19

Bimanual Hand Tracking based on AR-KLT

Hye-Jin Kim, Keun-Chang Kwak and Jae Jeon Lee
Intelligent Robot Research Division
Eelectronics and Telecommunications Resarsh Institute,
Korea

1. Introduction

Robot and human interaction has received a significant amount of attention in the robot vision research community in the past decades. This has been motivated by the desire of understanding human gesture/motion tracking and recognition. If you solve tracking problems under the circumstance of fast movement, occlusion, and illumination, then you need to complicate calculation, and hence the computational complex prevents to work in real time. For example, particle filter is an useful algorithm to track objects, even under occlusion and non-rigid motion difference. However, particle filter needs to enough samples to support reliability of the potential candidates of the target. There have done many works in hand tracking. To track hands in real time, Shan(Shan, 2004) made particle filter faster by reducing sample size according to mean shift. On the other hand, Kolsch (Kolsch&Turk, 2004) designed a fast tracking algorithm that combined Kanade-Lucas-Tomasi(KLT) flocks and k-nearest neighborhood.

Some papers concentrated on the particular properties of hands and their features. Non-rigidity of the hand causes difficulties to track because of non-linear dynamics of the articulation. Fei and Reid(Fei&Reid, 2003) dealt with deformation of the hand by constructing two models according to non-rigid motion from rigid motion. HLAC (Higher-Order Local Auto-Correlation) features of Ishihara (Ishihara&Otsu, 2004) achieved efficient information over time domain by Auto-Regressive model.

The size of interesting objects is another critical factor for tracking because if its size is too small or changes too fast, object tracking becomes very challenging problem. Francçis(Francçis,2004) dealt with blobs varying their resolution, hence made it possible to track the object with various size in the image sequence. Both hands tracking is simultaneously different from one hand tracking since features such as shape, color etc. between both hands is almost the same each other. Shamaie (Shamaie&Sutherland, 2003) built the model of the movements of bimanual limbs. However, the model needs large enough time to compute distance transform in the image. McAllister(McKenna et al., 2002) solved the both hands tracking by employing contour distance transform and 2D geometric model.

In this paper, we propose a new 2D both hands tracking algorithm based on the articulated structure of human body in real time. This method is efficient enough to perform in real time due to the limb model tracking. The model enables to deal with the deformation of hands and nonrigid motion because of the articulate structure of the arm for both hands.

Source: Scene Reconstruction, Pose Estimation and Tracking, Book edited by: Rustam Stolkin, ISBN 978-3-902613-06-6, pp.530, I-Tech, Vienna, Austria, June 2007

The model can be tracked by a linear line obtained from the regression of KLT features in order to represent the target information. Unlike Shamaie and McAllister, the proposed algorithm outperforms previous method in occlusion handling of both hands. For instance, some methods require restricting occlusion cases because similar features prevent a hand to differentiate from another. However, this method tracks superimposed hands correctly by virtue of its prediction of the moving direction.
In the next section, we will elaborate our proposed algorithm step by step. In the section 2 A-B, we will illustrate key algorithms to build our model. In Section 2.3, we give brief explanation about how to segment and extract hands from the background. The section 2.4 is dedicated to the dynamic model and the algorithms for occlusion detection and tracking. Some experimental results are presented in the section 3. Our contribution in hand tracking and conclusion are presented at the end of paper.

2. Articulate Hand Motion Tracking Method

2.1 Building the auto-regression model

Auto-regression model is one of dynamical mode that is a statistical framework for motion tracking. Through accumulated motion sequences, dynamical model obtains the information to predict motion in the next frame. Second-order auto-regression model is a special Markov process model with gaussian priors $\mathbf{X} \sim N(\overline{X}, \mathbf{V})$, the dynamical model

$$p(X(t_k) \mid X(t_{k-1})) \propto \exp\{-\frac{1}{2}\left\|B^{-1}(X(t_k) - AX(t_{k-1}))\right\|^2\} \tag{1}$$

Also the Markov process can be expressed in a generative form:

$$X(t_k) - \overline{X} = A_2(X(t_{k-2}) - \overline{X}) + A_1(X(t_{k-1}) - \overline{X}) + B_0 w_k \tag{2}$$

where A_2, $A_1$and B_0 are all $N_X \times N_X$ AR coefficients. We set the order of auto-regression model as 2 because it can handle motions with different velocity and noisy direction [8].

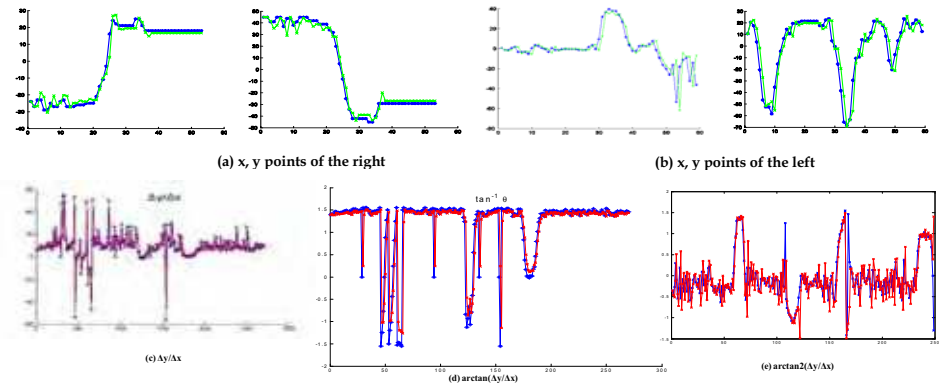

Figure 1. Learning and prediction accuracy using auto-regressive second-order model. Blue line stands for the original data and green and red lines are predicted data by AR2 model

AR2 coefficients were learned from the manually marked ground truth data. Training data consist of 100 samples and test data 53 samples with 7 dimensions with respect to x and y point of 2D hand, elbow and shoulder points, and a slope between an elbow and an hand. The AR2 model predicts most points well except for the gradients. Fig.1 (a) is the image of test samples (above row). Fig.1 (b) shows how to estimate the slope. We attempt to extract the slope in several ways: (1) $\Delta y/\Delta x$ (2) $\theta_1 = tan^{-1}(\Delta y/\Delta x)$ (3) $\theta_2 = tan_{\,2}^{-1}(\Delta y/\Delta x)$. The period of θ_1 ranges from $-\pi/2$ to $\pi/2$ and that of θ_2 ranges $-\pi$ to π. We emphasize on the gradient factor because it gives useful clues that a predicted hand belongs to which side when both predicted hands are crossed each other.

2.2 KLT features and linear regression

KLT features, named after Kanade, Lucas and Tomasi, provide steepest density gradients along the x and y directions (see [9]). The features are corner points with the largest eigenvalues. The size of each feature represents the amount of context knowledge and depends on two factors: quality level of a corner' intensity and minimum distance between corner points. To match the image I and J, the current and the next image, we minimize the error function ε by the following equation:

$$\varepsilon = \iint_W [J(x) - I(x)]dx \tag{3}$$

where W is the given feature window and w(**x**) is a weighting function. Minimizing Eq.(3), you find the A and d corresponding to the affine motion field and the translation of the feature window's center, respectively. The largest eigenvalue of A estimates feature quality. In the presented system, KLT features calculate their density gradient on the skin and motion image when the object has motion or on the skin image if there is no object to move.

(a) image frame (b) body structure (c) slope estimation

Figure 2. We define a body structure consists of hands, shoulders and elbows. The elbow points, green cross marks in (a), represent the base point of the arm's slope. In the first frame, the elbow points are assumed by the ratio of the length between a shoulder and an elbow and the length between the elbow and a hand. For the slope estimation, the difference between slope($\theta^*-\theta$)is considered to predict the next change of the slope.

By adjusting the feature size in the skin image or in the skin-motion image, KLT features can be spread out over the whole image plane. Therefore, we filtered KLT features using mean and variance constraint. That is, we removed all KLT features of which variances are more than 2.5 times the overall variance. Linear regression is applied to the filtered KLT features in order to get the slope and find the end point of each hand. Here, we note that the end point extraction needs a reference point because the slope and y-axis intercept need to fit the

exact tracking position by removing noisy data and abstract the structural information of the arm by linear regression. This bias will be adjusted and removed by the reference point. As you can see in Fig.2, we construct the arm model for reflecting articulate motion of hands in tracking issue. There are three points for each arm: the shoulder, elbow and hand points. We take each elbow point into the reference point instead of the shoulder point because if you set the shoulder point up as the reference point, then you may lose the elbow point and cannot figure out the status of arms : stretched out , curved and so on. Fig. 2(c) shows the elbow and hand points and the line between them and the slope of the line illustrating in Fig. 2(c) gives the directional information when the occlusions within hands and arms are detected.

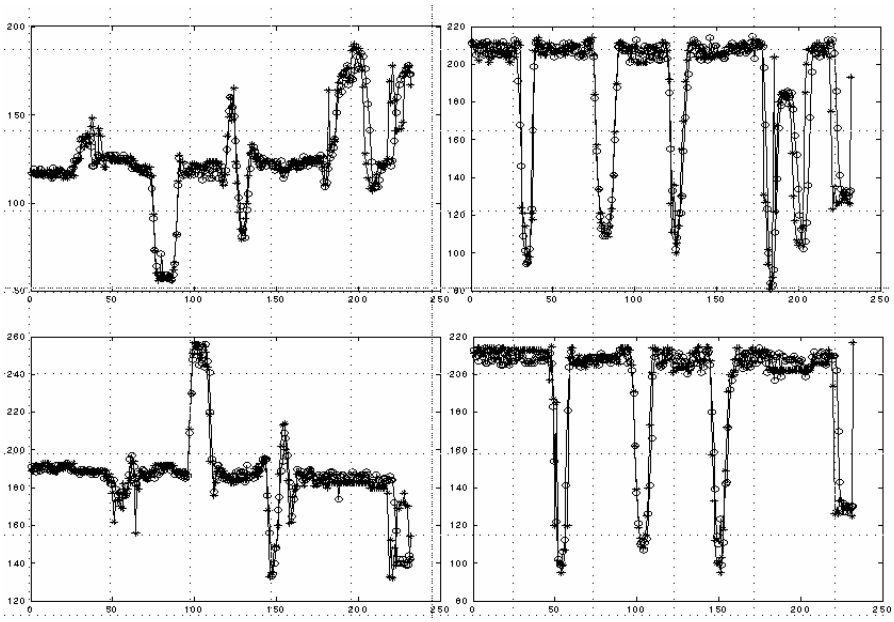

Figure 3. 'x-' line: original data, 'o-' line: tracked data. Left side plots: x-axis movement, right side plots:y-axis movement. Tracking results of the right (upper row) and left (lower row) hand.

Therefore, we know which hand is the right hand and which one is the left when one hand, completely or partially, over the other hand. In short, the regression of KLT features gives you the position of the hands and the direction to move.

2.3 Hand detection and pre-processing

Skin-color and motion cues are adopted for pre-processing the image. Motion cues are obtained from differentiating the current frame with the previous frame. Skin-color segmentation requires the following four steps.

1. Construct skin-color database with about one million size samples on RGB plan.
2. Generate non-skin color database.
3. Train the skin-color pixel and non-skin color pixel after transforming RGB spaces into YUV spaces.

4. Obtain the U-V image sequence along the Y plan.
5. Create the representative U-V lookup table of skin-color at the mean point of Y.
6. Find skin-color pixels in an input image using the U-V lookup table.

The fourth and fifth steps are essential steps to achieve real-time skin color segmentation. In the third step, skin-color detection scheme needs 256×256× 256 comparison per a pixel on the YUV space. However, it is revealed that the trained U-V ranges did not have much difference on Y-spaces. Therefore, U-V values are chosen at the average point of Y as the skin-color lookup table.

Finally, the interesting target the motion of skin-colored regions, the input image is processed by the logical AND operator between the color probability image and the difference image.

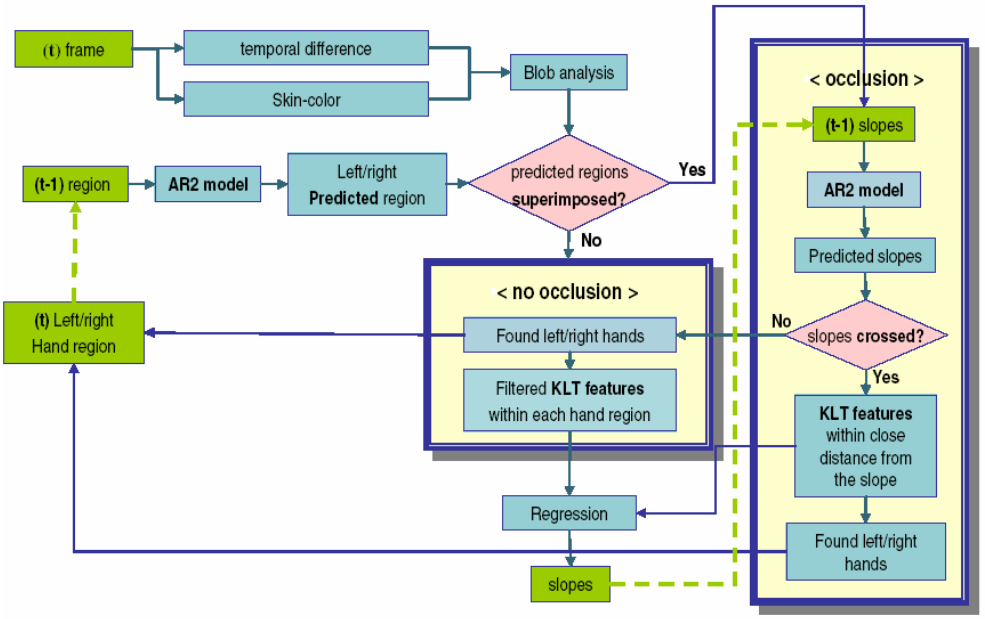

Figure 4. Overview of fast 2D both hands tracking with articulate motion prediction

2.4 A dynamic model for occlusion detection and tracking

In this section, detecting occlusion and tracking is dealt with for both hands at once as shown in Fig.4.

For tracking hands in an image, the limb model is useful to predict future movements of each hand and catch occlusion. Tracking is divided by two parts: crossed motion and uncrossed motion. Hand occlusion in the next frame can be detected by the following factors: (1)the size of superimposed region between predicted areas of both hands should be large enough; (2) the product of two slopes from left and right hands should be non-positive; (3) the amount of slope changes from one frame to a consecutive frame should be beyond threshold. If these three conditions are all satisfied, it is the alarm that two hands are crossed each other.

Detection and tracking issues highly depend on characteristic of targets. Therefore, it is hard to find targets such as both hands with similar color and similar shape. This fact invokes the need of special features that can decide whether a hand belongs to left or right one. The proposed method uses directivity of hands because the limb structure of human body enables to restrict discriminative movement for each hand. Directivity can be obtained by KLT features and its regression result as already shown in 2.2. When predictive both hands are occluded each other, KLT features are collected when they are close enough to the predicted linear line from the previous frame. Closeness is calculated by the following eq. (4). Where a line equation $ax + b - y = 0$ and a point $(x0, y0)$ are given, the distance d is obtained by

$$d = \frac{| ax_0 + b - y_0 |}{\sqrt{a^2 + b^2}} \tag{4}$$

The close KLT feature to the predicted line is highly possible a candidate of the target feature in the current frame. Therefore, the filtered KLTs are regressed in order to find the proper end point of the hand.

On the other hand, this method analyzes blobs on the skin and motion image since blobs segments generic features without domain-dependent information. Difficulties that use blobs are the change of size and the velocity of a object corresponding to a blob. Such changes can be serious under the Ubiquitous Robotic Companion (URC) circumstance where image transmission is usually much slower than other mediums such as USB camera because of server-robot transmission system. Fast movement and sudden magnification/reduction of a target leads to lose the target information, preventing from tracking. In the proposed method, the AR2 dynamic model is used for eliminate such risk because second-order of auto-regression can enlarge/abridge the search range of the target according to the status of the target movement. Moreover, the 2nd-order dynamic model gives the alarm of the occlusion. It is a cue of occlusion that each predictive region of both hands coincides with the same place. Tracking system selects occlusion process as shown in Fig.4 based on that cue.

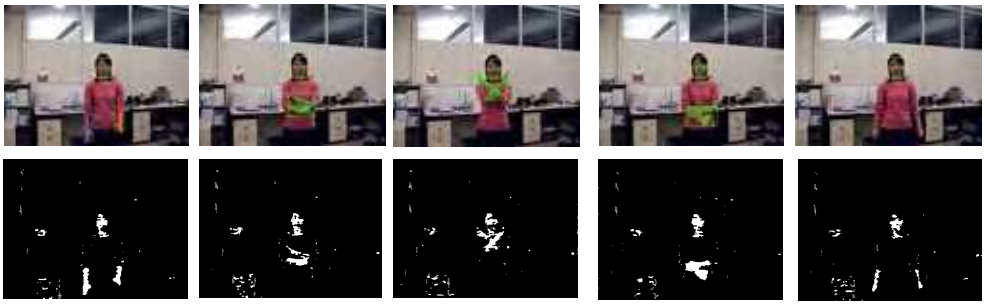

Figure 5. Both hands are crossed each other

3. Experimental results

Fig.1 (a) and (b) are the image of test samples (above row). We attempt to extract the slope in several ways: (1) $\Delta y/\Delta x$ (2) $\theta_1 = tan^{-1}(\Delta y/\Delta x)$ (3) $\theta_2 = tan_{\,2}^{-1}(\Delta y/\Delta x)$. The period of θ_1 ranges from $-\pi/2$ to$\pi/2$ and that of θ_2 ranges $-\pi$ to π. We emphasize on the gradient factor because it gives useful clues that a predicted hand belongs to which side when both predicted hands are crossed each other. Fig.1 (a) and (b) represented how well the learned AR2 parameters predicted $\Delta x/\Delta y$ coordinates of both hands. Especially in Fig.1 (b), predicted points were

well tracked even if fast movement - the rapid change at x or y coordinate axes of the hand occur. On the other hand, Fig.1(c)-(e) showed that the gradient was hard to make a pre-estimation. Although various approaches such as tangent and arc tangent were taken to calculate gradients, it is revealed that the gradient was very sensitive to the difference of the x coordination, Δx between (t-1) frame and (t) frame. For example, negligible Δx much less than 1 could cause remarkable change of its gradient but such difference in an image can be considered as roughly no change. In other words, although the hand stayed little motion along the x-axis in an image changes, robot considered it big hand movement while human-beings can ignore such changes. Therefore, the effort to reduce the effect of Δx was made by transforming the gradient $\Delta y/\Delta x$ into $\theta_1 = \arctan(\Delta y/\Delta x)$, or $\theta_2 = \arctan2(\Delta y/\Delta x)$. However, some parts still failed to get correct prediction because tangent and arc tangent is a trigonometrical function having own period. That is, prediction could not but be failed at the extreme point of its period, θ_1: $-\pi/2$ and $\pi/2$ and θ_2 : $-\pi$ and π, as shown in Fig.1(c)-(e). Despite of such restrictions of the slope prediction, the gradient information can provide the key clue that a hand belongs to left or right one. To adopt benefits of slope, tracking process was decomposed into two processes (see Fig.4). One is for the uncrossed hand tracking. Here, the slope information is kept in until the next frame. This process used x- and y-coordinates of both hands and confirmed the tracking result. Another handles the hand occlusion. That is, if the occlusion is detected by the AR2 model, then the previous slope for each hand is prepared for finding the correct hand position.

For the bimanual tracking, it is hard to figure out whether both hands are crossed each other as well as which hand is a left or right one because both hands' properties are almost the same. Our method proposes a good feature to discriminate two hands: directivity. The well-known law of inertia can tell that a hand belongs to a right or left hand because moving objects suddenly do not change its direction. The directivity can be obtained from the slope. Fig.4 shows that the slope gives a cue whether both hands are superimposed. According to this information, we can track both hands simultaneously as shown in Fig.5.

In order to measure the performance of the algorithm, 900 experiments were performed on many different hand shapes. The result of the experiments is listed in Table 1. Fig.3 shows a part of our experimental results. We performed the experiment using multimodal hand gesture database such as drawing 'O' and 'X', pointing left and right and so on. In Fig.3, the movement velocity along y-axis is higher than x-axis direction. Despite the velocity difference, our proposed algorithm adaptively found correct hand position whether its velocity is fast or slow.

Another important issue in tracking is that an algorithm can be simulated under the real time system. Wever, a robot for cheap practical use, has limited computing power, can transport an image through the internet only in 6.5 frames per a second on average without additional image processing. Furthermore, the target, hand, was often found out of detectable range because of slow image transportation. Under this circumstance we achieved real-time tracking in 4.5 frames per a second.

Side	Left	Right
Hand(x –axis)	94.39±0.62	94.84±0.56
Hand(y –axis)	93.01± 1.96	91.32± 1.75
Elbow(x-axis)	99.09±0.68	99.78±0.30
Elbow(y-axis)	99.77±0.33	99.57±0.61

Table 1. Tracking accuracy

4. Conclusion

Our ARKLT method is very useful for tracking and gesture recognition. As mentioned before, the ARKLT method consists of three points for each hand: the shoulder, elbow and hand. Since the model reflects the articulated motion of the human body which is restrained by the each limb's degree of freedom. That is, the possible region for hand movement is restricted in the elongated region of the shoulder and elbow movement. Therefore, the proposed method can devise effective prediction method, which enables to pre-detect crossing hands based on the body structure. In addition, the proposed method applies the KLT features and their regression line so that the body structure can effectively be fitted into the target. Also, the well-fitted KLT line can provide the exact point of a hand; meanwhile most tracking methods provide the broad region of the target. When it comes to practical uses such as gesture recognition, the find location of the target improves to draw accurate outcome, for example, gesture recognition.

5. Acknowledgement

This work was supported in part by IT R&D program of MIC & IITA [2005-S-033-03, Embedded Component Technology and Standardization for URC].

6. References

Blake, A. & Isard K. (2000) *Active Contours* Springer ISBN3-540-76217-5 Springer-Verlag Berlin Heidelberg New York

Francçis, A. (2004) Real-time multi-resolution blob tracking *IRIS Technical Report*

Shamaie, A & Sutherland A. (2003) A dynamic model for real-time tracking of hands in bimanual movements *Gesture Worshop*

Shan, C. et al., Real time hand tracking by combining particle filtering and mean shift *IEEE International Conference on Automatic Faces and Gesture Recognition*

Fei, H. & Reid, I (2003) Probablistic tracking and recgonition of non-rigid hand motion *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*

Shi, J. and Tomasi, T. (1994) Good feature to track *Proc. IEEE Conference on Computer Vision and Pattern Recognition*

Kolsch, M. & Turk, M. Fast (2004) Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration Proceeding of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPR'04)

McKenna, S. J. and McAllister, G. and Ricketts, I. W. (2002) Hand Tracking for Behavior Understanding *Image and Vision Computing* Vol. 20 pp 827-840

Ishihara, T. & Otsu, N. Gesture Recognition Using Auto-Regressive Coefficients of Higher-order Local Auto-Correlation Features *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition* 2004