Monocular Vision Based Relative Depth Estimation for Hand Gesture Recognition

Wei Fan[a], Li Chen[b], Wei Liu[a], Yuan He[a], Jun Sun[a] and Shutao Li[b]
[a]Fujitsu Research and Development Center Co., Ltd.
[b]College of Electrical and Information Engineering, Hunan University, Changsha, China 410082
{fanwei, willie, heyuan, sunjun}@cn.fujitsu.com
chen220li@163.com; shutao_li@yahoo.com.cn

Abstract

In this paper, we focus on real-time relative hand depth estimation with cluttered backgrounds and variable illumination for a target application to hand pushing and pulling detection. The task is characterized by a lack of consistent internal contrast in the hand combined with the complex background. We adopt a detection-and-registration strategy to predict frame-to-frame scaling factor and make decision by the cumulated relative scale changes. The registration accuracy is enhanced by an adaptive skin color segmentation step. Our system works effectively in real time and gives a low false alarm rate.

1 Introduction

One of the fundamental problems in vision is that of detecting in-depth motion of objects through sequences of images. The estimation of object-camera-distance has a number of important applications such as video surveillance, obstacle detection and human machine interface through gestures. Much research in depth detection has been conducted using stereo vision techniques which establishes correspondence between a pair of images acquired from two well-positioned cameras. Within this paper we present a single camera based depth estimation algorithm for measuring the relative position changes along the deep direction of rigid or non-rigid objects. The application considered here is estimating the relative hand depth in monocular video sequences, but the method is equally applicable to general object depth estimation.

Hand detection and tracking are far from trivial and the hand depth estimation is even more challenging. This is due to the variability of the possible hand gestures. Hands are complex, deformable objects that are very difficult to detect in dynamic environments with cluttered backgrounds and variable illumination.

There are only a few works [1][2][3] focusing on single camera based depth detection. They generally consist of two steps: target detection and depth estimation. Target detection is accomplished by discrimination between target and background using locally computed image features such as texture, edge, etc. As a result, these algorithms only work in narrowly defined environment, and usually do not give good predictive results when they are applied to targets highly resolved in complex background scenes. The method in [2] describes an approach to moving targets detection based on some heuristics: maximum velocity, small velocity changes, coherent, and continuous motion. The method in [3] describes a real-time depth detection system for in-vehicle surveillance.

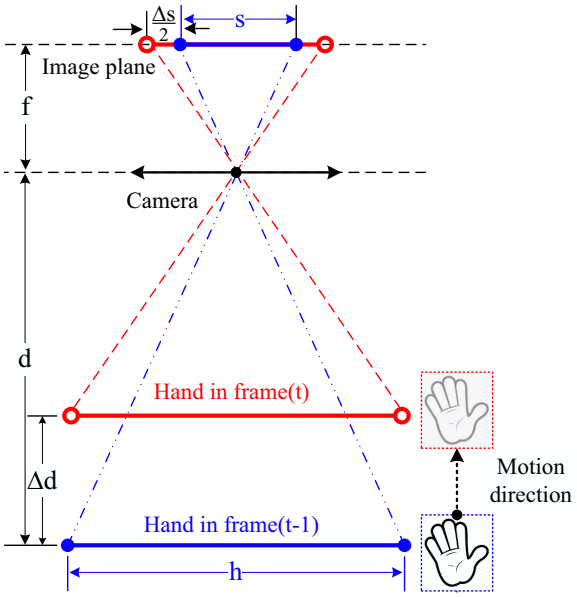

Figure 1. Imaging geometric model of one hand moving towards the deep direction

The above methods are not applicable to our task which is characterized by the significant deformation of hand and a lack of consistent internal contrast in the foreground combined with the complex background. Comparing with [2], we do not rely on explicit motion heuristics, and instead present a registration based strategy for depth estimation.

The remainder of this paper is structured as follows: Section 2 describes the details of the relative depth estimation algorithm which consists of hand detection, segmentation, registration and classification, respectively. Experimental results are presented in Section 3 and Section 4 concludes this paper.

2 The Proposed Approach

Given a sequence of hand images from a single camera, our goal is to estimate the relative hand depth change along the axis perpendicular to the image plane. Figure 1 shows the hand motion of a "push" gesture in two consecutive frames. By simple geometrical derivation, the relationship between the change of hand depth Δd and the change of apparent hand size Δs in the image plane can be written as

$$\frac{\Delta s}{s} = \frac{\Delta d}{d - \Delta d} \approx \frac{\Delta d}{d}, \qquad (1)$$

where the "≈" is reasonable since $\Delta d \ll d$ for two consecutive frames in a video. Equation 1 shows that the relative depth change of an object can be approximated by the relative size change of it's projected image.

Further derivation indicates that the apparent size change Δs is affected by the object-camera distance d as

$$\Delta s = \frac{\Delta d}{d^2} hf, \tag{2}$$

where h is the object size and f is the focal length. To guarantee good estimation precision of Δs, the operation distance d should not be too large. In Section 3, we demonstrate this effect by quantitative experimental results for two different distance settings.

The general framework of our approach to relative hand depth estimation is shown in Figure 2 which consists of four steps.

Step1. Applying a trained hand detector, the hand position is located in the image by a multi-scale sliding window approach.

Step2. The hand region is then segmented from the background clutter by online learning an adaptive skin color model.

Step3. The scale factor between hands extracted from two adjacent frames is estimated by a scale-space registration process.

Step4. Cumulated scale changes in a period are used to categorize the corresponding hand motion as pull or push gesture.

The following subsections present more details on the four constituent modules.

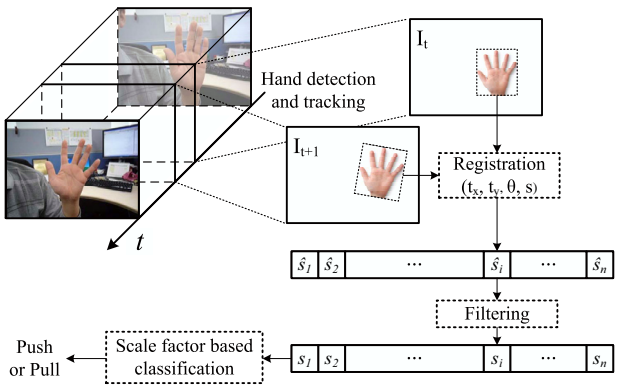

Figure 2. Overview of the relative hand depth estimation system

2.1 Hand detection and segmentation

The hand region is located by applying a Viola&Jones like object detector [5] which slides a window across the image, and applies a binary classifier to each such window. The classifier, which discriminates between the class or the background, is trained using standard machine learning techniques such as boosting or support vector machines. We calculate the values of Local Binary Patterns (LBP) [6] of all possible subrectangles inside the detection window to construct an over-complete feature pool. LBP features can act to encode ad-hoc domain knowledge that is difficult to learn using a raw and finite set of training data. Our empirical testing shows the number of LBP features needed for making a classifier is about half of the number of haar-like features [5] in terms of same accuracy. Figure 3 illustrates some detected hand postures which are trained separately. A tracking strategy is introduced in our framework which can recover the hand position up to 30 frames in case of detection failure.

Figure 3. Detected hands: fist, palm, v-sign

After hand is detected, we segment the foreground region which consists of all skin pixels from the background clutter. A good hand mask is of great importance to the registration step since it indicates the pixels which undergo notable transformation. We use an adaptive skin segmentation method to generate the hand mask. The skin model incorporates both foreground and background color likelihoods computed from two online histograms. We obtain superior segmentation results comparing with traditional Hue-thresholding based method [7]. (Figure 4)

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

Figure 4. Comparison of hand segmentation for non-skin colored background (left) and skin colored background(right), among which (a)(d) are original images, (b)(e) are segmentation results by [7] and (c)(f) are our results.

2.2 Scale space based registration model

In general, registration aims to align two images (the *reference* image I_r and the *observed* image I_o) by jointly estimating translation, rotation, and scale parameters. If I_r and I_o are defined as the temporally consecutive hand images, the predicted parameter s can be well used for relative depth estimation.

Considering the low-resolution of hand images, we adopt an area-based registration method as in [4] which directly matches pixel intensities of the two images as a whole. This is achieved by embedding a *scale-space* model into a non-linear least squares framework.

The relationship between the reference image and the observed image can be defined as follows

$$I_o(i,j) = \sum_{x,y} W(x,y;i,j,\tau) I_r(x,y), \tag{3}$$

where (x, y) represent the column and row indices of reference image I_r, and (i, j) are the column and row indices of observed image I_o. W is the weighting factor determined by the interpolation method.

The column vector $\tau = [t_x, t_y, \theta, s]^T$ denotes the registration parameters accounting for translation, rotation and scale changes between I_r and I_o. These parameters determine the location of the interpolation

operator. In this particular application the nearest neighbor interpolation strategy is chosen to minimize the computational load. W is calculated as follows.

$$W(x,y;x_c,y_c) = \begin{cases} 1 & |x-x_c|<0.5, |y-y_c|<0.5 \\ 0 & other \end{cases}, \quad (4)$$

where (x_c, y_c) is the transformed pixel coordinates of the observed image measured in sub-pixel accuracy. The relationships between (x_c, y_c) and (i, j, τ) are derived as follows

$$\begin{bmatrix} x'_c \\ y'_c \end{bmatrix} = \begin{bmatrix} x_0 + (i-1)\frac{1}{s} + \frac{1}{2s} \\ y_0 + (j-1)\frac{1}{s} + \frac{1}{2s} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (6)$$

Equation (5) defines a transformation from (i, j) to (x, y) using a single scale parameter s, where (x_0, y_0) are the coordinates of the left-top pixel of the observed image. Equation (6) expresses the affine transformation of the 2D coordinates by a rotation θ and horizontal and vertical translation (t_x, t_y).

Let $\mathbf{g}(\mathbf{x}, \tau)$ the vectorized form of the transformed version of an image $\mathbf{x}$ with registration parameters τ, the reconstruction error is calculated as follows:

$$\Delta\mathbf{y} = \mathbf{y} - \mathbf{g}(\mathbf{x}, \tau), \quad (7)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the reference image and the observed image respectively in raster scan order. Items in $\mathbf{g}(\mathbf{x}, \tau)$ are calculated according to equation (3).

We seek to find an estimate τ^* that minimizes the criterion function $J(\tau)$.

$$\begin{aligned} \tau^* &= \arg\min_{\tau} J(\tau) \\ &= \arg\min_{\tau} \frac{1}{2}\Delta\mathbf{y}^T\Delta\mathbf{y}. \end{aligned}$$

The solution of the above non-linear least squares problem can be obtained by a gradient descent procedure activated by setting an initial value τ_0 for τ^* according to the last estimated result.

In general, $\tau(k+1)$ is obtained from $\tau(k)$ by the equation

$$\tau(k+1) = \tau(k) - \eta(k)\nabla J(\tau(k)), \quad (8)$$

where η is the learning rate that sets the step size. The correction term $\nabla J(\tau)$ is calculated by

$$\nabla J(\tau) = (\frac{\partial \mathbf{g}}{\partial \tau})^t \Delta\mathbf{y}, \quad (9)$$

where $\frac{\partial \mathbf{g}}{\partial \tau}$ is the *Jacobian matrix* of $\mathbf{g}(\mathbf{x}, \tau)$ with respect to τ.

According to [8], the optimal learning rate $\eta(k)$ should be calculated in each step by

$$\eta(k) = \frac{\|J(\tau)\|^2}{J(\tau)^t \mathbf{H} J(\tau)}, \quad (10)$$

where $\mathbf{H}$ is the *Hessian matrix* of second partial derivatives $\partial^2 J/\partial\tau_i\partial\tau_j$ evaluated at $\tau(k)$. Note that the criterion function $J(\tau)$ is not quadratic, thus the $\mathbf{H}$ is not a constant and the optimal η is dependent of k. In real implementation, it takes less time to set $\eta(k)$ to a constant η that is smaller than necessary and make a few more iterations than it is to compute the optimal $\eta(k)$ at each step.

Based on the above derivation, the correction term in the iterative updating procedure $\tau \leftarrow \tau + \Delta\tau$ is approximated by

$$\Delta\tau = -\eta(\frac{\partial \mathbf{g}}{\partial \tau})^t \mathbf{M}\Delta\mathbf{y}. \quad (11)$$

Note we introduce a binary matrix $\mathbf{M}$ which restricts the registration to the region of interest for our target (i.e. the segmented hand area). This makes the registration more robust to background clutter and at the same time reduces the computational load.

2.3 Relative depth estimation

A sequence of scale factors $\{s_i\}$ is obtained from the registration process and a Kalman filter with constant velocity model is used to smooth this sequence. Figure 5 shows the predicted relative depth change $\{s_i/s_{i-1}\}$ of one person's hand when he makes pushing and pulling gestures several times in front of the camera. The ground-truth of the hand depth can be obtained from a consumer-level RGB-D camera such as Kinect. Notice the hand detection module works in a multi-resolution way and thus can give a rough scale estimation of the detected hand. The disadvantage of this approach is that (1) the scale can be estimated at only fixed levels and (2) no temporal information is used. The result is much noisy comparing with the prediction from the registration method.

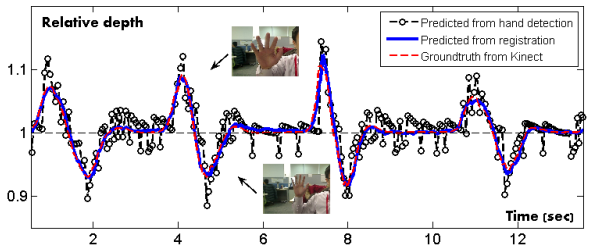

Figure 5. Relative depth estimation of one hand moving back and forth along the optical axis

Once the relative depth change is estimated from the scale factor by registration, some simple criteria can be devised to detect if a person pushes or pulls his hand in front of the camera. We assign a depth action label $l_i \in \{push, pull, uncertain\}$ to each frame I_i according to the magnitude of s_i, i.e.

$$l_i = \begin{cases} push & s_i > 1+\varepsilon_a \\ pull & s_i < 1-\varepsilon_b \\ uncertain & others \end{cases} \quad (12)$$

where ε_a and ε_b are small positive parameters controlling the classification margin. The recognition criteria for "push" and "pull" gestures are defined as follow.

T1: The minimum number of successive frames which show the same push/pull depth action labeled l_i.

T2: The minimum total scale change S_{total} along the entire sequence where $S_{total} = S_i * S_2 * \ldots * S_n$.

3 Experiments

We demonstrate the registration performance by simultaneously estimating translation, rotation, and scale parameters between two hand images captured at different times and use one for the observation and the other for reference. The transformation ground truth $(t_x = 20, t_y = -4, \theta = 19°, s = 1.25)$ is obtained by setting three pairs of corresponding points across the two images. The registration process is shown in Figures 6 and 7; with a rough initial estimate, this approach is able to find the correct registration parameters. Note in Figure 7 that without hand segmentation the observed image was registered to a wrong parameter configuration.

Figure 6. The updated estimates of the observed image parameters which eventually converges and matches the reference image.

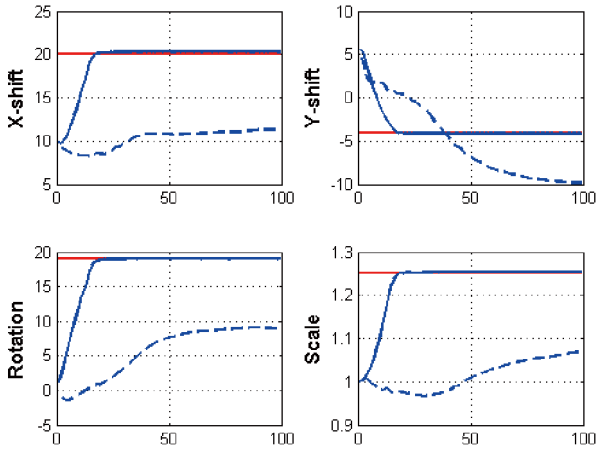

Figure 7. Estimated values for translation (x-shift, y-shift), rotation, and scale in each iteration using hand segmentation mask (blue solid line) or using full image (blue dashed line). The ground-truth values (red) are shown for reference.

To demonstrate the performance of our method, we collected a comprehensive dataset of hand videos from six persons in different poses. While collecting the data, no restriction was imposed on the environment. All experiments are carried out using 320×240 videos from a single camera directed at the user in a normal illumination condition. The system is implemented in C++ without code optimization and runs at 30 fps on a 2.53 GHz desktop PC. Table 1 shows the detection rates of *push* and *pull* gestures for palm and fist respectively, together with false alarm rates measured on hand movement samples without explicit depth change. The distance from users to camera is set to around 0.8 meter. When enlarging the working distance to 1.2 meters, we observed a notable performance decrease in detection rate ($Push$: 92.0% $\rightarrow$ 62.4%, $Pull$: 90.8% $\rightarrow$ 76.2% for palm posture). This result can be explained by Equation (2) in Section 2. We also did some experiments in an outdoor setting where illumination variation is large. Due to hand segmentation errors, the detection rate for both gestures decrease moderately - ($Push$: 92.0% $\rightarrow$ 78.8%, $Pull$: 90.8% $\rightarrow$ 81.3% for palm posture). Our method is robust to testing environments with a moderate tolerance to the variation of motion direction. The major error comes from a segmentation failure when hand overlaps with the face area and a registration failure due to out-of-plan hand rotation.

Table 1. Detection accuracy of *push* and *pull* with numbers in parentheses (#detected/#tested)

Posture	Push	Pull	False alarm
Palm	92.0%(80/87)	90.8%(79/87)	7.8%(4/51)
Fist	81.0%(64/79)	84.6%(66/79)	9.5%(7/74)

4 Conclusion and Future Work

In this paper, we present a single camera based relative hand depth estimation method with an application to *push* and *pull* gesture detection. We adopt a detection-registration strategy to predict frame-to-frame scaling factor, since our target is in low-resolution where salient feature points are difficult to extract accurately. The registration accuracy is enhanced by an adaptive skin color segmentation step. Our system works effectively in real time and gives a low false alarm rate.

Future research will involve studies on 1) extracting discriminating spatio-temporal features associated with object depth changes after sub-pixel accuracy alignment of object center and 2) exploring a more robust and effective classifier to recognize push and pull gestures from smoothed scale factor series.

References

[1] S. Min, A.Y. Ng,: "Learning 3-D Scene Structure from a Single Still Image," *IEEE ICCV*, 1–8, 2007.

[2] H. Guo, Y. Lu, S. Sarka: "Depth detection of targets in a monocular image sequence," *18th Digital Avionics Systems Conference*, vol.2, 8.A.2-1–8.A.2-7, 1999.

[3] J. Chen, J. Crossman, P. Richardson, L. Sieh: "Depth Finder, a real-time depth detection system for aided driving," *Intelligent Vehicles Symposium, Proceedings of the IEEE*, 122–127, 2000.

[4] J. Lee, S.S. Young, R. Gutierrez-Osuna: "An Iterative Image Registration Technique Using a Scale-Space Model," *Technical Report*, CSE Department, Texas A&M University, 2011.

[5] P. Viola, M., Jones: "Rapid object detection using a boosted cascade of simple features," *IEEE CVPR*, 511–518, 2001.

[6] T. Ojala, M., Pietikainen, T., Maenpaa: "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on PAMI*, vol.24, no.7, 971–987, 2002.

[7] F. Dadgostar, A. Sarrafzadeh: "An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods," *Pattern Recognition Letters*, vol.27, no.12, 1342–1352, 2006.

[8] R. O. Duda, P. E. Hart, D. G. Stork: *Pattern Classification (2nd Edition)*, John Wiley & Sons, 2000.