3D Hand Pose Detection

Ahmed ElSayed, Zhengping Wu, Ausif Mahmood
Department of Computer Science and Engineering
University of Bridgeport, Bridgeport, CT

Abstract

Hand pose detection is one of the new important research fields, especially because of its applications in the field of human machine interaction. It benefits deaf and mute people, who cannot use voice recognition systems. The main target of this paper is to design an intelligent, autonomous system that can be used for hand pose detection. The input to the system is from Microsoft Kinect's sensor for depth and color images, and the output is the hand pose that optimally matches the hand in the input image.

Proposed System

All possible hand projections can be formulated as interconnected tree of several levels. The first level can be all possible poses of the wrist hinge moves with a resolution of 10°, with the next level being possible poses for the upper level states with a small movement in the down hinges of the fingers. The following level can be the poses from the movement of upper hinges of the fingers and so on (Figure 1). All these projections arecollected by:
1)The projections of graphical hand model.
2)Collecting images for these possible poses.

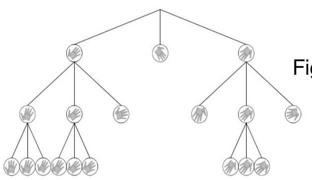

Figure 1: Expected Problem States

Figure 2 describes the designed system structure. The system consists of two stages. The first one is used to process the input image and prepare it for the match in the search stage. This first stage consists of segmentation and some image processing algorithms, which are used to separate skin areas in input image (Figure 3). The second stage of the system is a searching algorithm that will be used to search for the entire input area into stored states (which are already in the knowledgebase). Because of the big number of states in the search space, informal search using heuristic function (the matching between the input and stored state) is used.

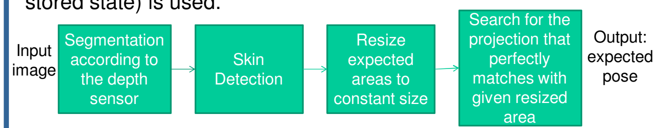

Figure 2: Proposed system structure

Figure 3: Extracted hand image of size 20x20 that is used in the matching process.

Searching and Matching Step:
First we give each one of the stored poses a name that relates it to its neighbor poses. This operation is done (naming them) by using gray code numbering system (the only difference between the poses and its neighbor is a change of one bit). The action of the searching algorithm is the change of one bit in the name of the current pose to check its neighbor.
Second, the function to be optimized is used that maximizes the normalized cross-correlation between the given input image and stored poses. The equation of the normalized cross-correlation function is given as:

$$\delta(u,v)=\frac{\sum_{x,y}\left(f(x,y)-\overline{f_{u,v}}\right)\left(t(x-u,y-v)-\bar{t}\right)}{\sqrt{\left(\sum_{x,y}\left(f(x,y)-\overline{f_{u,v}}\right)^2\right)\left(\sum_{x,y}\left(t(x-u,y-v)-\bar{t}\right)^2\right)}}$$

where :
t is the template image under test of a size m x n.
$f(x,y)$ is the captured image of a size m x n.
$\bar{t}$ is the mean of the template image.
$\overline{f_{u,v}}$ is the mean of $f(x,y)$ in the region under the template.
After exploring the known informed search algorithms, we found that Tabu Search (TS) is more suitable for the searching process. Figure 4 shows the TS Algorithm.

TabuSearch

Step 1. *Choose an initial solution f in S. Set $f^* = f$ and $k = 0$.*

Step 2. *Set $k = k+1$ and generate a subset V^* of solution in $N(f,k)$ such that*
either one of the Tabu conditions $t_r(f,m) \in T_r$ is violated $(r = 1,......,t)$
or at least one of the aspiration conditions $a_r(f,m) \in A(f,m)$ holds $(r = 1,.....a)$

Step 3. *Choose a best $j = f \oplus m$ in V^* (with respect to E or to the function $\tilde{E}$) and set $f = j$.*

Step 4. *If $E(f) < E(f^*)$ then set $f^* = f$.*

Step 5. *Update Tabu ad aspiration conditions.*

Step 6. *If a stopping condition is met then stop. Else go to Step2.*

Figure 4: Tabu Search Algorithm.

Results

The proposed system has been implemented using Matlab 7.11.0 (2010b) running on an HP machine with a dual core processor.
For the searching step (Tabu search algorithm), the system is implemented to terminate after a maximum of 1000 rounds or an error less than 0.15 (good match). The algorithm's average time to find the matched pose is about 1.4 second (and varies from 0.2 to 4 seconds / frame), with an error less than the threshold value (of order 0.07). This means that the proposed method can on the average process 1 frame every second, which is good with the Matlab framework, and this time can be reduced if implemented directly in C++.

Conclusions

This paper developed a 3D hand pose detection system that uses search techniques to find the pose that perfectly matches a hand in a given image. The proposed system consists of two stages, first the preparation stage, where the input image is prepared for the searching and matching process. The second stage is the searching process itself that maximizes the match between the input and one of the stored poses. The searching algorithm used in the optimization process is Tabu Search. The system performance can be improved by implementing it using one of the compiler based programming languages (Matlab uses interpreter during run time). One of the improvements that have been tried is using Matlab Parallel Processing toolbox for running the algorithm faster, but the Matlab Parallel Processing toolbox uses client server model in the parallel processing which adds more overhead on the original calculations. As a suggestion for future work, Task Parallel Library (TPL) in .NET framework, or OpenCL on the GPU can be used to improve the performance with new multicores computers.

References

[1] Stenger B., Thayananthan A., Torr P.H.S., Cipolla R., Model-based hand tracking using a hierarchical Bayesian filter, Pattern Analysis and Machine Intelligence, IEEE Transactions, Sep. 2006.
[2] B. Stenger. Model-Based Hand Tracking Using A Hierarchical Bayesian Filter. PhD thesis, University of Cambridge.
[3] S. Russel, P. Norvig, Artificial Intelligent: A Modern Approach, Prentice Hall; 3 edition, ISBN-10: 0136042597, Dec. 2009.
[4] Sadiq M. , Habib Youssef , Iterative Computer Algorithms with Applications in Engineering solving combinatorial optimization problems, IEEE Computer Society , ISBN 0-7695-0100-1, 1999.
[5] www.mathworks.com