The Effects of Capture Conditions on the CAMSHIFT Face Tracker

Michael Boyle
University of Calgary
2500 University Drive NW
Calgary AB T2N 1N4 CANADA
+1 403 220-3532
boylem@cpsc.ucalgary.ca

ABSTRACT

Face tracking—the continuous monitoring of head position, orientation, and geometry—has numerous practical applications for human-computer interaction, such as a perceptual form of multi-modal input. There are several non-invasive and computationally inexpensive techniques for face tracking that draw upon algorithms from computer vision. Of them, Bradski's CAMSHIFT algorithm is appealing because it requires minimal training. These techniques are particularly attractive in light of the growing installed base of fast desktop computers and cheap, low-end desktop digital video cameras. Low-end cameras, however, have characteristics that make them a poor fit for some such face tracking algorithms. In this paper, I introduce the problem of face tracking, provide an overview of the operation of CAMSHIFT as an example of a non-invasive vision-based face tracking algorithms, and describe my experiences attempting to employ video obtained from a low-end desktop digital video camera source in face tracking. I conclude this paper by offering conclusions and recommendations drawn upon my experiences.

Keywords

Face tracking; CAMSHIFT; digital video; computer vision.

1. INTRODUCTION

1.1 Motivation

Head and face tracking (henceforth, simply "face tracking") as a perceptual form of multi-modal input has many conceivable practical applications in HCI. Face position and orientation information is presently being applied to navigation and control in immersive virtual reality systems (e.g., caves). Similar applications exist for other sorts of highly visual, reactive environments, such as computer games, simulations, and visualizations of large data spaces. Head pose can also be used to garner crude estimates of focus of interest (FOI), that is, the location in one's workspace to which one's gaze is directed, although eye tracking is also needed when the particular application requires very accurate FOI estimates. This FOI information is very valuable, and can be readily applied to implicit user interfaces, in which human-computer dialogue evolves through the continual adjustment of computer operation in response to observations of human behaviour. Many things, from gaming to telecommunication to policing, stand to benefit from face tracking, so it is clear that this is a problem well motivated by practicality.

Although accurate measurements of face pose are best captured by specialized helmets [2] that are able to sample head position and orientation with six degrees of freedom (i.e., coordinates in the XYZ Cartesian space, pitch, yaw, and roll) hundreds of times a second, these and other hardware-based approaches are in many cases unfit for practical application. The sensitive hardware is expensive and has limited mobility, and thus is only suitable for tracking the face of a single individual in a small, specialized area. While this is fine for use with cavelets and similar VR environments, which are also highly specialized and immobile, the hardware is too cumbersome for most other applications.

Alternative approaches to tracking face pose and geometry are possible if one draws upon the techniques employed in computer vision. These techniques analyze video to locate the position and observe the orientation of faces found in the camera's field of view. They track by continually predicting and updating the pose estimates as each frame of video is encountered. The most accurate of such trackers require carefully fabricated camera angles and employ special markings on the face at various points [2]. However, as in the case of hardware-based trackers, these invasive techniques are too cumbersome to be employed in most HCI applications.

There are, however, many non-invasive vision-based techniques for tracking faces and recovering head pose and geometry information. It is these trackers that are the subject of this paper. Some require "training" on a per-person basis [2], others do not [1]. A face tracker that works well on any individual and needs little or no training is ideal for most HCI applications: to track a face requires only that it appear in the video sequence. Quite often, however, this ease of use must come at the cost of accuracy, and so the designer who wishes to leverage face tracking technologies will inevitably be forced to find a compromise—between accurate operation and the per-person investment of effort into training—that is most appropriate to the particular application.

2. CAMSHIFT

CAMSHIFT [1] is an example of a non-invasive, stochastic, model-free face tracker that incorporates computer vision techniques. Gary Bradski at Intel who evaluated its use in perceptual user interfaces developed it. The algorithm is representative of a class of algorithms that use the colour information in a video sequence to locate, and subsequently track a human face. Hunke and Waibel's algorithm [2] also falls into this class of face trackers. Many other "hybrid" face trackers employ this technique, as well. CAMSHIFT has certain properties that make it particularly suitable for HCI applications.

2.1 Colour model

To say that CAMSHIFT is "model-free" is not entirely accurate. It incorporates a probabilistic model based on colour. For every

possible colour, the algorithm assigns a probability that the colour will appear in the face to be tracked. Thus, the algorithm has a model of what a face should look like, at least in terms of its colour. This is typical of the entire class of algorithms of which CAMSHIFT is but one member. These algorithms use the model to compute the probability that a pixel in a frame in a video sequence is part of a face in the scene. The algorithm replaces each pixel in a video frame with the probability that it is a face-colour pixel, as given by a table look-up into the stochastic colour model. Figure 1 gives an example of a face and its corresponding face colour probability diagram. In this diagram, the brighter the pixel, the more likely it is to be part of a face.

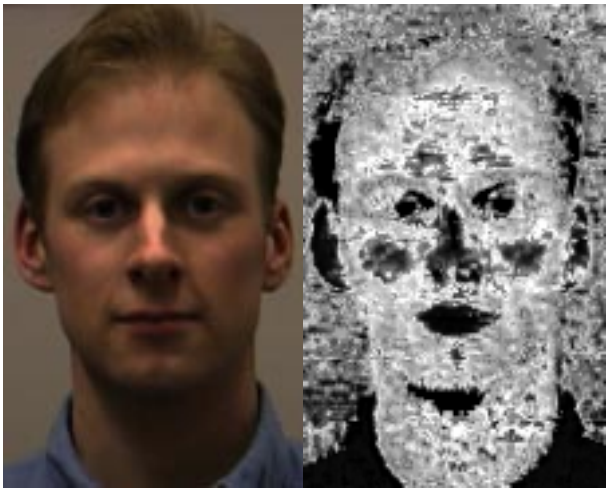

Figure 1. A face and its corresponding face colour probability diagram. Brighter areas indicate higher probability of being part of a face.

The histogram is computed by simply counting the number of pixels in an image of the face that have a given colour. The most frequently occurring colour is assigned the highest probability (1.0), and the probabilities of other colours are computed based on their frequency relative to that of the most frequently occurring colour.

There are several challenges to using colour as a stochastic model for face tracking. First, colour includes two components: chromacity (e.g., is it red or is it blue?), and luminosity (e.g., is it deep burgundy or bright candy-apple red?). Luminosity, in particular, fluctuates wildly in ordinary environments, even indoors. Second, not all people are the same colour: the colour model used by a face tracker must either be trained for a specific individual, or be flexible enough to accommodate individuals with varying skin colour. Worse, the skin colour of a particular individual may vary across the face (e.g., a pale-skinned man with a dark-coloured beard). Third, under extremely bright or extremely dim conditions, colour is very hard to accurately capture using common digital video equipment.

The CAMSHIFT algorithm is unique within this class because of the colour space it uses to compute this stochastic colour model. Most digital images are handled using the RGB colour space; but the individual R, G, and B components will vary widely under changing illumination conditions. To mitigate this problem, most stochastic, colour-based face trackers normalized the RGB colour components:

$$R' = \frac{R}{R+G+B}, G' = \frac{G}{R+G+B}$$

Bradski, in developing CAMSHIFT, however, observed that all humans (except albinos) are basically the same hue. Hence, he uses the HSV colour space as the basis for the colour model in CAMSHIFT. Hue, in particular, is ambiguously defined when the saturation or value are at either extreme. Thus, his model ignores pixels with very high or low saturation or value by assigning them zero probability. Similar truncations of a normalized RGB colour space appear in face trackers which use that colour model, instead.

As an acid-test to verify Bradski's claim that all humans are the same hues, the face colour probability histograms for some 12 individuals of varying racial backgrounds and skin colour (including Caucasian, Middle Eastern, East Asian, native North American, and African ethnicities, and, in fact, one albino) were captured and plotted (Figure 2). Indeed, with the exception of the albino and one other (Caucasian) subject, the histograms of the individuals very closely relate to one another. The large spike seen at the far right edge of Figure 2 was from the albino.

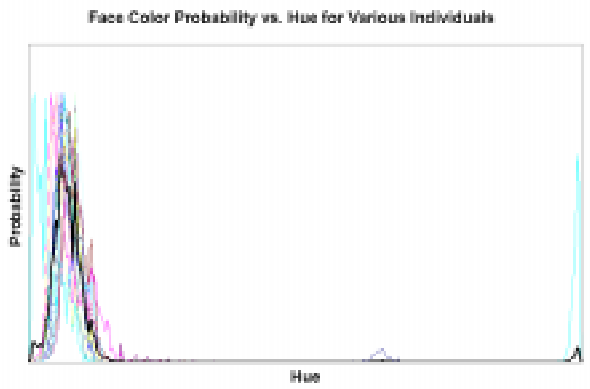

Figure ?. The face colour probabilities of various individuals extracted from an HSV colour model. The mean probability at each hue is shown as a heavier line.

Excluding these two outliers, the mean correlation coefficient between the histograms of any two of the individuals tested was 0.92. Single-factor ANOVA analysis at the 0.05 confidence level for a handful of these individuals shows that there are no statistically significant differences. Thus, we can safely conclude that Bradski's claim that all humans are basically the same hue holds some truth. This is important because now it may be possible to capture just one histogram and store it for use with all individuals.

2.2 Algorithm operation

After the face colour probability diagram for a frame in the input video sequence is computed, the main work of the CAMSHIFT algorithm takes place. As implied by the meaning of the CAMSHIFT acronym (Continuously Adaptive Mean Shift), it is a variation of the Mean Shift algorithm, a robust metric used by statisticians to compute the mean value of a variable, given the probabilities associated with each possible value.

The Mean Shift algorithm uses the zeroth and first moments of the face colour probability diagram to compute the centroid of an area of high probability. The algorithm requires the selection of the location and size of an initial search window. The zeroth moment

and first moments of the probabilities found in that region are then used to compute the centroid of a high-probability region within the search window. The search window is then centered over this centroid and the size is adjusted as a function of the zeroth moment. The algorithm repeats the computation of the centroid and repositioning and resizing of the search window until the result converges to some value, that is, changes less than some threshold fixed a-priori. Bradski recognizes that some implementations may chose to further limit the number of iterations to some maximum. Given a face colour probability diagram *I(x,y)* the zeroth and first moments are computed by the formulas given in Figure 3.

$$M_{00} = \sum_x \sum_y I(x,y)$$

$$M_{01} = \sum_x \sum_y yI(x,y)$$

$$M_{10} = \sum_x \sum_y xI(x,y)$$

$$(x_c, y_c) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right)$$

Figure 3. Equations used by CAMSHIFT to compute the zeroth and first moments and centroid of the face colour probability diagram *I(x,y)*.

CAMSHIFT uses the Mean Shift algorithm as its core, but expands upon it to use previously computed zeroth and first moment information to predict a good initial search window size and location. To accommodate motion, Bradski recommends setting the initial search window to an area slightly larger than that ordinarily computed by the Mean Shift algorithm. Bradski also points out the need to maintain an lower-bound on the size of the search window, otherwise the algorithm could degenerate to the smallest possible search window, and not correctly track a face as it moves.

CAMSHIFT takes the centroid (x_c, y_c) as the position of the face. The dimensions and its orientation are computed based on the second moments of *I(x,y)*, and are given by the equations in Figure 4. Together, CAMSHIFT can output from an input video sequence the position, dimensions, and orientation of a face in the scene. This information is popularly visualized as cross-hairs superimposed over the face, as in Figure 5. The angle at which the cross-hairs are set reflects the value of the roll computed by CAMSHIFT.

2.3 Modifications made

Bradski describes several "tweaks" or modifications to the algorithm to improve its performance. On the matter of ignoring pixels with extreme saturations or values, he suggests that pixels with saturations or values below 20% or about 80% of the maximum be discounted from the input

$$a = \frac{M_{20}}{M_{00}} - x_c^2$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c \right)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \quad \theta = \frac{1}{2}\tan^{-1}\left(\frac{b}{a-c} \right)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}$$

Figure 4. Roll *θ*, width *w*, and length *l* of the face as computed by CAMSHIFT.

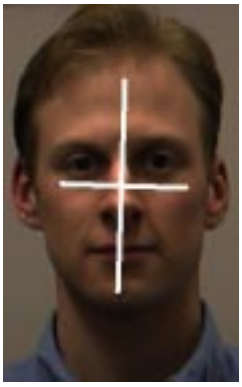

Figure 5. Visualization of the face position, dimensions, and orientation information output by the CAMSHIFT algorithm.

The size of the search window, both initially and as the Mean Shift component of CAMSHIFT proceeds, is very important. Bradski suggest setting the width *s*, to:

$$s = 2\left(\sqrt{\frac{M_{00}}{256}} \right)$$

The height of the search window is set to 1.2*s* because, Bradski argues, human faces tend to be slightly elongated along their vertical major axis. The zeroth moment is divided by 256 because Bradski uses 8-bit integers to store the probabilities, so that they occupy the range [0,255].

It was discovered, after implementing the CAMSHIFT algorithm as Bradski described it, that some modifications were necessary to the procedure by which the search window size is set. First, the search window was found to be too large; best results were obtained when using a value for *s* that is one-half that which Bradski suggested. Second, the behaviour of the algorithm once the track had been lost was found to be unsatisfactory.

When the track is lost, CAMSHIFT, as described by Bradski, hopelessly fixates on some small region of the background. Even if a face were to reappear in the scene, it would not influence the size of the search window, and thus the track would never be reacquired (or, more precisely, a new track would never be acquired).

The possibility that the track could be lost as people move in and out of the field of view is quite high in most HCI applications of face tracking, and so to mitigate this problem, I modified CAMSHIFT so that if the search window size degenerates to its minimum (3×3 pixels), then the search window is inflated to fill the entire image. Of course, if no face appears anywhere in the source image, the search window will continue to degenerate again, back down to its minimum. This cycle of degeneration-inflation-degeneration continues until eventually a face reappears in the scene and a new track can be acquired.

2.4 Application to an HCI problem

In his discussion of CAMSHIFT, Bradski evaluates the technique for use in a perceptual user interface. He specifically links face motion to control of a 3D flight simulation and a computer game. Bradski reported good success in his attempt to employ face tracking in an human-computer interface.

Originally, my intention was to apply this algorithm to the problem of managing sub-conversations in multiparty videoconferences. The basic idea is to split the screen into four regions, one for each of four outgoing videoconferencing links. If it were possible to determine which of the four remote parties a participant was presently looking at (thus talking with), then the audio of a that participant could be piped at its fullest quality to the interesting remote party, while the three other remote parties receive muffled or no audio. When the FOI could not be adequately mapped to a particular remote party, all would receive full quality audio. Thus, the system could support sub-conversations within a multiparty videoconferencing application.

However, two things became immediately apparent after using the Logitech QuickCam VC (a low-end desktop digital video camera) with CAMSHIFT. First, face pose and orientation is insufficient to recover even the gross FOI information demanded by this application. To highlight this, observe Figure 6. It shows a subject looking at each of eight different compass directions on a display centered around the camera. Note how terribly difficult it is to determine from the image the direction the subject is looking. That is, head pose alone is insufficient to recover FOI.

Second, low-end desktop digital video cameras and the conditions under which they are used have certain properties that will affect their use with face trackers. A comparison study grew out of these experiences to further explore this second point.

Figure 6. Shot taken from an over-the-monitor angle of a subject looking at each of eight different compass directions on the display. The mapping of direction the subject was looking at the time the snapshot was taken and position it appears in this figure has been randomized to further illustrate the difficulty of this problem.

3. COMPARISON STUDY

The remainder of this paper is a discussion of observations made after attempting to use a Logitech QuickCam VC—a low-end desktop digital video camera—with the CAMSHIFT face-tracking algorithm. This comparison study examines the effect on the CAMSHIFT algorithm's success of the characteristics of the camera used to capture live video for face tracking and the circumstances under which the video is captured. The parameters specifically addressed are:

- Frame size;
- Frame rate;
- Sensor element used (e.g., CCD, CMOS);
- M-JPEG compression[1] quality; and,
- Global illumination conditions (e.g., indoor/outdoor, dim/bright).

The two camera types examined are the Logitech QuickCam VC (as an example low-end camera) and the Canon XL-1 MiniDV camera (as an example of a high-end camera). The QuickCam uses CMOS-type sensor elements; compression is performed in hardware using the VIDEC codec, which is a variant of M-JPEG; the parallel port version offers limited bandwidth (942 kbps), and thus only small frame sizes and slow frame rates can be used. The Canon XL-1 uses a professional quality CCD sensor; the MiniDV format uses MPEG compression (which is similar to M-JPEG; and, as the video is written directly to MiniDV tape, it was digitized post-hoc using a Truevision Targa board at 24 fps and normal NTSC video frame sizes (720×486 pixels).

[1] Although there is no M-JPEG standard, the term M-JPEG typically refers to an encoding scheme which JPEG compresses each frame independently. Thus, an M-JPEG compression video stream is merely a sequence of JPEG-compressed frames.

The comparisons that follow should be regarded with some skepticism as the underlying data sets are small and the cameras being compared are at opposite ends of the performance spectrum. Still, it is important to recall that the fact that a difference between the parameters examined is not the important message to be communicated. Rather, it is the implications these differences have on the algorithms and their application that is of particular interest.

In most cases, video was captured under each of the comparison conditions. Then, CAMSHIFT was run on each video sequence, and the results were eyeballed for gross differences. In some cases, the impact the change in one parameter had on the face colour probability histograms generated under CAMSHIFT (using the HSV colour model) and other algorithms (using a normalized RGB colour model) was illustrated for comparison.

3.1 Sensor element used

3.1.1 Motivation

As previously explained, there are significant differences in the performance of higher-end cameras that use CCD-type sensor elements and lower-end cameras that use CMOS-type sensors. The CAMSHIFT and similar algorithms are particularly vulnerable to the poor quantum efficiency of CMOS sensing elements because they rely much more heavily on colour information, and thus require accurate colour reproduction. This comparison is motivated by the fact that CMOS cameras are significantly less expensive (and thus more prevalent in lower-end cameras) than CCD cameras, and so it is desirable to use CMOS cameras if the type of sensing element has no impact on the output of CAMSHIFT and other similar algorithms. This comparison is intended to establish if there is a difference in algorithm operation between the two camera types.

3.1.2 Methodology

Video sequences of the same individual were captured using both the XL-1 and the QuickCam cameras. The images were captured in front of the same location, at approximately the same angle (head-on, face-only shot), under very similar lighting conditions. Four tests of the CAMSHIFT algorithm were run. In each test, one of the input sequences was selected as the source for the face colour probability histogram, and one (possibly the same) sequence was selected as the input to be tracked. Face colour probability diagrams of the sequences under each of the four histogram source/input source conditions in both the HSV colour model and a normalized RGB colour model were computed.

3.1.3 Results

Figure 7 shows a sample of the output of CAMSHIFT under each of the four test conditions. Although the centroid of the face was correctly found (within reason) in all test conditions, the orientation and dimensions of the face were only computed correctly in under the QuickCam histogram source/QuickCam input source condition. In fact, the worst performance was obtained when the histograms were extracted from the XL-1 source, regardless of the actual input source.

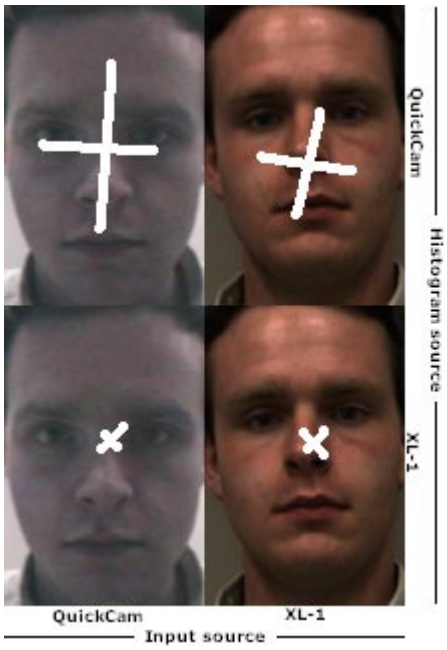

Figure 7. Output produced by CAMSHIFT algorithm under the four different histogram source/input source condition pairings.

This surprising result prompted further investigation of the histograms used during tracking. Figure 8 shows the face colour probability histogram computed from the QuickCam and XL-1 sources. Seemingly contrary to what was seen in the output of the CAMSHIFT algorithm, the histogram under the QuickCam case is extremely noisy; so much so, that the correlation coefficient between the two histograms is only 0.10, suggesting that they are unrelated. Indeed, a single factor ANOVA statistical test at the 0.05 confidence level shows that the histograms are significantly different. Figure 9 shows a face colour probability diagram under each of the four test conditions. The “noise” seen in the QuickCam histogram is readily apparent in these diagrams; the face is most clearly discernable under the XL-1/XL-1 test condition. If the histogram extracted from the QuickCam source was so noisy, why then was superior output obtained using it?

No satisfactory explanation could be found for this observed behaviour. Furthermore, attempts to reproduce the experiment using the faces of other individuals did not result in the same behaviour. For these other individuals, the XL-1/XL-1 condition produced output as good, or better than the QuickCam output. It appears as though the face used in initial testing produced an outlier effect.

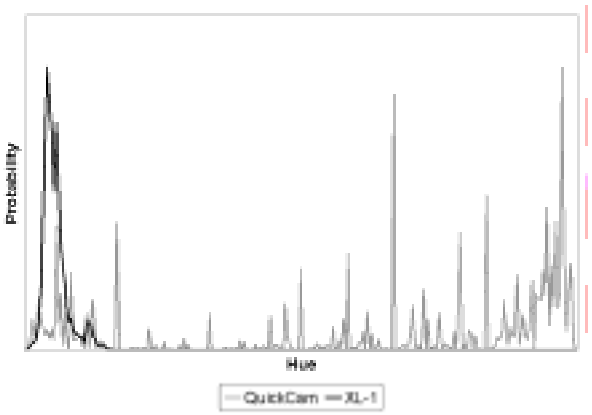

Figure 8. Face colour probability histograms extracted using an HSV colour model from video sequences captured using the QuickCam and the XL-1.

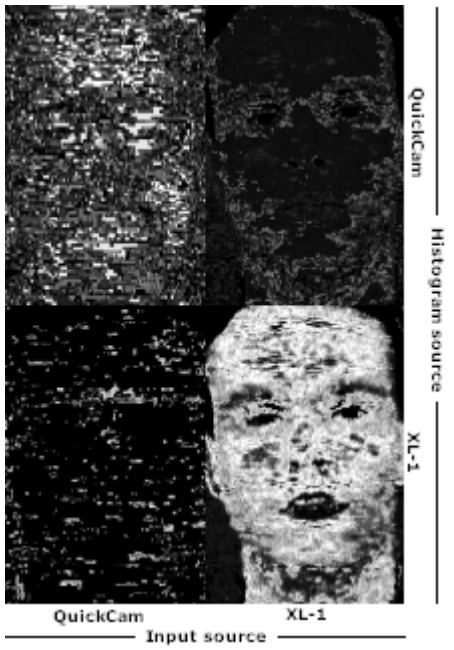

Figure 9. Face colour probability diagrams extracted under the four different histogram source/input source condition pairings, using the HSV colour model.

Lastly, the histograms collected using the HSV colour model were compared against those collected using a normalized RGB colour model. Figure 10 shows the histograms themselves, and it is readily apparent that those collected from the QuickCam source do not overlap with those collected from the XL-1 source. The face colour probability diagrams produced using the normalized RGB histograms are shown in Figure 11 and more vividly demonstrate the difference: a face cannot be discerned when the histogram is extracted from a camera different than the input source. Obviously, the differences that exist between CCD- and CMOS-type sensing elements are more pronounced when using a normalized RGB colour space.

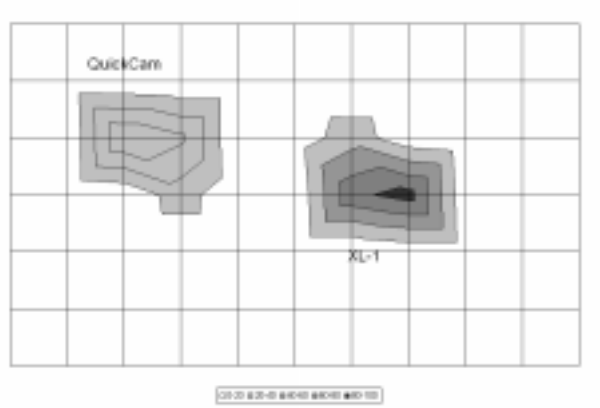

Figure 10. Face colour probability histograms extracted using a normalized RGB colour model from video sequences captured using the QuickCam and the XL-1.

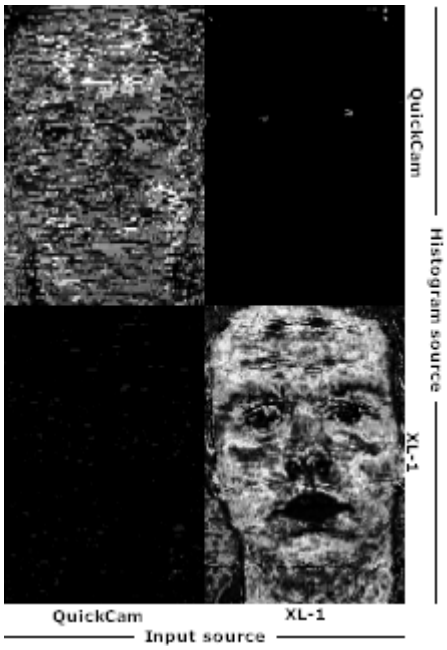

Figure 11. Face colour probability diagrams extracted under the four different histogram source/input source condition pairings, using a normalized RGB colour model.

3.1.4 Conclusions

We have found that the face colour probability histograms extracted from input taken from a CCD-type camera are far less noisy than the histograms extracted from input taken from a CMOS-type camera. This is the expected result, and the expected conclusion to be garnered from it is that one should prefer CCD-type cameras over CMOS-type cameras for face tracking purposes. Surprisingly, however, this is not true for all individuals, though it was true for all but one of the individuals used in informal comparison tests. The more interesting result is that we have also shown that using histograms extracted from images sourced at a CCD-type camera does not improve algorithm performance when the input actually used is from a CMOS-type camera. Best operation will be obtained if one extracts the histograms from the same model of camera as that which will be used for input.

Unfortunately, this requirement is tantamount to training, and thus negates a strong point in favour of the CAMSHIFT technique; until now, we have seen evidence that it would be possible to completely remove all training at the user's site, and instead use a single, generic face colour probability histogram for any arbitrary individual. One potential workaround not explored is to include a different generic face colour probability histogram calibrated for each model of camera (or, at least, type of sensor) and choose one of these at run-time depending on the actual camera used for input.

3.2 Frame size

3.2.1 Motivation

This comparison is motivated by two reasons. First, if algorithm output is independent of frame size, or, more precisely, the use of smaller frames results in negligible difference in algorithm output, then it would be preferable to use the smaller frame size because the video could then be processed at a higher frame rate (or, the processing at a given frame rate would require fewer resources). Second, it superficially appears as though the QuickCam has poor colour response at smaller frame sizes; the video appears richer in colour as the frames are made larger. This comparison is intended to establish if the observed difference in colour response is, in fact, real.

3.2.2 Methodology

Although the QuickCam can capture video at several different frame sizes, the video from the XL-1 could only be digitized at the normal NTSC video frame size. Thus, it was discounted from this comparison. Head-on, face-only video sequences from the QuickCam were produced at three different sizes: 640×480, 320×240, and 160×120 pixels. The face colour probability histograms produced from each source and the output of the CAMSHIFT algorithm were compared.

3.2.3 Results

Figure 12 shows compares the face colour probability histograms under each condition. Contrary to perception, there is little difference in these histograms; indeed, the mean standard deviation of the face colour probability at any hue was computed at 7.6 out of a maximum of 255.

The output produced under each case, however, differs rather dramatically. Figure 13 shows the output of CAMSHIFT when the input is derived from each of the frame sizes. It is apparent that smallest frame size produced the poorest output, even under such idealistic circumstances as those tested. This difference might be a result of the way the colours over a region of camera sensor elements are sampled to produce the smaller image. As it turns out, the sensing element on the QuickCam VC is only 352×288 pixels, and hence the largest format tested in this comparison is merely scaled up from a smaller source, while the smallest format tested is scaled down from a larger source. Although the down sampling does not appear to significantly affect the face colour probability histogram derived from the smaller format, the effect is significant enough to upset the CAMSHIFT algorithm output.

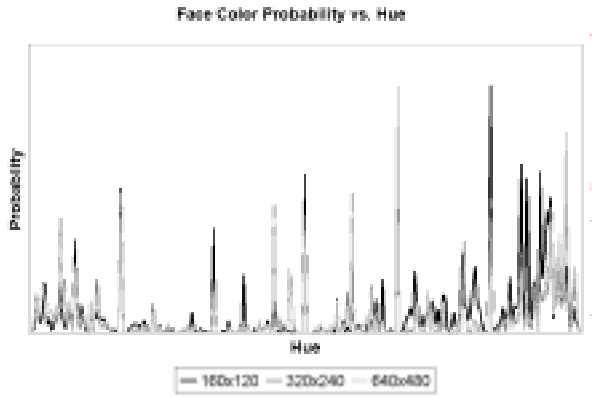

Figure 12. Face colour probability histograms for the same individual when source image captured using QuickCam at three different frame sizes.

Figure 13. Output of CAMSHIFT on (from left to right) 640×480, 320×240, and 160×120 pixel-sized sources.

3.2.4 Conclusions

Frame size will have an impact on the output of CAMSHIFT or other similar face tracking algorithms. Nothing is to be gained when using images that are scaled up from smaller sources; much is to be lost when using images that are scaled down from larger sources. It is best to choose an input frame size that matches the number of sensing elements on the camera surface.

Note that these results (and the recommendations generated from them) assume that field of view is independent of frame size, that is, no additional scene areas come into view by using larger frames. This may not be true for all cameras, particularly those that offer hardware zoom, tilt, and pan. It is certainly preferable in most HCI applications to use larger frame sizes if they afford a wider field of view.

3.3 Frame rate

3.3.1 Motivation

This comparison is motivated by two considerations: low frame rates increase the risk of losing the track, and low-end cameras cannot support high frame rates. A dilemma results, that pits the algorithm against the camera. Ideally, a lower frame rate is preferred if it does not substantially impair algorithm operation because it lessens data transfer and compute bandwidth requirements. Thus, this comparison is intended to establish the significance of impact the image frame rate has on the output of the CAMSHIFT and similar algorithms.

Most of the face trackers examined use prior estimates of face pose and geometry as initial predictions in a fashion consistent with other kinds of single hypothesis trackers (e.g., Kalman filter). If the frame rate is quite high, however, then the tracking algorithm on the host computer side becomes the source of a

bottleneck. If the frame rate is quite low, then the distance the centroid of the face being tracked can move between any two successive frames could end up being quite large. If the distance is too large, the track will be lost. In section 2.3 I mentioned that CAMSHIFT will hopelessly flounder indefinitely once a track is lost.

Low-end desktop digital cameras use slower data transfer interfaces like USB-1 or standard parallel port interfaces. At USB-1's fastest bus speed (12 Mbps) [4] an uncompressed 24 bpp QCIF (176×144 pixel) video sequence could be transmitted at a maximum of 20 fps; parallel ports are even slower, and could only accomadate a paltry 1.5 fps! Higher-end cameras use interfaces based on the IEEE 1394 standard, which has a peak bandwidth capacities of 400 Mbps, so that same uncompressed 24 bpp QCIF video sequence could be transmitted at 30 fps using only 22% of the bandwidth available on the IEEE 1394 bus. Realistically, most cameras incorporate hardware compression, and so the actual bandwidth requirements are in fact much lower: the parallel port QuickCam VC I used transmits up to 12 fps (1:8 compression). This hardware compression, though, becomes a major bottleneck in the video pipeline. Indeed, many low-end cameras lack the on-board compute muscle to compress video at 30 fps.

3.3.2 Methodology

A video sequences captured on the XL-1 (high-end camera) was digitized at 24 fps at 320×240 pixel frame size, and then reduced to 10 fps, 5 fps, and 1 fps. The video sequence was taken in an outdoor setting with adequate lighting. It begins with a close-up of the actors face, to seed the face colour probability histograms. The camera then zooms out to approximately 5 m away from the actor, who casually walks to the left and exits the scene. The CAMSHIFT algorithm was run on each sequence, and the results were compared against each other.

3.3.3 Results

In all cases, the CAMSHIFT algorithm continued to track the location (i.e., centroid) of face quite well after the camera zoomed out to a distance of 5 m. However, even at 24 fps, the roll computed varies quite widely when the size of the face within the image is decreased. Length and width information, while remaining roughly approximate to the observed dimensions of the face, is nonetheless skewed because of the miscalculated roll.

CAMSHIFT was found to adequately track the face in the test scene at 10 fps: the track was maintained so long as the face remained within the field of view. At the 5 fps and 1 fps frame rates, however, the track was lost and was not recovered. The track was lost at approximately the same time within the sequence (5 s) in all conditions. Figure 14 shows the frame at which the track was lost in the 1 and 5 fps conditions, and how at the 10 fps condition it was still preserved.

Further to this, a fourth frame rate condition was introduced at 8 fps, and the CAMSHIFT algorithm was run on it. As in the 5 fps case, the track was lost at close the 5 s time point.

Figure 14. Frame at which track was lost under (from top to bottom) the 1 fps, 5 fps, and 10 fps frame rate conditions.

3.3.4 Conclusions

Information about the orientation of the face (e.g., roll) computed by CAMSHIFT is unstable and unreliable when the size of the face within the image (i.e., the number of pixels that make up the face) is quite small. This is to be expected, however, as if we were to relate this back to the equations in Figure 4 used to compute the roll, we would see that as fewer pixels are used to compute the zeroth and second moments of the face colour probability diagram, the impact small errors in the image will have increases.

More importantly, however, we can see that the paltry frame rates offered by low-end digital cameras will fail to keep up to the motion within the scene in some cases. The video sequence tested here featured casual motions by a single actor; more realistic situations, which could feature distracters or occlusions, will most certainly fail if the frame rate is too low. For the motion examined in this study 10 fps was sufficient to maintain the track, yet 8 fps

was not. This is precariously close to the maximum frame rate offered by the QuickCam VC.

Two recommendations come out of this comparison. First, use the fastest frame rate possible: opt for a more expensive camera if it will offer faster frame rates (without compromising on other parameters discussed here). Second, evaluate the speed of the head motions you expect to encounter in the scenes you will capture, and evaluate your choice of frame rate carefully before committing to a decision.

3.4 M-JPEG compression quality

3.4.1 Motivation

We have already established that higher frame rates are preferable (section 3.3) but given the bandwidth constraints found in most low-end digital video cameras, compression must be employed. M-JPEG compression is particularly troublesome for CAMSHIFT and other such algorithms that build stochastic models out of colour information because the incredible compression rates obtained with this codec come at the expensive of chromacity information. The compromise on fidelity in the chromacity channel in favour of higher fidelity in the luminosity channel is based on the fact that the human visual perception system is more sensitive to local luminosity changes than to local chromacity changes. Ideally, we would like to compress the video as much as possible without substantially affecting algorithm output; this is particularly important in security applications that place the host computer at a significant distance from the actual camera.

3.4.2 Methodology

A single head-on face-only video sequence was captured using the Canon XL-1 and digitized to 640×480 pixels. Each frame in the sequence was then compressed using the IJG JPEG implementation [3] at 50% and 10% quality levels for a total of three different input sequences. (100% quality refers to the original, uncompressed input sequence.) The 50% compression quality level results in a 1:46 bit compression ratio, and 10% produces a 1:88 bit compression ratio. Figure 14 shows the three input sequences. JPEG compression artifacts are virtually indistinguishable at the 50% quality level, but are readily visible at the 10% level.

Figure 14. Source input video sequences, shown (from left to right) at uncompressed, 50%, and 10% JPEG compression quality levels.

Face colour probability histograms were extracted under each condition. The CAMSHIFT algorithm was run on each of the input sequences a total of three times, once for each of the three probability histograms extracted.

3.4.3 Results

Figure 15 shows the face colour probability histograms extracted under each of the three compression qualities. Generally, the histograms have high probability hues clustered in the same region, but both the 50% and especially the 10% quality histograms seem to suffer from large spikes. This visual similarity is deceptive, though: the correlation between the histogram taken from the uncompressed source and the histogram taken from the 50% compressed source is 0.85, but is 0.26 with the 10% compressed source's histogram.

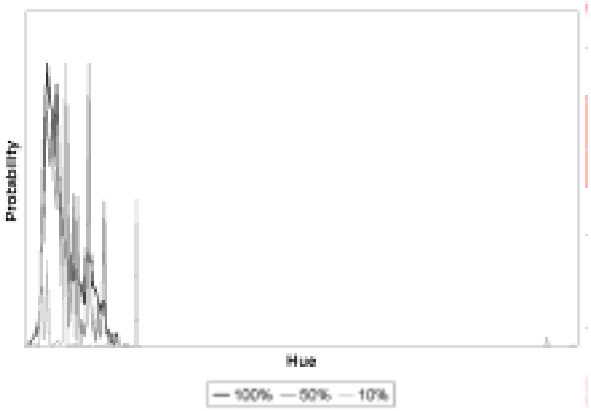

Figure 15. Face colour probability histogram extracted under each of three JPEG compression qualities: uncompressed (100%), 50%, and 10%.

Figures 16 and 17 show 3×3 "montages" of the face colour probability diagrams of each source input image when the histogram is seeded from each of source input images. Images in the same row are seeded with the same histogram; the rows are ordered (from top to bottom) as 100%, 50%, and 10%, and the columns are ordered (from left to right) as 100%, 50%, and 10%. The histograms in Figure 16 are computed from an HSV colour model; the histograms in Figure 17 are computed from a normalized RGB colour model. The extent to which chromacity information is lost during the JPEG compression process is immediately evident in these figures: note the extensive blocking on the 10% compressed images. Note also that under the normalized RGB colour model, hair was incorrectly given a high probability when the input is from a 10% quality compressed source, regardless of what quality is used to seed the histograms.

Figure 18 shows a 3×3 montage of the output produced under each of the histogram source/input source conditions. The ordering of histogram sources by row and input sources by column are identical to the previous two figures. What is immediately striking here is that in the first row, the output appears generally unaffected by the input source condition. The compression quality used on the histogram source appears to be the dominant influencing factor. Indeed, when the face colour probability histograms are seeded from a 10% compressed source, the CAMSHIFT algorithm is wholly unreliable.

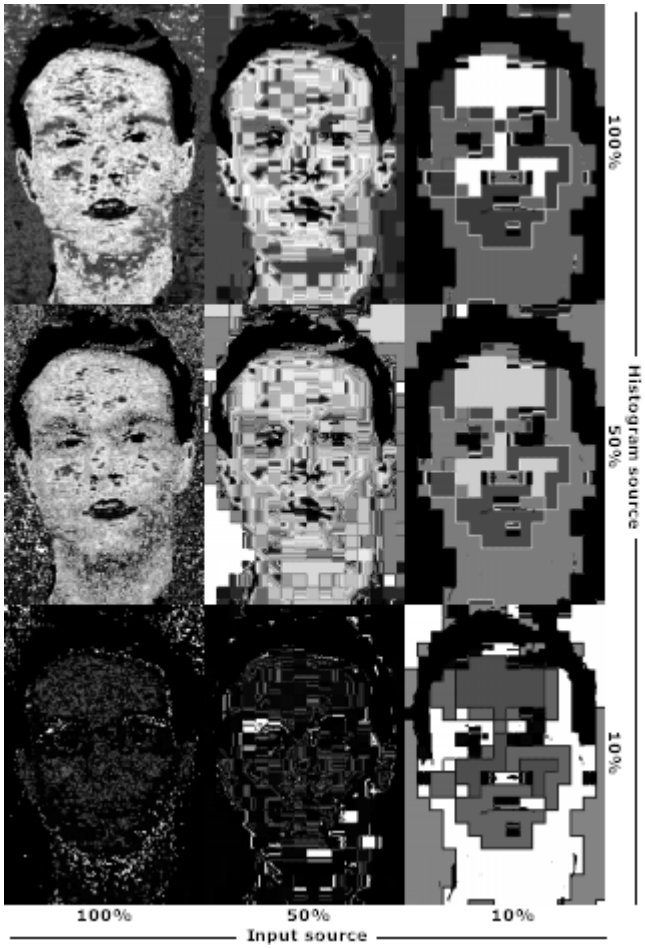

Figure 16. Face colour probability diagrams produced using an HSV colour model under each of the nine histogram source/input source pairings.

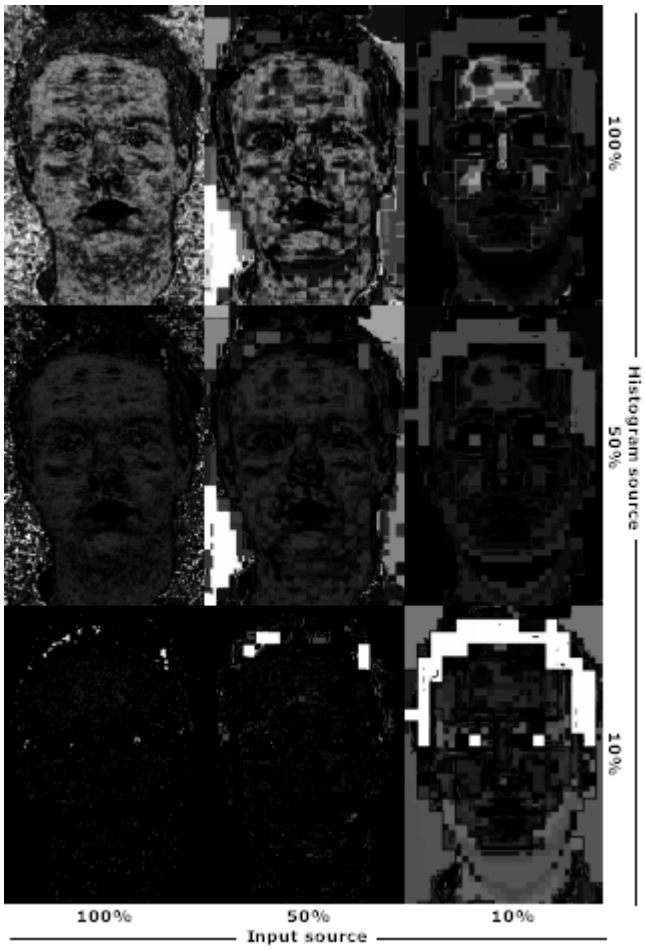

Figure 17. Face colour probability diagrams produced using a normalized RGB colour model under each of the nine

3.4.4 Conclusions

These results point to two significant conclusions. First, the HSV colour model touted by Bradski for use with the CAMSHIFT algorithm fairs much better on compressed images than the normalized RGB colour space used by nearly all other similar face trackers. Second, and likely resulting from the first, CAMSHIFT output is not substantially affected by input compression, so long as the face colour probability histograms are taken from uncompressed images. The recommendation that falls out of this second point is that although one should certainly seed the stored histograms from uncompressed images, some hardware M-JPEG compression (even a 1:40 bit compression ratio) is acceptable. Compression quality can be used as a balancing factor when considering frame size and rate against bandwidth considerations.

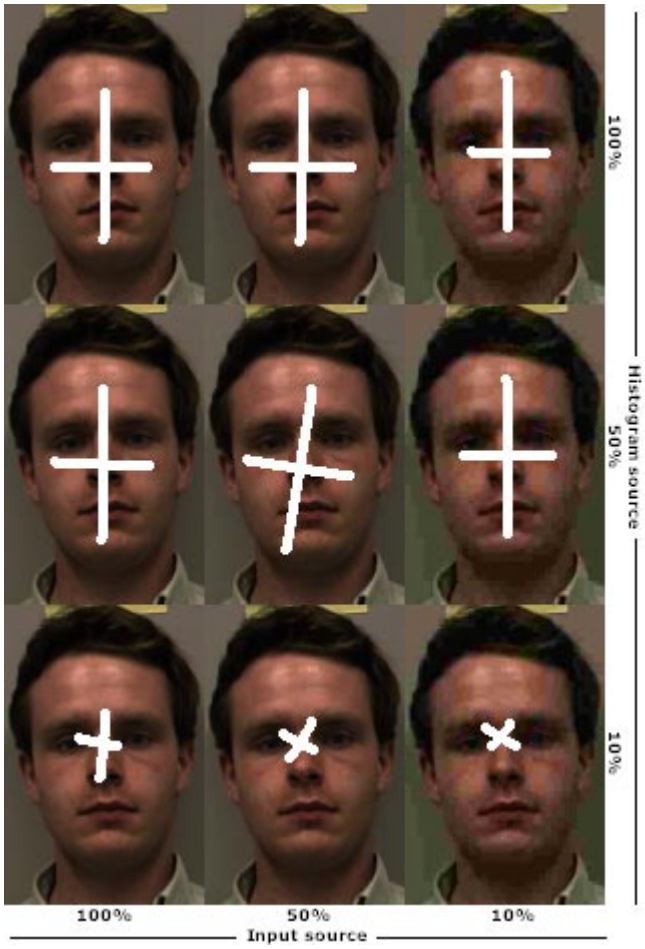

Figure 18. CAMSHIFT algorithm output under each of the histogram source/input source conditions.

3.5 Global illumination (spectra)

3.5.1 Motivation

It is well known that the EM spectrum of the light source used will affect the perceived colour of an object viewed under its light. Since CAMSHIFT and other such algorithms rely quite extensively on accurate colour reproduction, the effect that the spectra of the light sources used to illuminate a scene becomes a serious consideration. This comparison is intended to establish what impact the source of the light used to illuminate a scene has on CAMSHIFT performance.

3.5.2 Methodology

Using only the XL-1 camera (since it is portable), video sequences of an individual were captured under indoor and outdoor lighting conditions. The lighting indoors used so-called "cool" white fluorescent lighting operating at 60 Hz; the outdoor condition was taken on a sunny April afternoon with no artificial lighting. The face colour probability histograms under each condition were correlated against one another, and the CAMSHIFT algorithm was run on each sequence. Unfortunately, a photometer was not used during filming, and it may very well be the case that we are also testing for the effects of global illumination changes here. The same individual was used in both the indoor and outdoor scenes, but the individual filmed was forced to squint in the outdoor conditions because the sunlight was so bright. Consequently, face pose is different in the two scenes.

3.5.3 Results

The face colour probability histograms extracted under indoor and outdoor lighting conditions using the HSV and normalized RGB colour models are visualized in figures 19 and 21. The HSV colour model histograms (Figure 19) overlap a great deal; indeed, the correlation coefficient between the indoor and outdoor histograms is 0.91. This suggests that the HSV colour model is quite robust against the sort of large-scale source lighting emission spectra changes that we are testing for here. In contrast, however, we see very little overlap in the histograms computed using a normalized RGB colour model (Figure 20). The correlation coefficient between these two histograms was low: 0.21. This strongly suggests that the normalized RGB colour space is a poor choice when the input is subject to source lighting emission spectra changes. Figures 20 and 22 show comparative montages of the actual probability diagrams produced; the face is clearly discernable under all conditions in the HSV model-based montage, but barely visible when the indoor image probabilities are computed from the outdoor image under the RGB model and vice versa.

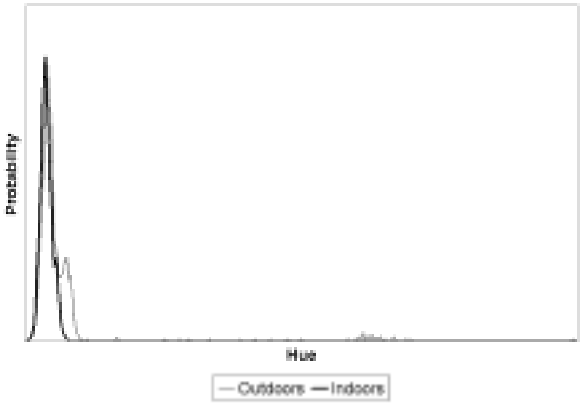

Figure 19. Face colour probability histogram extracted using the HSV colour model under indoor and outdoor lighting conditions.

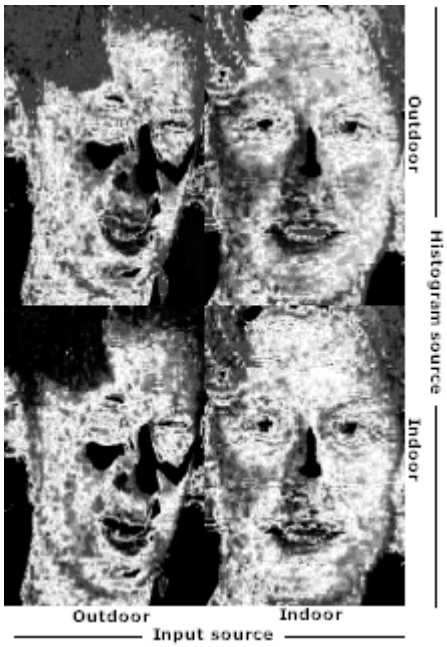

Figure 20. Face colour probability diagram montages for all histogram source/input source indoor/outdoor conditions under the HSV colour model.

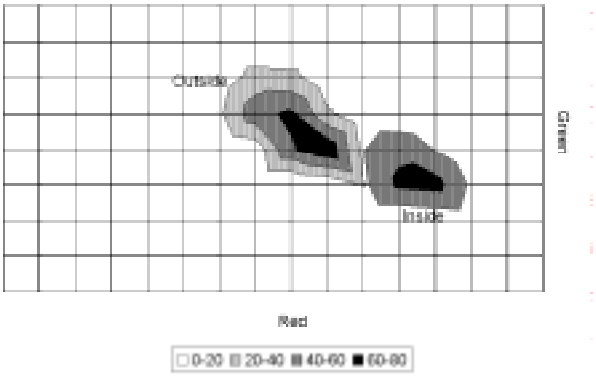

Figure 21. Face colour probability histograms extracted using a normalized RGB colour model under indoor and outdoor lighting conditions.

Quite expectedly, the robustness of the HSV colour model under the two lighting conditions translated to good CAMSHIFT output performance, regardless of which lighting condition was used to seed the histograms. Output was largely identical under either histogram source condition.

3.5.4 Conclusions

Clearly, if there is a risk of global illumination spectra changes, then it is preferable to use the HSV colour model as it performs well regardless of the spectra used when histograms are computed or when input is captured. That said, there is a small difference in the performance of the output under the different lighting conditions: the histograms taken under indoor fluorescent lighting suffer less from noise than those taken under natural sunlight, and so performance of the CAMSHIFT algorithm varied slightly. One possible recommendation out of this is to try and seed histograms under the same lighting spectra conditions as the input source.

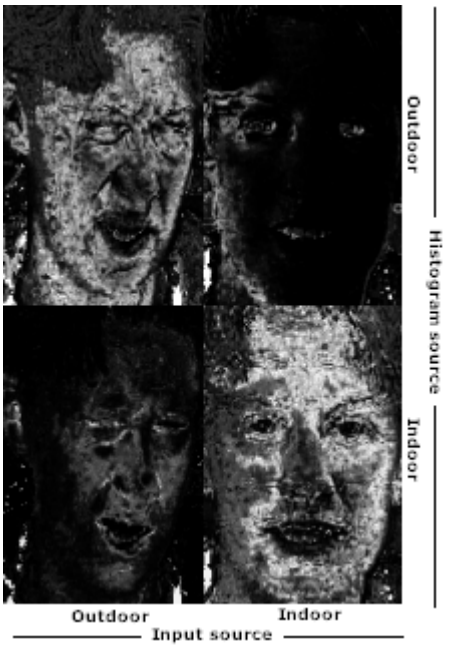

Figure 22. Face colour probability diagram montages for all histogram source/input source indoor/outdoor conditions under the HSV colour model.

3.6 Global illumination (brightness)

3.6.1 Motivation

Global illumination changes across a scene are very difficult to control; to improve the aesthetic results obtained from low-end desktop digital video cameras, many implement automatic white-balance and brightness controls. Thus, as the individual moves about the scene, the camera adjusts brightness and contrast in ways that affect the illumination across the entire scene.

These fluctuations in the brightness levels of the pixels in a video sequence are particularly troublesome to stochastic, colour-based face trackers. Brightness changes affect all three RGB components, and so algorithms that use the RGB color model normalize the space to mitigate this problem. Bradski recognized that hue is ambiguously defined under extremely dim and extremely bright lighting conditions, and thus CAMSHIFT ignores pixels that have very high or very low saturation values. This comparison is intended to compare the performance of CAMSHIFT under differing global illumination levels, and compare the use of the HSV model to the use of a normalized RGB color model in dim lighting conditions.

3.6.2 Methodology

Video sequences were captured in an indoor office setting using the Canon XL-1 and QuickCam cameras with the lights either off and on. These sequences were used to derive face colour probability histograms and diagrams, and compare the performance of the CAMSHIFT algorithm under each condition.

Two confounding factors impaired this test. First, a photometer was not used, and so it is not entirely certain that the light levels were the same under a given lighting condition for both cameras. Second, there was no way to turn off the QuickCam's automatic brightness control, and thus the sequence captured under it has been "artificially" brightened. Figure 23 shows a frame taken from each of the four source video sequences.

Figure 23. Source videos used in global illumination level comparisons: (from top left, to bottom right) QuickCam dim; QuickCam bright; XL-1 dim; XL-1 bright.

3.6.3 Results

The automatic brightness control feature of the QuickCam certainly paid off; the effect of the illumination change was not as vivid as it was in the video sequences captured using the Canon XL-1. The correlation coefficients in the face colour probability histograms extracted under each condition also reflect the powerful advantage auto-brightness correction gave the QuickCam.

3.6.4 Conclusions

Bradski, in his paper, suggests that auto-brightness correction be turned off during capture because it interferes with algorithm performance. These results clearly demonstrate that global brightness fluctuations can impair algorithm performance. However, Bradski's advice is only suitable under more-or-less controlled lighting conditions, such as typically office environment. If the area to be tracked suffers from highly variable brightness conditions, such as an office with a window, then Bradski's recommendation may cause us to fail to take advantage of added robustness made possible by automatic brightness control.

Regardless, from the comparisons of the face colour probability diagrams and the operation of CAMSHIFT algorithm, it appears that the HSV model it employs works better than a normalized RGB color model when global illumination levels are dim. Moreover, it is unreasonable to expect reliable CAMSHIFT behaviour when lighting levels are extremely dim. Tracking faces in dark environments could perhaps be better accomplished using cameras which detect infra-red radiation. An interesting project would be to apply the CAMSHIFT technique to input derived from an IR camera.

The important recommendation that comes out of this comparison, however, is that face colour probability histograms should be captured from a scene that has an extremely saturated, solidly coloured background as those pixels will be ignored in the computation of the histogram.

A possible optimization of this technique could arise if it were coupled with a background subtraction or video segmentation algorithm that sets scene background regions to pure black. In the videos used during this comparative study, it was observed that the matte gray paint on the wall was assigned a rather high face colour probability. Thus, these false negatives perturb CAMSHIFT's ability to track; background subtraction could eliminate this "noise" and improve CAMSHIFT's ability to track.

4. SUMMARY

Although CAMSHIFT appears well suited for use in many HCI applications of face tracking, it could not be successfully employed as a perceptual user interface to managing sub-conversations in a multiparty videoconferencing application because the head pose and orientation information its measures is too crude to accurately reconstruct focus of interest. Furthermore, attempts to implement CAMSHIFT were hampered by complications inherent in the use of a low-end desktop digital video camera as a source of live video for the algorithm. Such inexpensive cameras are becoming very popular and widely available, and thus are ideal for use in HCI applications.

A comparative study grew out of the failed attempt to apply CAMSHIFT to an HCI problem. This study looked at the output of CAMSHIFT using video captured under various conditions: camera type (e.g., CCD vs. CMOS), frame rate and size, global illumination conditions, and the quality of video compression. Although crude, the comparisons are instructive, and several important recommendation concerning the choice of camera, capture settings, and face tracking algorithm design. The recommendations are summarized in point-form below:

- Use a CCD-type camera, instead of a CMOS-type camera.
- Use the HSV colour model instead of a normalized RGB colour model for building stochastic face colour probability histograms because it eliminates the need for per-person training and is more immune to variations in global illumination brightness and spectra.
- Choose a frame size that closely matches the dimensions of the sensing element grid on the camera itself.
- Evaluate your choice of frame rate against the expected speed of head motion to be observed.
- Opt for a wider camera angle over larger face size.
- Opt for some hardware M-JPEG compression, if it allows you to boost frame size or rate. Up to 1:40 bit compression ratios were found to be acceptable.
- Turn on auto-brightness correction if you expect large or uncontrollable global illumination brightness changes. Turn it off if you expect lighting to be very constant.

5. REFERENCES

[1] Bradski, G. R.. Computer Video Face Tracking for use in a Perceptual User Interface. In *Intel Technology Journal Q2'98.*

[2] Hunke, M., and Waibel, A. *Face Locating and Tracking for Human-Computer Interaction.*

[3] Independent JPEG Group (IGJ) Web Site, *JPEG FAQ* (http://www.faqs.org/faqs/jpeg-faq/).

[4] USB Consortium Web Site, *Frequently Asked Questions* (http://www.usb.org/faq/ans2.html#q1).

[5] Yang, J. Wu, L., and Waibel, A. *Focus of Attention: Towards Low Bitrate Video Tele-Conferencing.*