A Flexible Gesture-Based User Interface for Information Displays

Kam Lai, Janusz Konrad, and Prakash Ishwar
Department of Electrical and Computer Engineering, Boston University
8 Saint Mary's St., Boston, MA 02215, USA
{laikamho,jkonrad,pi}@bu.edu

Robust - independent of user height, body shape, clothing
Flexible - easily adaptable to different scenarios
Simple - only 3 parameters to adjust

1. Introduction

We propose a real-time demonstration of the use of hand gestures to control information displays found in shopping malls, airports, bus/railway stations, universities, and command and control centers through a Microsoft Kinect camera. The underlying methodology for real-time gesture recognition that we have developed is flexible and can be easily adapted to control the new Metro interface of Xbox 360 or the upcoming Windows 8 computers.

2. Methodology

The underlying methodology for gesture recognition in our demonstration is the empirical covariance framework originally developed by Tuzel, Porikli and Meer [4] for object tracking and face detection, and later extended to action recognition by Guo, Ishwar and Konrad [2]. Recently, this framework was successfully applied to gesture recognition using the Kinect [3] and we adopt this method in the classification engine of our demonstration.

We use the Microsoft Kinect SDK to obtain the 3D coordinates of 20 skeleton joints in real time (30 fps) for a total of 1,800 floating-point numbers per second. In this demonstration, we focus on gestures with dominant arm and hand movements that mainly impact the following 4 joints: left elbow (le), left hand (lh), right elbow (re) and right hand (rh).[1] We note, however, that this can be easily extended to include more joints should a specific application require.

Features: We use the 3-D coordinates of each of the 4 joints with the spine joint (s) as the origin and further normalize it by the neck-to-spine distance d_{ns} in order to make it invariant to a person's absolute height. For example, for the left elbow's normalized X coordinate we use $d_{le}^X = (X_{le} - X_s)/d_{ns}$. The normalized X, Y, Z coordinates for each joint together with the capture-time t_n form a 13-dimensional feature vector:

[1] We have experimented with 6 joints by adding the left and right wrist joints; but thorough cross-validation tests did not reveal significant performance gains.

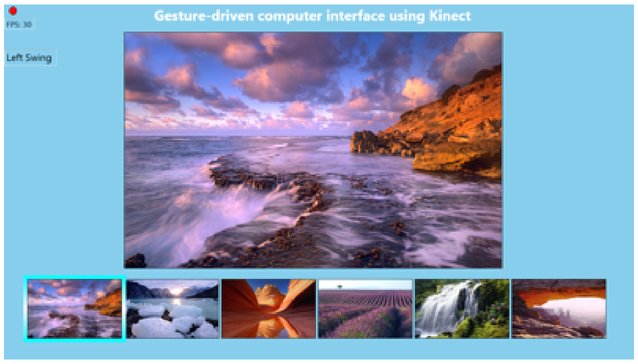

Figure 1. Screen-shot of our gesture-driven computer interface.

$$\boldsymbol{f}_n = [d_{le}^X, d_{le}^Y, d_{le}^Z, d_{lh}^X, d_{lh}^Y, d_{lh}^Z, d_{re}^X, d_{re}^Y, d_{re}^Z, d_{rh}^X, d_{rh}^Y, d_{rh}^Z, t_n]$$

Typically, we use $N = 30$ vectors $\boldsymbol{f}_n$ to capture a gesture that lasts for up to 1 second.

Representation: We capture the dynamics of a gesture (joint trajectories) by computing an empirical covariance matrix from N feature vectors: $C = 1/N \sum_{n=1}^{N} (\boldsymbol{f}_n - \boldsymbol{\mu})(\boldsymbol{f}_n - \boldsymbol{\mu})^T$, where $\boldsymbol{\mu}$ is the empirical mean of $\boldsymbol{f}$. Since the set of all covariance matrices of a given size does not form a vector space, common machine learning methods do not perform well. Therefore, we map the convex cone of covariance matrices to the vector space of symmetric matrices by using the matrix logarithm [1].

Classification: We use the nearest-neighbor (NN) classifier for simplicity. For a query hand gesture with covariance matrix $\mathcal{Q}$ and a hand-gesture dictionary of covariance matrices $\{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K\}$, NN classification is described by:

$$\text{label}(\mathcal{Q}) = \text{label}(\mathcal{C}_m), \quad m = \arg \min_{i=1,...,K} \rho(\mathcal{Q}, \mathcal{C}_i), \quad (1)$$

where $\rho(\mathcal{Q}, \mathcal{C}) = || \log(\mathcal{Q}) - \log(\mathcal{C}) ||_2$ is the Euclidean distance in log-covariance space.

3. Implementation

We have implemented our gesture recognition demo in C# using Visual Studio 2010 and the following libraries: Math.NET, InputSimulator, and Microsoft Kinect SDK. The demo easily runs in real time on any recent-generation laptop or desktop (2GHz+, 2GB+) with no need for multi-core processing.

3.1. Practical Considerations

For our demonstration, we will consider only single gestures and not a sequence of consecutive gestures. Furthermore, we will constrain the set of permissible single gestures to those that start with a significant hand movement, remain for a time period of $T = N \times 33$ ms (33 ms is the temporal sampling period of Kinect), and are followed by T seconds of "dead" time. We deem a hand movement significant when the average displacement of the 4 joints over 5 consecutive frames exceeds a threshold θ. To accurately capture a complete gesture, we use the joints from the triggering frame, the preceding 4 frames and the subsequent $N-5$ frames to build a bag of N feature vectors $\boldsymbol{f}_n$. The "dead" time prevents the algorithm from being spuriously triggered by a premature subsequent gesture (a small red/green disk on the screen provides feedback to the user: red = "wait", green = "go ahead").

3.2. System Functionality

Currently, our system allows recognition of up to 8 hand gestures in two modes: *fixed* and *programmable*. In the *fixed mode*, no gesture learning is needed as we have trained the system off-line on 20 individuals performing 5 times each of the 8 following gestures:

- right-forearm swing to left, left-forearm swing to right,
- right-forearm push and left-forearm push,
- right-forearm swing back and left-forearm swing back,
- spread out both arms horizontally,
- bring both arms together diagonally.

Each gesture is associated with a specific on-screen function, e.g., right-forearm swing to left changes focus to the right neighbor of the current icon, whereas a forearm push activates the current icon.

In the *fixed* mode, a gesture is recognized by first computing the logarithm of its empirical feature covariance matrix $\mathcal{Q}$ and then classifying it with the NN-classifier (1) using a fixed pre-computed dictionary of $K = 8 \times 5 \times 20 = 800$ log-covariance matrices. No training is needed but users have to use the 8 predefined gestures described above.

In the *programmable* mode, a user first trains the system by performing an arbitrary gesture of his/her choice and associating it with an on-screen function. During training, a covariance matrix is computed from the 4 skeleton joints and its logarithm is stored as a dictionary element. After training, the user performs gestures that are processed by the NN classifier (1). We are currently experimenting with an automatic dictionary adaptation. The main idea is to expand the dictionary (up to some maximum size) each time a gesture is recognized with high accuracy, i.e., when $\rho(Q, C) < \delta$, where δ is a small threshold. In the *programmable* mode, a user can "craft" the gesture to his/her own liking and needs, but other users may have difficulty following this specific gait.

3.3. User-Tunable Parameters

Our system has just 3 tunable parameters:

- θ: The threshold for the average displacement of 4 joints over 5 frames that adjusts the sensitivity of the system to the initial gesture speed.
- N or $T = N \times 33$ms: The maximum length of a gesture and also of the "dead" interval when the system waits. A smaller N can be useful for users who perform gestures more rapidly.
- δ: The gesture-acceptance threshold for dictionary expansion. A small δ assures that only gestures that are very similar to those in the dictionary (in terms of log-covariance) will be added. This parameter is optional; it is not needed if the user performs the same gesture several times during training.

4. Benefits and Extensions

The robustness and flexibility of our gesture-based computer interface stem from the employed feature-covariance framework; new gestures can be learned with no algorithm adjustment. The simple NN classifier works quite well but we hope to improve on it by employing kNN classification before mid-June. Although simple, short gestures are sufficient in the current demonstration, more complex, longer gestures may be needed if a richer set of on-screen functions are needed. Longer gestures can also serve as a unique biometric in other applications that we are currently pursuing and hope to demonstrate in the near future.

References

[1] V. Arsigny, P. Pennec, and X. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56(2):411–421, 2006. 1

[2] K. Guo, P. Ishwar, and J. Konrad. Action recognition in video by sparse representation on covariance manifolds of silhouette

tunnels. In *Proc. Int. Conf. Pattern Recognition (Semantic Description of Human Activities Contest)*, Aug. 2010. 1

[3] K. Lai, J. Konrad, and P. Ishwar. A gesture-driven computer interface using Kinect camera. In *Proc. Southwest Symposium on Image Analysis and Interpretation*, Apr. 2012. 1

[4] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. European Conf. Computer Vision*, May 2006. 1