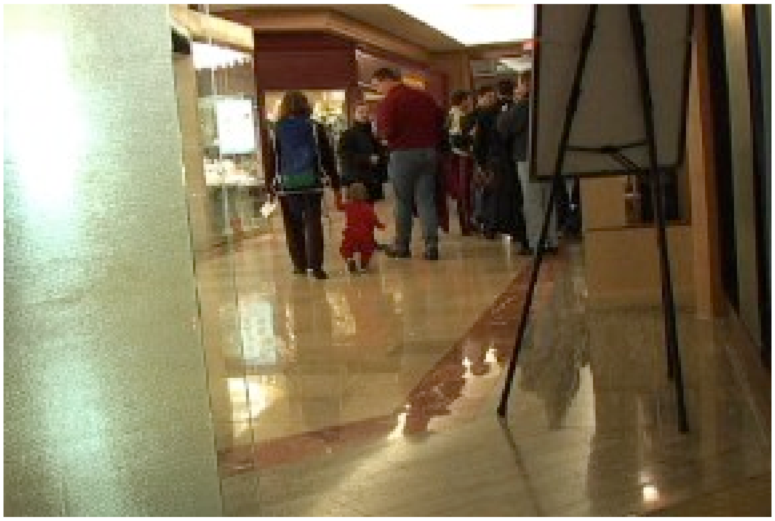
- Cardoor: tracks a man as he gets out of his car, shuts the door, and walks around and into a building. Another figure walks into the frame near the end. Bright lighting, sunny, outdoors, winter conditions. Lots of occlusion, and some camera movement, including panning, rotation, and zoom.

- Cards1-4: these four clips consist of figures walking through and interacting with an aisle of a supermarket. Some variation in the height of the figures. Actions include removing items from shelves and carrying objects. Bright, indoor lighting. Several well-lit, large faces are present in some of these clips. There is no camera motion, and the only occlusion occurs when figures cross in front of each other.

- Construction1-2: these two clips are of a construction crew repairing a road. Bright and sunny conditions, cold weather clothes. Scene is mostly backlit. Several interesting actions can be found: sweeping, shoveling, and raking of the ground. Second clip has a lot of occlusion as the same scene is filmed from behind a fence. Some camera zooming.



- Copley1-2: These are indoor clips of a large number of figures walking in a mall. Variety in age, there is a toddler in the first clip, and many different adults in both. The lighting is indoors, but not very bright, and the clothing is not as heavy as in other winter clips. No camera motion and a large amount of occlusion.



- Copley3-8: These six clips are all shot at about knee level. They are scenes from a busy crossway in a mall, with a large number of people in every clip. The lighting is brighter than copley1-2, but not very bright overall. Several people walk close to the camera, so that only torsos and upper legs are visible. Many different vectors of motion. Most figures are not wearing heavy jackets, but all are in at least long sleeve shirts and pants. Variation in most human characteristics, especially height and skin tone. Occlusion caused by figures.

- Copley9: same conditions as copley3-8, but shot from one floor above for variation in the viewing angle.

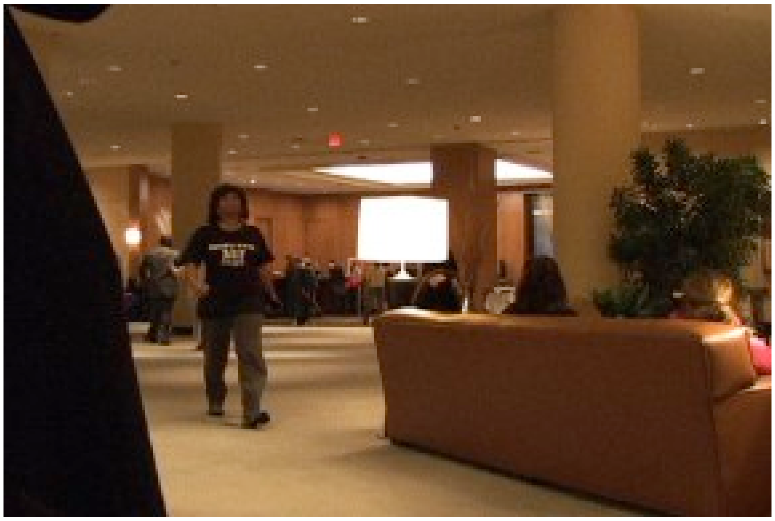
- Escalator1-2: these clips were an opportunity to have a vertically panning camera motion. Lighting is indoors but dimly lit, mostly from lamps at eye level. Contains variations in race and skin tone as well. Three figures make an interesting path around another group of figures. Same clothing as copley3-8.

- Foodcourt1-4: these four clips are good examples of both a looking-down viewing angle and interesting interactions, such as with a salad bar and a cash register. The scene is a large self-serve foodcourt. A large number of figures in many different kinds of clothing, at varying depths in the frame. Variation also exists in the level of zoom in each clip. Slight camera motion.

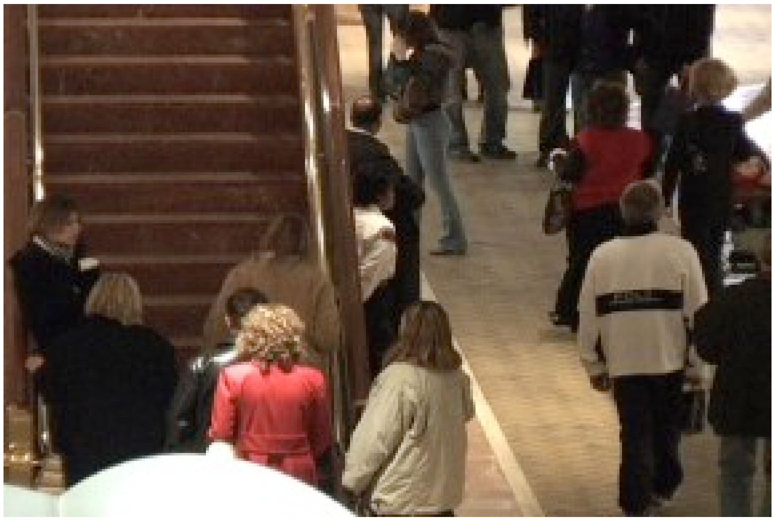
- Legs1-4: For these clips, the camera was placed at about ankle level on the side of a walkway. The scene is indoors but dimly lit. The figures are limited to about waist height at the most. This would particularly useful for studying the motion of the legs while walking. Different kinds of shoes are also present: boots, sneakers, heels, etc. No camera motion.

- Library1-4: these 4 clips are of a large entranceway to a library. Lighting is indoors and bright. Large variation in the types of people in the scene. Different motion vectors. Different poses. The third clip contains a woman with a seeing-eye dog. Good variation in the figures' depth in the frame. No camera motion.

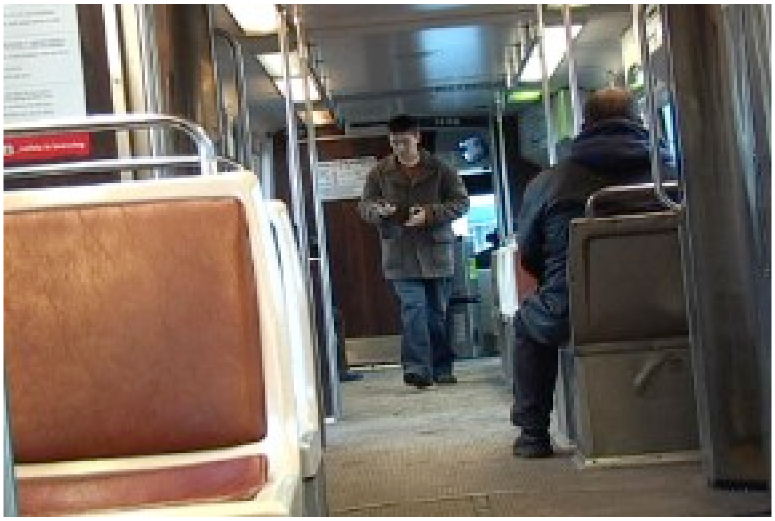
- Lobby1-2: Indoor, very dimly and backlit hotel lobby. Age variation, race, and especially shape variation. Child running in second clip. Clothing level ranging from jackets to t-shirts. No camera motion.



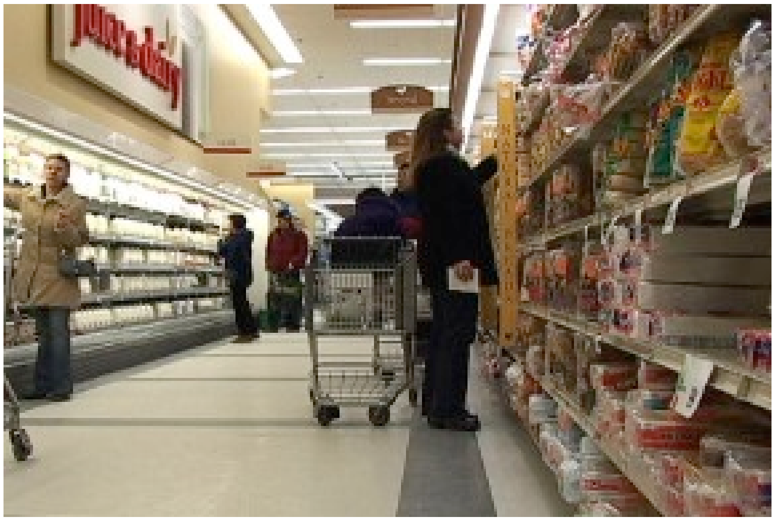
- Mallstairs1: This clip contains one of the largest numbers of figures. It is indoors, average lighting. Figures are both walking up and down a stairway, as well as walking flat along side it. With this number of people there is large variation in most human characteristics. A lot of camera motion, both panning and zooming. Viewing angle is also from above, as in foodcourt and copley9 sequences.



- Mbta1-9: this family of clips involves interactions and motion on a subway car. They contain 1 to 5 people. Because the train is moving, the shadows and lighting are always changing, creating some interesting effects as people go in and out of shadow and light. Motion of train also causes different kinds of walking than would flat ground. Age variation. Little camera motion.



- Milkaisle1-3: Scene is of a dairy aisle in a supermarket. Most figures are pushing shopping carts as they walk. Lighting is bright and indoors. Interaction with objects on the shelf and with the shopping carts. Figures walking almost directly towards and away from the camera. No camera motion, placed at waist level.



- Parkinglot1-3: These clips are the same conditions and people from the attitash4-5 clips. Parkinglot2 is an especially interesting clip, consisting of 8 figures of varying ages and sizes doing very different actions, including carrying objects, running, jumping off a fence and landing on the ground. Many different colors of clothes. Variety of poses. Some occlusion from objects. Slight camera motion.

- Runner: short (3 seconds) clip of a young woman running on the street. Outdoors, well lit. Cold weather, but tight clothing helps to display the motion of the limbs, especially legs. Some occlusion at the end, and the camera pans and rotates to track the subject.

- Shaws1: Clip consisting of four figures in a supermarket, a small corner of a bakery section. Indoors, average lighting. A couple object interactions. One figure has very bulky clothing. Some occlusion, only the head of one figure can be seen. Slight camera motion.

- Shaws2: Short clip of a man leaning on and pushing a shopping cart. Indoors, bright lighting. Some camera motion tracking the figure, and there is occlusion when the figure is behind the shopping cart.

- Shaws3-5: Camera was placed at eye level facing the refrigerated meat section of the supermarket. Scene is indoors, brightly light. A number of different types of people pass through the frame. Shaws4 and 5 contain a man interacting with a box (removing objects from and placing objects in) in some detail. Some occlusion. No camera motion.

- Shaws6, 8: Shaws6 has two figures, one of whom walks toward the camera and interacts with the objects nearby. Similar conditions to shaws3-5, different aisle. No camera motion or occlusion. Shaws8 is identical to shaws6, with more figures, and figures are pushing shopping carts.

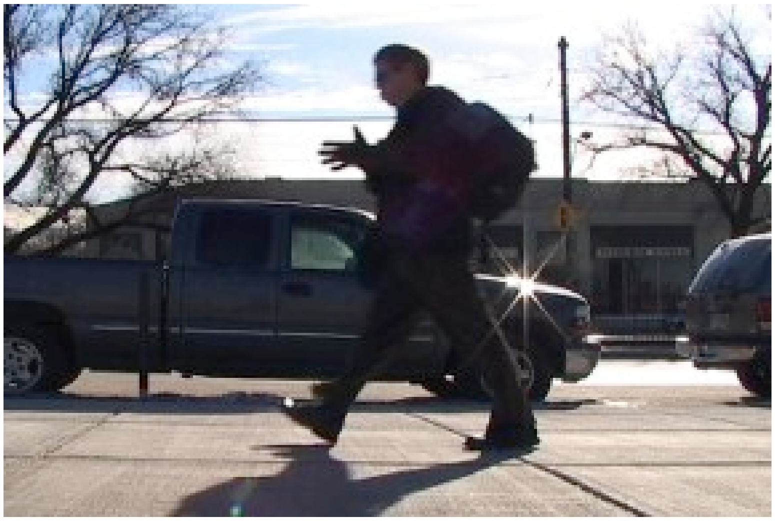
- Shaws7: Same location as shaws6, starts with a figure very close to the camera. Figure is carrying a pallet-like object on one shoulder. Grabs an object from off-camera and slowly walks away. This is the closest and largest face in the collection so far. Same conditions as shaws3-6. No camera motion or occlusion.

- Shaws9: Camera is held in the midst of a busy section of the supermarket as people walk closely by all around it. Indoors, brightly lit. Large variations in type of people and clothes. Some camera motion, including some jerky motion.

- Shawscorner1-2: Camera placed at intersection in supermarket. Many different vectors of motion and people changing direction. Large variation in clothing. Most are pushing shopping carts. Indoors, brightly lit. No camera motion.

- Skier1-4: these four clips follow a woman dressed in bright yellow ski clothes. She is wearing ski boots, which makes most of her motions very interesting. Note, these clips are out of order, skier4 was added later, but consists of the time before 2 and 3. Same conditions as the attitash clips. A lot of camera motion, panning, rotation, and zooming. A large amount of occlusion while tracking.

- Street1-7: Camera was placed at knee level facing a sidewalk as people walked by. Outdoors, bright day, but the scene is very backlit (the figures are almost silhouettes.) Winter clothing. Variation in the speeds of the walkers, but it is difficult to differentiate between people because of the lighting conditions. No camera motion or occlusion.



3. Video Annotation

The other aspect of our dataset is that it is human-annotated. We wanted to create data files consisting of annotations done by humans so that there would be a standard to benchmark tracking algorithms by. Ideally, these files would be annotated by many different people to average out any human differences or mistakes. There are many different kinds of annotations that we would like eventually have made, and two that we have implemented at this time. The two annotations that are currently implemented are full-figure tracking bounding boxes around each person in a frame, and face detection annotations, consisting of a face box and two eye points. Eventually, annotations will include:

- Head (not just face) location
- Individual limb and/or joint tracking
- Linking figure tracks from frame to frame
- Segmentation and Labeling of the actions

Because we wish to add future annotations, we have created our methodology to be incremental in nature. That is, both our tools and our data files are designed to allow additional annotations to be implemented on top of the current, existing annotations. Data files contain as many or as few types of annotations as have been completed for that sequence. If we wish to add another kind of annotation to the data file, it is simple matter of adding another element to our frame element in the XML tree. For example, once we have the figures annotated with bounding boxes around them, creating a figure track annotation is very easy, as we can write a function that just links up a bounding box in one frame to the corresponding bounding box in the next frame, and run that on top of the pre-existing annotation data.

The annotation tools created so far are Matlab functions: there is a function to do a particular kind of annotation on a single frame, and a main program that feeds each frame into the annotation functions. While it is possible to annotate every frame, we decided it was acceptable and much more feasible to annotate every 10th frame of a sequence. Therefore, the main program currently shows every 10th frame to the user and has them first annotate the all of the bounding boxes around the figures, and then annotate all the bounding boxes around the faces. Currently, there is a tool for annotating figure-tracking bounding boxes, and a tool for marking faces and eye locations. In both of these, if a particular frame has already been annotated, that data is displayed for the user, and they decide whether to keep the previous annotation or to redo the annotation for that frame. For rectangle-based annotations, the user selects points at the edge of a person, and the program constructs a bounding box based on the minimum and maximum X and Y values of these points. Keyboard commands signal to the program that the user has completed a frame, and wishes to go to the next. For rectangle-based annotating, we currently have undo and redo support for selecting the points used to construct the boxes. Eye points are selected by simply clicking on the eyes. The figures on the next page illustrate the tools in use.

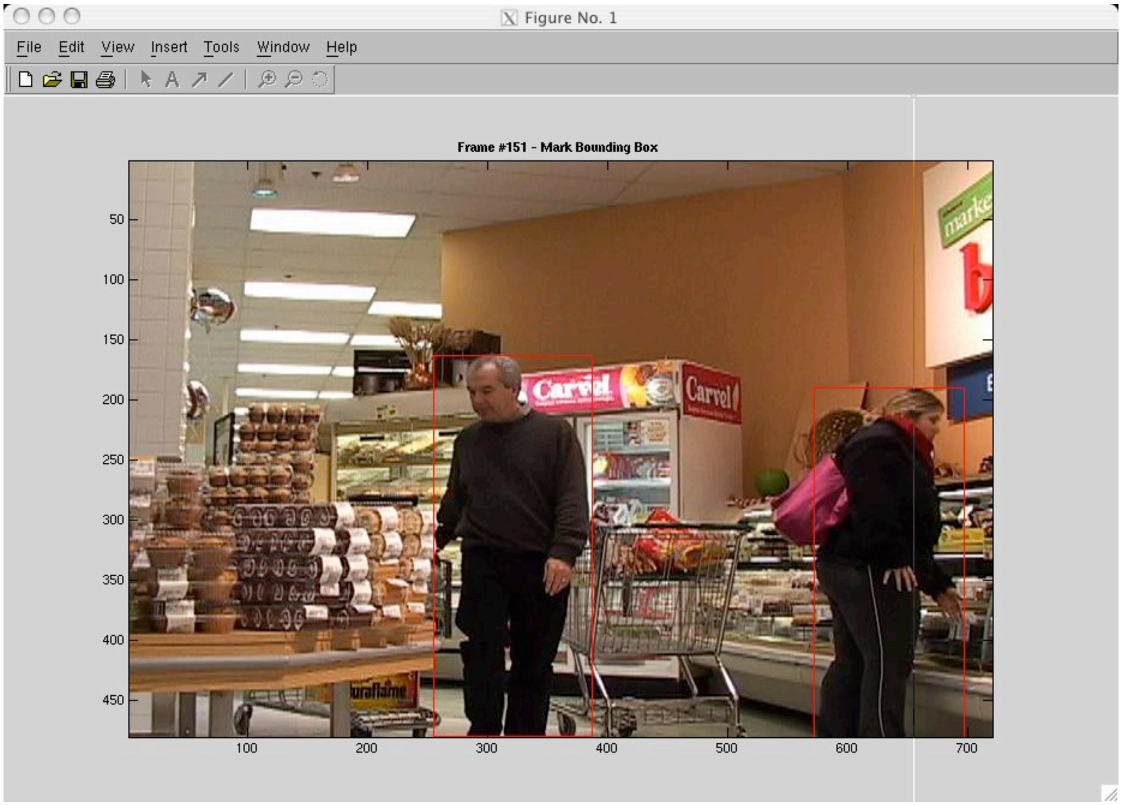


Figure No. 1
File Edit View Insert Tools Window Help
Frame #151 - Mark Bounding Box
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
700

Above: Bounding Boxes around subjects. Below: Face and Eye locations marked:

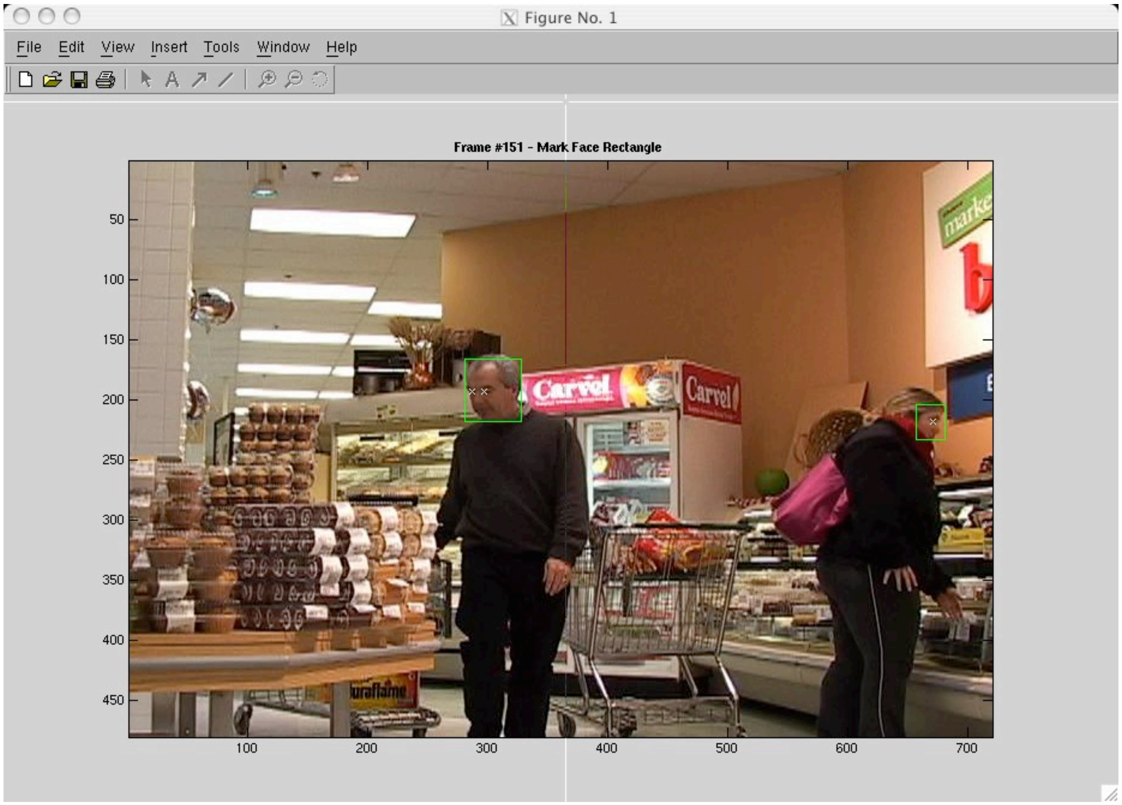


Figure No. 1
File Edit View Insert Tools Window Help
Frame #151 - Mark Face Rectangle
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
700

The data is stored in XML format, making it easy to read by both outside programs, and by humans themselves. There is one xml file for each sequence. All annotations for a particular sequence go in this file. Each frame is an element in the xml tree, and each annotation is a sub-element of the frame. Below is an example of an annotation file (just 1 frame is listed):

```
<?xml version="1.0" encoding="utf-8"?>
<sequence><!-- this is an xml representation of a sequence-->

  <sequenceheader isfaceannotated="true" numboundingboxes="2"
numcomplextracks="1" numfaces="0" numframes="61" numsimpletracks="1"
sequencedescription="girl running" sequencenum="1"><!-- Header info for a sequence--
></sequenceheader>

  <frame annotated="true" face-annotated="true" framenumber="1" numfaces="1"
numpeople="2"><!-- Frames have header info and people-->

    <person bbheight="153" bbwidth="79" bbxmin="3" bbymin="153"
complextracknum="1" personnumber="1" simpletracknum="1"/>

    <person bbheight="186" bbwidth="93" bbxmin="69" bbymin="155"
complextracknum="1" personnumber="2" simpletracknum="1"/>

    <face faceheight="22" facenumber="1" facewidth="19" facexmin="117"
faceymin="170" leftx="-1" lefty="-1" rightx="129" righty="178"/>
  </frame>
```

4. Benchmarking Face Detection

As an example of the potential use of this dataset, we used our annotation data to compare the performance of two different face detection algorithms. As a method of comparison of the effectiveness, we created precision-recall curves based on their results compared to the actual results from the same clips, human annotated. In detection, there are four possible cases to consider when evaluating an algorithms success rate.

A. True-Positive (Hit): The algorithm detected a face where there was a face annotated in the dataset.

B. False-Negative (Miss): The algorithm did not detect a face where there was one marked in the annotation.

C. False-Positive: The algorithm detected a face where there was no face marked in the annotation.

D. True-Negative: The algorithm did not detect a face where there was no face marked in the annotation.

PR curves use the first 3 of these cases to graph the effectiveness of the algorithm. The Y-axis is the precision, which is calculated as #A/(#A+#C). The X-axis is the recall (hit) rate which is calculated as #A/(#A+#B). Using different thresholds gives us the different data points necessary to make the curves. An ideal PR curve is one where the precision stays maximal for as far down the recall axis as possible. That is, the curve is 'pushed' to the upper-right corner of the graph.

Matt Veino's color-based face detection algorithm:

The first algorithm was created by Matt Veino as his Senior Thesis [Veino 2004]. The algorithm determines the "faceness" of areas of the image based on the A and B channels of the LAB color data of the image. It then finds the center of mass of the areas with the most faceness and returns those points as probable faces. We created simple Matlab functions to run this algorithm on our sequence of images, and compared the faces returned by the algorithm to the results from the annotated data set to get the precision and recall rates for different thresholds (ranging from .3 to .6). We evaluated on several sequences.

CMU face detection algorithm:

The other algorithm we evaluated was from the Robotics Institute at Carnegie Mellon University, developed by Henry Schneiderman and Takeo Kanade. They describe their algorithm as follows: “Our approach is to use statistical modeling to capture the variation in facial appearance. Currently, we use a set of models that each describe the statistical behavior of a group of wavelet coefficients.” – [Schneiderman/Kanade 2000] While the actual source code was not available, we were able to use an online demo of the algorithm to gather our results, located at: http://vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi.

The online demo is a CGI form that takes the address of an image on the web and a threshold, and runs the CMU face detection algorithm on the results overnight, before posting both the image with the faces highlighted and a data file with the coordinates of the faces it found. Because of the sheer amount of frames we wanted to run the algorithm on, doing this manually was out of the question. We created Perl scripts to automate both the submission of the form and the downloading of the results the next day, and used our workstation as a webserver so that the algorithm could download our images.

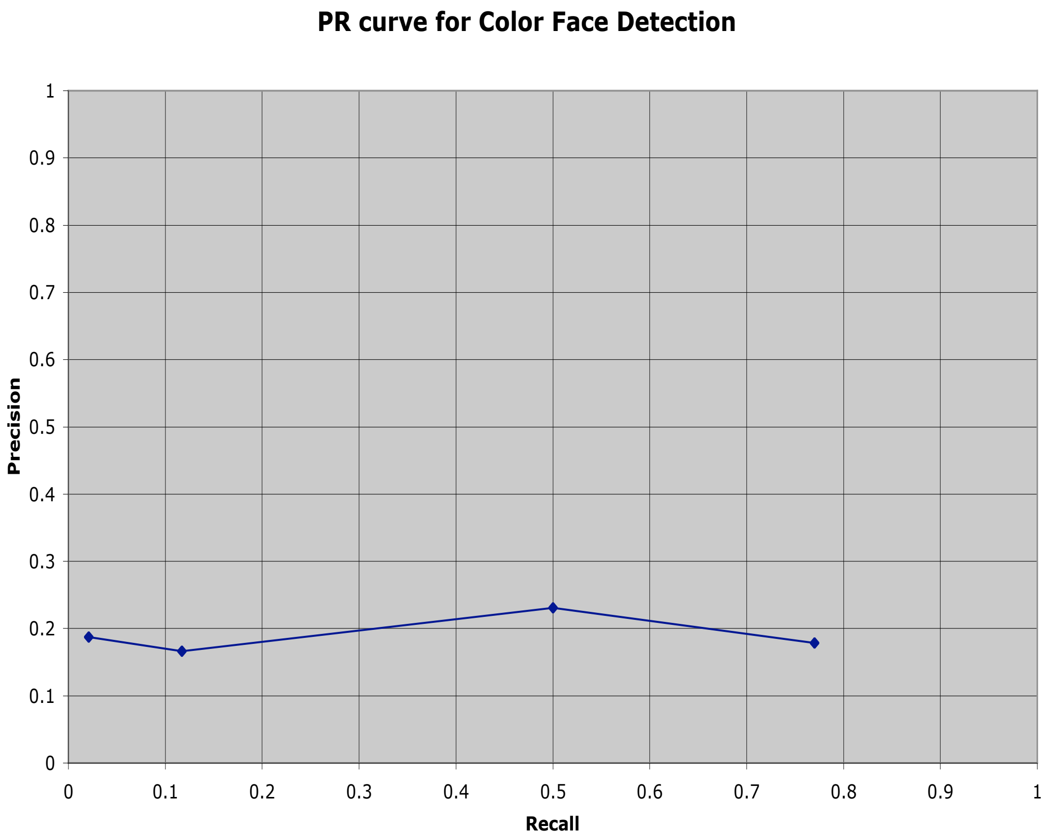
There were a few problems with using this algorithm, mainly that we could only run it once a day, and that it could not handle a large volume of submissions. Often, longer clips could not be completed without the algorithm stopping to work before finishing. Since we needed to run it on the same clip with different thresholds, it could take several days before we had results back for a single clip. For that reason, we were only able to completely evaluate the algorithm on 6 different clips, bicycle, bu1, bu2, shaws1, runner, and parkinglot2. We tested it with three different thresholds, 1.0, 1.75, and 2.5. According to CMU, the threshold affects the results in the following manner: "Decreasing the thresholds will increase the number of faces detected and the number of false detections."

Results of the Color Face Detector:

Numerical Results:

TP	FN	FP	Precision	Recall	Clip Name	Threshold
3	3	37	0.075	0.5	Runner	0.2
1	5	24	0.04	0.16666667	Runner	0.3
0	6	12	0	0	Runner	0.4
171	51	785	0.17887029	0.77027027	shaws1	0.3
88	88	293	0.23097113	0.5	shaws1	0.4
17	128	85	0.16666667	0.11724138	shaws1	0.5
3	141	13	0.1875	0.02083333	shaws1	0.6

Precision Recall Curve for Color Face Detector for the shaws1 clip:



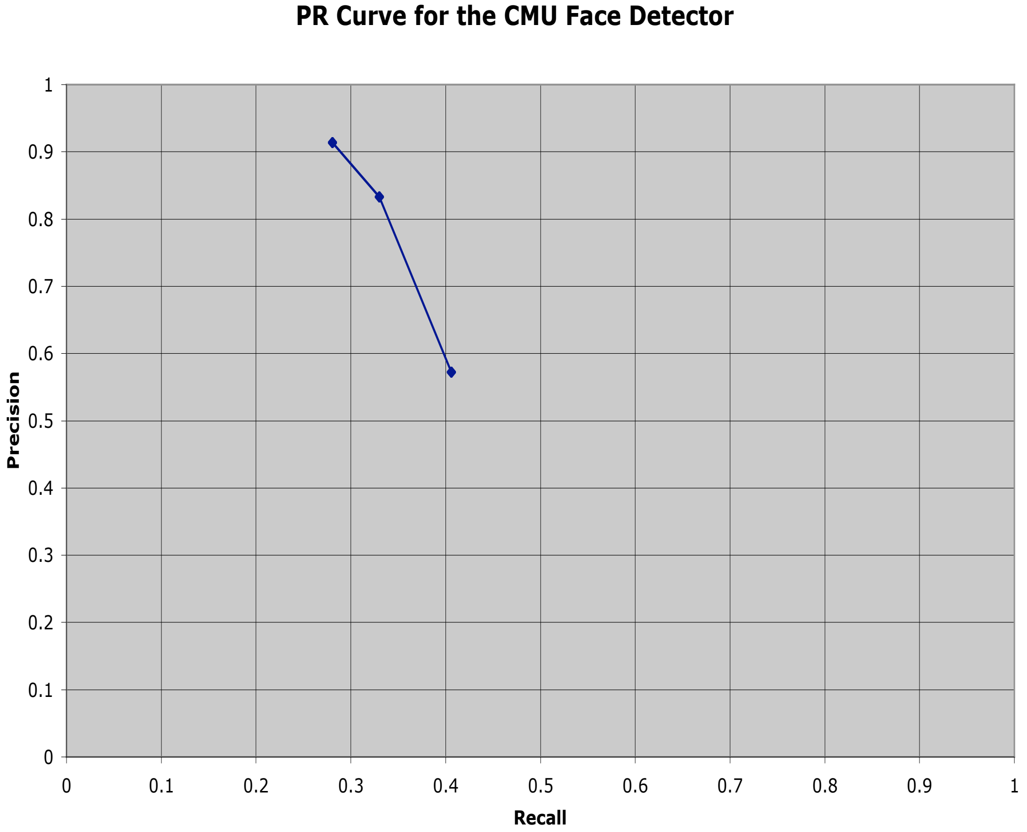
From the data, we can extrapolate several things about the Color Face Detector. First, it appears that the precision remains relatively constant across different thresholds. One possible explanation for this curve is that sometimes at more lenient thresholds the CFD finds multiple faces where only one face actually exists, lowering the number of false-positives, and therefore increasing the precision above what it should be. However, at stricter thresholds, the curve is in the correct form, as the multiple-face problem does not occur often. Also, we can see that we can increase the recall without affecting the precision too greatly, making the algorithm useful for finding a large number of faces.

Results of the CMU Face Detector**:**

Numerical Results:

A	B	C	Precision	Recall	Clip Name	Threshold
1	5	5	0.166667	0.167	CMU Runner	1
0	6	0	~	0	CMU Runner	1.75
0	6	0	~	0	CMU Runner	2.5
20	78	13	0.606061	0.204	CMU parkinglot2	1
13	85	0	1	0.133	CMU parkinglot2	1.75
8	90	0	1	0.082	CMU parkinglot2	2.5
78	66	44	0.639344	0.542	CMU Shaws1	1
71	73	12	0.855422	0.493	CMU Shaws1	1.75
65	79	3	0.955882	0.451	CMU Shaws1	2.5
0	9	3	0	0	CMU bicycle	1
0	9	0	~	0	CMU bicycle	1.75
0	9	0	~	0	CMU bicycle	2.5
24	22	27	0.470588	0.522	CMU Bu1	1
16	30	8	0.666667	0.348	CMU Bu1	1.75
12	34	5	0.705882	0.261	CMU Bu1	2.5
2	77	16	0.111111	0.025	CMU bu2	1
0	79	3	0	0	CMU bu2	1.75
0	79	0	~	0	CMU bu2	2.5
125	257	108	0.536481	0.327	All clips listed above	1
100	282	23	0.813008	0.262	All clips listed above	1.75
85	297	8	0.913978	0.223	All clips listed above	2.5
123	180	92	0.572093	0.406	All clips minus bu2	1
100	203	20	0.833333	0.33	All clips minus bu2	1.75
85	218	8	0.913978	0.281	All clips minus bu2	2.5

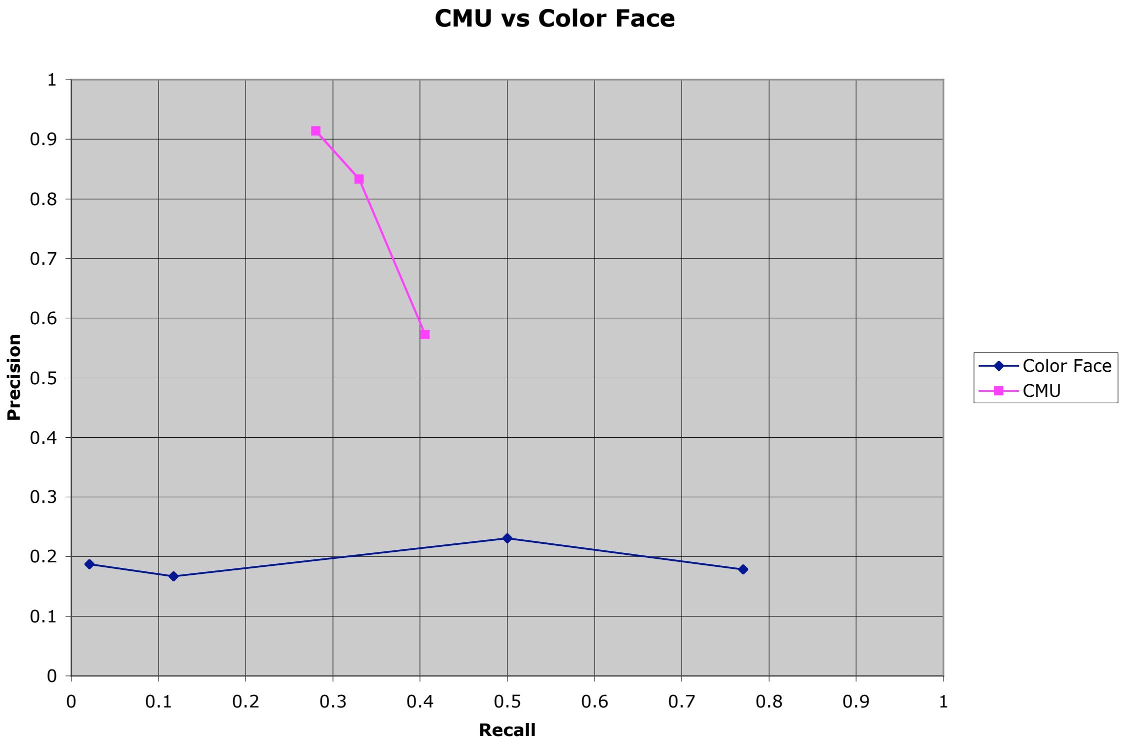
Precision Recall Curve for CMU Face Detector for the first five clips listed above:



As we can see from the data, the CMU Face Detector is heavily weighted towards precision. That is, the algorithm only detects faces it is very confident about, most likely the most obvious faces in a picture. There are not many false positives, but correspondingly, there is a large number of misses, most likely among the tougher to find or smaller faces. Therefore, one can be reasonably sure that something detected was in fact a face. One caveat with the algorithm is that it will not find very small faces, in fact, they have set a minimum size requirement of at least 20 pixels. Therefore, small faces in our annotation data will be not be detected by this algorithm and regarded as misses, reducing the recall rate.

Comparing the face detectors:

Overlay of PR curves for CMU and for Color Face:



When we look at the results together, we see how different the respective effectiveness's of the two algorithms are. Since the curves are so disparate, it is hard to say one algorithm is better than the other. Were we to compare two versions of the CMU detector, for example, we'd expect the curves to be much closer together and similar in shape, enabling us to more easily say one version is more superior (its curve pushed more to the upper right.) For these two algorithms though, we can only say that which algorithm is better depends on what the application is. For an application where you do not want to miss any faces, such as detecting terrorists, then the Color Face Detector might be the better algorithm. For an application where you don't want any false positives, then clearly the more precise CMU face detector is the better algorithm. An intriguing concept that this graph also illustrates about these two particular algorithms is that clearly combining the two approaches could be very effective, eliminating the weaknesses inherent in each.

5. Conclusion

While we have accomplished what we wanted to do for the scope of this project, there is more work to be done. Clearly, considering the size of the dataset we wish to establish, there is much more work to do in collecting video clips, especially in different seasons and locations than the ones that the majority of these clips were taken in. There are also many different types of annotations we would like to have tools for, so future work could involve developing these tools.

Overall however, we feel we accomplished the goals we set out to achieve at the start of this project. We have established the groundwork that will enable this dataset to grow quickly, as we have developed an intuitive methodology for turning raw footage into usable image sequences. The template for creating new kinds of annotation tools is present with the current figure and face location tools. We store our annotation data in a data format that is easily extensible and widely used. We have also demonstrated the usefulness of this project in evaluating algorithms. We were able to demonstrate objectively the effectiveness of two different detection algorithms, and have shown how this data set can be useful in evaluating many different computer vision problems.