Resolving Hand Over Face Occlusion

Paul Smith[1], Niels da Vitoria Lobo[1], and Mubarak Shah[1]

Computer Vision Lab, School of Computer Science University of Central Florida
{rsmith, niels, shah}@cs.ucf.edu

Abstract. This paper presents a method to segment the hand over complex backgrounds, such as the face. The similar colors and texture of the hand and face make the problem particularly challenging. Our method is based on the concept of an image force field. In this representation each individual image location consists of a vector value which is a nonlinear combination of the remaining pixels in the image. We introduce and develop a novel physics based feature that is able to measure regional structure in the image thus avoiding the problem of local pixel based analysis, which break down under our conditions. The regional image structure changes in the occluded region during occlusion. Elsewhere the regional structure remains relatively constant. We model the regional image structure at all image locations over time using a Mixture of Gaussians (MoG) to detect the occluded region in the image. We have tested the method on a number of sequences demonstrating the versatility of the proposed approach.

1 Introduction

The task of segmenting the hand over complex backgrounds such as the face is a challenging problem. The difficulty lies in the fact that the hand and head are similarly colored/textured regions. A necessary step for many HCI applications such as gesture recognition, pointing interfaces, hand pose recognition, and event detection is a reliable hand segmentation. Sign language recognition methods also need to first segment the hand over complex boundaries, such as the face. Some events like coughing, eating, and taking medication could be more easily recognized by segmenting the hand from the face. In short there are many applications that could benefit from having a robust segmentation of the hand over complex backgrounds.

We propose two main contributions in segmenting the hand over complex backgrounds such as the face. First we develop a new feature based on the force field image [10]. The force field image uses concepts from force field transformations used in physics. Basically, each image location is represented by a vector value which is a nonlinear combination of all other pixels in image. Their approach focused on a possible feature space for recognition of faces and uses single frames. The feature we develop is the distance traveled by test pixels placed in the force field. Our novel feature is able to model regional structural changes in the image over time. Local methods (pixel based) cannot resolve the occlusion because there is little change in local color when similarly colored objects occlude each other. Regional structure in the image does change when the hand occludes the face, although local pixel colors in the occluding region remain largely the same before and during the occlusion. By quantifying the regional structural change in an image over time we can resolve this kind of occlusion.

The second contribution is in presenting a method that is able to model our newly developed feature response over time and capture where and when occlusion is happening using a Mixture of Gaussians (MoG) modeling paradigm. We also clarify several concepts from [10] and give more details in using this image representation. An extension of the force field computation to video data is also given.

Section 1.1 gives previous work. Section 2 provides details on the image representation. Section 3 shows how formulating the problem using MoG can aid the task of segmenting the hand/face. Results are presented in Section 4, and then we conclude.

1.1 Previous Work

Much of the work in finding the hand in a complex background relies on colored markers [7] on the hands or requires the hand to be the only skin object in view [5]. Contour based approaches [1] [9] [11] and other edge based methods [14] rely on good edges separating the hand and head, which are often not present in such difficult occlusion. Active contour approaches [1] [11] require the hand shape change to be small. Our method has no such constraint. In [9], hand shape is estimated over a complex background by using a shape transition network with the attributes of contour, position, and velocity. They use a simple template based approach and skin color segmentation to find the hand during hand face occlusion. Their approach is sensitive to small changes in lighting, different skin colors, and requires small differences in the 2D hand shapes. Other color based approaches [3] [12] [14] would have difficulties in segmenting the hand over face. In [12] body parts are tracked using Bayesian Networks but the conditional probabilities are specified manually. Further, skin color is used to find the body parts. In [8] examples are given handling a few frames of occlusion using shape and color in a Bayesian framework, but it is unclear if it can withstand occlusion involving hundreds of frames (as our approach does). In [18] hand tracking is performed using eigen dynamics analysis, but the hand tracking system uses pretrained hand models. It is unclear how person-independent these models are.

In [2], a method is presented which uses multiscale features to find the hand. Color priors are used, requiring retraining for new people. This method will not work when the face is present because of the stronger blobs and ridges on the face. [19] performs well on segmenting hands, but the method requires that the hand cover a large portion of the image. Our image sequences frequently have only part of the hand in the image.

An Elastic Graph Matching approach is given in [15], which uses color models to find skin regions. It has problems when the illumination changes, as the skin color model fails. Each training image requires manual labeling of at least 15 node points. The approach has problems handling geometric distortions of the hand as does [16]. Our approach is not hand model based, so we do not have this limitation.

In [6] an approach is given that segments the hand from a complex background. They localize the hand using motion information and map this region to a fovea vector. The method does not extend to other people. There is significant change in hand size which our method can cope with. Most model based approaches presented above fail in the case where the hand is only partially visible in the image or for gestures not in the database.

Because of the similar colors of the hand and face, segmentation algorithms such as [4] will generally either under or over segment the hand/face occlusion. In principle, one can do tracking but then the question becomes how to initialize the tracking. Further, tracking methods generally fail when tracking across similarly colored regions.

Background subtraction [13] will not work in segmenting the hand over the face because even a slight movement of the head will trigger a large change of foreground pixels. Further, supposing the head was relatively fixed, the underlying problem with the RGB (and other color spaces) input domain is in the similarity of the head and hands. These methods cannot distinguish between the head and hand colors. Most background subtraction methodologies operate on RGB or some other color space (i.e. the input space is color information). When similarly colored objects, occlude each other the individual pixel values in the region of occlusion give little information considered individually because the objects are similarly colored. This presents difficulties for individual pixel based methods.

2 Potential and Force Images

We can define the smoothed potential at a given position, $\mathbf{r}_j$, with respect to position $\mathbf{r}_i$, in image I as

$$E_i(\mathbf{r}_j) = \frac{I(\mathbf{r}_i) + I'(\mathbf{r}_i)}{2 \cdot |\mathbf{r}_i - \mathbf{r}_j|} \tag{1}$$

where $\mathbf{r}_j$ is the image location in question and $I(\mathbf{r}_i)$ is the image intensity at position $\mathbf{r}_i$. I' represents the image intensity at the previous time instant. Because we are dealing with video data, we introduce temporal smoothing into the force field representation to account for spurious noise.

Equation 2 gives the potential energy for a particular image location. This computation is then performed for every location in the image. This gives the potential energy image. The total potential energy at location $\mathbf{r}_j$ is given by:

$$E(\mathbf{r}_j) = \sum_{\mathbf{r}_i \neq \mathbf{r}_j} E_i(\mathbf{r}_j) = \sum_{\mathbf{r}_i \neq \mathbf{r}_j} \frac{I(\mathbf{r}_i) + I'(\mathbf{r}_i)}{2 \cdot |\mathbf{r}_i - \mathbf{r}_j|} \tag{2}$$

2.1 Force Fields

To find the force exerted by all pixels at a particular image location $\mathbf{r}_j$ simply compute

$$F(\mathbf{r}_j) = \sum_{\mathbf{r}_i \neq \mathbf{r}_j} E_i(\mathbf{r}_j) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^2} = \sum_{\mathbf{r}_i \neq \mathbf{r}_j} \frac{I(\mathbf{r}_i) + I'(\mathbf{r}_i)}{2} \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \tag{3}$$

We can see that the force is a vector as it has magnitude and direction. These vector fields will be very important in the image representation. The units of pixel intensity, direction, and force are arbitrary as is the origin of the coordinate system. $\overline{F}(\mathbf{r}_j)$ is the normalized vector at $\mathbf{r}_j$. Examples of the potential and force fields are shown in Figure 1. Since the force fields are two dimensional the magnitude and direction are shown separately. The direction was quantized (for display purposes only) into 10 regions.

Fig. 1. Potential and Force Vector Fields for various input frames. Input images are shown on the top left. Potential image is next. Next is magnitude of the force field and the last contains the direction (quantized) of the force field.

2.2 Finding Potential Wells

Once the potential and force field images have been computed the well points(local extrema) are computed. This is done in an iterative fashion. Unit test pixels are placed uniformly (resulting in a rectangular grid of test pixels) throughout the image. They can be placed at every pixel, every other pixel etc. They are placed in the field and serve to capture the flow of the field. Suppose there are m test pixels $t_1, \ldots, t_m$. Since the position of each test pixel will change as it traverses the force field, we denote the initial location of t_i as $t_{i,0}$. To find any $t_{i,j}$ apply the recursive equations:

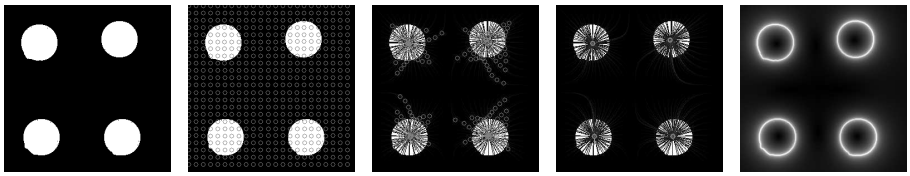
$$t_{i,0} = (x_i, y_i)$$

$$t_{i,j} = t_{i,j-1} + \overline{F}(t_{i,j-1}) \tag{4}$$

Where $\overline{F}(x)$ is the normalized vector at x, which is computed as $\overline{F}(x) = \frac{F(x)}{|F(x)|}$. Given a unit test pixel starting point, $t_{i,0}$, it goes through the force field until it stabilizes at a well point, denoted as $t_{i,N}$. Unit test pixels eventually reach stable points. In our examples N=500. Convergence was always reached well before N=500, but we could test for convergence to allow more than 500 iterations. The computation could be ended earlier if convergence is reached. Iterations needed for convergence depends on image size and the number of wells. Larger images or ones with fewer wells will need more iterations, but 500 iterations was sufficient for the 1000's of image we tested. Not all t_i end up at the same wells. The path that a test pixel takes is called a channel. It is easy to see that once two test pixels reach a common point, they both travel the same path from them on. Before deriving the distance traveled feature we would like to give some intuition as to what information in the image the force field is capturing and why it is useful in our problem domain. Equation 3 shows that the force field captures global structure, technically. However since the effect on the field is proportional to $\frac{1}{d^2}$ the net effect is that the force field captures regional image structure.

The potential image is a scalar at each pixel and it is a measure of the brightness of that region. The force field is a vector at each pixel location. It measures properties related to regional edge strength. It is not an edge detector, but it is related. The force field measures regional edge like structure in the image. The potential wells are those points in the force field where the net force is zero. Intuitively these are the points that seek to position themselves in between the regional edge structures of the image. A well

equalizes the force (regional structure) around itself. See Figure 2 for an example of a synthetic image demonstrating these ideas.



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

Fig. 2. This is a synthetic image used to give some intuition of the force field representation. (a) is the original image. (b) is the original image with the initial configuration of test pixels overlaid on it. (c)-(d) show the movement of the test pixels through the force field after 50 and 200 iterations. (e) shows the magnitude of the force field.

The force field captures regional structure and we can model this regional structure over time to detect structural changes in the image. Though the hand and head have similar color and texture, by analyzing regional image structure we are able to capture structural changes that are introduced when the hand enters the scene. We can see that other methods monitoring pixel wise information are not enough because of the similar texture of the hand and head. When the hand enters, the local structure would not change (i.e. the pixel values remain largely the same), but there is useful regional structure variation (we will show examples of this change in subsequent sections). We now detail how we model this changing force field over time.

3 Developing New Image Feature

The structure of these field lines for a particular image sequence are relatively constant until the hand (or anything else) enters the image. Once the hand enters a clear disturbance in the channels occurs in the region of occlusion. This hypothesis has been borne out in experiments on thousands of video frames. It is consistent with the fact that the force field is a measure of regional image structure. Figure 3 shows an example of this phenomenon. It can be seen that most of the channels are stable before and during the occlusion. We could show more examples, but due to space limitations, we will not. We next demonstrate how to measure and quantify this changing force field.

If test pixels are placed uniformly in each image we can measure the variation a certain test pixel exhibits in the distance it travels to a potential well. Since these distances remain relatively constant when there is no disturbance in the image (i.e. no hand/face occlusion), the distance that each test pixel travels can be modeled as a random variable with Gaussian distribution. When the hand enters, the wells and the distances that the test pixels travel will vary significantly. These will be the foreground channels, and they are somewhat analogous to foreground pixels in background subtraction.

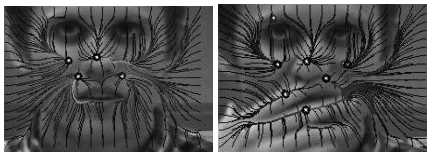


Fig. 3. Channels before and during occlusion. Notice that a disturbance in the channels can be seen in the lower left corner of the image, whereas the rest of the channels in the image are relatively stable.

The reason this occurs is that when another object is introduced, it has its own set of channels and wells. When the two objects merge, the channels and wells of both objects interact with one another. Although the hand and face are similar in color, the potential and force structure present in the image changes when another object enters the scene. Using the MoG modeling technique we are able to measure and localize this change, which allows us to find the boundary between the face and the hand. The distance from a test pixel start location to its final well position can be measured by computing

$$d = |t_{j,0} - t_{j,N}| \tag{5}$$

This is the new distance traveled in a force field feature. Other distance measures such as the arc length could be used. In any case, the distances the test pixels travel are relatively constant until the hand enters the facial region. We model the face before occlusion in terms of the distance traveled at each test pixel start location using a MoG.

Let us assume that in the first video frame for a particular test pixel t_j: $|t_{j,0} - t_{j,N}| = X_0$. In the next video frame for the same test pixel location we can compute $|t_{j,0} - t_{j,N}| = X_1$. Given the distance traveled history of a particular test pixel at location t_j: $X_0, X_1, \ldots, X_\tau$, we want to model this density as a mixture of K Gaussians. The current distance traveled by t_j, X_τ, at time τ, has probability

$$P(X_\tau) = \sum_{i=1}^{K} w_{i,\tau} \frac{1}{\sqrt{2\pi}\sigma_{i,\tau}} e^{\frac{-(X_\tau - \mu_{i,\tau})}{2\sigma_{i,\tau}^2}} \tag{6}$$

of belonging to the current model. $w_{i,\tau}$ is the weight of the i^{th} Gaussian, and $\mu_{i,\tau}$ and $\sigma_{i,\tau}$ are the mean and variance of the distribution all at time τ. If none of the Gaussian distributions match for this particular location t_j, the least likely distribution is replaced by the new distance. The distribution's mean is the distance traveled by t_j, $|t_{j,0} - t_{j,N}|$, with the weight of this distribution set low. At each time instant the weights of the K distributions are updated as

$$w_{i,\tau} = (1 - \alpha) w_{i,\tau-1} + \alpha(M_{i,\tau}) \tag{7}$$

with α set to a constant (learning rate) and $M_{i,\tau}$ being an indicator function which is 1 for the distribution that matched and 0 otherwise. The distribution i that matched the current distance observation has its mean and variance updated as

$$\mu_{i,\tau} = (1 - \rho)\mu_{i,\tau-1} + \rho X_\tau \tag{8}$$

$$\sigma_{i,\tau} = (1 - \rho)\sigma_{i,\tau-1}^2 + \rho(X_\tau - \mu_{i,\tau})^2 \tag{9}$$

In our case ρ is set to a constant. For notational convenience we denote t_{j_u} as the mean of the distribution that matched for test pixel t_j. Using this approach we are able to model the distances traveled by each test pixel in a coherent manner. The next task is to use these models to segment the hand from the face.

3.1 Extracting the Hand

There are two steps needed to extract the hand. We must first identify whether or not the frame has a hand in it. A good measure is when the maximally changing test pixel's distance from its distribution is much larger than its change in the previous frame. This indicates a large change in the image. Concretely, we say the hand has entered when

$$t_{l_\mu} - X_\tau > 3 \cdot (t_{l_{\mu-1}} - X_{\tau-1}), \tag{10}$$

$$\text{where } l = \underset{l}{\text{argmax}}\ t_{l_\mu} - X_\tau, l = x \tag{11}$$

t_{l_μ} is the mean of the distribution for t_l, and X_τ is the current distance traveled observation (computed as $|t_{l,0} - t_{l,N}|$ for t_l. $t_{l_{\mu-1}}$ and $X_{\tau-1}$ are the mean of the distribution and observation for t_l at the previous input frame.

The goal is to find the set **H** which is all the hand pixels. Initially set $\mathbf{H} \leftarrow t_{l,x}, \forall x \ni 1 \leq x \leq N$. This only gives one t_i and corresponding channel. To get the full hand, any test pixel which ended up at the same well is also assumed to be part of the hand. Further, any test pixel whose well is within β pixels is assumed to be part of the hand. Concretely, set

$$\mathbf{H} \leftarrow \mathbf{H} + t_{a,x}, \forall a, x \ni |t_{l,N} - t_{a,N}| \leq \beta, 1 \leq x \leq N \tag{12}$$

These test pixels and corresponding channels taken together segment the hand region. Once the hand enters the head region, the distances test pixels travel will vary greatly. This variation should not be learned, so the models are not updated after the hand enters the head region. Figure 4 shows three frames of the found channel lines. The final segmentation is achieved by finding the convex hull of this point set **H** and drawing the hull. Other methods could be used to improve the resulting contour. The full algorithm is given in Table 1. Detailed results are presented in Section 4.

For every frame

1. Compute force at every pixel using Equation 3
2. Place test pixels t_i uniformly and $\forall t_i$ compute Equations 4 and 5
3. $\forall t_i$Use Equations 6 - 9 to update online MoG models
4. Check for hand using Equations 10 and 11
5. If hand present, segment using Equation 12, find convex hull and display result

Table 1. Overall Algorithm

Fig. 4. Channels superimposed on hand region. These channels varied most from the previous model. A convex hull algorithm could be used to fill in this hand region.

4 Results

Our method was tested on 14 sequences involving hand/face occlusion for a total of 1800 frames. Not all of these frames contained hand over face occlusion. Of course the non-occlusion frames were needed in order to build the online distance models. Out of the 1800 frames, roughly one half contained hand over face occlusion. The method was successful under a variety of lighting conditions. We assume that the hand is initially not present (which allows us to build the model). In order to allow translational invariance and to have faster processing,we find the head region using [17] and only process these regions. We model every 5^{th} pixel in both directions for faster computation. More samples would increase segmentation rates and the contour accuracy. Figures 5 and 6 show results of the hand segmentation on different input sequences. We should note that in Figure 5 the head starts out frontal and then rotates to a half-profile position. Our method is able to cope with this type of rotation, after which the model starts to break down. Again to obtain the results we run a convex hull algorithm on the set **H**, described in Section 3.1, and show the hull. The algorithm was always able to determine when the hand entered the image using the steps in Section 3.1. Figure 7 shows a comparison between our proposed method, background subtraction [13], and mean shift segmentation [4] respectively. Our method and [13] give pixel wise segmentation, so comparison was straightforward and unambiguous. We felt it would be interesting to compare against general methods because our approach does not use hand color/shape to improve its decision, meaning it could possibly be applied in other contexts. Neither of these two other methods were successful in segmenting the hand from the face.

To quantify how well the algorithm performed we manually generated ground truth segmentations for two sequences. Comparisons of our method to ground truth and background subtraction [13] are shown in Table 2. Comparison was made pixel wise. For our method each pixel in the convex hull was counted as hand and each pixel outside was counted as non-hand. The true positive percentages for every frame were added and divided by the total number of frames. A similar method was used for the true negative rate. Our method outperformed [13] in all cases. While [13] segmented part of the hand, it found much of the head region as hand, indicated by the low true negative rate.

5 Conclusion and Future Directions

We have developed a method that is successfully able to segment the hand from the face. From a high level the method succeeds because we developed an image feature

Seq #	# Frames	Our Method TP %	Method in [13] TP %	Our Method TN %	Method in [13] TN %
1	44	80.04	72.00	97.11	74.12
9	150	79.53	73.15	96.58	72.19

Table 2. True positive (TP) and true negative (TN) segmentation % for the specified sequences.

Fig. 5. Hand Segmentation. This was a difficult sequence due to the large face rotation.

Fig. 6. Hand segmentation results for three sequences. Row 1 shows channel lines superimposed. Row 2 shows the convex hull. The sequence in Row 3 involves occlusion for over 300 frames.

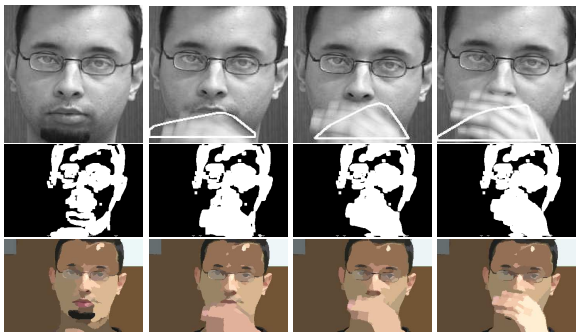


Fig. 7. Segmentation results for our method (row 1), [13] (row 2) and [4] (row 3).

which is based on regional information. During occlusion of head and hand, local pixel regions remain similar, but the regional image structure changes during the occlusion. Our method detects this change and is able to recover the occluding region. Our main contributions are in development of a novel feature: the distance traveled of a test pixel in the image force field. And in modeling the distance traveled using a MoG, which allowed us to capture occlusion information that is difficult to extract. We want to explore more the force field representation, and determine its limits in resolving occlusion. Better methods of segmentation using the MoG model could be explored. It would be useful to test how well the method resolves occlusion with other types of objects.

References

1. T. Ahmad, C. Taylor, A. Lanitis, and T. Cootes. Tracking and recognising hand gestures, using statistical shape models. *Image and Vision Computing*, 1997.
2. L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *Automatic Face and Gesture Recognition*, 2002.
3. L. Brthes, P. Menezes, F. Lerasle, and J. Hayet. Face tracking and hand gesture recognition for human-robot interaction. *International Conference on Robotics and Automation*, 2004.
4. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 2002.
5. Y. Cui and J. Weng. Learning-based hand sign recognition. *Automatic Face and Gesture Recognition*, 1995.
6. Y. Cui and J. Weng. Hand sign recognition from intensity image sequences with complex backgrounds. *Automatic Face and Gesture Recognition*, 1996.
7. J. Davis and M. Shah. Recognizing hand gestures. *ECCV*, 1994.
8. H. Fei and I. Reid. Joint bayes filter: A hybrid tracker for non-rigid hand motion recognition. *ECCV*, 2004.
9. Y. Hamada, N. Shimada, and Y. Shirai. Hand shape estimation under complex backgrounds for sign language recognition. *Automatic Face and Gesture Recognition*, 2004.
10. D. J. Hurley, M. S. Nixon, and J. N. Carter. Force field energy functionals for image feature extraction. In *IVC*, 2002.
11. M. H. Jeong, Y. Kuno, N. Shimada, and Y. Shirai. Recognition of shape-changing hand gestures. *IEICE Transactions Division D*, E85-D No. 10:1678–1687, 2002.
12. J. Sherrah and S. Gong. Resolving visual uncertainty and occlusion through probabilistic reasoning. *BMVC*, 2000.
13. C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 2000.
14. B. Stenger, A. Thayananthan, P. Torr, , and R. Cipolla. Hand pose estimation using hierarchical detection. *Intl. Workshop on Human-Computer Interaction*, 2004.
15. J. Triesch and C. von der Malsburg. A system for person-independent hand posture recognition against complex backgrounds. *TPAMI*, 2001.
16. J. Triesch and C. von der Malsburg. Classification of hand postures against complex backgrounds using elastic graph matching. *Image and Vision Computing*, 2002.
17. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.
18. H. Zhou and T. S. Huang. Tracking articulated hand motion with eigen dynamics analysis. *ICCV*, 2003.
19. X. Zhu, J. Yang, and A. Waibel. Segmenting hands of arbitrary color. *Automatic Face and Gesture Recognition*, 2000.