**Computer Vision Face Tracking For Use in a Perceptual User Interface (continued)**

    Page 4 of 11

**CAMSHIFT Derivation**

The closest existing algorithm to CAMSHIFT is known as the mean shift algorithm [2][18]. The mean shift algorithm is a non-parametric technique that climbs the gradient of a probability distribution to find the nearest dominant mode (peak).

**How to Calculate the Mean Shift Algorithm**

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence (or until the mean location moves less than a preset threshold).

**Proof of Convergence**   
Assuming a Euclidean distribution space containing distribution f, the proof is as follows reflecting the steps above:

1. A window *W* is chosen at size *s*.
2. The initial search window is centered at data point *pk*
3. Compute the mean position within the search window   
     
   The mean shift climbs the gradient of *f(p)*
4. Center the window at point
5. Repeat Steps 3 and 4 until convergence.

Near the mode so the mean shift algorithm converges there.

For discrete 2D image probability distributions, the mean location (the centroid) within the search window (Steps 3 and 4 above) is found as follows:

Find the zeroth moment

Find the first moment for *x* and *y*

Then the mean search window location (the centroid) is

where *I(x,y)* is the pixel (probability) value at position *(x,y)* in the image, and *x* and *y* range over the search window.

Unlike the Mean Shift algorithm, which is designed for static distributions, CAMSHIFT is designed for dynamically changing distributions. These occur when objects in video sequences are being tracked and the object moves so that the size and location of the probability distribution changes in time. The CAMSHIFT algorithm adjusts the search window size in the course of its operation. Initial window size can be set at any reasonable value. For discrete distributions (digital data), the minimum window size is three as explained in the Implementation Details section. Instead of a set or externally adapted window size, CAMSHIFT relies on the zeroth moment information, extracted as part of the internal workings of the algorithm, to continuously adapt its window size within or over each video frame. One can think of the zeroth moment as the distribution "area" found under the search window. Thus, window radius, or height and width, is set to a function of the the zeroth moment found during search. The CAMSHIFT algorithm is then calculated using any initial non-zero window size (greater or equal to three if the distribution is discrete).

**How to Calculate the Continuously Adaptive Mean Shift Algorithm**

1. Choose the initial location of the search window.
2. Mean Shift as above (one or many iterations); store the zeroth moment.
3. Set the search window size equal to a function of the zeroth moment found in Step 2.
4. Repeat Steps 2 and 3 until convergence (mean location moves less than a preset threshold).

In Figure 4 below, CAMSHIFT is shown beginning the search process at the top left step by step down the left then right columns until convergence at bottom right. In this figure, the red graph is a 1D cross-section of an actual sub-sampled flesh color probability distribution of an image of a face and a nearby hand. In this figure, yellow is the CAMSHIFT search window, and purple is the mean shift point. The ordinate is the distribution value, and the abscissa is the horizontal spatial position within the original image. The window is initialized at size three and converges to cover the tracked face but not the hand in six iterations. In this sub-sampled image, the maximum distribution pixel value is 206 so we set the width of the search window to be *2\*M0/206* (see discussion of window size in the Implementation Details section below). In this process, CAMSHIFT exhibits typical behavior: it finds the center of the nearest connected distribution region (the face), but ignores nearby distractors (the hand).

**Figure 4:** CAMSHIFT in operation down the left then right columns

Figure 4 shows CAMSHIFT at startup. Figure 5 below shows frame to frame tracking. In this figure, the red color probability distribution has shifted left and changed form. At the left in Figure 5, the search window starts at its previous location from the bottom right in Figure 4. In one iteration it converges to the new face center.

**Figure 5:** Example of CAMSHIFT tracking starting from  
the converged search location in Figure 4 bottom right

**Mean Shift Alone Does Not Work**   
The mean shift algorithm alone would fail as a tracker. A window size that works at one distribution scale is not suitable for another scale as the color object moves towards and away from the camera. Small fixed-sized windows may get lost entirely for large object translation in the scene. Large fixed-sized windows may include distractors (other people or hands) and too much noise.

**CAMSHIFT for Video Sequences**   
When tracking a colored object, CAMSHIFT operates on a color probability distribution image derived from color histograms. CAMSHIFT calculates the centroid of the 2D color probability distribution within its 2D window of calculation, re-centers the window, then calculates the area for the next window size. Thus, we needn't calculate the color probability distribution over the whole image, but can instead restrict the calculation of the distribution to a smaller image region surrounding the current CAMSHIFT window. This tends to result in large computational savings when flesh color does not dominate the image. We refer to this feedback of calculation region size as the Coupled CAMSHIFT algorithm.

**How to Calculate the Coupled CAMSHIFT Algorithm**

1. First, set the calculation region of the probability distribution to the whole image.
2. Choose the initial location of the 2D mean shift search window.
3. Calculate the color probability distribution in the 2D region centered at the search window location in an area slightly larger than the mean shift window size.
4. Mean shift to convergence or for a set number of iterations. Store the zeroth moment (area or size) and mean location.
5. For the next video frame, center the search window at the mean location stored in Step 4 and set the window size to a function of the zeroth moment found there. Go to Step 3.

For each frame, the mean shift algorithm will tend to converge to the mode of the distribution. Therefore, CAMSHIFT for video will tend to track the center (mode) of color objects moving in a video scene. Figure 6 shows CAMSHIFT locked onto the mode of a flesh color probability distribution (mode center and area are marked on the original video image). In this figure, CAMSHIFT marks the face centroid with a cross and displays its search window with a box.

**Figure 6:** A video image and its flesh probability image

**Calculation of Head Roll**   
The 2D orientation of the probability distribution is also easy to obtain by using the second moments during the course of CAMSHIFT's operation where (x,y) range over the search window, and I(x,y) is the pixel (probability) value at (x,y):

Second moments are

Then the object orientation (major axis) is

The first two Eigenvalues (major length and width) of the probability distribution "blob" found by CAMSHIFT may be calculated in closed form as follows [4]. Let

and

Then length l and width w from the distribution centroid are

When used in face tracking, the above equations give us head roll, length, and width as marked in Figure 7.

**Figure 7:** Orientation of the flesh probability   
distribution marked on the source video image

CAMSHIFT thus gives us a computationally efficient, simple to implement algorithm that tracks four degrees of freedom (see Figure 8).

**Figure 8:** First four head tracked degrees  
of freedom: X, Y, Z location, and head roll

**How CAMSHIFT Deals with Image Problems**   
When tracking color objects, CAMSHIFT deals with the image problems mentioned previously of irregular object motion due to perspective, image noise, distractors, and facial occlusion as described below.

CAMSHIFT continuously re-scales itself in a way that naturally fits the structure of the data. A colored object's potential velocity and acceleration scale with its distance to the camera, which in turn, scales the size of its color distribution in the image plane. Thus, when objects are close, they can move rapidly in the image plane, but their probability distribution also occupies a large area. In this situation, CAMSHIFT's window size is also large and so can catch large movements. When objects are distant, the color distribution is small so CAMSHIFT's window size is small, but distal objects are slower to traverse the video scene. This natural adaptation to distribution scale and translation allows us to do without predictive filters or variables–a further computational saving–and serves as an in-built antidote to the problem of erratic object motion.

CAMSHIFT's windowed distribution gradient climbing causes it to ignore distribution outliers. Therefore, CAMSHIFT produces very little jitter in noise and, as a result, tracking variables do not have to be smoothed or filtered. This gives us robust noise tolerance.

CAMSHIFT's robust ability to ignore outliers also allows it to be robust against distractors. Once CAMSHIFT is locked onto the mode of a color distribution, it will tend to ignore other nearby but non-connected color distributions. Thus, when CAMSHIFT is tracking a face, the presence of other faces or hand movements in the scene will not cause CAMSHIFT to loose the original face unless the other faces or hand movements substantially occlude the original face.

CAMSHIFT's provable convergence to the mode of probability distributions helps it ignore partial occlusions of the colored object. CAMSHIFT will tend to stick to the mode of the color distribution that remains.

Moreover, when CAMSHIFT's window size is set somewhat greater than the root of the distribution area under its window, CAMSHIFT tends to grow to encompass the connected area of the distribution that is being tracked (see Figure 4). This is just what is desired for tracking whole objects such as faces, hands, and colored tools. This property enables CAMSHIFT to not get stuck tracking, for example, the nose of a face, but instead to track the whole face.

    Page 4 of 11

* [Site Map](http://www.intel.com/siteindex.htm?iid=ftr+map)
* [RSS](http://www.intel.com/intel/rss.htm?iid=ftr+rss)
* [Jobs](http://www.intel.com/jobs/index.htm?iid=ftr+jobs)
* [Investor Relations](http://www.intc.com/index.cfm?iid=ftr+invrel)
* [Press Room](http://www.intel.com/pressroom/index.htm?iid=ftr+press)
* [Contact Us](http://www.intel.com/feedback.htm?iid=ftr+contact)
* [Terms of Use](http://www.intel.com/sites/sitewide/en_US/termsofuse.htm?iid=ftr+terms)
* [Terms of Service](http://www.intel.com/sites/sitewide/en_US/termsofservice.htm?iid=ftr+terms_svs)
* [\*Trademarks](http://www.intel.com/sites/sitewide/en_US/tradmarx.htm?iid=ftr+trademark)
* [Social Media Guidelines](http://www.intel.com/sites/sitewide/en_US/social-media.htm?iid=ftr+social)
* [Privacy](http://www.intel.com/sites/sitewide/en_US/privacy/privacy.htm?iid=ftr+privacy)
* ©Intel Corporation