JAIDM
Journal of AI and Data Mining
Vol. 2, No. 1, 2014, 15-24.

Applying mean shift and motion detection approaches to hand tracking in sign language

M.M. Hosseini*, J. Hassanian

Islamic Azad University, Shahrood branch, Shahroodt, Iran.

Received 25 February 2013; accepted 30 June 2013
**Corresponding author: hosseini_mm@yahoo.com (M.M. Hosseini).*

Abstract
Hand gesture recognition is very important to communicate in a sign language. In this paper, an effective object tracking and the hand gesture recognition method is proposed. This method is a combination of two well-known approaches, the mean shift and the motion detection algorithm. The mean shift algorithm can track objects based on a color, then when hand passes the face occlusion happens. Several solutions such as the particle filter, kalman filter and dynamic programming tracking have been used, but they are complicated, time consuming and so expensive. The proposed method is so easy, fast, efficient and low costly. The motion detection algorithm in the first step subtracts the previous frame from the current frame to obtain the changes between two images and white pixels (motion level) are detected by using the threshold level. Then the mean shift algorithm is applied for tracking the hand motion. Simulation results show that this method is faster than two times compared with the old common algorithms.

Keywords: *Hand tracking, Motion detection, Mean shift, Hand gesture recognition, sign language.*

1. Introduction

Recently, many research fields of object tracking have involved hand tracking known as a crucial and basic ingredient computer vision. Human beings can simply recognize and track an object immediately even in the presence of high clutter, occlusion, and non-linear variations in the background as well as in the shape, direction or even the size of the target object. However, hand tracking can be a difficult and challenging task for a machine. Tracking for a machine can be used to find the object states. Position, scale, velocity, feature selecting and many other important parameters obtained from a sequential series of images are included, so object tracking uses each image or incoming frame to obtain the weighting coefficients of the entire image. Therefore, it is necessary to specify the special target to track a desired object such as a specific hand. Many solutions are proposed to deal with a hand motion that they use some features to obtain targets such as colors, area motion and texture, in which [1-2] suggested a solution for an object tracking. Their suggested method converted color frames into gray level images, and then a kernel function is employed. Furthermore, the weights of pixels were obtained each frame. Their proposed method offered several advantages. For instance, it can be very resistant against difficulties such as partial occlusion, blurring caused by camera shaking, deformation of object position and any other sorts of translation. This is due to employing color information as feature vectors in the proposed technique. Wang at el [3] proposed a method based on Hidden Markov Model that they used a Cyber sensory glove and a Flock of Birds motion tracker to extract the features of American Sign Language gestures. The data received from the strain gages in the glove describe the hand shape while the data from the motion tracker describes the trajectory of hand movement. Cheng [4] notes that mean shift is

fundamentally a gradient ascent algorithm with an adaptive step size. Since Comaniciu [5], the first introduced mean shift-based on the object tracking, has proven to be a promising alternative for popular particle filtering based on the trackers. In this paper, our method is described in section 2 and the experiment result is shown in section 3, and finally, section 4 draws the conclusions.

2. The Proposed method

Using the hand gesture tracking through the proposed method uses the mean shift algorithm. Occlusion especially happens, when hand reaches the face. Some features are used in the mean shift algorithm such as colors. Since hand (target) and face have the same color, the program cannot recognize the target and the hand tracking faces the problem. Other researchers offered several solutions, such as [5] using the particle filter and [6] using the kalman filter. These methods are complicated and take long time to run and need a large memory, so advanced processor can just satisfy these needs and then they are so expensive. However, this paper suggests an easy and efficient method for solving these problems. This method limits the tracking range by a) using the mean shift algorithm for motion detection b) eliminating the constant point and then motion range specification, so tracking is done with high accuracy. In this combination method, the color feature is used as a hand distinguished feature. Here, a user in the first frame obtains the hand model as a main purpose of tracking and the color feature and the number of pixels can be extracted by using this model. Then some pixels whose color value is less than hand color value are eliminated with a specific threshold and remained pixels are weighted appropriate to the distance to the hand center (closer, much weighted).

Finally, a range motion is specified through using the motion detection and constant point illustration. Mean shift algorithm calculates the mean of pixels considering the weighting coefficient to obtain hand center. In frames that occlusion happens, motion detection makes track move, because constant points have already been eliminated and when new frames come, the target is obtained by using the proposed combination method again. The block diagram is shown in Figure1.

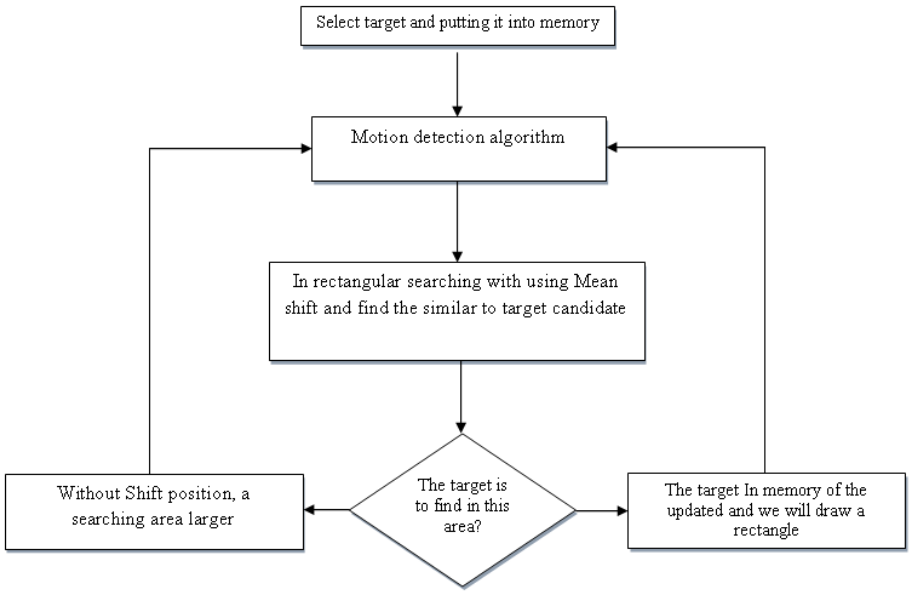

Figure 1.Complete block diagram of integration tracker.

2.1. Motion detection

Motion detection is a famous approach in object tracking that is in fact faster than mean-shift tracker [6]. Motion detection is the easiest of the three motion related detection: tasks, estimation, and segmentation. It results in identifying which image points, or even which regions of the image have moved between two time instants. Using motion detection algorithm can compare the first frame with the pervious one. It is used in video compression when it is necessary to estimate changes and to write only the changes, not the whole frame. This algorithm shows an image with white pixels (motion level) on the place where the current frame is different from the previous one. It is already possible to count these pixels, and if the

amount of these pixels becomes greater than a predefined threshold level, threshold is produced a motion event. Detecting the motion is calculated the distance in the luminance space between the current image I_k and the last aligned image I^*_{k-1}, obtaining the difference image DI_k, defined as [6]:

$$DI_k(x) = \begin{cases} m & if \ \left|I_k(x) - I^*_{k-1}(x)\right| > T_m \\ 0 & otherwise \end{cases} \quad (1)$$

Where *m* refers to a factor of increment in the motion, and T_m refers to a motion threshold. DI_k Contains the initial set of points that are candidate to relate to the Moving Visual Object. In order to consolidate the blobs to be detected, a 3×3 morphological closing is used to DI_k. Isolated detected moving pixels are discarded through applying a 3×3 morphological opening. The representation of the motion history image MH_k is then updated by multiplying the previous motion history representation MH_{k-1} with a decay factor, and by adding the difference image DI_k [6]*:*

$$MH_k = MH_{k-1} * DecayFactor + DI_k \quad (2)$$

Finally, all pixels of MH_k whose luminance is over a motion detection threshold (T_h) are considered as pixels in motion, these pixels can generate the detection image D_{H_k} defined as [6]:

$$D_{H_k}(x) = \begin{cases} 1 & if \quad MH_k(x) > T_h \\ 0 & otherwise \end{cases} \quad (3)$$

a 3×3 morphological closing is applied to the detection image D_{H_k} followed by a 3×3 morphological opening. It is used 3×3 morphology because it can be obtained more smooth and a proper edge from the pictures that cannot be taken by applying 4×4 or 8×8 morphology. Therefore, this solution results in high accuracy in tracking and then the best result.

2.2. Mean shift

In this section mean shift algorithm is adequately presented.

2.2.1. Target representation

At first, a feature space is chosen to characterize the target [7]. The reference *target model* is represented by its pdf q in the feature space. For example, the reference model can be chosen to be the color pdf of the target. The target model can be considered as centered at the spatial location 0, without any general loss. In the subsequent frame a *target candidate* is defined at location y , and is characterized by the pdf $p(y)$. Both pdf-s are estimated from the data. Discrete densities, i.e., m -bin histograms should be used to satisfy the low computational cost imposed by real-time processing. Therefore we have [7]

Target model:
$$q = \{q_u\}_{u=1,2,...,m} \qquad \sum_{u=1}^{m} q_u = 1 \quad (I)$$

Target candidate:
$$p(y) = \{p_u(y)\}_{u=1,2,...,m} \qquad \sum_{u=1}^{m} p_u = 1 \quad (\Pi)$$

The histogram is enough for our purposes but it is not adequate [8]. Other discrete density estimates can be also applied. We will denote by the formula below.

$$\rho \equiv \rho\left[p(y), q\right] \quad (4)$$

A similarity function exists between p and q . The function $\rho(y)$ plays the role of likelihood and the local maxima in the image shows the presence of objects in the second frame having representations similar to q defined in the first frame. If only spectral information is used to characterize the target, the similarity function can have large variations for adjacent locations on the image lattice and the spatial information is lost. To realize the maxima of such functions, gradient-based optimization procedures are hard to apply and only an expensive exhaustive search can be used. We regularize the similarity function by masking the objects with an isotropic kernel in the spatial domain. When the kernel weights, carrying continuous spatial information are used in defining the feature space representations, and $\rho(y)$ becomes a smooth function in y .

2.2.2. Target model

A target is represented by an ellipsoidal or rectangular region in the image (Figure 2). To eliminate the effect of different target dimensions, all targets are first normalized to a unit circle. This is achieved by independently rescaling the row and column dimensions with h_x and h_y .

Figure 2. Determine target in the first frame (reference image).

Let $\{x_i^*\}_{i=1,2,...,n}$ be the *normalized* pixel locations in the region defined as the target model. The region is centered at 0. An isotropic kernel, with a convex and monotonic decreasing kernel profile $k(x)$, assigns smaller weights to pixels farther from the center. The robustness of the density estimation increased by using these weights since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or interference from the background.The function $b: R^2 \to \{1,2,...,m\}$ associates to the pixel at location x_i^* the index $b(x_i^*)$ of its bin in the quantized feature space. The probability of the feature $u = 1,2,...,m$ in the target model is then calculated as following [7]

$$q_u = C\sum_{i=1}^{n} k(\left\|x_i^*\right\|^2)\delta\left[b(x_i^*) - u\right] \qquad (5)$$

Where 1 is the Kronecker delta function. The normalization constant C is obtained by imposing the condition $\sum_{u=1}^{m} q_u = 1$, from where [7]

$$C = \frac{1}{\sum_{i=1}^{n} k(\left\|x_i^*\right\|^2)} \qquad (6)$$

Since the summation of delta functions for $u = 1,2,...,m$ is equal to one.

2.2.3. Target candidates

Suppose that this name $\{x_i\}_{i=1,2,...,n_h}$ is the *normalized* pixel locations of the target candidate, centered at y in the current frame (Figure 3). The normalization is inherited from the frame containing the target model. Using the same kernel profile $k(x)$, but with bandwidth h , the probability of the feature $u = 1,2,...,m$ in the target candidate is given by [7]

$$p_u(y) = C_h\sum_{i=1}^{n_h} k(\left\|\frac{y - x_i}{h}\right\|^2)\delta\left[b(x_i) - u\right] \qquad (7)$$

Where

$$C = \frac{1}{\sum_{i=1}^{n_h} k(\left\|\frac{y - x_i}{h}\right\|^2)} \qquad (8)$$

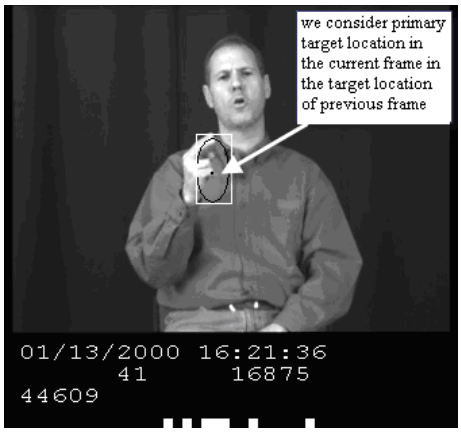

Figure 3. Target candidate in the second frame of the figure 2.

Is the normalization constant Point out that C_h does not depend on y , since the pixel locations x_i are organized in a regular lattice and y is one of the lattice nodes.

C_h is calculated for specific kernel and different values of h . The number of pixels considered in the localization process is obtained according to h.

2.2.4. Similarity function smoothness

The kernel profile $k(x)$ gives its properties to the similarity function (4).When the target model and candidate are expressed according to (5) and (7). A variable kernel profile leads to a variable similarity function and efficient gradient based optimization procedure can be utilized to find its maxima. The continuous kernel represents an interpolation process between the location and image structure. The used target representations do not limit the

similarity and then various functions can obtain ρ . In [9], an experimental evaluation of different histogram similarity has been shown where.

2.2.5. Metric based on Bhattacharyya Coefficient

The similarity function defines a distance between a target model and candidates [7]. This distance should have a metric structure to accommodate comparisons among various targets. We define the distance between two discrete distributions as a formula 9,

$$d(y) = \sqrt{1 - \rho\left[p(y), q \right]} \quad (9)$$

Where we chose formula 10.

$$\rho(y) \equiv \rho\left[p(y), q \right] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \quad (10)$$

The sample estimate of the Bhattacharyya coefficient between p and q [10, 7]. The Bhattacharyya coefficient is a divergence-type measure [11] that has a straightforward geometric interpretation. It is the cosine of the angle between the *m*-dimensional unit vectors $(\sqrt{p_1}, \sqrt{p_2}, ..., \sqrt{p_m})^T$ and $(\sqrt{q_1}, \sqrt{q_2}, ..., \sqrt{q_m})^T$. The fact that H and C are distributions is thus explicitly taken into account by representing them on the unit hyper sphere. At the same time, we can interpret (10) as the (normalized) correlation between the vectors $(\sqrt{p_1}, \sqrt{p_2}, ..., \sqrt{p_m})^T$ and $(\sqrt{q_1}, \sqrt{q_2}, ..., \sqrt{q_m})^T$. Properties of the Bhattacharyya coefficient such as its relation to the Fisher measure of information, quality of the sample estimate, and explicit forms for various distributions are shown in [12, 10].

The statistical measure (9) has several desirable properties:

1. It imposes a metric structure (see Appendix). The Bhattacharyya distance [13, p.99] or Kullback divergence [14, p.18] are not metrics when they violate at least one of the distance axioms.
2. It has an obvious geometric interpretation. Note that the L_p histogram metrics including histogram intersection [15]) do not enforce the conditions $\sum_{u=1}^{m} q_u = 1$ and $\sum_{u=1}^{m} p_u = 1$.
3. It utilizes discrete densities, and therefore, it is invariant to the scale of the target (up to quantization effects).
4. It is valid for arbitrary distributions, thus being superior to the Fisher linear discriminate, which yields useful results only for distributions that are separated by the mean-difference [13, p.132].
5. It approximates the chi-squared statistic, while avoiding the singularity problem of the chi square test when comparing empty histogram bins [16].

Divergence based measures were already used in computer vision. The Chern off and Bhattacharyya bounds have been applied in [17] to determine the effectiveness of edge detectors. The Kullback divergence between joint distribution and product of marginal (e.g., the mutual information) has been used in [18] for registration. Information theoretic measures for target distinctness were discussed in [19].

2.2.6. Target Localization

The distance (9) should be minimized as a function of y to find the location corresponding to the target in the current frame [7]. The localization procedure starts from the target position in the previous frame (the model) and searches in the neighborhood. Since our distance function is smooth, the procedure uses gradient information provided by the mean shift vector [20]. More involved optimizations based on the Hessian of (9) can be used [21].

Color information was chosen as the target feature. However, the same framework can be utilized for texture and edges, or any combination of them. In the sequel, it is assumed that the following information is available: (a) detection and localization in the initial objects frame to track (target models) [22, 23]; (b) periodic analysis of each object to account for possible updates of the target models due to significant changes in color [24].

2.2.7. Distance Minimization

Minimizing the distance (9) is equal to maximizing the Bhattacharyya coefficient $\rho(y)$ [7]. The search for the new target location in the current frame starts at the location y_0 of the target in the previous frame. Therefore, the probabilities $\{p(y_0)\}_{u=1,2,...,m}$ of the target candidate at location y_0 in the current frame first have to be computed. Using Taylor expansion around the values $p_u(y_0)$, the linear approximation of the Bhattacharyya coefficient (10) is derived after some manipulations as [7]

$$\rho[p(y),q] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(y_0)q_u} + \frac{1}{2}\sum_{u=1}^{m} p_u(y)\sqrt{\frac{q_u}{p_u(y_0)}} \quad (11)$$

The approximation is satisfactory. When the target candidate $\{p_u(y)\}_{u=1,2,...,m}$ does not change drastically from the initial $\{p_u(y_0)\}_{u=1,2,...,m}$, that is, most often a valid assumption between consecutive frames. The condition $p_u(y_0) > 0$ (or some small threshold) for all $u = 1,2,...,m,$ can be enforced by not using the feature values in violation. Recalling (7) results in [7] – you should illustrate the formulas below not just simply dropping them down the text!!!!

$$\rho[p(y),q] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(y_0)q_u} + \frac{C_h}{2}\sum_{u=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (12)$$

Where

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{q_u}{p_u(y_0)}}\delta\left[b(x_i) - u\right] \quad (13)$$

Therefore, to minimize the distance in the formula (9), the second term in (12) has to be maximized, the first term being independent of y. See that the second term represents the density estimate computed with kernel profile $k(x)$ at y in the current frame, with the data being weighted by w_i (13). The mode of this density in the local neighborhood is the sought maximum that can be found applying the mean shift procedure [20]. In this procedure the kernel is recursively moved from the current location y_0 to the new location y_1 according to the relation

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \quad (14)$$

Where $g(x) = -k'(x),$ assume that the derivative of $k(x)$ exists for all $x \in [0,\infty),$ except for a finite set of points. The complete target localization algorithm is presented in the following.

The target model $\{q_u\}_{u=1,2,...,m}$ and its location y_0 in the previous frame.

1. Initialize the location of the target in the current frame with y_0, compute $\{p_u(y_0)\}_{u=1,2,...,m}$, and evaluate $\rho\left[p(y_0),q\right] = \sum_{u=1}^{m}\sqrt{p_u(y_0)q_u}$
2. Obtain the weights $\{w_i\}_{i=1,2,...,n_h}$ according to (13).
3. Find the next location of the target candidate according to (14).
4. Compute $\{p_u(y_1)\}_{u=1,2,...,m}$, and evaluate $\rho\left[p(y_1),q\right] = \sum_{u=1}^{m}\sqrt{p_u(y_1)q_u}$
5. While $\rho\left[p(y_1),q\right] < \rho\left[p(y_0),q\right]$

   Do $y_1 \leftarrow \frac{1}{2}(y_0 + y_1)$

   Evaluate $\rho\left[p(y_1),q\right]$
6. If $\|y_1 - y_0\| < \varepsilon$ Stop.

Otherwise set $y_0 \leftarrow y_1$ and go to Step 2.

2.2.8. Implementation of the Algorithm

The stopping criterion threshold ε used in Step 6 is obtained by constraining the vectors y_0 and y_1 to be within the same pixel in *original* image coordinates [7]. A lower threshold leads to the sub pixel accuracy. From real-time constraints (i.e., uniform CPU load in time), we also limit the number of mean shift iterations to $N_{\max}$, typically taken equal to 20. In practice, the average number of iterations is much smaller than about 4.

Implementation of the tracking algorithm can be much simpler than what is presented above. The Step role 5 is only to avoid potential numerical problems in the mean shift based maximization. These problems can appear because of the linear approximation of the Bhattacharyya coefficient. However, a large set of experiments tracking different objects for long periods of time has shown that the Bhattacharyya coefficient computed at the new location y_1 failed to increase in only 0.1% of the cases. Thus, the Step 5 is not used in practice,

and as a result, there is no need to evaluate the Bhattacharyya coefficient in Steps 1 and 4.
We only iterate by computing the weights in Step 2 in the practical algorithm deriving the new location in Step 3, and testing the size of the kernel shift in Step 6. The Bhattacharyya coefficient is computed only after the algorithm completion to evaluate the similarity between the target model and the chosen candidate.
Kernels with Epanechnikov profile [20, 7] are recommended to be used.

$$k(x)=\begin{cases}\frac{1}{2}C_d^{-1}(d+2)(1-x) & \\ 0 & \end{cases} \quad (15)$$

$if\ x \le 1$

$otherwise$

In this case, the derivative of the profile, $g(x)$, is constant and (14) reduces to

$$y_1=\frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (16)$$

i.e., a simple weighted average.
The maximization of the Bhattacharyya coefficient can be also interpreted as a matched filtering procedure. In fact, (10) is the correlation coefficient between the unit vectors $\sqrt{q}$ and $\sqrt{p(y)}$, representing the target model and the candidate. Thus the mean shift procedure finds the local maximum of the scalar field of correlation coefficients. Will call the *operational basin of attraction* the region in the current frame in which the new location of the target can be found by the proposed algorithm. This basin is at least equal to the size of the target model due to the use of kernels. In other words, if in the current frame, the center of the target remains in the image area covered by the target model in the previous frame, and the local maximum of the Bhattacharyya coefficient is a reliable indicator for the new target location. We assume that the target representation provides sufficient discrimination, such that the Bhattacharyya coefficient presents a unique maximum in the local neighborhood. The mean shift procedure finds a root of the gradient as location function that can also correspond to a saddle point of the similarity surface. The saddle points are unstable solutions, and since the image noise acts as an independent perturbation factor across consecutive frames, they cannot affect the tacking performance in an image sequence. The best algorithm for mean shift is shown in (Figure 4)

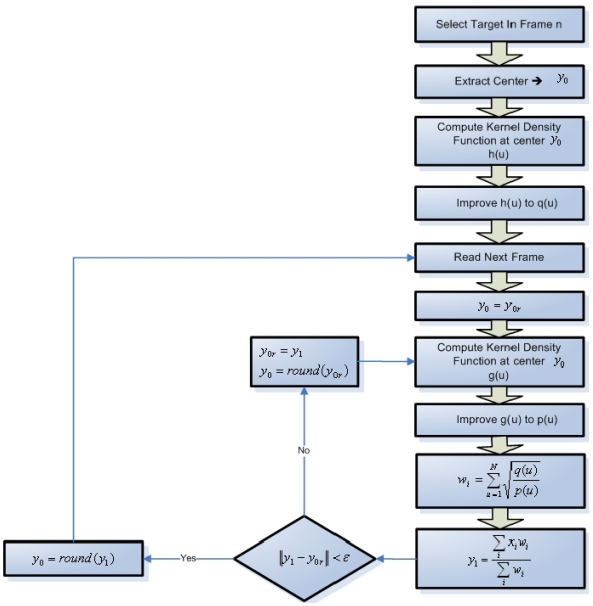

Figure 4.Complete block diagram of the mean shift tracker.

3. Databases

All presented databases in this section are easily available for much research in linguistics and recognition. The data were collected and recorded by Boston University, the database subsets build up benchmark databases that can be used for the automatic recognition of isolated and a continuous sign language, respectively, and so were they defined at the RWTH Aachen University. We briefly describe some commonly used statistical measures w.r.t. automatic recognition in the following: running words are the total number of words in the corpus unique words, which determine the vocabulary size singletons. These are words (or word tuples) that occur only once zerogram-, unigram-, bigram-, trigram- language models describe different linguistic contexts.

3.1. RWTH-BOSTON-104

The RWTH-BOSTON-104 database is based on the sign language published by the national gesture reorganization and the sign language of Boston University. This database has basically registered for studying the language structure and grammar of America sign language (ASL). The RWTH-BOSTON-104 database consists of 201 annotation videos from ASL sentences. These sentences were produced by 3 people (1 man and 2 women) and videos have been taken by the 4 cameras at the same time, 2 of them are located on the forward and show the person forward view, 1 of them is lied

beside the person and the last one records the face picture. Videos are taken by speed of 30 frames per second with high resolution. All videos are Gray – Scale except camera related to face picture.

4. Experiments and Results

In this section, the results of applied algorithms on different images have been showed In Figure 5. This shows hand tracking for the usual image by applying just mean shift algorithm. Here 5 frames (Frame 21, 38, 40, 41 and 45) from 76 frames are determined. In this case, hand movements are slow and also resolution and quality of image are acceptable. As illustrated, this algorithm could not track the hand as an object tracking properly.

Figure 6 shows hand tracking for the usual image by applying just motion detection algorithm. This algorithm could not track just hand as an object tracking. As seen in some frames face, the other hand has been chosen as a target.

Figure 7 shows hand tracking for the usual image by applying the combination of motion detection and mean shift algorithms. This algorithm could track just hand as an object tracking better than above algorithms.

Figure 8 shows hand tracking for the complicated image by applying the combination of motion detection and mean shift algorithms. In Figure 8, the resolution and quality of images are not so proper, images are noisy and blurring, and hand movements are faster than the old ones but as seen, this algorithm could track the hand as an object tracking so efficiently. These pictures have been taken from the deaf forum Iran's site but the old pictures are related to the main database, RWTH-BOSTON-104.

Figure 5. Hand tracking for the usual image by applying just mean shift algorithm.

Figure 6. Hand tracking for the usual image by applying just motion detection algorithm.

Figure 7. Hand tracking for the usual image by applying the combination of mean shift and motion detection algorithms.

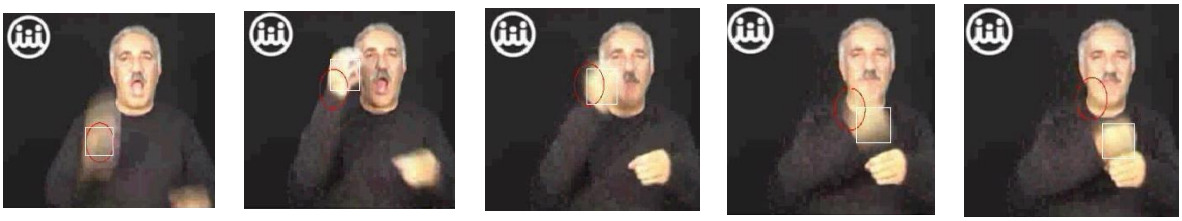

Figure 8. Hand tracking for complicated image by applying mean shift and motion detection algorithms.

This algorithm in spite of some other famous algorithm used for target tracking, such as dynamic programming tracking [25], which is proper in real time works. Due to using the mean shift algorithm, the less computation is needed to track the target and it is faster than other common algorithms, such as it is 2 times faster than dynamic programming tracking when we use the same processor. Also, it needs less memory to track the target; therefore, it is useful for implementation practically.

5. Conclusions

Mean shift algorithm is just based on the color feature and sometimes loses the hand completely, so it always cannot track the hand correctly such as complicated images that the hand is lost because face and hand skin have the same color and then occlusion happens. Mean shift algorithm cannot consider the multimode levels, so it converges on the local maximum and cannot track the hand efficiently if there is the same color object with the hand. Motion detection algorithm is based on the object motion and cannot track just the hand as a target. The old solutions are complicated, time consuming and so expensive. However, the proposed method is so easy, fast, and efficient and low costly. The used combination method can track the hand properly when there are noise and parasite in the background. Because, it is able to consider the multimode levels and detect the hand as an object tracking. Simulation results showed our method could track more appropriately than the both of mean shift and motion detection algorithms. It is so efficient even in complicated images.

References

[1] Birchfield, S., 1998. "Elliptical Head Tracking Using Int ensity Gradients and Color Histograms," Proc.IEEE Conf. ComputerVision and Pattern Recognition, pp: 232-237.

[2] Black, M. and D. Fleet, 2000. "Probabilistic Detection and Tracking of Motion Boundaries," Int'l J.Computer Vision, 38(3): 231-245.

[3] Honggang Wang, Ming C. Leu and Cemil OZ. "American Sign Language Recognition Using Multi-dimensional Hidden Markov Models," journal of information science and engineering 22, 1109-1123 (2006).

[4] Y. Cheng. "Mean shift, mode seeking, and clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, Vol.17, No. 8, pp. 790-799.

[5] D. Comaniciu, V. Ramesh, P. Meer. "Kernel-based object tracking". IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, Vol. 25, No. 5, pp. 564-577.

[6] Ruiz-del-Solar, J. and Vallejos, P.: Motion Detection and Tracking for an AIBO Robot Using Motion Compensation and Kalman Filtering. RoboCup 2004 Symposium, Lecture Notes in Computer Science (accepted).

[7] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.

[8] D. W. Scott, Multivariate Density Estimation. Wiley, 1992.

[9] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in Proc. 7th Intl. Conf. on Computer Vision, Kerkyra, Greece, 1999, pp. 1165–1173.

[10] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," IEEE Trans. Commun. Tech., vol. 15, pp. 52–60, 1967.

[11] J. Lin, "Divergence measures based on the Shannon entropy," IEEE Trans. Information Theory, vol. 37, pp. 145–151, 1991.

[12] Djouadi, O. Snorrason, and F. Garber, "The quality of training-sample estimates of the Bhattacharyya coefficient," IEEE Trans. Pattern Anal. Machine Intell., vol. 12, pp. 92–97, 1990.

[13] K. Fukunaga, Introduction to Statistical Pattern Recognition. Academic Press, second edition, 1990.

[14] T. Cover and J. Thomas, Elements of Information Theory. John Wiley & Sons, New York, 1991.

[15] M. Swain and D. Ballard, "Color indexing," Intl. J. of Computer Vision, vol. 7, no. 1, pp. 11–32, 1991.

[16] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," Kybernetika, vol. 34, no. 4, pp. 363–368, 1998.

[17] S. Konishi, A. Yuille, J. Coughlan, and S. Zhu, "Fundamental bounds on edge detection: An information theoretic evaluation of different edge cues," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, 1999, pp. 573–579.

[18] P. Viola and W. Wells, "Alignment by maximization of mutual information," Intl. J. of Computer Vision, vol. 24, no. 2, pp. 137–154, 1997.

[19] J. Garcia, J. Valdivia, and X. Vidal, "Information theoretic measure for visual target distinctness," IEEE Trans. Pattern Anal. Machine Intell., vol. 23, no. 4, pp. 362–383, 2001.

[20] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 5, pp. 603–619, 2002.

[21] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, Numerical Recipes in C. Cambridge University Press, second edition, 1992.

[22] Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in IEEE Workshop on Applications of Computer Vision, Princeton, NJ, 1998, pp. 8–14.

[23] M. Black and D. Fleet, "Probabilistic detection and tracking of motion boundaries," Intl. J. of Computer Vision, vol. 38, no. 3, pp. 231–245, 2000.

[24] S. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixture models," Image and Vision Computing Journal, vol. 17, pp. 223–229, 1999.

[25] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, H. Ney: Tracking Using Dynamic Programming for Appearance-Based Sign Language Recognition. In Proceedings of the 7th International Conference of Automatic Face and Gesture Recognition, Southampton, UK, 2006.