BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

GESTURE PASSWORDS:

CONCEPTS, METHODS, AND CHALLENGES

by

JONATHAN WU

B.S./M.S, Carnegie Mellon University, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2016

Approved by

First Reader

Prakash Ishwar, PhD
Associate Professor of Electrical and Computer Engineering
Associate Professor of Systems Engineering

Second Reader

Janusz Konrad, PhD
Professor of Electrical and Computer Engineering

Third Reader

Vivek Goyal, PhD
Associate Professor of Electrical and Computer Engineering

Fourth Reader

Margrit Betke, PhD
Professor of Computer Science

It marks a big step in your development when you come to realize that other people can help you do a better job than you could do alone.

Andrew Carnegie

Acknowledgments

I would first like to thank both my research advisors, Prof. Prakash Ishwar, and Prof. Janusz Konrad for their support and guidance over the last 5 years. I was fortunate to receive their mentorship and advice, which ultimately helped sculpt and refine my methodology and perspective towards research. I truly appreciate the opportunities they've given me, as well as their encouragement over the years to pursue excellence in research. I would like to thank them for their patience and willingness to spend the extra time and effort to encourage and aid me throughout my doctoral studies. Truly, without their support, none of this would have been possible. I also would like to thank my committee members, Prof. Vivek Goyal, and Prof. Margrit Betke for contributing time from their schedules and giving feedback and suggestions. Additionally, I would like to thank Prof. Stan Sclaroff for letting us borrow his GTX Titan Z, as well as the Nvidia Hardware Grant for providing us with a GTX Titan X.

From the Information Data Science (IDS) group, I would also like to acknowledge some current and former members: Weicong Ding, Tolga Bolukbasi, Wenbo He, Amanda Gaudreau, Ji Dai, Feng Nan, Jiawei Chen, and Andrew Cutler. I would also like to make note of my high school and undergraduate collaborators: Luke Sorenson, Lucas Liang, James Christianson, Jonathan Kim, and Kristi Richter. Finally, I would like to thank my family for providing and supporting me all my life.

This work was not possible without the support of the National Science Foundation (NSF) under award CNS-1228869. The views in this document are those of the author's and does not represent the NSF.

GESTURE PASSWORDS:

CONCEPTS, METHODS, AND CHALLENGES

JONATHAN WU

Boston University, College of Engineering, 2016

Major Professors: Prakash Ishwar, PhD
Associate Professor of Electrical and Computer Engineering
Associate Professor of Systems Engineering

Janusz Konrad, PhD
Professor of Electrical and Computer Engineering

ABSTRACT

Biometrics are a convenient alternative to traditional forms of access control such as passwords and pass-cards since they rely solely on user-specific traits. Unlike alphanumeric passwords, biometrics cannot be given or told to another person, and unlike pass-cards, are always "on-hand." Perhaps the most well-known biometrics with these properties are: face, speech, iris, and gait. This dissertation proposes a new biometric modality: gestures.

A gesture is a short body motion that contains static anatomical information and changing behavioral (dynamic) information. This work considers both full-body gestures such as a large wave of the arms, and hand gestures such as a subtle curl of the fingers and palm. For access control, a specific gesture can be selected as a "password" and used for identification and authentication of a user. If this particular motion were somehow compromised, a user could readily select a new

motion as a "password," effectively changing and renewing the behavioral aspect of the biometric.

This thesis describes a novel framework for acquiring, representing, and evaluating gesture passwords for the purpose of general access control. The framework uses depth sensors, such as the Kinect, to record gesture information from which depth maps or pose features are estimated. First, various distance measures, such as the log-euclidean distance between feature covariance matrices and distances based on feature sequence alignment via dynamic time warping, are used to compare two gestures, and train a classifier to either authenticate or identify a user. In authentication, this framework yields an equal error rate on the order of 1-2% for body and hand gestures in non-adversarial scenarios. Next, through a novel decomposition of gestures into posture, build, and dynamic components, the relative importance of each component is studied. The dynamic portion of a gesture is shown to have the largest impact on biometric performance with its removal causing a significant increase in error. In addition, the effects of two types of threats are investigated: one due to self-induced degradations (personal effects and the passage of time) and the other due to spoof attacks. For body gestures, both spoof attacks (with only the dynamic component) and self-induced degradations increase the equal error rate as expected. Further, the benefits of adding additional sensor viewpoints to this modality are empirically evaluated. Finally, a novel framework that leverages deep convolutional neural networks for learning a user-specific "style" representation from a set of known gestures is proposed and compared to a similar representation for gesture recognition. This deep convolutional neural network yields significantly improved performance over prior methods.

A byproduct of this work is the creation and release of multiple publicly available, user-centric (as opposed to gesture-centric) datasets based on both body and

hand gestures.

Contents

List of Tables

List of Figures

List of Abbreviations

ASL	.....	American Sign Language
BLD	.....	BodyLogin Dataset
BLD-M	.....	Body Login Dataset: Multiview
BLD-PBD	.....	Body Login Dataset: Posture, Build, Dynamics
BLD-SS	.....	Body Login Dataset: Silhouettes vs Skeletons
CCE	.....	Correct Classification Error
CCR	.....	Correct Classification Rate
CNN	.....	Convolutional Neural Network
CRF	.....	Conditional Random Field
DCT	.....	Discrete Cosine Transform
DTW	.....	Dynamic Time Warping
EER	.....	Equal Error Rate
FAR	.....	False Acceptance Rate
FRR	.....	False Rejection Rate
GEI	.....	Gait Energy Image
GMM	.....	Gaussian Mixture Model
HOG	.....	Histogram of Oriented Gradients
HMM	.....	Hidden Markov Model
LOP	.....	Local Occupancy Patterns
LOOCV	.....	Leave-One-Out Cross Validation
LDS	.....	Linear Dynamical System
MKL	.....	Multiple Kernel Learning
NN	.....	Nearest Neighbor
PC	.....	Personal Computer
RGB	.....	Red Green Blue Color Model
ROC	.....	Receiver Operating Characteristic
SDK	.....	Software Development Kit
SVM	.....	Support Vector Machine
t-SNE	.....	t-Distributed Stochastic Neighbor Embedding

Chapter 1

Introduction

1.1 Motivation

Traditional biometrics have been plagued by the use of inherently nonrenewable information. For instance, having to change or replace a person's face, iris, fingerprint, or speech is inconvenient and difficult. A compromised biometric is not necessarily rare. Faces are open public information and are vulnerable to being photographed, fingerprints are easily left on surfaces, and speech can be recorded and replayed. Thus, a renewable biometric, one that could be easily changed if compromised, would be invaluable.

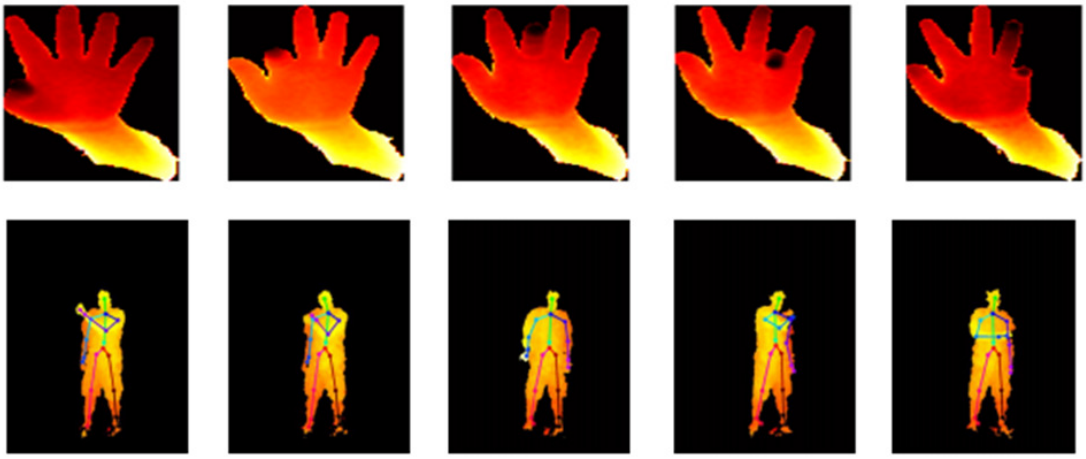


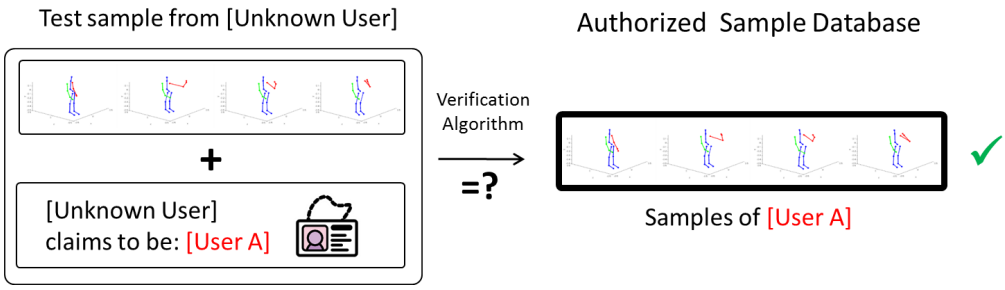
Figure 1·1: Example depth map sequences of a body and a hand gesture as captured by a Kinect camera.

Gesture is a new emerging biometric modality that is partially renewable. A

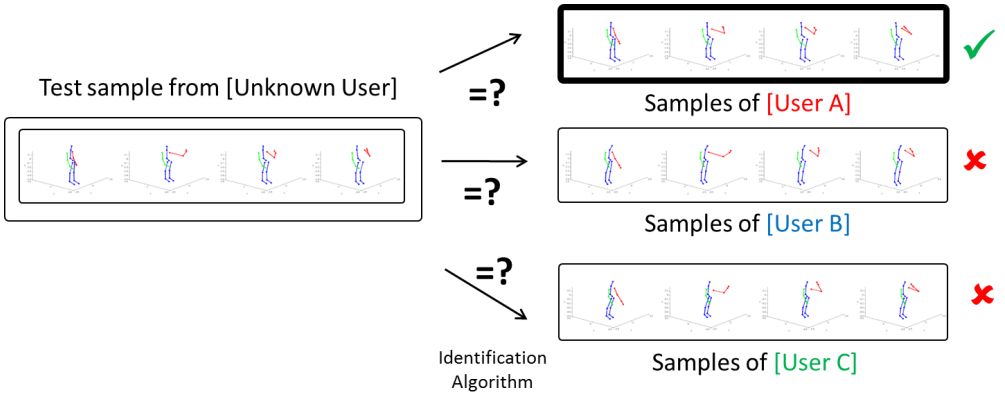
gesture is a short, few seconds long, body motion that contains static anatomical information, and changing (dynamic) information. This thesis considers both full-body gestures, such as a wave of the arms, and hand gestures, such as a subtle curl of the fingers and palm. These gestures are typically only a few seconds long and are performed in front of a stationary camera starting from a resting, neutral position. Should a gesture ever be compromised, a user can *intentionally* select a new gesture.[1] As a gesture consists of both static and dynamic information, the dynamic portion can be altered. These gesture "passwords" can be presented to an access control system to either *identify* who a person is, or to *authenticate* (verify) whomever he/she claims to be (see Figure 1·2). These two primary access scenarios which we consider for evaluating biometric performance, are commonly known as: identification and authentication (verification).

Further, ongoing advances in depth capturing technologies, such as the Kinect v1 and v2 (Kin, 2014), have made acquiring quality biometric information based on body gestures widely accessible. In fact, ubiquitous depth sensor integration is expected in next-generation devices (smartphones, PCs, and tablets). One significant advantage of a depth sensor is its resistance to spoofing and evasion since 3-D information is required from its users. For example, unlike in face recognition, a photograph for spoofing would no longer work (due to its flat surface) and instead a 3-D molded mask would be required (much to the inconvenience of would-be attackers). This combination of renewability and inherent spoofing resistance has motivated us to carry out research in gesture passwords.

[1] In contrast to gait which is only unintentional user motion.



Authentication access scenario.
User provides gesture password (shown as a skeletal joint sequence) and purported identity.



Identification access scenario.
User only provides gesture password (shown as a skeletal joint sequence).

Figure 1·2: Two of the most common access control scenarios we consider.

1.2 Contributions

At the highest level, this dissertation introduces and validates a new modality, intentional body gestures, for the purpose of authenticating or identifying a person. Since this problem has not been tackled before, it necessitated both the generation of new datasets and a systematic quantitative study of the security performance of the proposed algorithms.

At a finer level, this dissertation makes the following contributions:

Contribution 1 - Gesture Representations: We introduce a novel framework for acquiring, representing and evaluating gesture passwords for either authentication or identification. This framework leverages information obtained by an RGB/depth-sensing camera, such as the Kinect, from which silhouette or skeletal features are extracted. We propose two distance metrics for comparing gestures: the log- euclidean distance between feature covariance matrices and distance based on feature sequence alignment via dynamic time warping. This contribution has been reported in the following works: (Wu et al., 2013; Wu et al., 2014a; Wu et al., 2015).

Contribution 2 - Gesture Component Decomposition and Valuation Framework: We propose a novel framework for first decomposing gestures into posture, build, and dynamics and then realistically re-synthesizing them with one or more components suppressed. This framework enables a fair evaluation of the contributions of different gesture components to authentication and identification performance. This contribution has been reported in the following work: (Wu et al., 2014b).

Contribution 3 - Gesture Degradations and Threats: Models and Performance Evaluation Framework: We study two major classes of threats to gesture-based authentication and identification with the help of real-world test subjects. The first class, intrinsic threats, contains self-induced degradations to the gesture password. This can be due to personal effects (outerwear or belongings) or due to a user's inability to accurately reproduce a gesture after a long period of time. The second class, extrinsic threats, focuses on spoof attacks. This contribution has been reported in the following works: (Wu et al., 2014a; Wu et al., 2014c).

Contribution 4 - User-centric Datasets: We create four new real-world datasets for the express purpose of evaluating the identification and authentication performance of body and hand gestures. Unlike datasets for gesture recognition that are gesture-centric and contain a high number of gestures per user, our datasets uniquely focus on being user-centric, and all contain a high number of users per gesture. In addition to this trait, we collect gesture samples under a multitude of conditions such as: personal effects, the passage of time, inclusion of copycats, and the usage of multiple camera sensors. This contribution has been reported in the following works: (Wu et al., 2014a; Wu et al., 2014b; Wu et al., 2014c; Wu et al., 2015).

Contribution 5 - User Gesture Style for Authentication and Identification: We develop a novel framework for authentication and identification based on a user's gesture style which is a set of common traits to gestures by the same user. This framework is based on deep convolutional neural networks, specifically, a two-stream convolutional network which uses both the spatial and the temporal information in a gesture. This contribution has been reported in the following work: (Wu et al., 2016).

1.3 Layout of Thesis

The following is the outline of the rest of the thesis.

Chapter 2 provides a brief overview and background of topics related to the gesture modality.

Chapter 3 introduces and defines the proposed user recognition framework and the metrics that are used to evaluate it.

Chapter 4 discusses the datasets that were collected to support the experimental evaluations in this work.

Chapter 5 focuses on studies dealing with *body* gestures. These studies pertain to the importance of dynamics, the robustness of the modality towards threats and degradations, and the value of additional viewpoints.

Chapter 6 evaluates the biometric performance of *hand* gestures.

Chapter 7 explores learning user-specific gesture "style" using deep convolutional networks.

Chapter 8 summarizes the conclusions and contributions of this dissertation and outlines possible directions for future work.

Chapter 2

Background

In this chapter, works are reviewed that are similar in nature to gesture-based authentication. As there is fundamentally no prior work in this biometric modality, instead, techniques that can be adapted or reapplied for gesture access control are described. The following is a short overview of these topics.

2.1 Gesture Recognition

Gesture authentication is perhaps most similar to gesture (action) recognition. In both problems, users perform a gesture with *intent* in front of a sensor. In authentication, the goal is to find or authenticate the user (analyze information specific to a user), and in recognition, the goal is to find the gesture (analyze information specific to a gesture). For example, in gesture recognition, information related to the angular velocities of the joints holds much more importance than information pertaining to a user's body build and shape. This is because the angular velocities of the joints are more *gesture* specific, than *user* specific.

First, we discuss methods for gesture recognition based on depth sensors. One particular advantage of the Kinect is that skeletal joint information can be estimated directly from the depth maps through pose estimation (Shotton et al., 2011; Shotton et al., 2013). As a result of this, many features have been proposed for recognition based on skeletal joints.

Xia et al. (Xia et al., 2012) proposed binning skeletal joints into 3-D spherical

coordinate bins, which could be used as a histogram feature. Wang et al. (Wang et al., 2012b) proposed using local occupancy patterns (LOP) as features which are computed by binning point-cloud values around calibrated skeletal joints. Ohn-Bar and Trivedi (Ohn-Bar and Trivedi, 2013) proposed using histogram of oriented gradients (HOG) around each skeletal joint and pairwise affinities between skeletal joint angles as features. Ofli et al. (Ofli et al., 2013) proposed using linear dynamical systems (LDSs) to model 3-D joints at several spatio-temporal scales on skeletal joints.

To compare or classify these features, methods such as dynamic time warping (DTW) (Reyes et al., 2011), hidden Markov models (HMMs) (Lv and Nevatia, 2006), conditional random fields (CRFs) (Han et al., 2010), and multiple kernel learning (MKL (Ofli et al., 2013), have been applied.

2.2 Biometrics

There are two categories of biometrics: physiological and behavioral.

Physiological biometrics are based on a person's physical traits. These are the well-known, "traditional" biometrics such as face, fingerprint, and iris.

Behavioral biometrics are based on a person's habits (their trends, patterns and "style"). These are signatures, keystrokes, or gait (walking) of an individual. Behavioral biometrics are quite similar to gestures and we discuss them in some detail below.

Gait

Gait is the unique shape and motion of an individual walking that can be used for identification. Using gait as a biometric has in recent years gained traction due to its properties being recognizable from a distance. This long-range recognition is

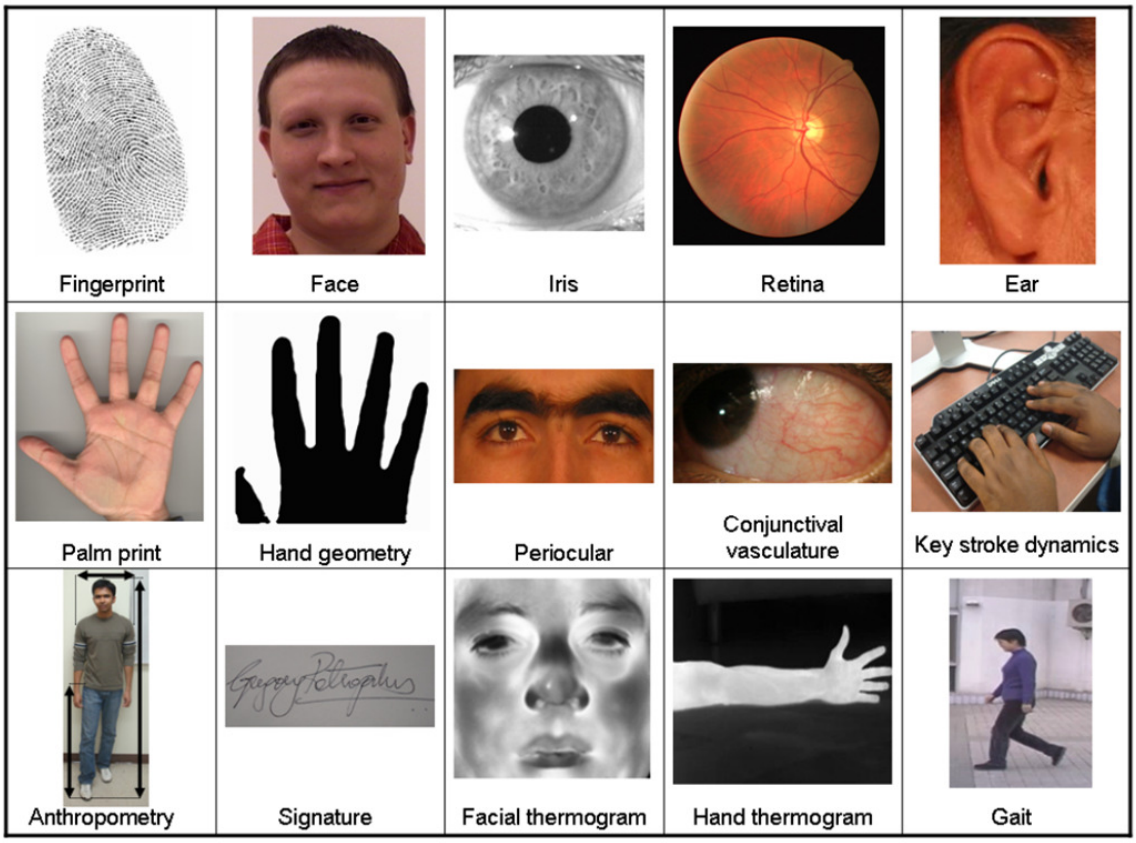


Figure 2·1: A listing of a few common biometrics (both physical and behavioral). Figure from (Jain et al., 2011).

desirable as it does not require the subject to be cooperative (or aware) of the identification. The most common approach for gait recognition is to perform silhouette extraction (through background subtraction), and extract various features that can be used for classification (such as through HMMs, and GMMs). Features such as optical flow (Little and Boyd, 1998), GEI (Han and Bhanu, 2006) (gait energy image – the averaged silhouette intensity), and biped models (Zhang et al., 2004) (coarse pose estimation of the lower body) are among the most popular.

On-line Signature

On-line signatures expand upon the classic pen and paper signature for authentication. Instead of only considering the static image of a signature, dynamic information such as pen x-y position, x-y speed, and tip pressure (how hard a hand is pressing) can be captured when a signature is performed on a touchscreen or tablet. There are primarily two classes for on-line signature features. The first class consists of "feature-based" scalars of global properties such as: the maximum and minimum x-y pen velocity, or the standard deviation of the pen's x-y acceleration. The second class consists of "function-based" vectors of time-dependent sequences: pen trajectory, velocity, acceleration, force, or pressure (Lei and Govindaraju, 2005). "Feature-based" information can be compared using distances such as weighted Euclidean distance and Mahalanobis distance. Elastic matching (predominantly, variants of DTW (Faundez-Zanuy, 2007; Kholmatov and Yanikoglu, 2005)) and statistical modeling (HMMs) have also been used to match "function-based" information.

Chapter 3

Framework

This chapter presents a framework for access control through authentication or identification. First, we discuss types of access control and their performance metrics. Next, we detail the framework used to represent, and compare gesture sequences that have been captured from the Kinect. We use two features for gestures: silhouette shapes and skeletal joints. After defining these features, we propose various distances to compare pairs of gesture samples. Finally, we use an authorized set of samples to train a classifier to determine access.

3.1 Evaluating Access Control

The performance of an access control system can be evaluated in one of two scenarios: *authentication* or *identification* (Jain et al., 2011). This section describes each of these access scenarios and their associated performance metrics.

3.1.1 Recognition Algorithm

A simple yet powerful recognition algorithm such as nearest-neighbor (1-NN) can be used to determine access based on suitable distance measures. If a query sample is sufficiently close in distance (falls within a determined cutoff threshold) to an enrolled sample, access or identity can be granted. Based on this algorithm, measures of security can be calculated for a given access scenario.

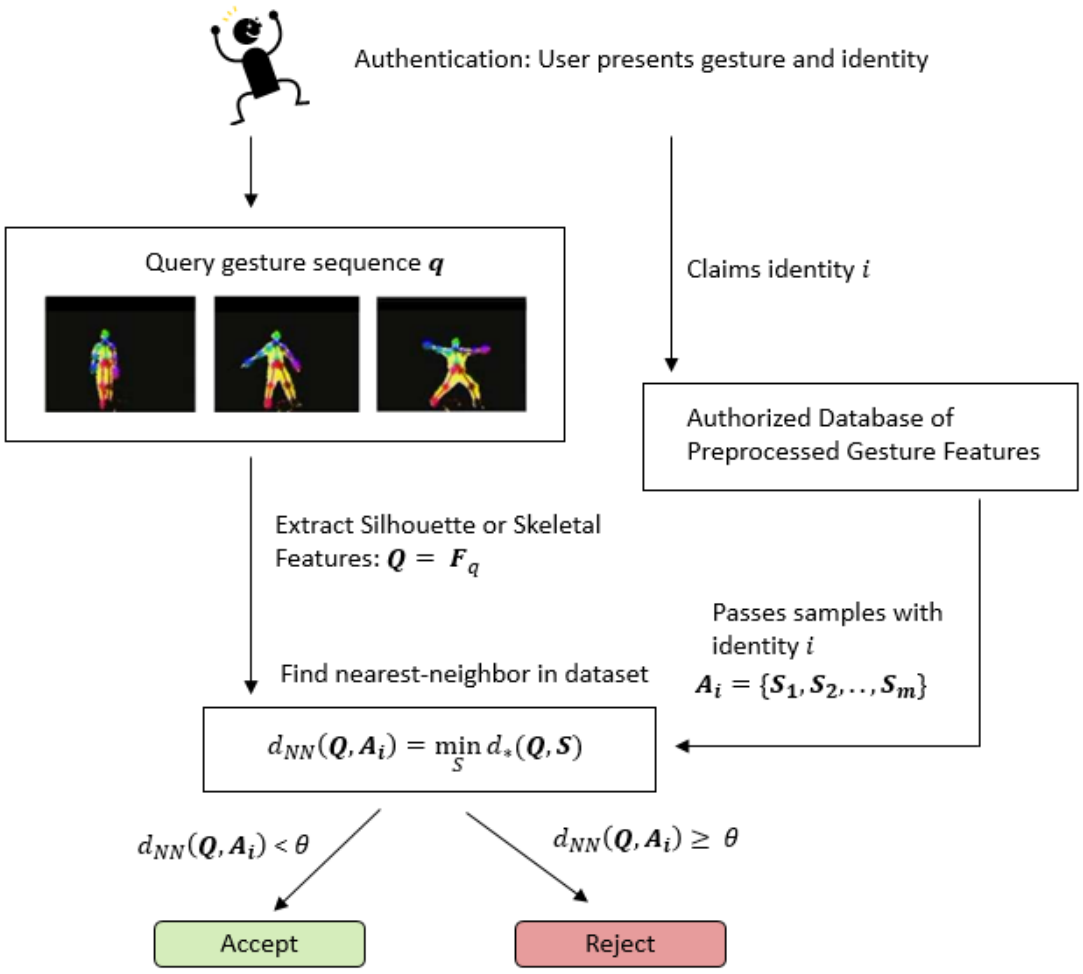


Figure 3·1: System diagram of a user performing authentication.

3.1.2 Authentication

This thesis primarily considers access control performance for *authentication*. In authentication, a user provides two pieces of information: his/her claimed identity and a biometric (see Figure 3·1). If the biometric closely matches an enrolled sample of the given identity (one-to-one-user match), the user is allowed access. Otherwise, he/she is rejected.

Two kinds of errors are considered in this case: false acceptance and false re-

jection. The false acceptance rate (FAR) is the rate at which *unauthorized* users are allowed access. The false rejection rate (FRR) is the rate at which *authorized* users are denied access. In any practical system, FAR and FRR will have trade-offs. One can find these trade-offs by applying various acceptance thresholds across the system. A common metric of performance is the equal error rate (EER) which occurs when FAR and FRR are equal. This process is briefly recapped below.

Let $\mathcal{A}_i = \{\mathbf{S}_1, \ldots \mathbf{S}_m\}$ be a set containing m gesture samples from a single authorized user i. Let $\mathcal{U}_i$ be a set of gesture samples that do not come from authorized user i. The FRR is found by comparing samples in $\mathcal{A}_i$ amongst themselves (each sample in $\mathcal{A}_i$ is treated as a query sample $\mathbf{Q}$). This is done using leave-one-out cross validation (LOOCV) such that each sample is compared to the set $\mathcal{A}_i \backslash \{\mathbf{Q}\}$, i.e., with the query itself removed. The FAR is found by comparing samples in $\mathcal{U}_i$ to samples in $\mathcal{A}_i$ (each sample in $\mathcal{U}_i$ is treated as a query sample). A nearest-neighbor criterion $d_{NN}(\cdot, \cdot)$ is used to compare a single query sample $\mathbf{Q}$ to the authorized set $\mathcal{A}$.

$$d_{NN}(\mathbf{Q}, \mathcal{A}_i) \quad = \min_{\mathbf{S} \in \mathcal{A}_i} d_*(\mathbf{Q}, \mathbf{S}).$$

where d_* is one of the following distances (discussed in Section 3.3):

- $d_{U.tri.Eucl}$ (Upper Euclidean)
- $d_{Hier.Eucl}$ (Temporal Hierarchy Euclidean Distance)
- $d_{Log-Eucl}$ (Log Euclidean Distance)
- d_{DTW} (Dynamic Time Warping)

Then, for a given threshold value θ, the FAR and FRR are calculated by:

$$FRR(\mathcal{A}_i, \theta) = \frac{\sum_{\mathbf{Q} \in \mathcal{A}_i} \mathbf{1}(d_{NN}(\mathbf{Q}, \mathcal{A}_i \backslash \{\mathbf{Q}\}) \geq \theta)}{|\mathcal{A}_i|}$$
$$FAR(\mathcal{A}_i, \mathcal{U}_i, \theta) = \frac{\sum_{\mathbf{Q} \in \mathcal{U}_i} \mathbf{1}(d_{NN}(\mathbf{Q}, \mathcal{A}_i) < \theta)}{|\mathcal{U}_i|}$$

where the indicator function $\mathbf{1}(condition)$ equals 1 if the '$condition$' is true and equals 0 otherwise.

EER for the pair $(\mathcal{A}_i, \mathcal{U}_i)$ is found by first computing the FAR-FRR pairs for different thresholds θ. Then, the EER is determined as the location on the boundary of the convex hull of the FAR-FRR pairs where FAR equals FRR. In practice, this EER does not lie directly on a FAR-FRR pair that corresponds to a single classifier (e.g., a 1-NN decision rule with a single decision threshold). Rather, the EER reflects two classifiers (e.g., two 1-NN decision rules with two different decision thresholds), where each classifier is chosen at random with a fixed probability (Scott et al., 1998).

This computation is repeated for each authorized user who each has his/her own unique set $(\mathcal{A}_i, \mathcal{U}_i)$. Effectively, each user in a dataset will have their own EER. If these values are averaged across users, an average EER can be computed. Such a score can be considered to represent the scenario where each user has their own accept/reject threshold θ_i. If a global accept/reject threshold θ_{global} is desired across *all* users, a "global" EER can be computed. In this case, θ is taken across all $(\mathcal{A}_i, \mathcal{U}_i)$ pairs simultaneously.

3.1.3 Identification

Access control can also be considered in the context of *identification*. In *identification*, a user presents his/her biometric sample to a system which retrieves an enrolled identity through a one-to-many-user match (see Figure 3·2). This is called the closed-set identification problem, i.e., classification under the assumption that

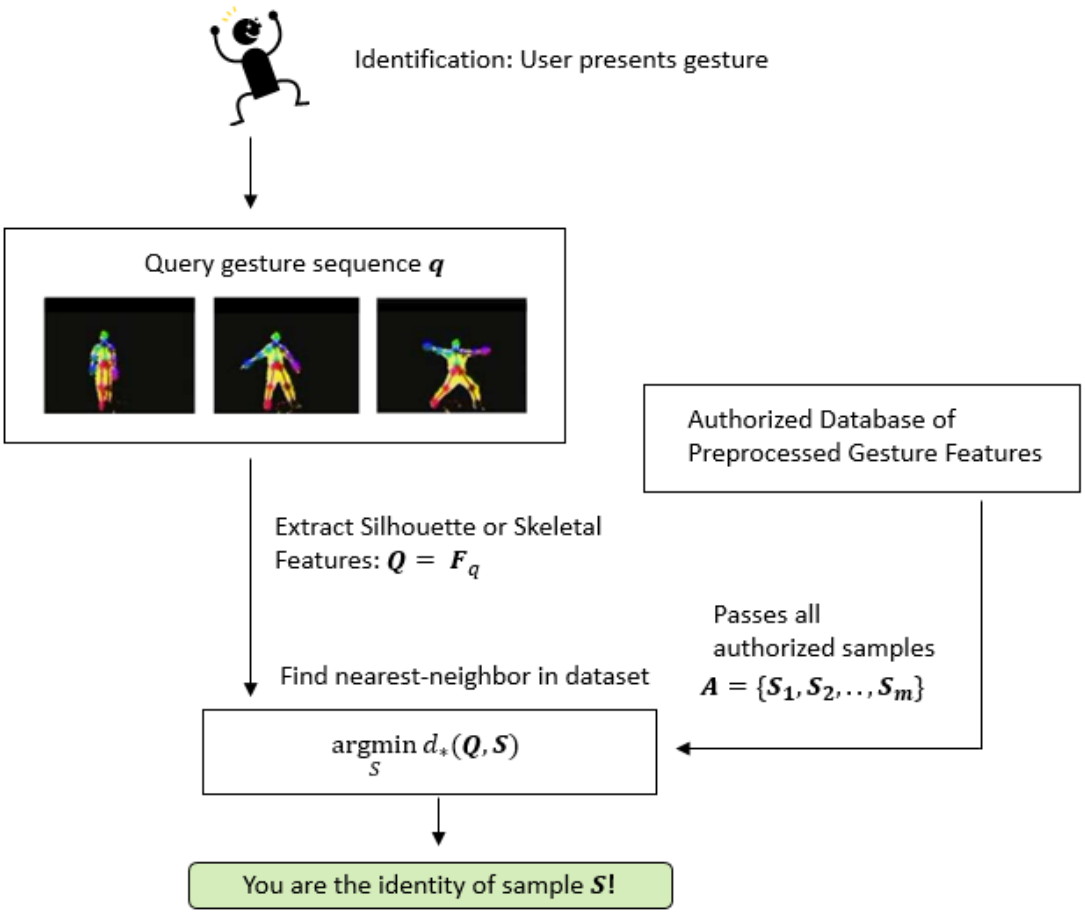


Figure 3·2: System diagram of a user performing identification.

the query user's identity is enrolled, i.e., no possibility of rejection. The correct classification rate (CCR), and its complement error, CCE = 1-CCR, can be used to express accuracy. This value is also computed with LOOCV, where each user is labeled with the identity of his/her nearest-neighbor match.

3.2 Representation

The following sections describe how silhouette or skeletal feature matrices $\mathbf{F}$ are computed. These features are used to represent each gesture sample, and will later

be used to compute distances between pairs of samples.

3.2.1 Silhouette Features

A compact silhouette representation can be extracted from any gesture sequence consisting of depth frames.

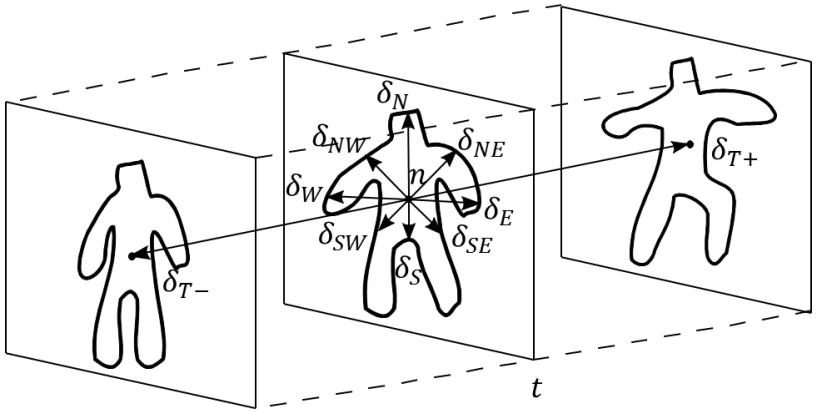


Figure 3·3: Each silhouette pixel n has an associated 13-dimensional feature vector which consists of the pixel position (x, y, t) and 10 directional distances δ_* to the silhouette boundary.

First, a sequence of binary silhouettes of a user performing a gesture is extracted from a sequence of depth frames. This is accomplished by differencing each depth map from a known depth map background. The largest 3-D connected component with 18-connected pixel connectivity in the 3-D sequence of binary silhouettes is taken to be a user's silhouette tunnel.

In order to extract features from this silhouette tunnel, an approach similar to the one used by Lai and Guo et al. (Lai et al., 2012; Guo et al., 2013) is adopted. Let $n = 1, ..., N$ index all N pixels within the silhouette tunnel and let (x, y, t) denote the space-time coordinates of pixel number n. The following 13-dimensional feature vector $\mathbf{f}^n$ is computed at each silhouette pixel and captures the shape and dynamic

characteristics of a gesture (Figure 3·3):

$$\mathbf{f}^n \quad = \quad \mathbf{f}(x, y, t) \quad := \quad [x, y, t, \delta_E, \delta_W, \delta_N, \delta_S, \delta_{NE}, \delta_{SW}, \delta_{SE}, \delta_{NW}, \delta_{T+}, \delta_{T-}]' \quad (3.1)$$

where δ_{DIR} denotes the Chebyshev distance between the space-time coordinates of pixel n and those of the farthest silhouette boundary pixel in direction $_{DIR}$. The first 8 directions are in the x, y spatial domain (4 cardinal directions and 4 inter-cardinal directions), and the last 2 are in the temporal domain (forward and backward in time).

The above procedure, after visiting all N pixels of a silhouette tunnel, produces a 13 by N matrix $\mathbf{F} = [\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^N]$ as the final representation.

3.2.2 Skeletal Features

A gesture can also be described as a sequence of a user's skeletal joints in rectangular coordinates across time. The advantages of using skeletons (pose estimation) over silhouettes, are two fold: (i) skeletal data is sparse yet informative and (ii) skeletal data is relatively insensitive to changes in clothing, personal effects, and lighting conditions. Conveniently, the Kinect SDK (Kin, 2014; Shotton et al., 2011) provides rectangular coordinates of 20 skeletal joints of the human body for each frame at 30 frames per second. These coordinates are extracted from each depth frame and correspond to the following locations: head, neck, spine, center hip, and left and right versions of the hand, wrist, elbow, shoulder, hip, knee, ankle and foot (Figure 3·4). A skeleton's evolution in time can be represented as a sequence of features $\mathbf{f}^t$ as follows:

$$\mathbf{f}^t := [\mathbf{s}_1^t, \mathbf{s}_2^t, ..., \mathbf{s}_{20}^t]' \quad t = 1, ..., T, \quad (3.2)$$

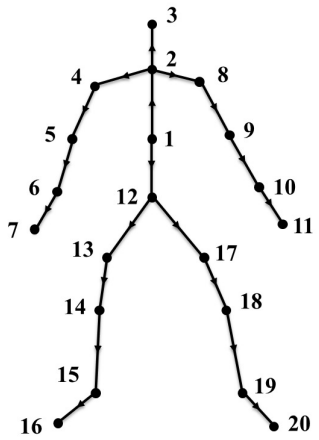


Figure 3·4: Example of a skeleton produced by the Kinect SDK.

where $\mathbf{s}_i^t = (x_i^t, y_i^t, z_i^t) \in \mathbb{R}^3$ is a 1×3 row vector which denotes the $x - y - z$ coordinates of the i-th skeletal joint in frame number t and T is the total number of frames.

Let $\mathbf{F} = [\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^T]$ denote a 60 by T matrix that becomes the final representation of this procedure.

3.2.3 Normalization

Due to the nature of gesture dynamics and body build, individual feature elements (e.g., coordinates $\mathbf{s}_i^t$, and distances along cardinal directions δ_{DIR}) may have significantly different dynamic ranges. A feature element with a large amplitude would then influence an overall error metric more than a feature with a small amplitude. In order to avoid developing a complicated metric with unequal weights for individual elements, the approach of Hussein *et al.* (Hussein et al., 2013) is adopted where the matrix $\mathbf{F}$ is normalized along rows (the time-wise or pixel-wise dimension) as follows:

$$\mathbf{F}_{norm}[i,j] = \frac{\mathbf{F}[i,j] - \min\limits_k \mathbf{F}[i,k]}{\max\limits_k \mathbf{F}[i,k] - \min\limits_k \mathbf{F}[i,k]} \tag{3.3}$$

where $\mathbf{F}[i, j]$ denotes the value in the i-th row and j-th column of $\mathbf{F}$. The above normalization ensures that the values of all feature elements are contained within the dynamic range $[0, 1]$.

3.2.4 Covariance Descriptor

A gesture sequence can be also viewed as a "bag of features," where each pixel (silhouette) or frame (skeleton) is associated with an $h \times 1$ feature vector. An $h \times h$ empirical covariance matrix C of a collection of feature vectors (normalized according to (3.3)) provides a low-dimensional, second-order representation of the entire feature vector collection:

$$C := \frac{1}{N} \sum_{1}^{N} (\mathbf{f}_{norm}^{n} - \boldsymbol{\mu})(\mathbf{f}_{norm}^{n} - \boldsymbol{\mu})^{T}, \tag{3.4}$$

where $\boldsymbol{\mu}$ is the empirical mean of normalized feature vectors $\mathbf{f}_{norm}^{n}$. For silhouettes, $h = 13$, and N is the total number of pixels in the silhouette tunnel. For skeletons, $h = 60$ and N is the number of frames in the skeletal sequence. Since C is a symmetric matrix, its upper-triangular part of size $(h^2 + h)/2$ can be used as an independent gesture descriptor.

3.3 Distance Measures

There are numerous ways to measure distances between two gesture sequences. In the following section, we describe how to compute the distance between pairs of covariance matrices and how to compute dynamic time warping distance. These distances represent the possibilities for d_* as used in Figure 3·1 and 3·2, which are used to compare biometric samples in authentication and identification.

3.3.1 Distances between Covariance Matrices

Upper-Triangular Euclidean Distance

Perhaps the simplest way to measure the distance between two covariance matrices of normalized features is to compute the Euclidean distance (Frobenius norm) between them. Since covariance matrices are symmetric, the upper-triangular portion contains all the information. Hence, one basic distance function between (normalized) covariance matrices that is considered is the Euclidean distance between their upper-triangular parts. This is denoted by $d_{U.tri.Eucl}(\cdot,\cdot)$.

Euclidean Distance with Temporal Hierarchies

A key problem with the covariance descriptor is that the *ordering* of pixels or frames does not matter or is irrelevant. If this order were to be scrambled, the covariance matrix would remain unchanged. In order to emphasize the importance of frame ordering in a gesture, Hussein *et al.* (Hussein et al., 2013) suggested using a hierarchical computation of covariance descriptors across temporal windows at various scales. In this way, given any scrambling of the frames, the covariance matrices across smaller time windows would be different. For example, consider this idea for 3 temporal levels. At level i, 2^{i-1} equal-length, non-overlapping windows are computed across the entire sequence. For example, at the 3^{rd} level in hierarchy there would be 4 equal-length windows each of length $N/4$ (temporal ranges: 1 to $\lfloor \frac{N}{4} \rfloor$, $\lfloor \frac{N}{4} \rfloor + 1$ to $\lfloor \frac{N}{2} \rfloor$, $\lfloor \frac{N}{2} \rfloor + 1$ to $\lfloor \frac{3N}{4} \rfloor$, and $\lfloor \frac{3N}{4} \rfloor + 1$ to N). All these covariance matrices can be computed quickly through the use of *integral signals* (Hussein et al., 2013; Tuzel et al., 2008).

For each covariance matrix that is computed from the temporal hierarchy, the upper triangular portion serves as our gesture descriptor, and all these descriptors are concatenated into one long gesture descriptor vector. For the case of 3

layers, there are 7 covariance matrices. Each upper triangular matrix is of length $(60^2 + 60)/2 = 1,830$, which concatenated together, yields a total length of $7 \times 1,830 = 12,810$. Thus, for a single gesture sequence from one Kinect camera using a 3-layer temporal hierarchy a descriptor of length 12,810 will be generated. A Euclidean norm between these concatenations of upper-triangular parts can be used as a distance. This is denoted by $d_{Hier.Eucl}(\cdot, \cdot)$.

Log-Euclidean Distance

The log-Euclidean distance between two covariance matrices C_1, and C_2 proposed by Arsingy *et al.* (Arsigny et al., 2006) is defined as follows:

$$d_{Log-Eucl}(C_1, C_2) := ||\log(C_1) - \log(C_2)||_2, \tag{3.5}$$

where $||\cdot||_2$ denotes the matrix Frobenius norm and $\log(C) := V\tilde{D}V'$ where $C = VDV'$ is the eigen-decomposition of covariance matrix C and $\tilde{D}$ is obtained from D by replacing its diagonal entries with their logarithms. This distance is a Riemannian metric on the manifold of covariance matrices. The basic intuition of this distance is to convert the space of covariance matrices (forms a convex cone) to a vector space, and then take a norm in the transformed Euclidean space. Additional properties of this distance can be found in (Arsigny et al., 2006).

3.3.2 Dynamic Time Warping

Dynamic time warping (DTW) can be used to measure the distance between two gesture sequences of possibly different durations. DTW is a non-linear alignment algorithm that is relatively popular and has been extensively used in the literature (Ding et al., 2008; Keogh, 2002; Ratanamahatana and Keogh, 2004). For skeletal features, a modified version of this algorithm has been designed for this problem

as detailed below.

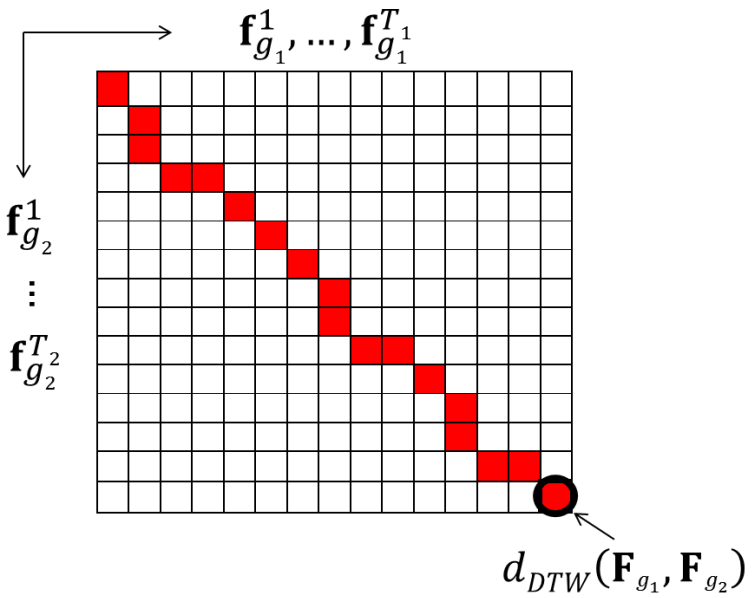


Figure 3·5: Visualization of the path P between two sequences.

Let $\mathbf{F}_{g_1} = [\mathbf{f}_{g_1}^1, \cdots, \mathbf{f}_{g_1}^{T_1}]$ and $\mathbf{F}_{g_2} = [\mathbf{f}_{g_2}^1, \cdots, \mathbf{f}_{g_2}^{T_2}]$ be two feature matrices of skeletal features corresponding to gestures g_1 (T_1 frames long) and g_2 (T_2 frames long). A distance based on the cost of *aligning* $\mathbf{F}_{g_1}$ and $\mathbf{F}_{g_2}$ can be computed from a $T_1 \times T_2$ cost matrix. Let the cost matrix's (i, j)-th entry be the cost of aligning the skeletal feature in frame-i of gesture g_1 with the skeletal feature in frame-j of gesture g_2:

$$\text{cost}(\mathbf{f}_{g_1}^i, \mathbf{f}_{g_2}^j) = \sum_{p=1}^{20} ||\mathbf{s}_{p,g_1}^i - \mathbf{s}_{p,g_2}^j||_2.$$

An admissible alignment scheme is a path $\mathbf{P}$ through the cost matrix defined as follows

$$\mathbf{P} = \{(i_k, j_k), k = 1, \ldots, K : i_1 = j_1 = 1, i_K = T_1,$$
$$j_K = T_2, \forall k, i_{k+1} - i_k, j_{k+1} - j_k \in \{0, 1\}\}$$

where $\max(T_1, T_2) \leq K \leq T_1 + T_2$ is the path-length. The cost of a path is defined

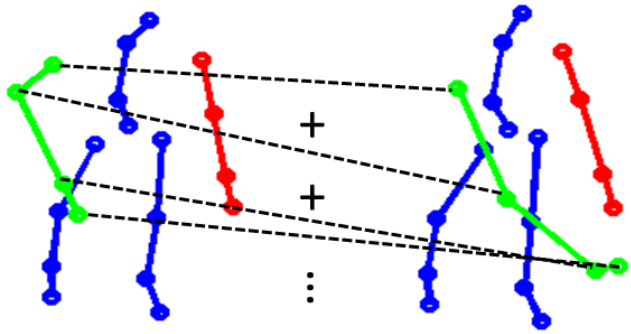


Figure 3·6: Visualization of the modified DTW cost function.

as follows:

$$\text{pathcost}(\mathbf{P}, \mathbf{F}_{g_1}, \mathbf{F}_{g_2}) = \sum_{(i_k, j_k) \in \mathbf{P}} \text{cost}(\mathbf{f}_{g_1}^{i_k}, \mathbf{f}_{g_2}^{j_k})$$

The path of interest is the one with the least cumulative cost. This path can be solved recursively using dynamic programming in quadratic time. The final cost is defined as follows:

$$d_{DTW}(\mathbf{F}_{g_1}, \mathbf{F}_{g_2}) = \min_{\mathbf{P}} \text{pathcost}(\mathbf{P}, \mathbf{F}_{g_1}, \mathbf{F}_{g_2})$$

Chapter 4

Gesture Datasets

4.1 Motivation for New Datasets

The problems of gesture recognition and gesture-based authentication are similar in the sense that they both involve users performing gestures. However, in the former problem the goal is to recognize the gesture regardless of the user, whereas in the latter problem the goal is to recognize the user regardless of the gesture. Although it might seem that a given dataset of gestures can be used interchangeably for studying both problems, e.g., analyzing user authentication performance using a gesture recognition dataset, this is not the case.

Datasets for gesture recognition are typically gesture-centric meaning that they have high number of gestures per user (many gestures to classify, few users performing them) whereas studying authentication requires the opposite, namely a user-centric dataset which has a high number of users per gesture. This issue is highlighted in Table 4.1, where we compare gesture recognition datasets. Notably, many of these datasets contain less than 20 users. In cases where there are more than 20 users, the data has been collected in such a way that there is either not enough users performing each gesture or the data contains dataset bias due to gestures being performed continuously standing in-place. By standing in-place, each user's lower body posture does not significantly change which can cause dataset bias. In gesture recognition, this is typically not an issue, as the same user will

never be seen in *both* training and testing. However, for cases of user recognition, this causes a significant issue, as the same user is almost always seen in *both* training and testing.

With these goals and issues in mind, three datasets were collected:

- BodyLogin Dataset: Silhouettes vs Skeletons (BLD-SS) (Wu et al., 2014a)
- BodyLogin Dataset: Posture, Build, Dynamics (BLD-PBD) (Wu et al., 2014b)
- BodyLogin Dataset: Multiview (BLD-M) (Wu et al., 2014c)
- HandLogin Dataset: In-air Hand Gestures (Wu et al., 2015)

Each of these datasets focuses on a different aspect of gesture-based user recognition, and thus have been recorded under different scenarios as detailed in the following sections.

4.2 Acquisition Methodology

Three of the aforementioned datasets (BodyLogin) have been collected with a Kinect v1 sensor. These three datasets each recorded a subject from a forward-facing Kinect sensor approximately 2 meters away. The Kinect v1 sensor captures a 640×480 depth image and skeletal joint coordinates (pose estimation through the SDK (Kin, 2014)) at 30 fps.

The last dataset, HandLogin, has been collected with a Kinect v2 sensor facing the ceiling. The Kinect v2, a time-of-flight depth sensor, is used to acquire a 512x424 depth image of each gesture sample at 30 fps. Each *hand* gesture sample is recorded in near proximity to the sensor (approximately 50 cm) so as to maximize hand detail.

Table 4.1: A comparison of mostly *body* gesture recognition datasets using either depth or mocap (motion capture) sensors. $^{\diamond}$In CMU Mocap, all users do not perform all gestures (some gesture types only have a single user performing it). ‡In MSRC-12, gestures are performed continuously in a long sequence, one after another causing inherent biases.

Dataset	# of Users	# of Gestures	Data Type
CMU Mocap (CMU, 2003)	>100	109$^{\diamond}$	Mocap
HDM05 (Müller et al., 2007)	5	>70	Mocap
MSRAction3D (Li et al., 2010)	10	20	Kinect v1 Depth
HumanEva I/II (Sigal et al., 2010)	4/2	6/1	RGB + Mocap
MSRC-12 (Fothergill et al., 2012)	30	12‡	Kinect v1 (Skeletons Only)
MSRGesture3D (Wang et al., 2012a)	10	12	Depth
MSRDailyActivity3D (Wang et al., 2012b)	10	16	Kinect v1 Depth
Berkeley MHAD (Ofli et al., 2013)	12	11	Multimodal (Depth + Mocap)
BodyLogin Combined (Ours)	40	5	Kinect v1 Depth
Handlogin (Ours)	21	4	Kinect v2 Depth

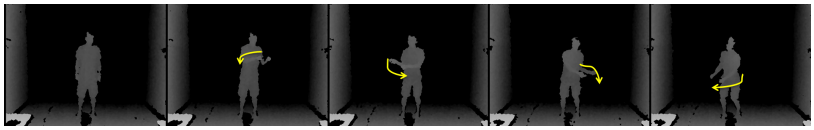
At some point in each dataset, users were instructed to perform a pre-defined gesture. Each user was instructed how to perform each gesture type through a text and video prompt (a multi-modal instruction scheme). In the literature, a multi-modal instruction scheme is known to improve gesture reproducibility over a single-modal instruction scheme (e.g., text or video only) (Fothergill et al., 2012). In order to strive for realistic intra-class variability and reduce pose bias, users were instructed to leave (for approximately one minute) and re-enter the recording area between gesture samples.

4.3 BodyLogin Dataset: Silhouettes and Skeletons

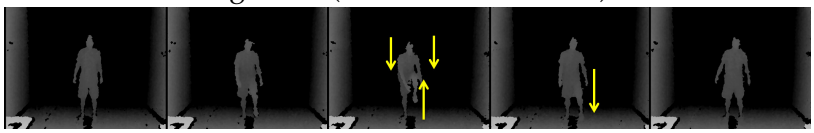
The BodyLogin Dataset: Silhouettes and Skeletons (BLD-SS) is used to compare the performance of silhouette and skeletal features in real world scenarios. Over

a two week, two session period (one session per week), gesture samples from 40 different college-affiliated users (27 males, 13 females) primarily 18-33 years old were collected. Each subject was asked to perform 2 unique short gestures (approximately 3 seconds long), each with 20 samples. In total, about 1.4 hours of data were recorded, with each user averaging 2 minutes of data (each sample about 3 seconds long). Both gestures involved motion in the upper and lower body (Figure 4·1):

- ***S gesture***: drawing an "S" shape with both hands,
- ***User-defined gesture***: user chooses his/her own gesture (no instruction).



S gesture (common to all users)



User-defined gesture: knee lift (will vary between users)

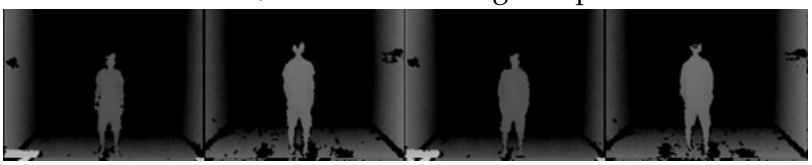
Figure 4·1: Snapshots of the gestures each user performed in our dataset (Kinect depth shown).

Degradations

Four different types of gesture scenarios were considered in BLD-SS:

- No degradations
- Personal effects
- User Memory
- Gesture Reproducibility

Before/after user wearing backpack



Before/after user wearing a sweatshirt

Figure 4·2: Views of users wearing personal effects. Left two images are before personal effects (front and back Kinect views shown), the right two images are after personal effects (front and back Kinect views shown).

In the case of *personal effects*, users either wear or carry something during their gestures. Half of our users were told to wear heavier clothing, and the other half were told to carry some type of a bag. Users wore a variety of heavier clothing: sweatshirts, windbreakers, and jackets. They carried backpacks (either on a single shoulder or both), messenger bags, and purses.

The impact of *user memory* was tested by collecting samples a week after a user first performed a gesture. Users were first asked to perform the gesture without any video or text prompt. After a few samples were recorded, the user was shown a prompt and asked to perform the gesture again. These last samples measured *reproducibility*. Of the 20 samples recorded for each gesture, each of the described scenarios had 5 samples recorded. Table 4.2 shows a summary of the degradation scenarios that were used for each gesture.

This dataset has been made available at `http://vip.bu.edu/projects/hcis/body-login/datasets/silhouettes-vs-skeletons/`

Table 4.2: Details of recording procedure for BLD-SS and BLD-M. Users were instructed to "reset" initial position between gesture performances. Half of the users wore coats, and the other half carried bags.

Gesture:	S gesture	User-defined
Session I:	1. Observe video and text description of gesture	1. Create custom gesture
	2. No degradation: Perform gesture normally (5 times)	2. No degradation: Perform gesture normally (5 times)
	3. Personal effects: Wear a coat, or carry a bag	3. Personal effects: Wear a coat, or carry a bag
	4. Perform gesture with personal effect (5 times)	4. Perform gesture with personal effect (5 times)
Session II (a week after):	1. Memory: Perform gesture from memory (5 times)	1. Memory: Perform gesture from memory (5 times)
	2. Observe video and text description of gesture	2. Observe video of prior performance from Session I.
	3. Reproducibility: Perform gesture (5 times)	3. Reproducibility: Perform gesture (5 times)

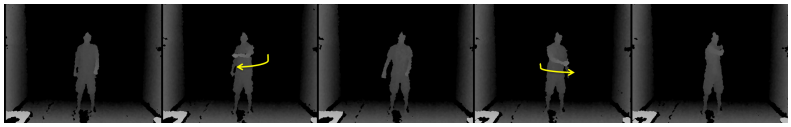
4.4 BodyLogin Dataset: Posture, Build and Dynamics

BodyLogin Dataset: Posture, Build and Dynamics (BLD-PBD) is a Kinect dataset that contains gestures across two sessions. In the first session, users recorded gestures under normal conditions. In the second session, users were matched to attack targets and were made to spoof another's gestures. This dataset was used to evaluate the effects of user-specific posture, build and dynamics in the context of authentication.

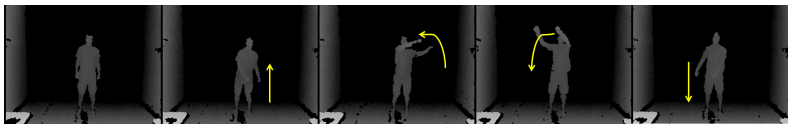
In total, about 1.8 hours of data were recorded, with each user averaging 3 minutes of data (each sample about 3 seconds long). There are 20 samples per gesture (10 own and 10 attack) for each user. Users were all college-affiliated (25 males, 11 females) mostly in the age range of 18-33 years. The three gestures, designed to be

of increasing complexity, involved movement in both the upper and the lower body (Figure 4·3):

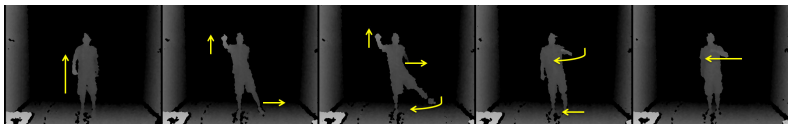
- ***Left-right gesture***: user reaches right shoulder with left hand, and then reaches left shoulder with right hand,
- ***Double-handed arch gesture***: user draws an arch from left to right with both hands,
- ***Balancing gesture***: user first raises right arm forward while pulling left arm back, then balances by forward sweeping left leg while simultaneously tucking left arm in and bringing right arm to rest.



Left-right gesture



Double-handed arch gesture



Balancing gesture

Figure 4·3: Snapshots of the gestures each user performed in our dataset (Kinect depth shown).

In order to facilitate our gesture spoofing study (Section 5.5 where extrinsic attacks are considered), each of the 3 gestures of each user was matched to an *attack target* after the first session. Attack targets were found by comparing the "centroid" samples of each user. If $\mathcal{A} := \{\mathbf{S}_1, \ldots, \mathbf{S}_m\}$ denotes a set of user's first-session sam-

ples (feature matrices) of a given gesture, then the user's "centroid" sample is defined for that gesture as follows:

$$\mathbf{S}_{centroid} = \arg\min_{\mathbf{S}\in\mathcal{A}} \sum_{i=1}^{m} d_{DTW}(\mathbf{S}, \mathbf{S}_i). \tag{4.1}$$

A user's attack target (in the second session), for a given gesture, is the owner of the closest centroid sample (nearest-neighbor) for that gesture. The aim of matching attackers to their "easiest victims" is threefold: 1) all participants can serve as attackers in the study 2) no participant is asked to attack more than one user which balances the burden across all participants, and 3) the odds of users succeeding as attackers are improved, which somewhat compensates for the lack of experience and the limited practice-time available for an attack. Under this matching scheme, vulnerable users get attacked more often than others and very distinct users never get attacked. Furthermore, attackers may end up attacking up-to three distinct users (one for each gesture). Most users had at least one attacker. The most attackers a user had was seven.

This dataset has been made available at `http://vip.bu.edu/projects/hcis/body-login/datasets/posture-build-dynamics/`

4.5 BodyLogin Dataset: Multiview

BodyLogin Dataset: Multiview (BLD-M) is a multiple-viewpoint Kinect dataset that contains gestures recorded under various degradations. This dataset was used to evaluate the value of additional Kinect viewpoints in gesture authentication.

This dataset is an extension of BLD-SS to multi-view. As with BLD-SS, over a two week, two-session period (one session a week), gesture samples from 40 different college-affiliated users (27 males, 13 females) primarily 18-33 years old were collected. Subjects were asked to perform 2 unique short gestures (approximately

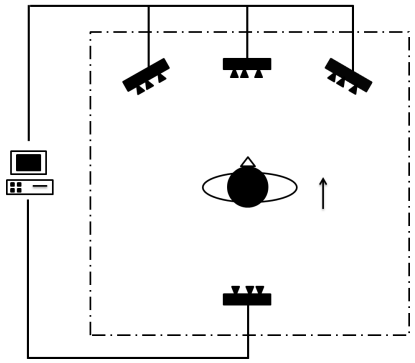


Figure 4·4: A setup with four Kinect cameras. Three Kinects (left, center, and right) were placed in front of the user, at offset angles, and one was placed behind the user (back). Skeletal estimation can be performed *independently* from each viewpoint.

3 seconds long), each with 20 samples. In total, approximately 1.4 hours of data were recorded, with each user averaging 2 minutes of data (each sample about 3 seconds long). As with BLD-SS, they performed the S gesture and User-defined gesture, each with the same degradations. However, the **main difference** is that the dataset was captured with 3 additional Kinects. 3 Kinects were placed facing the user and 1 Kinect was directly behind (Figure 4·4). Of the forward-facing Kinects, 2 were offset by about 35 degrees to the side, with 1 device directly in front. All devices were set up approximately 2 meters away from the user. Users were primarily facing the center camera for the duration of the performed gesture. All the Kinects were connected to a single PC to assure time synchronization. Captured frames were synced to the frame-rate of the center Kinect.

This dataset has been made available at `http://vip.bu.edu/projects/hcis/body-login/datasets/multiview/`

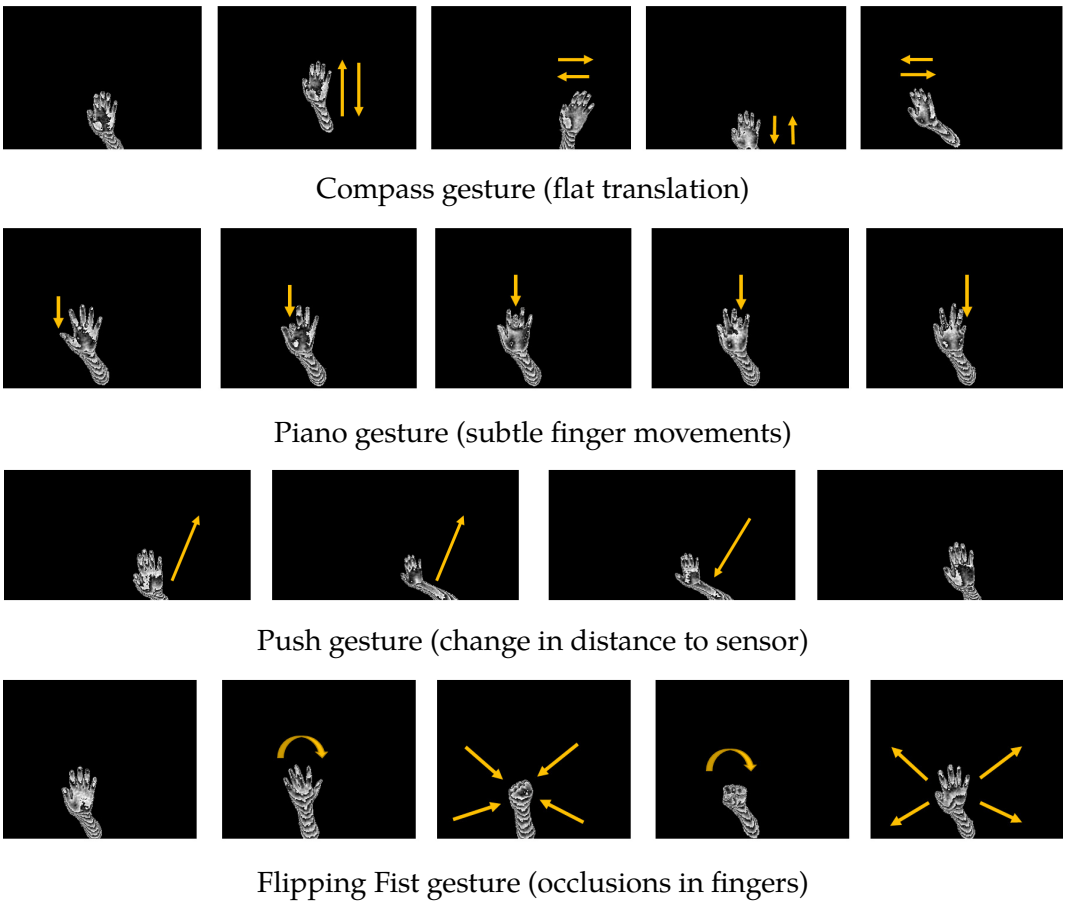


Figure 4·5: Kinect v2 depth images of the 4 gestures used for authentication. For visualization, images have been cropped, and only show the lower 4-bits of the 16-bit depth image.

4.6 HandLogin Dataset: In-air Hand Gestures

The HandLogin dataset was collected to evaluate the biometric performance of in-air hand gestures. Aiming to create a user-centric dataset (as the dataset from a related work was not publicly available (Aumi and Kratz, 2014)), hand gestures were collected from 21 college-affiliated users consisting of 15 males, and 6 females. Each user was asked to perform 4 unique types of hand-gestures, each type designed to be a few seconds in duration. Ten samples of each gesture type were recorded, with users instructed to leave and re-enter the recording area to reduce arm- and

hand-pose biases between samples. In total, approximately 0.7 hours of data were recorded.

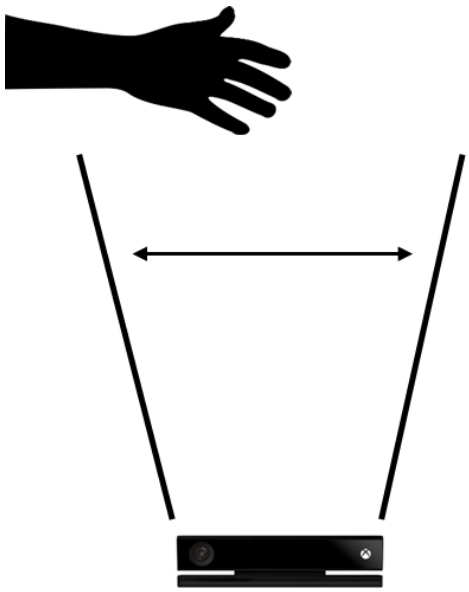


Figure 4·6: Visualization of HandLogin camera setup. Kinect v2 camera points towards the ceiling.

All 4 gestures were performed with the right hand starting in a "rest" position: the hand extended downwards on top of a ceiling-facing Kinect sensor, with fingers spread comfortably apart. This avoids the notorious "gorilla arm" issue, where users would need to maintain their hand in a vertical front-to-parallel position, instead using a more comfortable horizontal down-to-parallel position (see Figure 4·6). The orientation of our sensor was designed to mimic an authentication terminal, where typically only a single user is visible. The gestures that have been recorded:

Compass: users trace the compass directions of North, East, South and West with an open hand with the restriction that after each compass direction has been reached, the user must return to the center position before tracing a subsequent

direction; this gesture evaluates planar translation,

Piano: users use their fingers to "press" the keys of an imaginary keyboard – fingers are pressed one-by-one starting from the thumb and ending with the pinky; this gesture evaluates subtle fingertip movements,

Push: users pull the arm back, and push towards the sensor; this gesture evaluates depth translation,

Flipping Fist: users first flip the hand over and close it into a fist, and then flip the fist over and open it back to the starting hand pose in front of the sensor; this gesture evaluates the effect of occlusions and more sophisticated fingertip motion.

This dataset has been made available at `http://vip.bu.edu/projects/hcis/hand-login/dataset/`

4.7 Dataset Statistics

In both the aforementioned datasets, the duration of a single gesture sample is usually only a few seconds long. User-specific durations for fixed gesture types are shown in Figure 4·7 and 4·8 as stacked histograms. For the most part, these datasets consider gestures that are intentionally short. This is useful as performing a gesture that is too long becomes harder to remember and repeat. Further, having to perform a long gesture can become too prohibitive and too inconvenient over other alternative forms of access control. It is important to note that gestures that are of longer duration do not necessarily yield better performance. There can be scenarios where a gesture is too hard to remember, or too hard to replicate consistently, which can result in poor biometric performance relative to a shorter and simpler gesture.

All of the four aforementioned datasets have been made publicly available on the lab website. Table 4.3 shows the most recent snapshot of public access to this

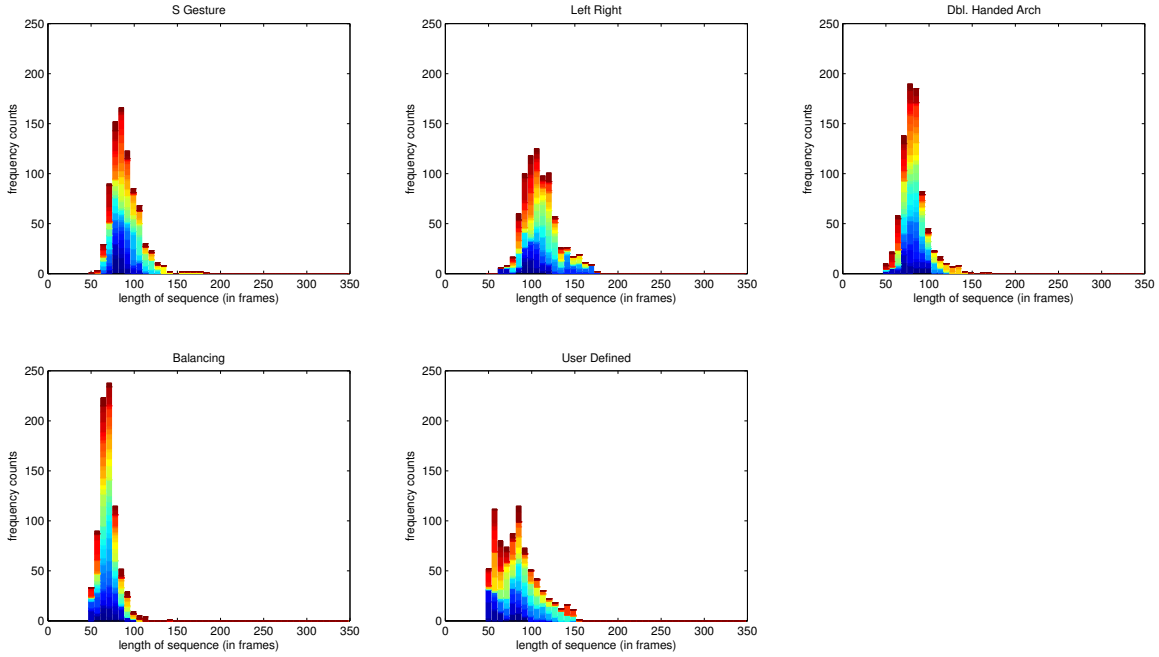


Figure 4·7: Stacked histograms representing the lengths of the 5 gestures in BodyLogin. Gestures were collected at 30 frames per second.

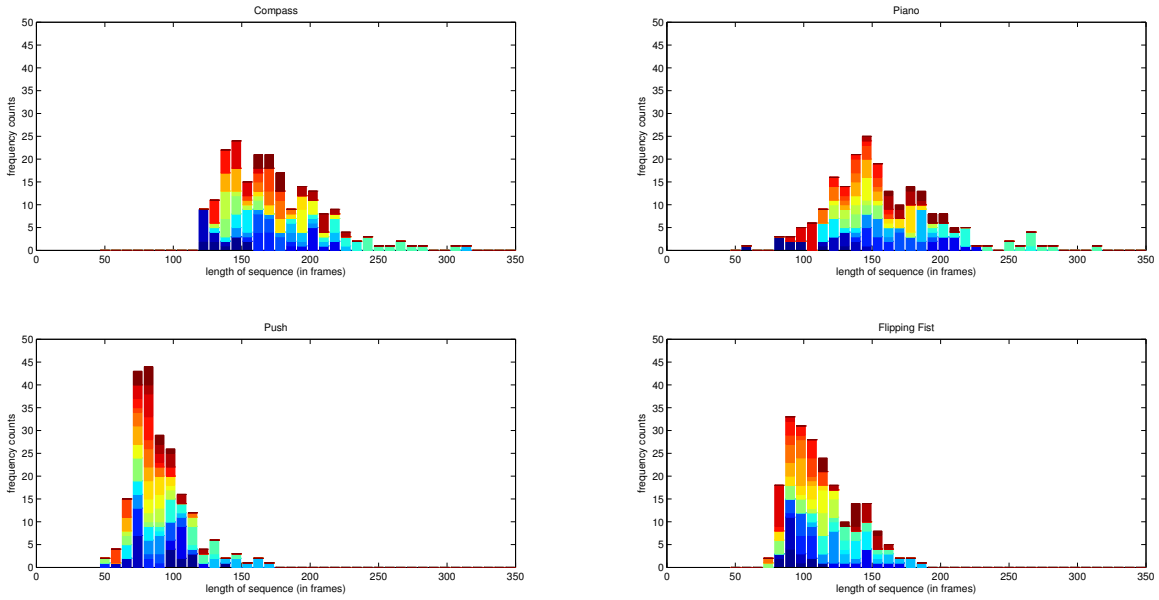


Figure 4·8: Stacked histograms representing the lengths of the 4 gestures in HandLogin. Gestures were collected at 30 frames per second.

resource.

Table 4.3: Current dataset downloads as of April 2016. Downloads are a count of *unique* and *registered* users not originating from a BU domain.

Dataset	**# of Downloads**	**Year Released**
BodyLogin: Silhouettes and Skeletons	8	Fall 2014
BodyLogin: Posture, Build, and Dynamics	11	Fall 2014
BodyLogin: Multiview	15	Summer 2014
HandLogin	3	Fall 2015

Chapter 5

Body Gestures

5.1 Introduction

In this chapter, three studies pertaining to *body gestures* are explored. First, a body gesture is decomposed into components – posture, build and dynamics. Subsequently, the impact of each component on biometric performance is evaluated. Next, various potential performance-reducing "threats" are discussed. One threat that is considered are the user induced degradations to the gesture sample. For example, these are cases where a user performs a body gesture while carrying a bag or wearing a heavy coat, or needs to recall and perform a gesture again after a period of time. Other threats considered are those caused by *other* users through spoof attacks. The impact on performance is evaluated when users are able to have access to one's personal gesture information, and attempt to "replay" one's gesture in front of a sensor. These threats are primarily investigated by changing the testing samples to magnify the threat under consideration. Finally, the biometric performance of additional camera viewpoints is explored. This analysis of multiple viewpoints also considers situations with user-induced threats.

5.2 Value of Posture, Build, and Dynamics

This section seeks to answer the following questions:

> *Which component of a gesture carries the weight of security performance? Is it the static information of the user (pose and build), or is it the dynamics (style of movements) the user performs and has full control over?*

In this section, a novel method is developed to suppress the effects of specific types of information in a gesture. A gesture can be represented as containing three types of user-specific information: initial body posture, limb proportions (build) and controlled user-dynamics. Each of these components can be isolated and suppressed by using a spherical coordinate representation of skeletal limb vectors. Initial posture can be suppressed by setting the initial limb vectors to a *standard initial posture* obtained by averaging the initial posture across all users. Build can be suppressed by setting all limb proportions to *standard limb proportions* obtained by averaging the limb proportions across all users. Finally, dynamics can be removed by ignoring all but the first frame of a gesture sequence. In the following sections, these information suppression processes are detailed. Further, by suppressing single or combination of components in a gesture, component-by-component security performance can be evaluated to see which component(s) are the most valuable. The analysis will show that the components rank in the order: dynamics, posture, build in terms of best authentication performance. The empirical results shown in this section are based on BLD-PBD dataset as discussed in Chapter 4.

5.3 Gesture Components

5.3.1 Extraction

A skeleton can be represented in multiple ways. The simplest and most direct representation is as a tuple of 20 *joint* vectors in rectangular coordinates as discussed in Section 3.2.2. An alternative representation which is more convenient for isolating the individual effects of initial posture, limb proportions, and dynamics, is as a tuple of 19 *limb* vectors together with one reference joint vector (see Figure 3·4). In this context, it is useful to view the skeleton as a rooted tree with, for concreteness, the spine joint (joint number 1) as the root (or the reference joint) of the tree and the outgoing connected joints as the children. By knowing the coordinates of root at time t as $\mathbf{s}^t_{spine} := \mathbf{s}^t_1$, and the outgoing edge vectors from the root, the entire skeleton can be reconstructed. If the limb connecting joints i and j at time t is denoted by the limb vector $\mathbf{v}^t_{i,j} := \mathbf{s}^t_j - \mathbf{s}^t_i$, then the feature vector (see Equation 3.2) can be alternatively represented as follows:

$$\mathbf{f}^t \equiv \{\mathbf{s}^t_{spine}, \mathbf{v}^t_{i,j}, i, j = 1, \ldots, 20 : i < j, (i,j) = \text{limb}\}. \tag{5.1}$$

To ensure that the initial position of the reference joint remains the same across different repetitions of a gesture, $\mathbf{s}^1_{spine}$ is subtracted from all the joint vectors across all the frames, or equivalently, $\mathbf{s}^1_{spine}$ is subtracted from only the reference joint vector across all frames in the limb vector representation (5.1). By doing this, the spine joint in the first frame is ensured to always be at the origin of the coordinate system.

Let $r^t_{i,j}, \theta^t_{i,j}$, and $\phi^t_{i,j}$ denote, respectively, the radius, azimuth angle, and elevation angle of the limb vector $\mathbf{v}^t_{i,j}$, i.e., the spherical coordinates of $\mathbf{v}^t_{i,j}$. Rectangular and spherical coordinates are information-equivalent representations of a vector

and one can readily convert from one set of coordinates to another, i.e.

$$\mathbf{v}_{i,j}^t = (x_i^t - x_j^t, y_i^t - y_j^t, z_i^t - z_j^t) \leftrightarrow (r_{i,j}^t, \theta_{i,j}^t, \phi_{i,j}^t).$$

The **initial posture** for each gesture can be thought of as the unintentional, habitual orientation of one's body parts. This orientation can be described via the azimuth and elevation angles of all skeletal edges in the first frame:

$$\{(\theta_{i,j}^1, \phi_{i,j}^1), i, j = 1, \ldots, 20 : i < j, (i,j) = \text{limb}\}.$$

Subsequent postures in the sequence pertain to the gesture's **dynamics**. The **limb proportions** describe the shape of a user's body (user build) regardless of the gesture that he/she performs. Ideally, limb lengths should not change across frames. However, the estimates of joint coordinates produced by the Kinect SDK are not perfect. This issue is addressed by computing the *average* length of each limb in a gesture across all frames and dividing it by the average length of the spine limb as follows:

$$\bar{r}_{i,j} = \frac{\sum_{t=1}^{T} r_{i,j}^t}{\sum_{t=1}^{T} r_{spine}^t} \tag{5.2}$$

where $r_{spine}^t = r_{1,2}^t$ is the spine limb length (Figure 3·4) and $\bar{r}_{i,j}$ is the limb proportion for limb (i,j). Thus, for a given gesture sequence with 20 skeletal joints, there will be 19 limb proportions,

$$\bar{\mathbf{r}} := \{\bar{r}_{i,j}, i, j = 1, \ldots, 20 : i < j, (i,j) = \text{limb}\},$$

where $\bar{r}_{1,2} = 1$. From the above discussion it follows that any gesture can be represented as the combination of three sets of values: initial posture, $\{(\theta_{i,j}^1, \phi_{i,j}^1), \forall i, j : (i,j) = \text{limb}\}$, limb proportions $\bar{\mathbf{r}}$, and dynamics $\{(\theta_{i,j}^t, \phi_{i,j}^t), \forall i, j : (i,j) = \text{limb}, t =$

$2, \ldots T\}$.

5.3.2 Suppression

This approach to study the individual and combined effects of initial posture, limb proportions and dynamics on user authentication performance is to first transform a given set of gestures to new ones (in rectangular coordinates) in which one or more gesture components (initial posture, limb proportions, dynamics) are either retained or suppressed and then evaluate the authentication performance on the transformed set of gestures.

The advantage of this approach, is that it allows us to use a single classifier and a single common feature space, namely the rectangular skeletal coordinates, for all the component combinations. If separate classifiers were developed for each combination of components (which live in different feature spaces), it would be unclear whether any performance differences are due to the components or/and the specific classifiers.

Suppressing Initial Posture

To remove the effects of user-specific initial posture, a limb-specific angular offset is introduced, $(\Delta\theta_{i,j}^{offset}, \Delta\phi_{i,j}^{offset})$, to every single frame. The goal of this is to orient the initial posture (1st frame), to a *standard* initial posture. As a result, this also re-orients subsequent frames in a sequence. The standard initial posture can be found by averaging the initial posture angles across all samples of all the users to yield $(\theta_{i,j}^{1,standard}, \phi_{i,j}^{1,standard})$. The angular offsets are then the angular differences between the standard initial posture and the user's initial posture:

$$(\Delta\theta_{i,j}^{offset}, \Delta\phi_{i,j}^{offset}) = (\theta_{i,j}^{1,standard} - \theta_{i,j}^{1}, \phi_{i,j}^{1,standard} - \phi_{i,j}^{1}) \tag{5.3}$$

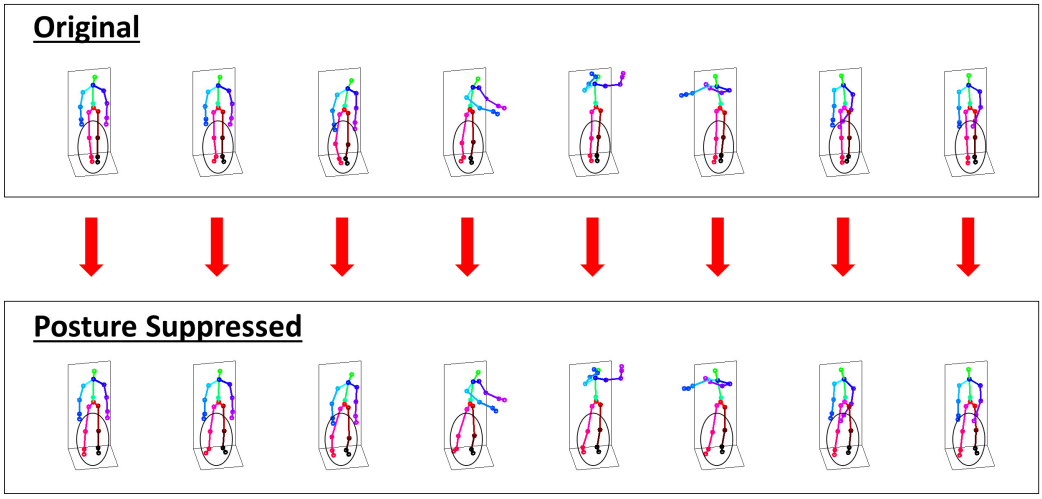


Figure 5·1: A skeletal sequence before and after initial body posture suppression. Note the change in leg posture.

A transformed gesture (in rectangular coordinates) with the initial posture suppressed (i.e., standardized) is then obtained by adding the angular offsets to the spherical coordinates of all frames and converting the result back to rectangular coordinates:

$$(\bar{r}_{i,j}, \theta^t_{i,j} + \Delta\theta^{offset}_{i,j}, \phi^t_{i,j} + \Delta\phi^{offset}_{i,j}) \rightarrow (\mathbf{v}^{t,noposture}_{i,j}). \tag{5.4}$$

A typical effect of initial posture suppression is shown in Figure 5·1.

Suppressing Limb Proportions (Build)

To remove the effects of limb proportions, the radial distances (limb length proportions) are replaced with a set of *standard* limb proportions. Standard limb proportions are found by averaging the limb proportions across all samples of all users to obtain $\bar{\mathbf{r}}^{standard}$. A transformed gesture (in rectangular coordinates) with the limb proportions suppressed (i.e., standardized) is then obtained by replacing the radial distances (limb length proportions) with the standardized limb length proportions

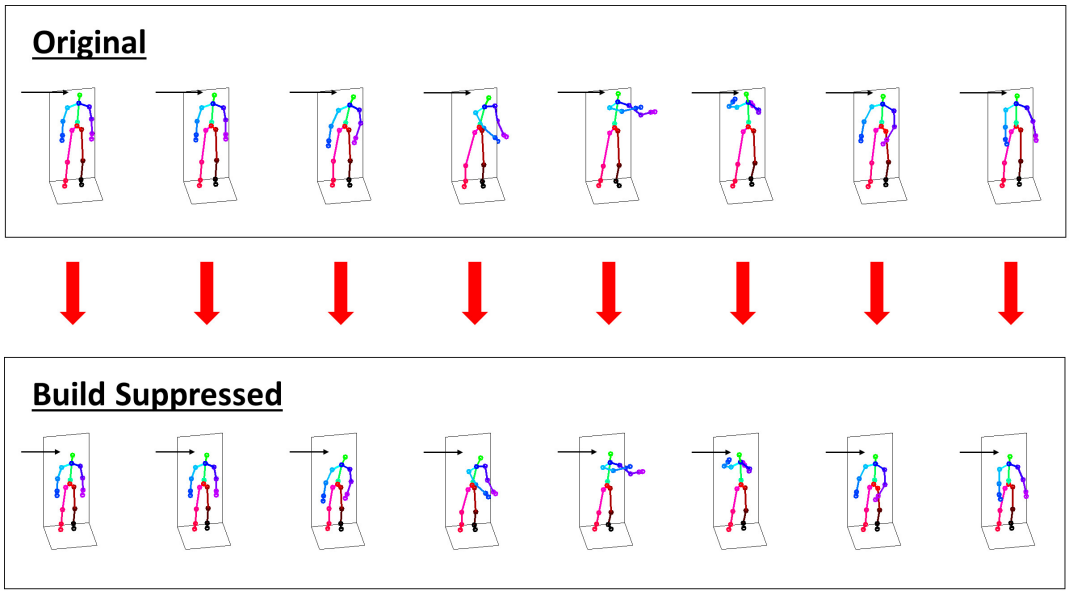


Figure 5·2: A skeletal sequence before and after body build suppression. Note the change in user height.

in all frames and converting the result back to rectangular coordinates:

$$(\bar{r}_{i,j}^{standard}, \theta_{i,j}^{t}, \phi_{i,j}^{t}) \rightarrow (\mathbf{v}_{i,j}^{t,nobuild}) \tag{5.5}$$

A typical effect of body build suppression is shown in Figure 5·2.

Suppressing Dynamics

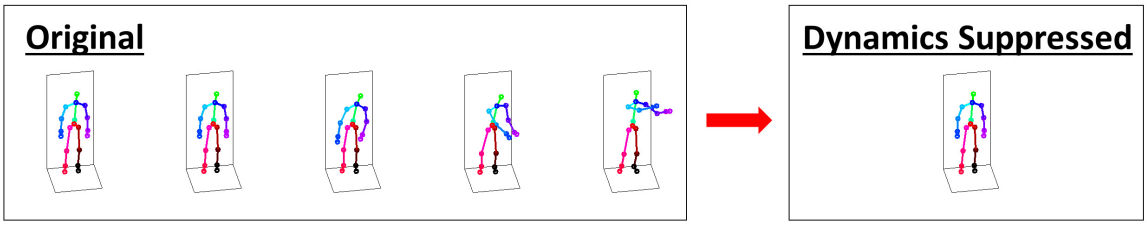


Figure 5·3: A skeletal sequence before and after dynamics suppression.

The suppression of dynamics is quite straightforward: just keep the first frame and discard the others. The effect of this suppression is visualized in Figure 5·3.

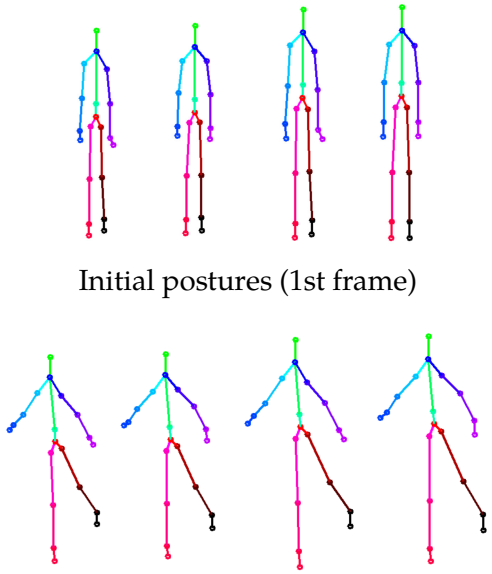


Figure 5·4: Skeletons with various components suppressed. Top row suppressions (left to right): Dynamics, Dynamics+Posture, Dynamics + Build, Dynamics + Posture + Build (standard initial posture and standard build). Bottom row suppressions (left to right): Nothing, Posture, Build, Posture + Build.

Suppressing Component Combinations

Above, we described how to remove each of the three types of information. To remove more than one type of information at a time, one only needs to combine the procedures proposed above for the information that is to be removed. Table 5.1 describes various combinations of information that are evaluated. The case where all components are suppressed is not evaluated as all gesture samples would be identical. Figure 5·4 shows a few samples of skeletons for gestures constructed using this methodology.

Table 5.1: Various combinations of components considered when reconstructing gesture sequences.

Information Suppressed	Initial Posture	Limb Proportions	Dynamics
Nothing	✓	✓	✓
Dynamics	✓	✓	
Build	✓		✓
Posture		✓	✓
Dynamics+Build	✓		
Dynamics+Posture		✓	
Posture+Build			✓

Table 5.2: User authentication EER (average of user-specific EERs) with zero effort attacks when various components are suppressed (please see Table 5.1 for component combinations). The best-performing EERs for each gesture are in boldface.

Information Suppressed	Left-right	Double-handed arch	Balancing
Nothing	**1.97%**	**0.25%**	**0.68%**
Dynamics	3.83%	3.01%	2.12%
Build	2.09%	0.38%	1.20%
Posture	3.75%	0.61%	1.30%
Dynamics +Build	4.29%	4.88%	3.72%
Dynamics +Posture	8.22%	4.76%	4.39%
Posture +Build	6.91%	0.91%	3.22%

Results: Effects of Posture, Build, and Dynamics

Authentication EER is computed for all 36 users from first-session samples of BLD-PBD dataset for each of the 3 gestures. This is equivalent to considering all 36 users as performing zero-effort attacks against one another in the worst case scenario when they all select the same gesture. Only skeletal information was used in this analysis (silhouettes are not considered). Each skeletal gesture sequence was compared using our variant of dynamic time warping. Feature normalization was not

performed (normalization may suppress some component). The 7 combinations of gesture components that are described in Section 5.3.1 were applied to each of the 3 gestures, as shown in Table 5.2. If each user has a different gesture, the EER performance would only be better (lower) than the values shown here.

In terms of gestures, the "double-handed arch" performs best, followed by "balancing," and then the "left-right" gesture. The "left-right" gesture should be expected to perform the worst as it is the least sophisticated (complex) of the three gestures. The "balancing gesture" was originally expected to perform the best due to its high complexity (it requires hand-leg coordination and body balancing). Surprisingly, it was only second-best. This can be explained, in retrospect, by the difficulty of reliably reproducing a complex gesture which has the effect of increasing the FRR and thereby the EER. So while complex gestures may be psychologically appealing as having higher discriminative power, they may actually be counterproductive because they can be difficult to reproduce.

In terms of gesture components, the suppression of dynamics has the single largest impact on the EER for every gesture followed by, somewhat surprisingly, the initial posture, and finally build. For example, for the "double-handed arch" gesture, the EER increases by 2.76% (from 0.25%) when the dynamics are suppressed, by 0.36% when the posture is suppressed, and by 0.13% when the build is suppressed (Table 5.2). Clearly dynamics play an important role. However, the role of posture and build is not insignificant. For instance, for the "left-right" gesture, the EER with posture and build retained but with dynamics suppressed is 3.83% which is lower than 6.91% when only dynamics are preserved. When all components are used, the EER is 1.97%. Similarly for the "balancing" gesture the EER with only posture and build (no dynamics) is 2.12% which is smaller than 3.22% when only dynamics are preserved. When all components are used, the EER is 0.68%. Thus,

while dynamics is the most significant component of the three, the combination of all components results in a significant improvement.

5.4 Intrinsic Threat Model: Performance Under Degradations

An intrinsic threat occurs when an authorized user is responsible for a security vulnerability. This type of threat can be analyzed by adding samples with user-induced degradations into the testing set. Degradations due to personal effects, such as when users either wear or carry something during their gestures, are considered. For example, users could be wearing thick sweatshirts or carry backpacks. This is an important degradation to consider for daily-use, e.g., should users perform well with this degradation it would show promise that this modality maintains security performance while being convenient. Degradations due to time are also considered.

This section seeks to answer the following questions:

> *After a period of time, such as a week, would users be able to recall a gesture successfully? Further, if users are shown their own previously-performed gesture, can they reproduce it successfully?*

Results: Silhouettes vs Skeletons

Conveniently, the BLD-SS dataset contains 2 different gestures (S gesture and User-defined) that are recorded with the 3 degradations of interest. To analyze this dataset, multiple representations and distance functions were tried to see which worked best across the degradations. In particular, the focus was to see whether the silhouette representation or the skeletal representation was more resilient to intrinsic degradations. The distance methods applied to each feature representation evaluated are shown in Figure 5·5.

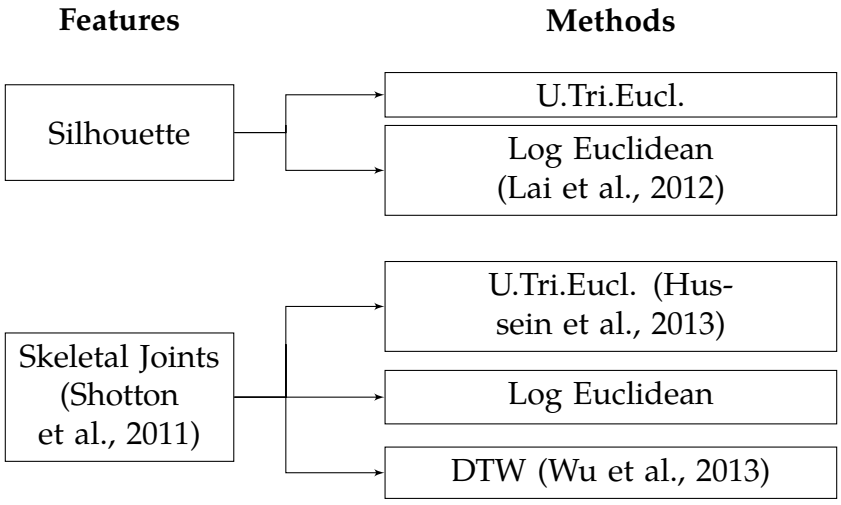


Figure 5·5: Overview of the features and methods studied for intrinsic threats. Both silhouette and skeletal joint features are extracted from depth maps generated by the Kinect sensor.

Table 5.3: Global equal error rate (EER) for the problem of user authentication. Smaller is better. The best results for silhouettes and for skeletons are shown in boldface. **SvS$^{\triangle}$**: denotes the error difference between the *best* performing result from skeletons vs. silhouettes (best skeleton - best silhouette). Everything* contains samples both with and without degradations.

Gesture	**Train-set/Test-set**	**User Authentication Equal Error Rate (EER)**						
	Data Type:	**Silhouette**		**Skeletal**			**SvS$^{\triangle}$ Diff.**	**Skel. Better**
	Distance Metric:	**Log-Eucl.**	**U.Tri. Eucl.**	**Log-Eucl.**	**U.Tri. Eucl.**	**DTW**		
S Gesture	No degrad./No degradations	3.46%	**2.70%**	9.30%	7.79%	**5.26%**	2.56%	
	No degrad./Personal-effects	**11.13%**	12.97%	12.94%	10.67%	**6.56%**	-4.56%	✓
	No degrad./User memory	**17.62%**	19.79%	24.13%	13.61%	**13.42%**	-4.20%	✓
	No degrad./Reproducibility	**20.16%**	20.22%	24.09%	**14.04%**	16.60%	-6.13%	✓
	No degrad./All the above	**14.12%**	15.12%	18.94%	12.09%	**11.16%**	-2.96%	✓
	Everything*/Everything*	**2.85%**	4.35%	13.94%	7.14%	**4.49%**	1.64%	
	Column Averages:	11.56%	12.53%	17.22%	10.89%	9.58%	-2.27%	✓
User-defined	No degrad./No degradations	**1.12%**	1.69%	19.19%	1.56%	**0.30%**	-0.82%	✓
	No degrad./Personal-effects	**2.51%**	4.65%	21.69%	3.31%	**0.68%**	-1.83%	✓
	No degrad./User memory	**12.14%**	13.82%	27.54%	**2.07%**	2.97%	-10.07%	✓
	No degrad./Reproducibility	**12.86%**	15.00%	25.38%	3.70%	**2.09%**	-10.77%	✓
	No degrad./All the above	**7.28%**	8.53%	24.07%	2.93%	**1.70%**	-5.58%	✓
	Everything*/Everything*	**0.73%**	1.45%	23.83%	1.76%	**0.45%**	-0.29%	✓
	Column Averages:	6.11%	7.52%	23.62%	2.55%	1.36%	-4.89%	✓

Table 5.4: Correct classification error (CCE) for the problem of closed-set identification. All users who query the system are *known* before-hand to exist in the system (no reject option). Please see the caption of Table 5.3 for explanations.

Gesture	Train-set/Test-set	User Closed-set Identification Error (1 - CCR)						
	Data Type:	Silhouette		Skeletal			SvS$^{\triangle}$ Diff.	Skel. Better
	Distance Metric:	Log-Eucl.	U.Tri. Eucl.	Log-Eucl.	U.Tri. Eucl.	DTW		
S Gesture	No degrad./No degradations	**2.50%**	2.50%	4.50%	7.00%	**1.00%**	-1.50%	✓
	No degrad./Personal-effects	**16.00%**	27.00%	10.50%	14.50%	**5.50%**	-10.50%	✓
	No degrad./User memory	**42.50%**	57.50%	28.00%	32.00%	**21.00%**	-21.50%	✓
	No degrad./Reproducibility	**44.00%**	55.50%	37.50%	33.50%	**29.00%**	-15.00%	✓
	No degrad./All the above	**26.25%**	35.63%	20.13%	21.75%	**14.13%**	-12.13%	✓
	Everything*/Everything*	**1.88%**	3.25%	2.25%	2.25%	**1.00%**	-0.88%	✓
	Column Averages:	22.19%	30.23%	17.15%	18.50%	11.94%	-10.25%	✓
User-defined	No degrad./No degradations	**1.00%**	3.00%	7.50%	**0.00%**	**0.00%**	-1.00%	✓
	No degrad./Personal-effects	**3.06%**	5.10%	13.27%	3.06%	**1.02%**	-2.04%	✓
	No degrad./User memory	**19.00%**	24.00%	24.00%	**4.00%**	5.00%	-15.00%	✓
	No degrad./Reproducibility	**21.00%**	23.00%	21.61%	**2.01%**	3.52%	-18.99%	✓
	No degrad./All the above	**11.06%**	13.82%	16.60%	**2.26%**	2.39%	-8.79%	✓
	Everything*/Everything*	**0.25%**	0.75%	5.91%	0.63%	**0.13%**	-0.13%	✓
	Column Averages:	9.23%	11.61%	14.81%	1.99%	2.01%	-7.66%	✓

Training and Testing Scenarios

Six types of training and testing scenarios are considered for evaluation. Five scenarios only train with clean, "No degradations" user samples. These five scenarios are then tested with user samples either from "No degradations," "Personal effects," "User memory," "Reproducibility," or "All the above" (all degradations – does not include "No degradations"). The final, sixth scenario "Everything" considers when degradations are included in training data. In this case, "Everything" trains and tests off *all* gesture samples with and without degradations. Tables 5.3 - 5.4 show these scenarios in identification and authentication for the S gesture and User-defined gesture.

Discussion of Results

Impact of Gesture Complexity

Overall, across both tables, User-defined gestures outperform S gestures for every scenario. This is expected, as it is harder to distinguish between users performing the same gestures. For User-defined gestures, each user has a distinguishable descriptor mean. Whereas, for S gestures, the descriptor means are similar. Introducing degradations, causes a *more* detrimental effect in the latter, as can be seen in our results.

Impact of Degradations

Unsurprisingly, introducing any degradation into the testing data causes an overall drop in performance. In particular, the effect of time (second-session gestures recorded after a week) has a greater impact than personal-effects. From our results, silhouette features are more adversely impacted by degradations than skeletons. If we look at the impact of degradations for silhouette features using log-euclidean

distance, the EER increases by around fivefold on average. Shape changes from either the user performing a gesture differently, or having a different silhouette from the inclusion of a bag or coat, has a great impact on silhouette features.

Skeletal features on the other hand, are not as impacted, as the joints tend to lie within stable "centered" regions within the silhouette. If we look at the impact of degradations for skeletal features using upper triangular euclidean distance, the EER increases by around a quarter on average. However, this can be a "double-edged sword." Although skeletal features are more robust to shape deformations, silhouettes may carry more information. This is seen to be the case with the S gesture where silhouette features outperform skeletons when the training data contains samples of the type of degradations that can appear during testing.

Sensitivity to Sample Reproducibility

There are a few peculiarities in our results. In S gestures, there is a decrease in performance from user memory to reproducibility, while in the User-defined case there is an increase. Upon closer inspection of the data, we noticed that a few users performed a mirror-image of their gestures during the user memory degradation tests. Gestures that were normally left to right, were performed right to left. This tended to happen more frequently in User-defined gestures. When users were shown the gesture again, user performance improved – which is what is seen for the case of User-defined gestures. The reverse is true in S gestures, which is attributed to a slight difference in protocol. Whereas in the User-defined case the user's original performance from a week earlier was shown, in the S-gesture case a generic action recording (from an individual who was not part of the dataset) was shown. From these empirical results, it would seem that users tend to perform differently when instructed to replicate themselves or another person.

Impact of Representation Metrics

In terms of metrics, silhouettes tend to have the best results with the Log-Euclidean distance. For skeletons, however, DTW empirically outperforms covariance metrics in most cases. This may be because the lengths of the gesture sequences are short (about 3 seconds or 90 frames) in comparison to the feature dimensionality (60). For silhouettes, features exist on a *per pixel* level (order of millions) whereas skeletal features only exist at a *per frame* level (order of hundreds). If the number of frames in the sequence were much longer (or a temporal hierarchy were to be considered), covariance metrics may outperform DTW.

Impacts Overall

Overall, in the idealized scenario for individual-user authentication (Table 5.3) where each user has a customized gesture and the training set contains examples of all degradations ("Everything"), the EER is as low as 0.45% (Skeletons with DTW) across 40 users. On the average, the best skeletal results outperform the best silhouette results by 4.89% EER. Even when all users perform the same gesture (S gesture), the EER is as low as 2.85%. For this gesture, on the average, the best skeletal results outperform the best silhouette results by 2.27% EER. Furthermore, the skeletal features always outperform silhouette features when training data contains no samples with degradations, but testing data does.

For closed-set identification (Table 5.4) our empirical results indicate that skeletal features always outperform silhouettes, but the improvement is small when the training set is representative of the testing set (No degradations/No degradations and Everything/Everything).

Although our conclusions about the improved performance with skeletal features cannot be drawn in every scenario, our results indicate that skeletal features

are always preferable when training data lacks degradations that are present during testing. Thus, in situations where large quantities of varied training data cannot be obtained, skeletal features may be preferable. Further, our results indicate that the most significant degradation of the ones considered are time-related. Possibly, this leaves room for either procedure improvement (selecting more "memorable" gestures, more training, more frequency of gesture use, etc.) or algorithmic improvement.

5.5 Extrinsic Threat Model: Performance Under Spoof Attacks

An extrinsic threat occurs when another user is responsible for a security vulnerability, such as from a spoof attack. This naturally raises the following question:

How easy is it to spoof someone's gesture?

This question can be somewhat answered by using minimal effort impersonation attacks (Gafurov et al., 2007). These are when amateur attackers are trained to mimic an authorized user with limited training time, basic knowledge of the system, and a set number of attack trials. This threat can be more effective when the mimicking target is matched to a user most similar to the attacker. Conveniently, the BLD-PBD dataset can be used as it contains such types of attacks.

The attacks in the BLD-PBD dataset limited the time to study a target to 1 minute and permitted 10 trials (10 recorded samples) by each attacker on a single target per gesture. For practice, attackers were allowed to view a looping video recording of their target's "centroid" sample from the first session. Attackers were given a chance to "mirror" the gesture by being shown streaming video of their practice. Once they were comfortable or a minute had elapsed, the spoof attempts were recorded.

Effects of Amateur Spoofing Attacks

Table 5.5: EER shown for matched zero-effort attacks, and matched spoofing attacks. Results are shown for when no information is suppressed (Nothing), and when user-unique initial posture and build information are removed.

Gesture	Information Suppressed	Matched Zero-Effort EER	Matched Spoof EER	$\mathrm{EER}_{\mathrm{Spoof}} - \mathrm{EER}_{\mathrm{Zero-Effort}}$
Left-right	Nothing	2.78%	2.35%	-0.43
	Posture+Build	7.33%	10.28%	+2.95
Double-handed arch	Nothing	1.24%	1.13%	-0.11
	Posture+Build	3.78 %	4.22 %	+0.44
Balancing	Nothing	2.66%	2.06%	-0.60
	Posture+Build	5.60%	6.36%	+0.76

Using the same procedure as in Section 5.2 on the BLD-PBD dataset, each sample is represented by a skeletal sequence and is compared to another sample using the nearest-neighbor DTW distance. In order to evaluate spoofing attacks, EER is considered in two contexts: matched zero-effort EER and matched spoofing EER. In computing matched zero-effort EER, only samples from the first session belonging to the pool of authorized users who will be attacked in the second session (approximately 16 users attacked for each gesture) are used. For each authorized user, unauthorized samples are only considered from users who will attack them in the second session. As we only use first-session samples, all these unauthorized samples are "matched" zero-effort attacks.

Following this train of thought, the matched spoofing EER is computed across the same authorized users with the only difference being unauthorized samples that are now second-session spoof attacks instead of first-session ones. These results are shown in Table 5.5.

Additional insight into this threat can be found by using the work of Section 5.2. By utilizing component suppression, we can determine how resilient each compo-

nent is to spoofing. In fact, when static information is suppressed (build and posture) leaving behind only dynamic information, it can be seen that attackers are more successful and security performance decreases. This shows that with training, attackers can replicate the dynamics of a gesture, but they are not quite able to imitate the static components of a gesture.

Discussion of Results

Intuitively, one would expect the EER to increase after a matched spoofing attack relative to a matched zero-effort attack. Surprisingly, for this dataset, the EER performance actually slightly improves for all 3 gestures. This suggests that it is non-trivial for lay persons to effectively copy a user's gesture even when they are explicitly asked to attack their most vulnerable target and they have the opportunity to practice using a video recording of their target performing his/her gesture.

Despite the unexpected decrease in EER of the matched spoofing attack relative to the matched zero-effort attack, interestingly, the EER based on dynamics alone, i.e., with posture and build suppressed actually increases consistently across all three gestures (see the last column of Table 5.5). This suggests that training does improve the ability of a lay user to copy the *dynamics*. Thus, body build and initial posture offer a limited but non-negligible level of protection against spoofing attacks.

5.6 Value of Multiple Viewpoints

Adding more sensors can help improve the security performance of a system. For gestures, this addition has the natural advantage of capturing motion that may be occluded in a single viewpoint (for example arms behind one's back). The question to answer is:

How much improvement can be gained from these additional sensors, and are they worth it?

The multi-view skeletal-based user samples from BLD-M can be used to answer this question.

Introducing additional viewpoints requires modifications to the framework. In particular, some sort of fusion scheme is necessary. Two simplistic schemes are used to approach this problem: score- and feature-based fusion.

5.6.1 Score Fusion

In score fusion, each Kinect viewpoint is considered to be an *independent* system. Each system computes an individual score for a given query gesture against a template from the enrollment database, and an aggregate score across all viewpoints is used to determine an acceptance or rejection.

For a given gesture g_1, multiple feature matrices $\mathbf{F}_{g_1,v}$ are considered, where v denotes one of V viewpoints. Since the viewpoints are considered to be independent, only features from the same v are compared. This means only distances of the form: $d_v = d_*(\mathbf{F}_{g_1,v}, \mathbf{F}_{g_2,v})$ are computed. Thus, given a set of V viewpoints, there will be a set of scores $S = \{d_1, d_2, ...d_V\}$ consisting of distances from each viewpoint for any two gesture sequences. To get a fused score, one of the following operations on the set S is applied: *min*, *mean*, *median*.

5.6.2 Feature Fusion

In feature fusion, *concatenation* is considered: combining features *before* a distance score is computed. To achieve this, the feature vectors $\mathbf{F}_{g_1,v}$ across all V viewpoints is simply concatenated. For example:

$$\mathbf{F}_{g_1,global} = [\mathbf{F}_{g_1,1}; \mathbf{F}_{g_1,2}; ...\mathbf{F}_{g_1,V}]$$

yielding a new singular feature of size $60V \times T$ for skeletal features (see Section 3.2.2), where T is the number of frames. If the temporal hierarchy-based covariance descriptor described in Section 3.3.1 is used, a final upper-triangular descriptor would be of length $7(60^2V^2 + 60V)/2$. Naturally, given a single descriptor, there will only be one score, so no subsequent score fusion step is necessary.

5.7 Effects of Multiview

The final score obtained through aggregation of individual scores or through feature concatenation is used to evaluate authentication and identification performance. Although more sophisticated fusion techniques could be applied, we believe the key insights into the benefits of using multiview data would remain unchanged.

For analysis, intrinsic attacks are considered, as before in Section 5.4, but in the context of multiple viewpoints. Extrinsic attacks (spoofing) have not been specifically analyzed with multiple viewpoints, but the overall conclusions are expected to be similar. Skeletal sequences were taken from multiple views in the BLD-M dataset, and were converted into temporal hierarchy-based covariance descriptors. The upper-triangular Euclidean distance $d_{Hier.Eucl}(\cdot, \cdot)$, was subsequently taken between sequences.

Table 5.6: Equal error rate (EER) for authentication shown to 1 digit of precision. Smaller is better. FRR, and thereby EER, is rounded off to the nearest accuracy margin which is the fraction of 1 over the number of positive samples in each train-set (this margin is $1/200 = 0.5\%$ for all but "Everything*" which is $1/800 = 0.125\%$). The best results for single-viewpoint and multi-viewpoint are shown in boldface. **B.M-B.S** denotes the error difference between the *best* performing result from multi-viewpoint vs. single-viewpoint (best multi - best single). **M-C**, similarly denotes the error difference between the mean multi-viewpoint scheme and the center camera (mean scheme - center). "Everything*" contains samples with and without degradations.

User Authentication Equal Error Rate (EER)

Gesture	Train-set/Test-set	Single-Viewpoint			Multi-Viewpoint Fusion				Multi minus Single	
	Camera/Method:	Left	Right	Center	Min	Mean	Median	Concat	B.M-B.S	M-C
S Gesture	No degrad./No degrad.	**4.0%**	5.5%	9.0%	**5.0%**	5.5%	**5.0%**	6.5%	1.0%	-3.5%
	No degrad./Personal effects	**7.5%**	**7.5%**	9.0%	7.5%	7.5%	**7.0%**	8.0%	-0.5%	-1.5%
	No degrad./User Memory	**11.5%**	**11.5%**	14.0%	11.5%	**10.5%**	11.0%	11.0%	-1.0%	-3.5%
	No degrad./Reproducibility	**11.5%**	12.5%	13.0%	**10.5%**	**10.5%**	11.0%	11.0%	-1.0%	-2.5%
	No degrad./All of the above	**9.5%**	**9.5%**	12.0%	**8.5%**	9.0%	9.0%	9.5%	-1.0%	-3.0%
	Everything*/Everything*	4.1%	**4.0%**	6.1%	3.9%	3.9%	**3.8%**	4.1%	-0.3%	-2.3%
	Column Averages:	8.0%	8.4%	10.5%	7.8%	7.8%	7.8%	8.4%	-0.2%	-2.7%
User-defined	No degrad./No degrad.	1.5%	2.0%	**1.0%**	1.0%	**0.5%**	**0.5%**	1.0%	-0.5%	-0.5%
	No degrad./Personal effects	**1.5%**	2.0%	2.0%	1.5%	**1.0%**	1.5%	2.0%	-0.5%	-1.0%
	No degrad./User Memory	2.5%	**1.5%**	2.5%	2.0%	**1.5%**	**1.5%**	2.0%	0.0%	-1.0%
	No degrad./Reproducibility	3.5%	**2.0%**	3.5%	3.0%	**2.0%**	2.5%	3.5%	0.0%	-1.5%
	No degrad./All of the above	2.5%	**2.0%**	2.5%	2.0%	**1.5%**	2.0%	2.0%	-0.5%	-1.0%
	Everything*/Everything*	1.5%	**1.0%**	1.4%	1.3%	**0.9%**	**0.9%**	1.4%	-0.1%	-0.5%
	Column Averages:	2.2%	1.8%	2.1%	1.8%	1.2%	1.5%	2.0%	-0.5%	-0.9%

Table 5.7: Correct classification error (CCE) for closed-set identification shown to 1 digit of precision. All query samples are assumed to have been enrolled into the system beforehand. See the caption of Table 5.6 for explanations.

User Closed Set Identification (CCE)

Gesture	Train-set/Test-set	Single-Viewpoint			Multi-Viewpoint Fusion				Multi minus Single	
	Camera/Method:	**Left**	**Right**	**Center**	**Min**	**Mean**	**Median**	**Concat**	**B.M-B.S**	**M-C**
S Gesture	No degrad./No degrad.	**1.5%**	3.0%	3.5%	**2.0%**	2.5%	2.5%	2.5%	0.5%	-1.0%
	No degrad./Personal effects	6.5%	**6.0%**	11.1%	7.0%	6.5%	7.0%	**6.0%**	0.0%	-4.5%
	No degrad./User Memory	19.5%	20.0%	**17.5%**	18.0%	15.5%	16.5%	**14.5%**	-3.0%	-2.0%
	No degrad./Reproducibility	28.5%	26.0%	**22.5%**	26.5%	21.5%	22.0%	**20.5%**	-2.0%	-1.0%
	No degrad./All of the above	14.0%	13.8%	**13.6%**	13.4%	11.5%	12.0%	**10.9%**	-2.8%	-2.1%
	Everything*/Everything*	1.3%	**1.1%**	1.3%	**0.5%**	0.8%	1.3%	0.6%	-0.6%	-0.5%
	Column Averages:	11.9%	11.7%	11.6%	11.2%	9.7%	10.2%	9.2%	-2.4%	-1.9%
User-defined	No degrad./No degrad.	**0.0%**	0.5%	0.5%	**0.0%**	**0.0%**	**0.0%**	**0.0%**	0.0%	-0.5%
	No degrad./Personal effects	0.5%	**0.0%**	1.0%	**0.0%**	0.5%	1.0%	0.5%	0.0%	-0.5%
	No degrad./User Memory	**1.5%**	2.5%	3.5%	3.0%	1.5%	2.0%	**1.0%**	-0.5%	-2.0%
	No degrad./Reproducibility	**1.0%**	**1.0%**	1.5%	1.0%	**0.0%**	1.0%	0.5%	-1.0%	-1.5%
	No degrad./All of the above	**0.8%**	1.0%	1.6%	1.0%	**0.5%**	1.0%	0.5%	-0.3%	-1.1%
	Everything*/Everything*	**0.1%**	0.3%	0.1%	**0.0%**	0.1%	0.1%	0.1%	-0.1%	0.0%
	Column Averages:	0.7%	0.9%	1.4%	0.8%	0.4%	0.9%	0.4%	-0.2%	-0.9%

Various results for authentication and identification are shown in Tables 5.6 and 5.7, respectively. Due to degraded skeletal pose estimates from the rear-facing Kinect, only the frontal 3 cameras are considered in our multi-viewpoint evaluations. Under our methodology, including the 4-th camera does not improve results.

At a high level, looking at the row "Everything" in Table 5.6, S gestures are outperformed by User-defined gestures. This should come as no surprise, as it should be harder to distinguish between users when they *all* perform the same gesture. In consequence, it is not surprising that the presence of degradations introduced into the test set causes a more significant performance drop for S gestures than for User-defined gestures. The introduction of any degradation causes a performance drop, although more for some degradations than others. In particular, time-related degradations (samples after a week), as seen in "User Memory" and "Reproducibility" rows, produce a larger drop than "Personal effects" degradation. Overall, these results are consistent with the conclusions drawn in Section 5.4.

It turns out that for results based on multiple viewpoints, the center camera is not always the best performing one across all test-set scenarios; the side cameras (left and right) consistently outperform the center camera. For the S gesture, the training sample that is closest to the test sample belongs to the center camera for only about 22% of the test samples ($\sim$32% match to the left and $\sim$46% to the right). For the User-defined gesture, about 39% of the test samples find their best match among the center training samples, about 28% with the left, and about 33% with the right. This can be explained as follows. If a part of the body is occluded during a gesture in one viewpoint, another camera may be able to see the gesture more clearly without this occlusion. Inherently, this shows the value of multiview acquisition during enrollment and testing.

If one multi-viewpoint fusion method out of all the methods applied had to be

picked, the mean score fusion would be chosen. This is because it performs the best across both gesture test-sets (mean fusion scores are bolded the most). Thus, in comparison to the single-viewpoint **authentication** setup which only consists of a single *centered* camera, an average EER decrease of 2.7% (∼26% relative improvement) and 0.9% (∼43% relative improvement), for the S and User-defined gestures can be found due to multi-viewpoint mean fusion. Similarly, in comparison to the single-viewpoint **identification** setup, an average CCE decrease of 1.9% (∼16% relative improvement) and 0.9% (∼68% relative improvement), for the S and User-defined gestures, is found due to multi-viewpoint mean fusion. In every testing scenario that is compared to the center viewpoint, multiple viewpoints always outperform – they are always more informative.

Even if the best performing single-viewpoint camera is compared against the best performing fusion scheme, separately for each of the six training/testing scenarios, an overall improvement in performance by using multiple viewpoints is still found. Specifically, the average EER decreases by 0.2% (∼3% relative improvement) and 0.2% (∼33% relative improvement), and average CCE decreases by 2.4% (∼23% relative improvement) and 0.2% (∼32% relative improvement), respectively for the S and User-defined gestures.

Finally, a finer perspective of the benefit of multiple viewpoints can be obtained by examining the ROC curves for single-view and multi-view authentication for "No degradations/All of the above" scenario shown in Figure 5·6.

5.8 Concluding Remarks

This chapter investigated multiple aspects of using body gestures as a biometric. Specifically:

- showing that dynamic information is invaluable to authentication perfor-

mance (ranked the highest in our methods over posture and build).

- a way to decompose body gestures into posture, build and dynamics.
- an investigation into possible degradations that a user can induce such as having a bag, wearing a coat, or having to perform a gesture after a period of time.
- showing that skeletal features tend to be more robust over silhouette features against user degradations.
- an investigation into spoof attacks.
- showing that dynamic information is susceptible to spoof attacks, but build and posture are less so.
- an investigation into how much there is to gain by adding additional camera viewpoints.

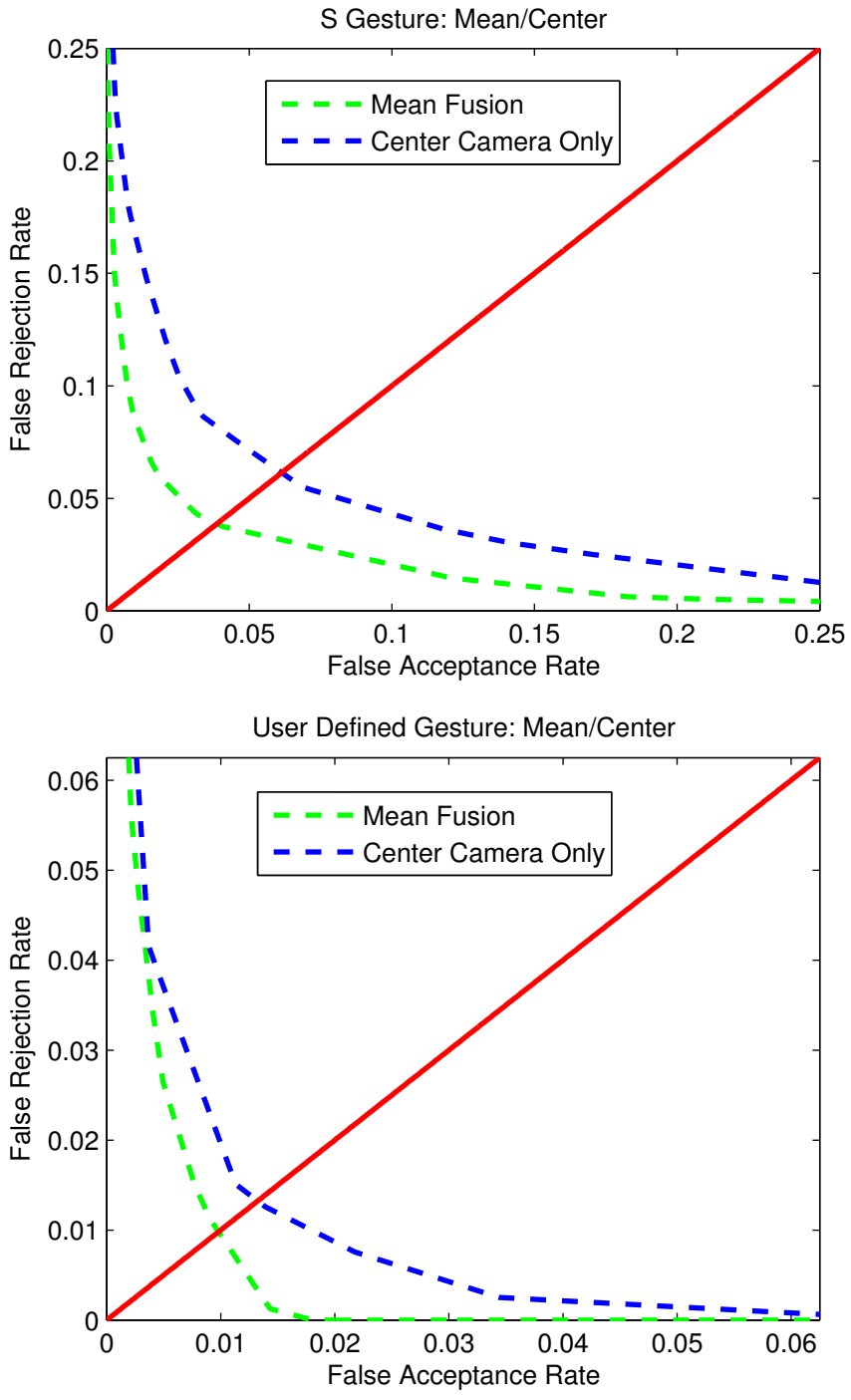


Figure 5·6: Convex hull of the ROC curves illustrating the EER improvement from using multiple views: mean fusion is compared to a single view (center). These results correspond to the EER values from the train-set/test-set "No degradations/All of the above," in Table 5.6 for both the S gesture and User-defined gesture.

Chapter 6

Hand Gestures

6.1 Introduction

Figure 6·1: Depth images of in-air hand gestures captured with Kinect v2 that is pointing towards the ceiling.

This chapter focuses on evaluating the biometric potential of in-air hand gestures. A novel approach is proposed for user access by leveraging a temporal hierarchy of depth-aware silhouette covariances, which is an extension and improvement to the methods proposed in previous chapters. Further, the usefulness of shape and depth information is investigated in this modality, as well as the importance of hand movement when performing a gesture. The empirical results described in this chapter are based on the HandLogin dataset described in Chapter 4. By exploiting both shape and depth information our method attains an average 1.92% Equal Error Rate (EER) on a dataset of 21 users across 4 predefined hand-gestures. This method consistently outperforms related methods on this dataset.

6.2 Related Work

Much like for body gestures, there has been extensive work in hand-gesture recognition with depth-sensors such as the Kinect (Ren et al., 2011b; Ren et al., 2011a; Suarez and Murphy, 2012; Wang et al., 2012a; Kurakin et al., 2012), but there has been little work in authentication. Perhaps the work most closely related is that of Aumi and Kratz (Aumi and Kratz, 2014) who propose using dynamic time warping (DTW) across six 3-D fingertip and palm coordinates (coarse hand pose-estimation) from a depth-image for hand-gesture authentication. Similar "signature"-type gestures have also been proposed using an accelerometer, gyroscope and touchscreen on a mobile phone (Patel et al., 2004; Farella et al., 2006; Liu et al., 2009; Sae-Bae et al., 2012). Many of these methods only use a single-coordinate in space and propose elastic matching algorithms (much like DTW) for authentication. As a result, physiological and anatomical shape information of the hand is lost in all these approaches. Although not the focus of this chapter, authentication with American Sign Language (ASL) using RGB frames (one frame per signed letter) has been investigated using features such as color histogram, DCT, and entropy (Gupta et al., 2013; Fong et al., 2013).

6.3 Extended Silhouette Representation

A few modifications are done to the feature representation originally proposed in Chapter 3. Specifically, depth is added to the feature vector (see Equation 3.1), and a spatial-hierarchy is proposed in addition to the temporal hierarchy. The entire method is briefly recapped below.

As before, a compact silhouette representation can be extracted from any gesture sequence consisting of depth frames. A sequence of binary silhouettes of a

hand-gesture is extracted by thresholding the difference between each depth frame and known depth background. Afterwards, the largest two-dimensional, component with 8-connected pixel connectivity of each frame is assumed to belong to the user's hand silhouette tunnel. Let $n = 1, ..., N$ index all N pixels within the silhouette tunnel and let (x, y, t) denote the space-time coordinates of pixel number n. Similarly to Equation 3.1, the following 14-dimensional feature vector $\mathbf{f}^n$ is computed at each silhouette pixel which captures the shape, depth, and dynamics of a gesture:

$$\mathbf{f}^n = \mathbf{f}(x, y, t) := [x, y, t, z, \delta_E, \delta_W, \delta_N, \delta_S, \delta_{NE}, \delta_{SW}, \delta_{SE}, \delta_{NW}, \delta_{T+}, \delta_{T-}]^T \quad (6.1)$$

where z is the depth value at (x, y, t), and d_{dir} denotes the Chebyshev distance between a pixel n and its the nearest silhouette boundary pixel in direction dir. The first 8 directions are in the x, y, spatial plane (4 cardinal directions and 4 intercardinal directions), and the last 2 are in the temporal direction (forward and backward in time). Further, let $\mathbf{F} = [\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^N]$ denote a 14 by N matrix that is computed from any silhouette tunnel. The features are rescaled by normalizing each row of $\mathbf{F}$ to range from 0 to 1 as done before (see Section 3.2.3).

6.4 Covariance Descriptor

The aforementioned silhouette representation can be seen as a "bag of features" since each silhouette pixel has an associated 14×1 feature vector. A 14×14 empirical covariance matrix C of the collection of feature vectors can be used to provide a low-dimensional, second-order descriptor:

$$C := \frac{1}{N} \sum_{n=1}^{N} (\mathbf{f}^n - \boldsymbol{\mu})(\mathbf{f}^n - \boldsymbol{\mu})^T, \quad (6.2)$$

where $\boldsymbol{\mu}$ is the empirical mean of feature vectors $\mathbf{f}^n$. Since C is a symmetric matrix, the upper-triangular portion of size $(14^2+14)/2 = 105$ can be used as an equivalent gesture descriptor. Further, a simple way to compare the distance between the two descriptors is to compute the Euclidean distance (Frobenius norm) between them. This is denoted by $d_{U.tri.Eucl}(\cdot,\cdot)$.

6.5 Adding a Temporal Hierarchy

A temporal hierarchy can be incorporated to assign importance to a specific order of frames in a gesture (see Chapter 3). As before (see Section 3.3.1), 3 temporal hierarchy levels are applied. The upper triangular portions of all covariance matrices computed from this hierarchy are concatenated together to yield one long gesture descriptor. With 3 levels, this yields 7 covariance matrices, giving a final descriptor length of $105 \times 7 = 735$. The Frobenius norm between two descriptors of this type is used as the distance metric.

6.6 Incorporating Hand Morphology

Figure 6·2: A complete hand silhouette is partitioned into 3 "sub"-silhouettes using per-frame adaptive depth thresholding. The thenar eminence can be seen in the lower-left region of the second silhouette.

Additional biometric information may be found by leveraging hand morphology (such as the thenar eminence in Figure 6·2). An investigation was done to understand whether silhouette parts (perhaps pertaining to the hand's morphology)

could be used to improve authentication performance. To achieve this goal, a crude segmentation algorithm was applied, based on depth, to partition a hand silhouette into multiple ("sub"-silhouette) regions. Across a sequence of frames, these mutually-exclusive regions form additional "sub"-silhouette tunnels. To generate these tunnels, multiple frame-dependent depth thresholds r are found as follows.

Consider the frame t, where K segmented silhouettes need to be found ($K = 3$ is used). Let $(r_{t,1}, r_{t,2}, \cdots, r_{t,K+1})$, be ordered depth-thresholds with the property that the range between any consecutive threshold pair $(r_{t,i}, r_{t,i+1})$ contains a fraction $\frac{1}{K}$ of all the depth values in the given frame. Using these thresholds, the segmentation at frame t associated with the i-th threshold pair yields frame t in the i-th "sub"-silhouette tunnel. As these segmentations can be noisy, connected components with less than 20 pixels were removed.

A covariance matrix was computed for each of these "sub"-silhouette tunnels. For fusion, the Frobenius norms (of covariance matrix differences between the query and enrolled samples) is averaged across 4 tunnels: the 1 full-silhouette and the 3 i-corresponding "sub"-silhouette covariances. Further, each of these matrices can use a temporal hierarchy as described in the next section.

Discussion of Results

Entry control performance is evaluated in the context of user authentication (Jain et al., 2011).

Table 6.1 shows the authentication EER and confidence interval for 4 methods. The first method is a baseline, which is used to highlight the improvements that can be gained from incorporating depth-information and leveraging a temporal hierarchy. In this method, we use a single silhouette tunnel and a 13×13 covariance matrix that does not use the depth value z from the feature vector in Equation (6.1).

Table 6.1: Average of user-specific EER and the confidence interval for various methods and gestures. The best-performing EER for each gesture is in boldface. Please refer to Section 6.6 for additional information about these methods.

Method	Gesture Used				
	Compass	Piano	Push	Flipping Fist	Average
1. Baseline	2.97%±0.69%	4.32%±0.99%	5.94%±1.31%	3.85%±0.91%	4.27%±0.51%
2. Temporal Hierarchy	1.98%±0.57%	2.90%±0.82%	5.28%±1.14%	2.40%±0.73%	3.14%±0.43%
3. Additional Tunnels + Temporal Hierarchy	**0.44%±0.16%**	**1.29%±0.46%**	**4.89%±0.95%**	**1.05%±0.54%**	**1.92%±0.35%**
4. First Frame of Silhouette Tunnel	5.33%±1.04%	7.00%±1.30%	7.94%±1.37%	6.44%±1.27%	6.68%±0.62%

Subsequent methods use the full 14×1 feature vector (with z). The second method, incorporates the temporal hierarchy representation as described in Section 6.5. The third method, incorporates the temporal hierarchy representation and additional silhouette tunnels (a total of 4) from Sections 6.5 and 6.6. The last method (fourth) shows the value of motion in a gesture. Since, all gestures start with the right-hand in a neutral position, using only the first frame will have no dynamics. Further, there is no time information in the first frame, as the features associated with (t, d_{T-}, d_{T+}) in the covariance matrix become irrelevant.

Incorporating additional silhouette tunnels from depth-information and the temporal hierarchy representation yields the best result on average (method 3) with an 1.92% EER. Comparing this to the baseline with a 4.27% average EER, there is a 2.35% EER reduction.

The gestures in order of performance from best to worst are: Compass, Flipping Fist, Piano, then Push. The push gesture always performs the worst. This is believed to be due to poorer hand segmentations that are prevalent in this gesture. A few users were noted as pulling the arm back too close to the body, resulting in silhouettes that sporadically include portions of the chest. Since only depth information is used in this approach, it is difficult to differentiate between the body

parts. Using RGB information may be useful in this case since skin pigment should be easy enough to differentiate from clothing.

As expected, method 4, using only the first frame performs the worst out of all the methods. This enforces the notion that there exists a unique behavioral movement in each user's hand gesture.

It is important to point out that the Flipping Fist gesture performs quite well (2nd best). This result indicates that occluded shapes of the hand (such as a fist where all five fingers are hidden still contains useful biometric information). This suggests that hand gestures need not be limited to cases where all fingers are visible, and that in these cases, leveraging features based on shape and depth (as features based on fingertip locations are ill-defined) is useful for authentication.

6.7 Concluding Remarks

This chapter investigated the use of silhouette and depth representations for user authentication from in-air hand gestures. Compared to our baseline silhouette covariance approach described in previous chapters, our depth-enhanced representation reduces the EER by over 2%. Furthermore, the value of movement is demonstrated; without movement (authentication from hand shape only) the EER increases by almost 5%. This is important since hand motion is a renewable component of an in-air hand gesture and can be easily replaced if compromised, thus leading to enhanced security.

Chapter 7

Deep Learning and Gesture Styles

In this chapter, deep learning methods are explored in the context of gesture biometrics (for both hand and body). The following are the key contributions:

- Development of a two-stream convolutional network for user identification and authentication based on body and hand gestures.
- Assessment of the value of dynamics for user identification and authentication.
- Evaluation of the user-insensitive representation for gesture recognition, and gesture-insensitive representation (style) for user identification and authentication.

This approach is validated on two biometrics-oriented datasets (BodyLogin and HandLogin), and one gesture-centric dataset (MSRAction3D).

7.1 Motivation for Learning Gesture Style

Prior chapters evaluate gesture biometric performance by matching each user to a single gesture motion – effectively associating each user with a fixed gesture "password". A user is expected to recall and replicate this specific gesture for subsequent security access. In this chapter, this assumption is generalized, by learning a gesture "style," across a bank of trained gesture motions. Effectively, rather than focusing on identifying a user performing a specific "password," the goal is to identify

a user across a *set* of gestures, in-effect learning a user-specific *gesture style*. Recent advances in deep convolutional neural networks are leveraged in this section, and are shown to be able to outperform methods proposed in prior chapters for user recognition. Further, performance is evaluated against non-trained gestures, as well as when users are not trained in the initial network. For evaluation, body- and hand-based gestures from depth maps acquired by Kinect sensors (v1 and v2) (Kin, 2014) are focused on (Figure 7·1).

7.2 Related Work

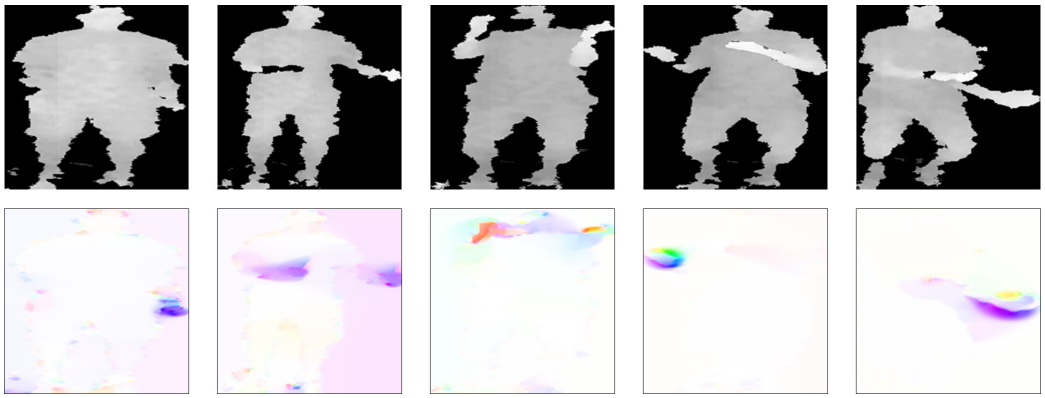
Perhaps the closest to the goal of this work can be found in (Kviatkovsky et al., 2015), where action-specific metric learning from normalized joint positions of the body was used to predict identity from a pool of known actions. Our work differs in that user identity is learned directly from depth images (end-to-end), without the need to have pose estimates of body joint positions. We use depth maps and associated optical flow, which can be useful in cases where skeletal pose estimation is not reliable or available (such as for hand poses).

7.3 Convolutional Neural Networks

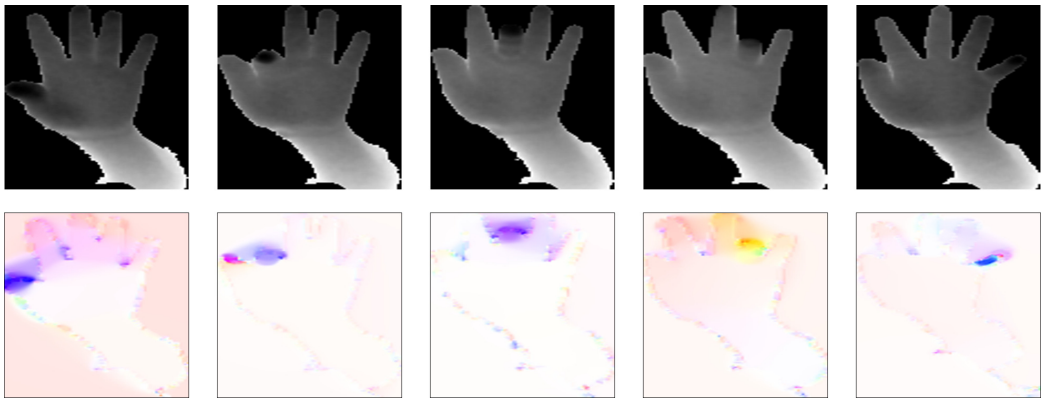
Deep convolutional neural networks (CNNs) have become very successful in vision tasks involving single still images. One of the contributions of this chapter is in adapting such CNNs to gesture-based biometrics where both static limb proportions as well as gesture dynamics (style) come into play.

The goal of CNNs is to learn a large set of kernel weights optimal for a particular loss function. Within this domain, several single-image network architectures have been proposed, such as: AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015), and VGGNet (Simonyan and Zisserman, 2014b). These networks gen-

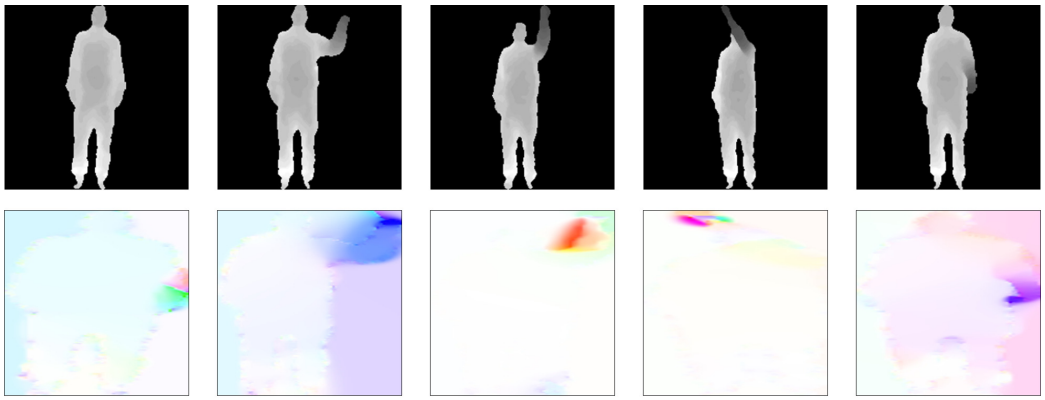
erally vary in the number of layers, number of kernels, and size of kernels.



BodyLogin: Full-body gestures (captured with Kinect v1)



Handlogin: In-air hand gestures (captured with Kinect v2)



MSRAction3D: Full-body gestures (captured with Kinect v1)

Figure 7·1: Examples of normalized depth images and corresponding colored optical-flow (Liu, 2009) for body and hand gestures captured using various depth sensors. Hue indicates optical flow orientation, and saturation indicates magnitude.

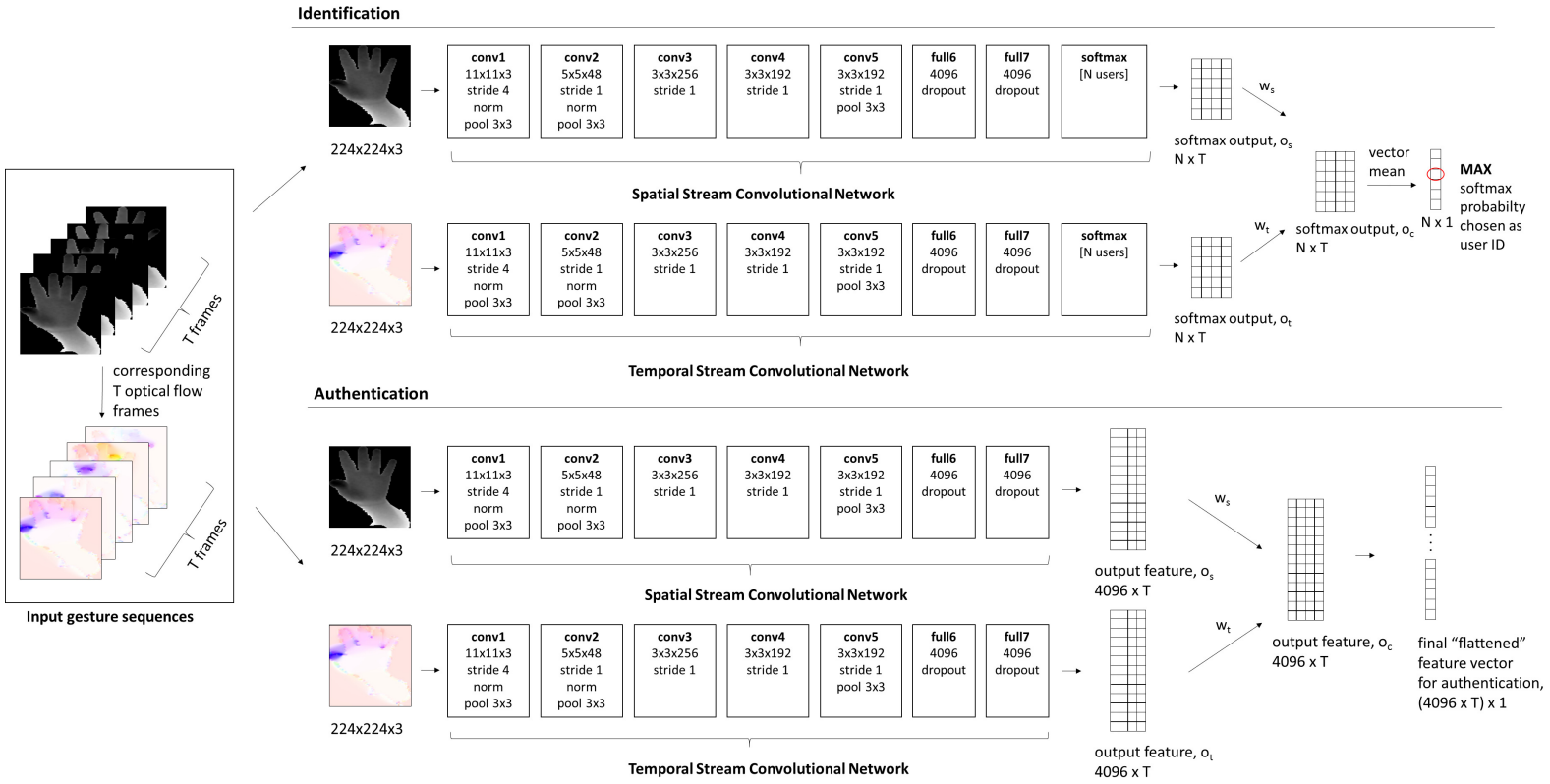


Figure 7·2: A visualization of how a deep network is used for user identification and authentication. In identification (top), a network is fine-tuned using gesture depth-map frames and optical flow. In authentication (bottom), weights are borrowed from an identification network, and the fully-connected layer output is used as authentication features.

In this section, AlexNet is used to analyze the biometric performance of gestures. AlexNet (Krizhevsky et al., 2012) is an eight-layer deep convolutional network consisting of five convolutional and three fully-connected layers (the last of which is a soft-max layer). This network is adapted to gesture sequences by using a variant of the two-stream convolutional network architecture proposed in (Simonyan and Zisserman, 2014a). Two-stream convolutional networks, as the name implies, train two separate convolutional networks: one for spatial information, and a second one for temporal information. Although such networks were originally intended for RGB images, they have been adapted to handle depth maps.

The first network is a "spatial stream" convolutional network (Figure 7·2) where a stream of T input depth-map frames, extracted from the input video through uniform temporal sub-sampling, are mapped to a stream of T output feature vectors (o_s) by passing each frame one-by-one through this network.

The second network is a "temporal stream" convolutional network that takes a sequence of colored, optical flow frames as input. Optical flow (Liu, 2009) is computed for each pair of adjacent depth-map images (depth-map values are treated as luminance values). The computed optical flow vectors are mapped into polar coordinates and then converted to hue (vector's angle), and saturation (vector's magnitude), with a fixed brightness (Figure 7·1). Much like in the first network, this stream of T input optical flow frames is mapped to a stream of T output feature vectors (o_t) by passing each colored optical flow frame one-by-one through the "temporal-stream" network.

A simple convex combination of the outputs of both networks is used to yield a single output o_c which is then used for performance evaluation:

$$o_c = w_s o_s + w_t o_t,$$

where $w_s \geq 0$ is the spatial stream network weight, $w_t \geq 0$ is the temporal stream network weight, $w_s + w_t = 1$, and o_s and o_t are the respective network outputs. When $w_s = 1, w_t = 0$, only information from the spatial-stream network is used, and when $w_s = 0, w_t = 1$, only information from the temporal-stream network is used. These results are reported for a wide sampling range of (w_s, w_t) weight pairs. A schematic visualization of such a network is shown in Figure 7·2.

7.3.1 CNNs for Identification and Authentication

Identification: The use of this network for *closed-set identification* (given a gesture, *identify* a user from a set of known users) is straightforward. During training (see Section 7.3.2), gesture sequences are broken up into single frames to be trained standalone. During testing, the mean of the soft-max probability outputs across T frames (o_c) is used. As described in the previous section, o_c is a weighted combination of the softmax probabilities for an input across two networks. This yields a single soft-max probability vector of length N (given N users to identify), and the component with the largest probability identifies the user. Although not the main focus of this section, gesture recognition uses the same structure where N is the number of gestures rather than the number of users to identity.

Authentication: In *authentication* (given a gesture, is a user who (s)he claims to be?), the output features from the "full7" layer of the network trained for identification (Figure 7·2) are used. This avoids having to train a separate authentication network for each user which is very expensive computationally and there are not enough training samples for each positive class represented by an authentic user. In this approach, for T frames that are uniformly sampled from a gesture sequence, two features of dimension $4096 \times T$ are extracted yielding o_s and o_t, whose linear

combination gives o_c (4096 is the length of the last fully connected layer). Since there is no built-in classification in this approach (no soft-max layer), these features are used as inputs to a two-class classification algorithm for authentication, e.g., based on nearest-neighbor or SVM. The intuition behind this idea is that, given enough users to identify, the network will naturally learn a user-separating feature space, which can be leveraged for authentication.

The parameters and training of all the elements of our networks are discussed in the next section.

7.3.2 Network Implementation Details

Typically, there are not enough training samples in gesture datasets to train all the weights of a deep convolutional network from scratch. Therefore, the common practice is followed to "pre-train" the network (Donahue et al., 2014; Karayev et al., 2014) using weights from another network with sufficient data. In our case, the dataset "pre-training" an AlexNet is ImageNet (Russakovsky et al., 2015) (resulting in a network with a soft-max loss function that classifies RGB images into 1000 classes) to initialize the weights in our 5 convolutional layers (conv1 to conv5). Although our modality is different, as we use depth images and colored optical flow (instead of RGB), initializing with ImageNet weights is still effective. Our fully-connected layers are trained from scratch, with weights initialized to be zero-mean Gaussian with a small standard deviation of 0.001. In all our networks, a batch size of 256 images is used. The spatial stream networks are started with a learning rate of 0.003, decreasing this rate by one-tenth every $3,000$ iterations until a total of $12,000$ iterations are completed. The temporal stream networks are started with a learning rate of 0.001, decreasing this rate by one-tenth every $1,000$ iterations until a total of $6,000$ iterations are completed. Dropout is set to at 0.5 in the fully-connected layers

of both networks.

The entirety of all the networks are implemented using Caffe (Jia et al., 2014; Wang et al., 2015) on a single Titan Z GPU.

7.4 On Gesture Datasets

This method is evaluated on 3 publicly-available datasets. Two of these datasets (Chapter 4) were designed for user authentication and identification (collected with the intention of maximizing the number of users): BodyLogin and Handlogin. The third one, was designed for gesture recognition (collected with the intention of maximizing the number of gesture/action types).

HandLogin (Wu et al., 2015) is a dataset containing in-air hand gesture sequences of 21 users, each performing 4 different gestures that are recorded by a Kinect v2 sensor (see Chapter 4 for details).

BodyLogin (Wu et al., 2014a; Wu et al., 2014c; Wu et al., 2014b) is a full-body multi-view dataset containing gesture sequences of 40 users performing 5 different gestures that are recorded by Kinect v1 sensors. The complete BodyLogin that is used here is the *combination* of all 3 body datasets described in Chapter 4. Four of these gestures are pre-defined and the fifth gesture is created by the user (user-defined). In this study, training and testing use samples across all degradations, and only from those of the center camera viewpoint.

MSRAction3D (Li et al., 2010; Wang et al., 2012b) is a full-body single-view dataset containing motion sequences of 10 users, performing 20 different actions in front of a Kinect v1 sensor. Each subject performs each action 2 or 3 times, with a total of 567 depth map sequences. Actions in this dataset are quite varied, for example: arm waves, hammer motions, catches, punches, symbol drawings, kicks, tennis swings, golf swings, and jogging. Although in (Li et al., 2010) the actions are

split into 3 subsets for evaluation, this work considers all the actions at once, which is a more difficult scenario.

The depth data from all datasets are first background-subtracted (background frames are given) and then normalized and re-sized using bicubic interpolation to 224×224 pixels as shown in Figure 7·1. For BodyLogin, this causes some clear geometric distortions which can be seen.

Discussion of Results

In authentication, the ℓ_2 distance is used between the features of gesture sequences (flattened vectors of length $4096 \times T, T = 50$). Here, EER is again used as a measure of authentication performance (see Section 3.1.2).

In all the experiments, the deep network approach is benchmarked against re-implemented depth- silhouette covariance features as proposed in Section 6.3 and in the paper (Wu et al., 2015). This benchmark method is not based on convolutional neural networks.

User Identification: The system attempts to identify a user across a whole pool of possible gestures. Performance is tested both when a gesture has been seen by the system and also when it has not. The latter case evaluates how well the learned model *generalizes* to gestures that have not been part of the training set. If it performs well, our model would have, in effect, learned a specific "style" with which a user performs gestures, not just the specific gestures a user performs.

Table 7.1: User identification results for BodyLogin and HandLogin.

Dataset	*Scenario*	**User Identification CCE (%)**					
		Weighted Convnets (w_s, w_t)					Baseline
	Training / Testing Gestures	$\longleftarrow$ Spatial				Temporal $\longrightarrow$	(Wu et al., 2015)
		$(1,0)$	$(\frac{2}{3},\frac{1}{3})$	$(\frac{1}{2},\frac{1}{2})$	$(\frac{1}{3},\frac{2}{3})$	$(0,1)$	
HandLogin (21 users, 4 gestures)	1. All / All	0.24%	0.24%	**0.24%**	0.71%	4.05%	6.43%
	2. All but Compass / Compass	**2.38%**	2.86%	4.76%	8.57%	36.19%	82.38%
	3. All but Piano / Piano	1.91%	**0.48%**	1.43%	1.91%	12.86%	68.10%
	4. All but Push / Push	**44.29%**	49.05%	54.29%	67.62%	77.14%	79.52%
	5. All but Fist / Fist	16.67%	**15.71%**	17.14%	20.00%	31.43%	72.38%
BodyLogin (40 users, 5 gestures)	1. All / All	0.05%	0.05%	0.05%	**0.05%**	5.01%	1.15%
	2. All but S / S	**0.75%**	1.00%	1.25%	1.75%	16.75%	75.75%
	3. All but Left-Right / Left-Right	**0.88%**	1.25%	1.50%	1.88%	11.50%	80.88%
	4. All but 2-Hand Arch / 2-Hand Arch	0.13%	0.13%	**0.13%**	0.38%	6.25%	74.50%
	5. All but Balancing / Balancing	**9.26%**	10.01%	13.27%	19.52%	45.06%	77.97%
	6. All but User Defined / User Defined	**5.28%**	5.53%	6.16%	8.54%	22.49%	71.61%

Table 7.2: User identification on MSRAction3D. (Kviatkovsky et al., 2015) performs user identification on skeletal pose-estimates derived from depth-maps.

Dataset	**User Identification CCE (%)**				
	Weighted Convnets (w_s, w_t)			Baselines	
	$\longleftarrow$ Spatial		Temporal $\longrightarrow$	(Wu et al., 2015)	(Kviatkovsky et al., 2015)
	$(1,0)$	$(\frac{1}{2},\frac{1}{2})$	$(0,1)$		
MSRAction3D	0.0%	**0.0%**	0.53%	13.6%	7.0%

Results for both the BodyLogin and Handlogin datasets are shown in Table 7.1. The first row of this table ("All / All") refers to a scenario when the network has been trained with samples from all gestures. In this row, the dataset is split into one half for training and the other half for testing, where each half contains samples from all gestures. The remaining rows in the table are for scenarios when the network has been trained on some gestures while tested on a different unseen gesture. For example, for "All but Fist / Fist" the network has been trained on "Compass," "Piano" and "Push" but tested on "Fist." In Table 7.2, the results are reported for user identification on MSRAction3D dataset. Here, only one sample of each action is used for training, and remaining 1-2 samples are used for testing. This is the same as the row ("All / All") in Table 7.1, where training is with samples from all gestures. In addition to our silhouette covariance benchmark from Section 6.3 and (Wu et al., 2015), this work also compares to a method that uses skeletal joint estimates and a distance metric based on skeletal coordinates (Kviatkovsky et al., 2015).

Suppression of Dynamics in User Identification: In order to understand the impact of dynamics in our deep network representation, empirically, the effect of "removing" it is studied. Although a similar study was done in Section 5.2 and paper (Wu et al., 2014b), that was based on pose-estimated skeletons. Our study is based on depth maps. This work considers *both* the input to the temporal stream network, as well as the input to the spatial stream network as containing full dynamic information. To suppress the impact of dynamics, the temporal network is completely removed, and only the first 3 depth-map frames are used (out of typically hundreds of frames, spanning the time duration of less than a tenth of a second) as input to the spatial stream network. In Table 7.3, the empirical performance of dynamics suppression is assessed for the two-stream approach as well as for the approach

from Section 6.3 and paper (Wu et al., 2015) which has been reimplemented for this experiment.

Table 7.3: Results for the suppression of dynamics in user identification: only first 3 frames of each depth-map sequence are used for training and testing, and the temporal stream is disabled ($w_s = 1, w_t = 0$).

Dataset	*Scenario*	**User Ident. CCE (%)**	
	Data Used	Spatial	(Wu et al., 2015)
HandLogin	All frames	**0.24%**	6.43%
	No dynamics	**1.90%**	9.29%
BodyLogin	All frames	**0.05%**	1.15%
	No dynamics	**1.00%**	32.60%

Table 7.4: User authentication results for BodyLogin and HandLogin.

Dataset	*Scenario*	**User Authentication EER (%)**					
		Weighted Convnets (w_s, w_t)					Baseline
	Users	$\longleftarrow$ Spatial				Temporal $\longrightarrow$	(Wu et al., 2015)
		$(1, 0)$	$(\frac{2}{3}, \frac{1}{3})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{3}, \frac{2}{3})$	$(0, 1)$	
HandLogin	Leave 5 persons out	2.52%	**2.20%**	2.71%	4.09%	6.50%	11.45%
BodyLogin	Leave 10 persons out	2.76%	2.45%	**1.99%**	3.07%	8.29%	3.46%

User Authentication: Here, the system attempts to verify a user's query gesture and claimed identity against a pool of known gestures (all gestures of the claimed identity). As it is impractical to train a deep network for each user, this work instead trains an identification network first and uses it as a *feature extractor* for authentication (see Section 7.3). In these experiments, one-fourth of the user pool is "leave-out" for testing, and the remaining three-fourths are used for training an identification network (for feature extraction). For BodyLogin, this is leave-10-persons-out and for HandLogin this is leave-5-persons-out cross-validation. In the benchmark authentication method, covariance features from the test samples are used. The average EER across 4 "leave-out" folds for authentication is shown in Table 7.4 for Bodylogin and HandLogin.

Gesture Recognition: Here, the system attempts to recognize the gesture type performed across a pool of users. While in user identification the system is trying to learn the user-identity irrespective of which gestures the user performs, in gesture recognition the system is trying to learn the gesture irrespective of the users who perform them. Thus, similar to how gestures are "leave-out" in user identification, users are "leave-out" in gesture recognition. Specifically, half of the user pool is "leave-out" for testing, and the remaining half is used for training a gesture recognition network. For MSRAction3D, the common cross-validation approach of leave-5-persons-out is followed as done in (Oreifej and Liu, 2013), and in BodyLogin[1] and Handlogin, leave-20-persons-out, and leave-10-persons-out (half of each dataset population), is performed respectively. The results for gesture recognition are reported in Table 7.5.

[1]Of the 5 gesture classes in BodyLogin, 4 gesture classes are shared across users, and 1 is not, being user-defined. This means that in leave-persons-out gesture recognition, the fifth gesture class will not have samples of its gesture type in training. As a result, the fifth gesture class is expected to act as a "reject"/"not gestures 1 - 4" category for gesture recognition.

Table 7.5: Gesture recognition results. For each dataset, leave-$(N/2)$-persons-out cross-validation is performed, where N is equal to the total number of users in the dataset.

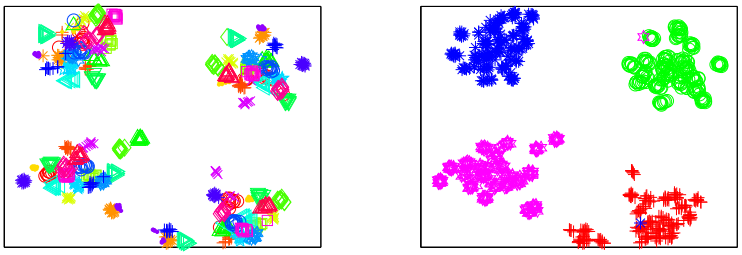
Dataset	**Gesture Recognition CCE (%)**			
	Weighted Convnets (w_s, w_t)			Baseline
	$\longleftarrow$ Spatial Temporal $\longrightarrow$			Wu (Wu et al., 2015)
	$(1, 0)$	$(\frac{1}{2}, \frac{1}{2})$	$(0, 1)$	
HandLogin	15.00%	6.82%	10.91%	**0.91%**
BodyLogin	21.10%	**15.09%**	20.35%	15.44%
MSRAction3D	44.36%	36.00%	40.36%	**25.45%**

Discussion: The above results demonstrate a significant decrease in error when using deep networks compared to benchmark methods in user identification (all 3 datasets) and authentication (HandLogin and BodyLogin).[2]. This decrease is most striking in identification, when gestures are tested that have not been used in training the network. In stark contrast to the CNN features proposed in our work, the covariance features proposed in (Wu et al., 2015) do not generalize well *across* gestures, i.e., when gestures that are not part of the training set appear in the test set. This can be seen most clearly by examining the CCE values for the "Compass" gesture in Table 7.1. The CCE for covariance features is as high as 82.38% while it is only 2.38% for our CNN features.

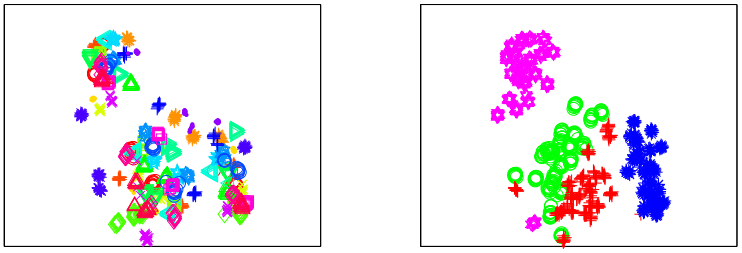
This cross-gesture generalization capacity of CNNs is also observed in the t-SNE embeddings (Van der Maaten and Hinton, 2008) of the "full7" layer outputs (Figures 7·3-7·5). In the two-dimensional t-SNE plots of the "full7" layer outputs of CNNs, users tend to cluster together whereas gesture types are mixed within each cluster. However, in the corresponding two-dimensional t-SNE plots of the covariance features, gesture types tend to cluster together with users mixed within each cluster. Figure 7·3(a) shows the feature embedding for our baseline, which fa-

[2]Due to the general lack of per-user samples in MSRAction3D (as it is a gesture-centric dataset), results are not reported for authentication and leave-gesture-out for identification

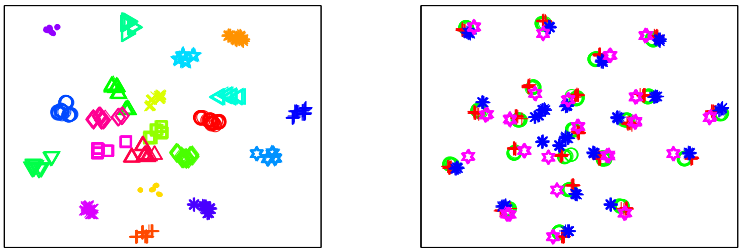
vors clustering by gesture type. Figures 7·3(b)-(d) show the feature embeddings for our convolutional networks. In 7·3(b), the pre-trained embedding from ImageNet tends to favor clustering points by gesture type. After fine-tuning for identification in 7·3(c), clustering by user identity can be seen. Fine tuning for gesture recognition in 7·3(d) causes even more compact clustering by gesture type.



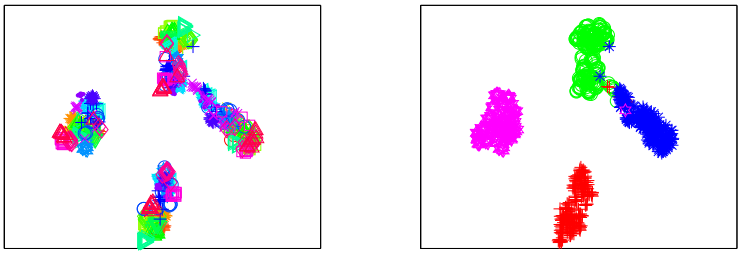
(a) HandLogin silhouette-covariance features (Wu et al., 2015)



(b) HandLogin pre-trained "full7" features (no fine tuning)

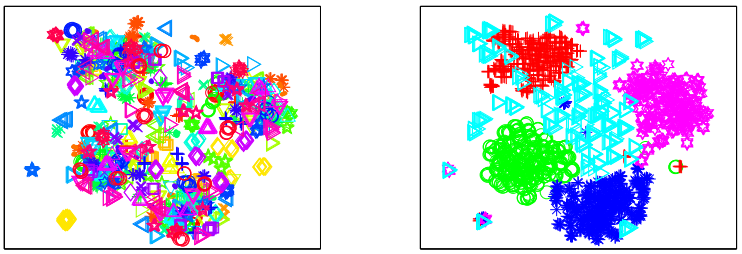


(c) HandLogin *user identification* fine-tuned "full7" features

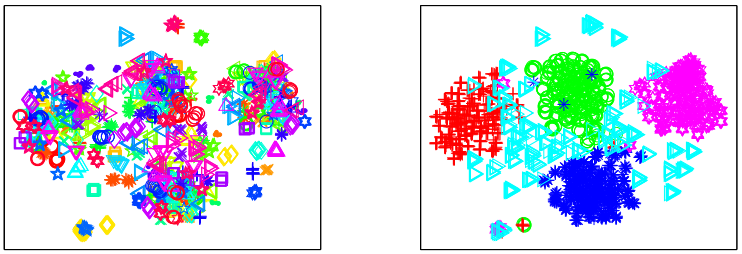


(d) HandLogin *gesture recognition* fine-tuned "full7" features

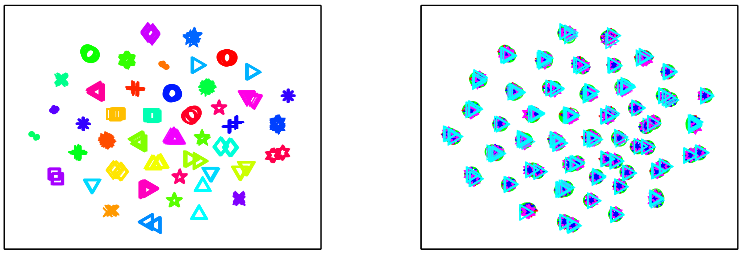
Figure 7·3: 2-D t-SNE embeddings of features for the HandLogin dataset. Left-column plots are color-coded by user, whereas those in the right column are color-coded by gesture type. A single marker represents a single gesture sequence. These figures show the t-SNE embeddings of the last fully-connected layer's output from our convolutional networks, and those from our baseline, silhoutte-covariance features.



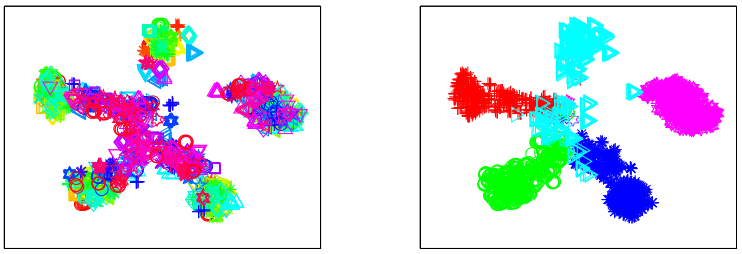
(a) BodyLogin silhouette-covariance features (Wu et al., 2015)



(b) BodyLogin pre-trained "full7" features (no fine tuning)

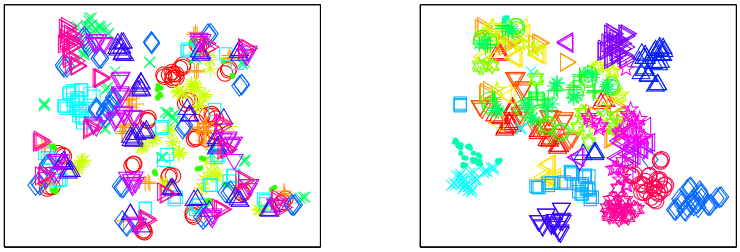


(c) BodyLogin *user identification* fine-tuned "full7" features

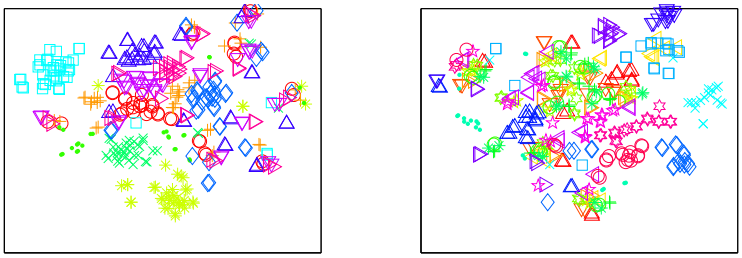


(d) BodyLogin *gesture recognition* fine-tuned "full7" features

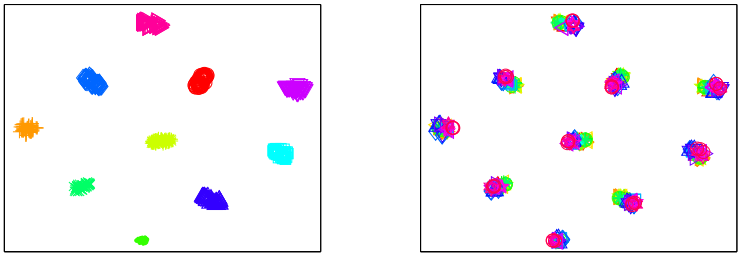
Figure 7·4: 2-D t-SNE embeddings of features for the BodyLogin dataset. For additional information, please see Figure 7·3. The cyan marker denotes user-defined gestures where any motion is allowed; it is not expected to cluster tightly.



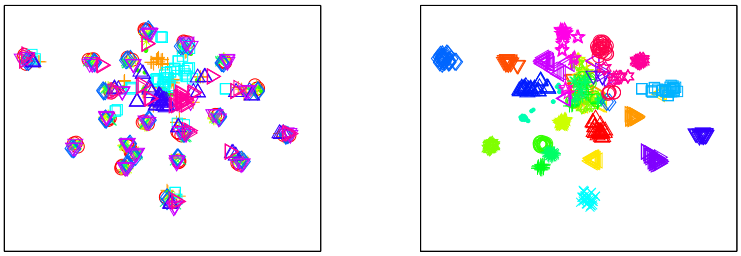
(a) MSRAction3D silhouette-covariance features (Wu et al., 2015)



(b) MSRAction3D pre-trained "full7" features (no fine tuning)



(c) MSRAction3D *user identification* fine-tuned "full7" features



(d) MSRAction3D *gesture recognition* fine-tuned "full7" features

Figure 7·5: 2-D t-SNE embeddings of features for the MSRAction3D dataset. For additional information, please see Figure 7·3.

There are, however, cases where our network does not generalize well across gestures, e.g., the "Push" gesture. This lower performance may occur because the trained gestures are significantly different in form and dynamics from the other gestures. The "Push" gesture contains variations in *scale* whereas the other gestures do not. The "Fist" gesture contains motion that completely occludes the shape of the hand, which is not in the other gestures. The "Balancing" gesture includes leg movements, not so for other gestures. For the most part, this type of result is to be expected. It will always be difficult to generalize to a completely unknown gesture that has little-to-no shared components with trained gestures.

For identification on MSRAction3D, there is a 0% classification error. Though seemingly surprising, this result might be attributed to the dataset collection procedure. In MSRAction3D, gesture samples from a user are extracted by partitioning one long continuous video into multiple sample parts. Though not an issue for gesture recognition (as the same user will never be in *both* training and test sets due to "leave-persons-out" testing), this can result in biases for user recognition. This bias stems from almost identical, partially-shared body postures across samples, which the deep network learns very well. The aforementioned issue is avoided in BodyLogin and HandLogin, as there is a "reset" procedure between samples, since samples are **not** recorded from one long continous sequence (users leave and re-enter the room between samples).

For authentication, the differences are far less dramatic, but CNN features still yield a decent decrease in EER. In both scenarios, the smaller the value, the better the performance (small EER and CCE is desired).

Across all our results, the temporal stream is complementary to the spatial stream for user identification, authentication, and even gesture recognition. That is, having a temporal stream weight $w_t \neq 0$, will not degrade performance. The

only exception to this, is when *information* is not seen in the training phase such as in leave-gesture-out results for user identification in Table 7.1. The reduced performance due to the inclusion of the temporal stream is not entirely surprising, as there are body/hand motions in testing that have not been seen in training (unseen optical flow vectors). As a result, this ends up generalizing poorly, whereas the static poses from the spatial network still fare quite well. Across all experimental results, a simplistic weighted average of $(\frac{1}{2}, \frac{1}{2})$ is perhaps the best option.

Our experiments involving dynamics suppression in user identification (Table 7.3) confirm that motion plays a crucial role; it can reduce the mis-identification rate from 1 error in 100 attempts to 1 error in 2,000 attempts (for BodyLogin). This conclusion is consistent across both methods evaluated.

In gesture recognition, our deep learning approach slightly outperforms the non-CNN approach on BodyLogin, but is outperformed on the other datasets. This could be due to the size of the dataset. Notably, BodyLogin is our largest dataset with the most samples (≈4000 gesture sequences, ≈150 frames each), and can beat our baseline. This is larger than both HandLogin (≈840 gesture sequences, ≈150 frames each) and MSRAction3D (≈600 gesture sequences, ≈35 frames each) combined. As the CNN approach outperforms the baseline in all other experiments, this perhaps suggests that with fewer samples it is easier to discriminate between users, than it is to discriminate between gestures. Overall, we believe that on larger datasets such as BodyLogin, deep learning will likely outperform the baseline.

7.5 Concluding Remarks

This chapter investigated the use of two-stream convolutional networks for learning user-specific gesture "styles" across gestures. Previous chapters as well as most of the state-of-the-art work assume a single gesture password per user and per-

form poorly when gesture types that are not encountered in the training set appear during testing. The proposed CNN-based features are able to effectively generalize across multiple types of gestures performed by the same user by implicitly learning a representation that depends only on the intrinsic "style" of each user as opposed to the specific gesture as demonstrated across multiple datasets.

A key practical outcome of this approach is that for authentication and identification there is no need to retrain a CNN as long as users do not use dramatically different gestures. With some degradation in performance, a similar new gesture can still be used for convenience.

Chapter 8

Conclusions

In this chapter, we summarize contributions of this dissertation, and discuss potential future research directions.

8.1 Contributions

This thesis explored the ins and outs of using body- and hand-based "gesture passwords" as a biometric. It was motivated by the desire to have a convenient, renewable, spoof-resistant biometric that could leverage recent advances in depth-sensing cameras.

To evaluate the effectiveness of this modality, a framework based on both silhouette shape information and skeletal pose information was proposed.

To understand the importance of gesture dynamics, a technique was proposed to decompose any gesture into posture, build, and dynamics components. Evaluation of this decomposition revealed that dynamic information is invaluable to authentication performance, ranking highest among all 3 components if used individually.

To test the robustness of the modality, possible real-life "threat" scenarios were proposed and evaluated. Specifically, these scenarios investigated what happens to biometric performance when the gesture sample becomes degraded (user having personal effects, or re-using the system after a period of time), or a fake "spoof" gesture sample is submitted instead. This study ended up showing that the frame-

work is rather resistant to spoofing, and the greatest challenge is from the effect of time (how easily a user can recall his/her own motion after a week or two). It also showed that skeletal features tended to be more robust than silhouette-based features against all degradations.

Finally, the performance gain that could be realized by including additional cameras was explored as well.

This dissertation also explored an alternative approach to gesture biometrics based on learning a user-specific gesture "style." To this end, a framework leveraging deep convolutional neural networks was proposed. Drops in performance when unseen gesture types and users were given as input to the network were also investigated. Results indicate that the performance of such networks is rather resilient to variations in gestures that are similar in style. Finally, the user-specific "style" representation was compared to the corresponding representation for gesture recognition to reveal further insights into the inner workings of the network.

To support all these experiments, 4 datasets were collected, processed and evaluated, and eventually made available on-line.

Overall, the key contributions of this dissertation can be summarized as follows:

- The development of a novel framework for authentication or identification of body- and hand- gestures (Chapter 3, 5, 6). This contribution has been reported in the following works: (Wu et al., 2013; Wu et al., 2014a; Wu et al., 2015).

- An extensive study of various strengths and weaknesses of body gestures as a biometric (Chapter 5). This contribution has been reported in the following works: (Wu et al., 2014a; Wu et al., 2014b; Wu et al., 2014c).

- An exploratory analysis of learning a gesture "style" across a bank of known gestures and a comparison to gesture recognition (Chapter 7). This contribution has been reported in the following works: (Wu et al., 2016).

- The creation of 4 novel datasets for identification and authentication, and making them available on-line (Chapter 4). This contribution has been reported in the following works: (Wu et al., 2014a; Wu et al., 2014b; Wu et al., 2014c; Wu et al., 2015).

8.2 Future Work

Below, a couple of interesting directions that can be pursued by using the work presented in this dissertation are briefly discussed.

Recurrent Neural Networks: Long Short Term Memory

Although dynamic information is leveraged in the two-stream convolutional neural network approach, temporal relationships are not explicitly modeled in the architecture.

It may therefore be desirable to have an architecture that is capable of detecting and learning informative short and long-term temporal relationships in sequential data. Long short term memory (LSTM) type recurrent neural networks are capable of learning such complex temporal relationships. In action recognition, these models have combined with convolutional neural networks to great success (Donahue et al., 2015). For user recognition with gestures, it would be interesting to see if modeling these temporal relationships can yield an improved biometric performance.

Learning Similarity Metrics via Siamese Networks

Learning a similarity metric for authentication/verification is another way to evaluate pairs of samples. Rather than using a fixed or hand-crafted similarity metric (e.g., Euclidean distance, cosine distance, DTW distance, and so forth), a metric can instead be learned directly from the data. This can be done by evaluating a large set of sample pairs that have matching and non-matching labels. Here, the true label of the user is not necessary.

A pair of convolutional neural networks whose weights are shared can be used in a siamese architecture (Chopra et al., 2005) to learn this metric. The output of both these networks can be combined through a contrastive loss function (Hadsell et al., 2006) which emulates the function of a mechanical spring between pairs of points. This loss function encourages a mapping that pulls similar pairs of samples closer together in space, while pushing away dissimilar samples. A specific "spring tension" is defined for a given contrastive loss function. This method has been deployed to great success in face recognition in (Taigman et al., 2014) and (Sun et al., 2014).

Another novel extension would be to use *gesture*-based convolutional networks, such as the two-stream networks in a siamese architecture for verification. The challenge, however, would be training such networks effectively over small datasets such as those currently available for gesture biometrics. In contrast to gesture biometric datasets which are small, face recognition datasets easily have hundreds to thousands of unique users and their corresponding samples to train from. A possible solution to this lack of samples would be to pre-train the siamese network from a relevant related problem *à la* domain adaption.

References

(2003). CMU Motion Capture Database. http://mocap.cs.cmu.edu/.

(2014). Kinect for Windows. http://www.microsoft.com/en-us/kinectforwindows/.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2006). Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421.

Aumi, M. and Kratz, S. (2014). Airauth: evaluating in-air hand gestures for authentication. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, pages 309–318. ACM.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546. IEEE.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, pages 647–655.

Farella, E., O'Modhrain, S., Benini, L., and Riccó, B. (2006). Gesture signature for ambient intelligence applications: a feasibility study. In *Pervasive Computing*, pages 288–304. Springer.

Faundez-Zanuy, M. (2007). On-line signature recognition based on vq-dtw. *Pattern Recognition*, 40(3):981–992.

Fong, S., Zhuang, Y., and Fister, I. (2013). A biometric authentication model using hand gesture images. *Biomedical Engineering Online*, 12(1):111.

Fothergill, S., Mentis, H. M., Kohli, P., and Nowozin, S. (2012). Instructing people for training gestural interactive systems. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM.

Gafurov, D., Snekkenes, E., and Bours, P. (2007). Spoof attacks on gait authentication system. *IEEE Transactions on Information Forensics and Security*, 2(3):491–502.

Guo, K., Ishwar, P., and Konrad, J. (2013). Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494.

Gupta, A., Arora, A., and Juneja, B. (2013). Tag: A two-level framework for user authentication through hand gestures. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 503–509. IEEE.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE.

Han, J. and Bhanu, B. (2006). Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322.

Han, L., Wu, X., Liang, W., Hou, G., and Jia, Y. (2010). Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing*, 28(5):836–849.

Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2466–2472. AAAI Press.

Jain, A. K., Ross, A. A. A., and Nandakumar, K. (2011). *Introduction to Biometrics*. Springer.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *MM'14: Proceedings of the 2014 ACM Conference on Multimedia*, pages 675–678. ACM.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., and Winnemoeller, H. (2014). Recognizing image style. In *Proceedings of the British Machine Vision Conference. BMVA Press*.

Keogh, E. (2002). Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pages 406–417.

Kholmatov, A. and Yanikoglu, B. (2005). Identity authentication using improved online signature verification method. *Pattern Recognition Letters*, 26(15):2400–2408.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1097–1105.

Kurakin, A., Zhang, Z., and Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1975–1979. IEEE.

Kviatkovsky, I., Shimshoni, I., and Rivlin, E. (2015). Person identification from action styles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 84–92.

Lai, K., Konrad, J., and Ishwar, P. (2012). Towards gesture-based user authentication. In *Proceedings of the IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 282 –287.

Lei, H. and Govindaraju, V. (2005). A comparative study on the consistency of features in on-line signature verification. *Pattern Recognition Letters*, 26(15):2483–2489.

Li, W., Zhang, Z., and Liu, Z. (2010). Action recognition based on a bag of 3d points. In *2010 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–14.

Little, J. and Boyd, J. (1998). Recognizing people by their gait: the shape of motion. *Videre: Journal of Computer Vision Research*, 1(2):1–32.

Liu, C. (2009). Beyond pixels: exploring new representations and applications for motion analysis. Unpublished doctoral dissertation, Massachusetts Institute of Technology.

Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. (2009). uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675.

Lv, F. and Nevatia, R. (2006). Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Computer Vision–ECCV 2006*, volume 3951-3954 of Lecture Notes in Computer Science, pages 359–372. Springer.

Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. (2007). Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn.

Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley MHAD: A comprehensive multimodal human action database. *IEEE Workshop on Applications of Computer Vision*, 0:53–60.

Ohn-Bar, E. and Trivedi, M. M. (2013). Joint angles similarities and HOG2 for action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 465–470. IEEE.

Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723.

Patel, S., Pierce, J., and Abowd, G. (2004). A gesture-based authentication scheme for untrusted public terminals. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pages 157–160. ACM.

Ratanamahatana, C. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Ren, Z., Meng, J., Yuan, J., and Zhang, Z. (2011a). Robust hand gesture recognition with kinect sensor. In *MM'11: Proceedings of the 2011 ACM Conference on Multimedia*, pages 759–760. ACM.

Ren, Z., Yuan, J., and Zhang, Z. (2011b). Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *MM'11 : Proceedings of the 2011 ACM Multimedia Conference*, pages 1093–1096. ACM.

Reyes, M., Dominguez, G., and Escalera, S. (2011). Featureweighting in dynamic timewarping for gesture recognition in depth data. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1182–1188. IEEE.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Sae-Bae, N., Ahmed, K., Isbister, K., and Memon, N. (2012). Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 977–986. ACM.

Scott, M. J., Niranjan, M., and Prager, R. W. (1998). Realisable classifiers: Improving operating performance on variable cost problems. In *Proceedings of the British Machine Conference*, pages 304–315.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. (2013). Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840.

Sigal, L., Balan, A. O., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27.

Simonyan, K. and Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 568–576.

Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Suarez, J. and Murphy, R. (2012). Hand gesture recognition with depth images: A review. In *2012 IEEE RO-MAN*, pages 411–417.

Sun, Y., Chen, Y., Wang, X., and Tang, X. (2014). Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1988–1996.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708.

Tuzel, O., Porikli, F., and Meer, P. (2008). Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1713–1727.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

Wang, J., Liu, Z., Chorowski, J., Chen, Z., and Wu, Y. (2012a). Robust 3d action recognition with random occupancy patterns. In *Computer Vision–ECCV 2012*, volume 7573-7574 of Lecture Notes in Computer Science, pages 872–885. Springer.

Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012b). Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297.

Wang, L., Xiong, Y., Wang, Z., and Qiao, Y. (2015). Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

Wu, J., Christianson, J., Konrad, J., and Ishwar, P. (2015). Leveraging shape and depth in user authentication from in-air hand gestures. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3195–3199. IEEE.

Wu, J., Ishwar, P., and Konrad, J. (2014a). Silhouettes versus skeletons in gesture-based authentication with kinect. In *Proceedings of the IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*.

Wu, J., Ishwar, P., and Konrad, J. (2014b). The value of posture, build and dynamics in gesture-based user authentication. In *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*.

Wu, J., Ishwar, P., and Konrad, J. (2016). Two-stream cnns for gesture-based verification and identification: Learning user style. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Wu, J., Konrad, J., and Ishwar, P. (2013). Dynamic time warping for gesture-based user identification and authentication with kinect. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2371–2375.

Wu, J., Konrad, J., and Ishwar, P. (2014c). The value of multiple viewpoints in gesture-based user authentication. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 90–97.

Xia, L., Chen, C.-C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 20–27. IEEE.

Zhang, R., Vogler, C., and Metaxas, D. (2004). Human gait recognition. In *2004 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 18–18. IEEE.

CURRICULUM VITAE

Jonathan Wu

jonwu@bu.edu

U.S. Citizen

Education

Boston University **Boston, MA**
Ph.D Candidate in Electrical and Computer Engineering 2011 - Present
Expected Graduation: Spring 2016

- Proposed Thesis: *Gesture Passwords: Concepts, Methods, and Challenges*
- Advsiors: Janusz Konrad and Prakash Ishwar

Carnegie Mellon University **Pittsburgh, PA**
M.S. in Electrical and Computer Engineering 2011
B.S. in Electrical and Computer Engineering 2007 - 2011

Professional Experience

Boston University, Visual Information Processing Lab **Boston, MA**
Graduate Research Assistant Fall 2011 - Present

- Extensively studied the feasibility of using gestures as a biometric using machine learning and computer vision algorithms such as deep learning
- Mentored several undergraduates and high schoolers on various computer vision projects that have been successfully deployed or have resulted in a peer-reviewed publication

IBM Smarter Cities Challenge Boston **Boston, MA**
Student Collaborator Summer 2012

- Collaborated with a Boston University, City of Boston, and IBM team to plan and prototype smarter technical solutions for the city pertaining to traffic and green policy

NASA Jet Propulsion Laboratory **Pasadena, CA**
Summer Intern, Mobility and Robotic Systems Summer 2011

- Developed tracking algorithms optimized to run on a miniature unmanned aerial vehicle
- Benchmarked and modified template matching algorithms through ROS for tracking

The MITRE Corporation **Bedford, MA**
Systems Intern, Comp. Vis. & Parallel Computing Summer 2010
Software Intern, Software Engineering Summers 2008, 2009

- Developed an image registration system that merges a full resolution narrow field of view onto a wide one
- Wrote, constructed, and maintained a pan-tilt servo system that performs synchronized camera capture and long-range scene scanning
- Built a distributed multi-node GPS simulator using MATLAB to generate various GPS metrics
- Designed web apps for weather data using MySQL, JSP and JavaScript

Collaborative Innovation Center **Pittsburgh, PA**
Undergraduate Researcher, Cylab Biometrics Lab Fall 2008 - Spring 2010

- Ported iris segmentation from MATLAB to C++ using OpenCV
- Built a system to perform unsupervised facial clustering and subsequent recognition on various videos

---

Technical Skills and Honors

Programming: C, C++, C#, Java, JavaScript, MATLAB, LaTeX, Vim

Honorable Mention CISE BU Scholars Day 2013
Carnegie Institute of Technology Dean's List Spring 2009, Fall 2010
AP Scholar with Distinction Award 2007

Academic Experience

Graduate Teaching Fellowships (Boston University) **Boston, MA**

- Introduction to Embedded Systems Spring 2012
- Introduction to Software Engineering Fall 2011

Professional Service

- Reviewer, IJCAI 2016
- Reviewer, Pattern Recognition Letters (PRL) 2015
- Reviewer, WIFS 2013

Publications

2016

- **J. Wu**, P. Ishwar, and J. Konrad, "Two-Stream CNNs for Authentication and Identification: Learning User Gesture Style," in *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) Biometrics Workshop, 2016. **(under review)**
- **J. Wu**, P. Ishwar, and J. Konrad, "Towards Learning Gesture Styles from Depth Maps using Deep Learning for Authentication and Identification," in *IEEE International Conference on Image Processing* (ICIP), 2016. **(under review)**
- J. Chen, **J. Wu**, K. Richter, J. Konrad, and P. Ishwar, "Estimating Head Pose Orientation Using Extremely Low Resolution Images," in *IEEE Southwest Symposium on Image Analysis and Interpretation* (SSIAI), 2016.

2015

- **J. Wu**, J. Christianson, J. Konrad, and P. Ishwar, "Leveraging Shape and Depth in User Authentication from In-Air Hand Gestures," in *IEEE International Conference on Image Processing* (ICIP), 2015.
- J. Dai, B. Saghafi, **J. Wu**, J. Konrad, and P. Ishwar, "Towards Privacy- Preserving Recognition of Human Activities," in *IEEE International Conference on Image Processing* (ICIP), 2015.
- J. Dai, **J. Wu**, B. Saghafi, J. Konrad, and P. Ishwar, "Towards Privacy- Preserving Activity Recognition Using Extremely Low Temporal and Spatial Resolution Cameras," in *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) Analysis and Modeling of Faces and Gestures (AMFG) Workshop, 2015.

2014

- **J. Wu**, P. Ishwar, and J. Konrad, "The Value of Posture, Build and Dynamics in Gesture-based User Authentication," in *IEEE International Joint Conference on Biometrics* (IJCB), 2014.
- **J. Wu**, J. Konrad, and P. Ishwar, "The Value of Multiple Viewpoints in Gesture-based User Authentication," in *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) Biometrics Workshop, 2014
- **J. Wu**, P. Ishwar, and J. Konrad, "Silhouettes Versus Skeletons in Gesture-based Authentication with Kinect," in *IEEE Conference on Advanced Video and Signal-Based Surveillance* (AVSS), 2014.

2013

- **J. Wu**, J. Konrad, and P. Ishwar, "Dynamic Time Warping for Gesture-based User Identification and Authentication using Kinect," in *IEEE International Conference on Acoustics Speech Signal Processing* (ICASSP), 2013.