# The Enhancement of Low-Level Classifications for Ambient Assisted Living

Rachel GOSHORN [a], Deborah GOSHORN [b] and Mathias KÖLSCH [c]
[a] *Systems Engineering Department, Naval Postgraduate School, U.S.A.*
[b] *Computer Science and Engineering Department, University of California, San Diego, U.S.A.*
and [c] *Computer Science Department and MOVES Institute, Naval Postgraduate School, U.S.A.*

**Abstract.** Assisted living means providing the assisted with custom services, specific to their needs and capabilities. Computer monitoring can supply some of these services, be it through attached devices or “smart” environments. In this paper, we describe an ambient system that we have built to facilitate non-verbal interaction that is not bound to the traditional input means of a keyboard and mouse. We investigated the reliability of hand gesture behavior recognition, from which computer commands for AAL communications are interpreted. Our findings show that hand gesture behavioral analysis reduces false classifications while at the same time more than doubling the available vocabulary. These results will influence the design of gestural and multimodal user interfaces.

**Keywords.** human-computer interaction, hand postures, hand gestures, user interface, ambient intelligence, posture recognition, gesture recognition, smart environments, computer vision, human body tracking, hand gesture behaviors, Ambient Assited Living (AAL), interpreted computer commands

## 1. Introduction

In a variety of situations gestural communication is either preferable over verbal communication or advantageous if used in a multimodal combination with voice. For example, noisy environments might render voice recognition unreliable. People with speech impairments might have difficulties communicating verbally. And some intentions are best communicated multimodally, best illustrated in Bolt’s “Put That There” elaboration [1]. In addition, people may be bedridden or elderly living alone at home, and may need assistance in communicating and controlling various devices.

Through behavior analysis of human hand movements over time, observed through vision sensor data, interpretation of communication commands can be carried out and communicated for human computer interaction to enable smart environments. A great need for smart environments exists with those needing assistance, such as the elderly at home, others bedridden, etc. In this, vision sensor data is providing the means to become aware of the surrounding environment, “ambient intelligence”, and through hand gestures, providing the human computer interaction, smart environments are enabled. Combining these two worlds of "ambient intelligence" and "smart environments", those needing help at home are assisted; in other words "ambient assisted living (AAL)" is provided. In AAL, people can for example be assisted in turning devices on and off, carry out phone calls (e.g. emergency calls), change the television channel, etc. Fig. 1 shows the overall systems view of AAL discussed in this paper. If these people needing assistance could communicate commands to devices, using their hands, it would remove the need for tools and remote controls, and allow for hands free communications. If remote controls were required, and for example these fell under the bed where a person is bedridden, they would need some way to communicate.

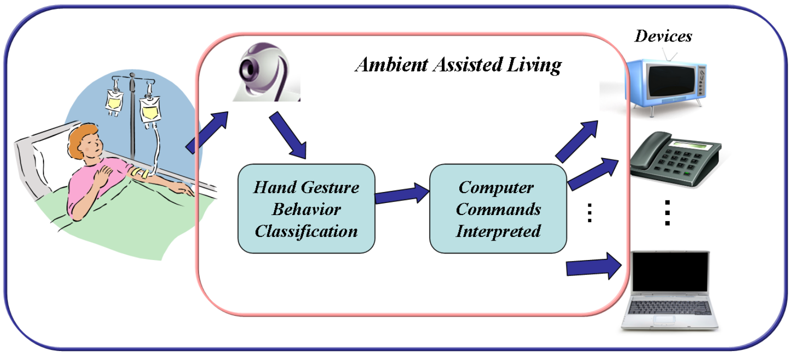
Devices
Ambient Assisted Living
Hand Gesture Behavior Classification
Computer Commands Interpreted

**Figure 1.** Overall systems view of an Ambient Assisted Living (AAL) system.

This paper will demonstrate the high level classification of hand gesture behaviors, based on sequences of hand postures over time. The hand gesture behaviors are then interpreted as various control commands for various computer devices. The levels of analysis for AAL, are shown in a pyramid process in Fig. 2.

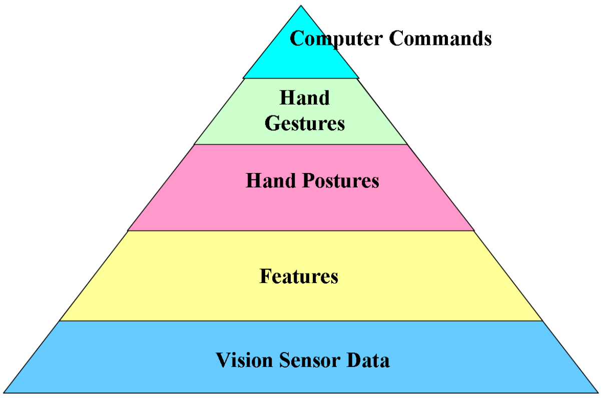

**Figure 2.** Overall pyramid process of behavior analysis and interpetation, from video to hand postures, to hand gesture behaviors to computer commands interpreted to enable Ambient Assisted Living (AAL).

In this paper, we demonstrate the use of hand gestures as a robust input for AAL. Based on an alphabet of atomic hand postures illustrated in Fig. 3, we compose a small “gesture” vocabulary composed of posture sequences. The postures are observed with ceiling-mounted cameras and recognized with the “HandVu” library. We then use a robust hand gesture behavior classification method [5,4] to distinguish the gestures. In an AAL system, these hand gesture behaviors are then interpreted as computer commands to control various devices of interest. In an experiment, we demonstrate improved recognition performance over individual hand postures, and providing additional computer commands per gesture (versus being limited in commands to the fixed set of postures), thus making the system robust for application in AAL. The overview of the AAL systems experiment focus can be seen in Fig. 4.

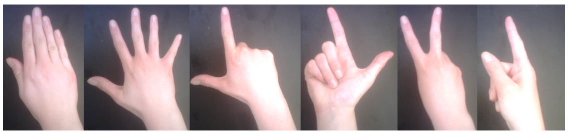

**Figure 3.** Close-up view of the six hand postures, from left to right: *closed*, *open*, *Lback*, *Lpalm*, *Victory*, *sidepoint*.

After reviewing the related work in the following section, Sec. 3 will introduce the posture recognition library, the robust hand gesture classification method for syntactic analysis is descried in Sec. 4 and the experimental design and results in Sec. 5. The last two sections cover the summary and conclusions.

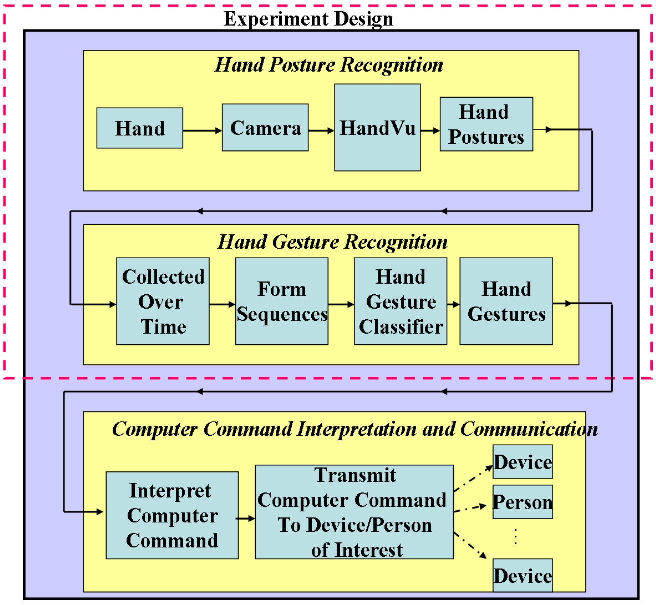

**Figure 4.** Technical Overview of Ambient Assisted Living (AAL) system, with the experiment focus highlighted.

## 2. Related Work

Smart environments have long attracted people's interests. In many ways, rooms have become more aware of its inhabitants: consider motion-activated doors, lights that turn off automatically if noone is present, and even the thermostat "senses" the environment. For active user interaction with the environment, there's the clap switch that turns on and off electricity to your favorite household appliance at the clap of a hand. However, networked systems that can continuously monitor and react to a person's state are still the dreams of researchers.

One of the earliest demonstrations of the of gestural and multimodal human-computer interaction was Bolt's 1980 "Put That There" article [1]. An early user interface implementation using temporal gestures was shown by Pausch and Williams [17] in 1990, making use of a tracked data glove. Various researchers and now also commercial products employ handheld devices instead of bare-hand gestures: The XWand [24], a stick the size of a remote control, enables natural interaction with consumer devices through gestures and speech. One can point at and gesture or speak commands to, for example, your TV or stereo. Earlier, Kohtake et al. [10] showed a similar wand-like device that enabled data transfer between consumer appliances such as a digital camera, a computer, and a printer by pointing the "InfoPoint" at it. Probably the most popular device for gestural HCI as of late is Nintendo's "Wii Remote" which makes use of sensors in its game controller to estimate position and orientation.

Computer vision as sensing technology has advantages over sensors embedded in handheld devices due to its silent, unobtrusive operation that enables unencumbered interaction. (Your hands are not likely to get lost between the sofa cushions.) There is a vast body of research on hand detection, posture recognition, hand tracking, and trajectory-based gesture analysis. Early work on utilizing computer vision to facilitate human-computer interaction in relatively unconstrained environments include work by Freeman et al. [3]. In their implementation, a variety of image observations including edges and optical flow allow distinction of hand gestures and full-body motions. Ong and Bowden [16] show methods for real-time hand detection and posture classification. 3D pointing directions as discussed in [1] can be determined with methods such as those described by Nickel et al. [15]. Recognition of a select vocabulary of the American Sign Language using temporal hand motion aspects has been demonstrated by Starner and Pentland [19]. Wachs et al.'s Gestix [22,23] combines many of these technologies to create a user interface suitable to interaction in antiseptic surgical environments where keyboards can not be used. An analysis of and recommendations for using gesture recognition for user interaction can be found in a book chapter by Turk [20].

Behavior classification has grown to be a popular area of research in the computer vision area. There are several approaches to classifying high-level behaviors of vision-based data. If taken the approach of decomposing behavioral classification into two stages: (1) first, a low-level event classifier based on features of raw video data, and (2) a high-level behavioral classifier based on sequences of events that were outputted from the first stage, then a behavior classifier can be thought of as a classifier of sequences of symbols (events). Sequential classification has been an interest in several applications such as genetic algorithms, natural language processing, speech recognition, and compilers. For computer vision applications, behavior classification has been mostly popular using state-space models, like Hidden Markov Models (HMMs). However, when certain sequences of low-level data inherently fall into meaningful behaviors, Ivanov and Bobick [7] conclude that using syntactic (grammar-based) structural approaches can outperform more statistically-based approaches such approaches as HMMs . Ivanov and Bobick [8] use a stochastic context-free grammar for behavior modeling (hand gesture and behavior modeling in car parking).

Others have designed specific deterministic finite state machines for a for classifying behaviors such as airborne surveillance scenarios [2], but are not able to handle noisy data. To fix this problem, augmented finite state machines can be used, so that noisy data can still be parsed and accepted by each finite state machine representing behavior. In [4,5], such a novel robust sequential classifier is developed and proved to classify behaviors on noisy data in various applications, such as modeling behaviors in freeway traffic, human behavioral patterns in a lab room, and signal processing patterns seen in communications channels for distinguishing types of transmitted signals. An extended version of this classifier [6] is used to classify hand gestures based on three hand postures, assuming the hand postures were classified ahead of time and were heavily mislabeled. The data was simulated with to be heavily noisy to demonstrate the classifier's robustness and ability to correct errors from a poor low-level posture classifier. This paper further improves this extended classifier (as the high-level gesture behavior classifier) as described in Sec. 4 and runs it on real data (posture labels) outputted from the real-time posture recognition classifier discussed in Sec. 3.

## 3. Hand Posture Recognition

This section introduces HandVu, a library and program to recognize a set of six hand postures in real-time in video streams. Its three main components are described in the following subsections: hand detection, 2D hand tracking, and posture recognition. HandVu's vision processing methods typically require less than 100 milliseconds combined processing time per frame. They are mostly robust to different environmental conditions such as lighting changes, color temperature and lens distortion. HandVu performs fast, accurate, and robust enough for a user interface's quality and usability. HandVu's output includes which of the six postures was recognized or, if none was recognized, an "unknown posture" identifier. This data was fed into the classification method described in This section introduces HandVu, a library and program to recognize a set of six hand postures in real-time in video streams. Its three main components are described in the following subsections: hand detection, 2D hand tracking, and posture recognition. HandVu's vision processing methods typically require less than 100 milliseconds combined processing time per frame. They are mostly robust to different environmental conditions such as lighting changes, color temperature and lens distortion. HandVu performs fast, accurate, and robust enough for a user interface's quality and usability. HandVu's output includes which of the six postures was recognized or, if none was recognized, an "unknown posture" identifier. This data was fed into the classification method described in Sec. 4.

### *3.1. Hand Detection*

HandVu's hand detection algorithm detects the hand in the *closed* posture based on appearance and color. It uses a customized method based on the Viola-Jones detection method [21] to find the hand in this posture and view-dependent configuration. This posture/view combination is advantageous because it can be distinguished rather reliably from background noise [12].

Upon detection of a hand area based on gray level texture, the area's color is compared against a user-independent histogram-based statistical model of skin color, built from a large collection of hand-segmented pictures from many imaging sources (similar to Jones and Rehg's approach [9]). If the amount of skin pixels falls below a threshold the detection is rejected. These two image cues combined reduce the amount of false detections to about a dozen per hour video.

### *3.2. Hand Tracking*

Next, the hand's motion is tracked in the video stream. To that end, the system learns the *observed* hand color in a histogram, hence adjusting to user skin-color variation, lighting differences, and camera color temperature settings. Hand tracking uses the "Flock of Features" approach [13] which calculates the optical flow for small patches and occasionally resorts to local color information as backup. This multicue integration of gray-level texture with textureless color information increases the algorithm's robustness, permitting hand tracking despite vast and rapid appearance changes. It further alleviates interdependency problems with staged-cue approaches, it improves the robustness, and increases confidence in the tracking results.

### *3.3. Posture Classification*

The algorithm's last stage attempts to recognize various predefined postures. A posture in our sense is a combination of a hand/finger configuration and a view direction, allowing for the possibility to distinguish two different views of the same finger configuration such as *Lback* and *Lpalm*. The focus of the recognition method is on reliability, not expressiveness. That is, it distinguishes a few postures reliably and does not attempt less consistent recognition of a larger number of postures. HandVu's recognition method uses a texture-based approach to fairly reliably classify image areas into seven classes, six postures and "no known hand posture." The confusion matrix of a video-based experiment is shown in Fig. 7 and described to more detail in [11].

A two-stage hierarchy achieves both accuracy and good speed performance. In the first step, a detector looks for any of the six hand postures without distinguishing between them. This is faster than executing six separate detectors because different postures' appearances share common features and can thus be eliminated in one classification. In the second step, only those areas that passed the first step successfully are investigated further. Each of the second-step detectors for the individual postures was trained on the result of the combined classifier which had already eliminated 99.999879% of image areas in a validation set  [14,12,13]

After a successful classification, the tracking stage is initialized again (the Flock of Feature locations and the observed skin color model). HandVu is largely user independent and not negatively influenced by different cameras or lenses.

Results from HandVu are sent as "Gesture Events" to the gesture classification module in a unidirectional TCP/IP stream of ASCII messages in the following format:

```
1.2 timestamp obj_id: tracked, recognized,...
```

```
..., "posture" (xpos, ypos) [scale, unused]\r\n
```

The two identifiers “tracked” and “recognized” are boolean values indicating the tracking and recognition state of HandVu.

More detailed descriptions of HandVu’s architecture [14], robust hand detection, [12] and hand tracking[13] are available elsewhere.

## 4. Hand Gesture Behavior Recognition Classifier

Sequences of hand postures, over time, compose a hand gesture behavior, which are then interpreted as a computer command (as represented in the overall system, described in Sec. 1 and Fig. 4.). This section describes the hand gesture recognition classifier theory, implementation, and cost autmation method used within the hand gesture behavior recognition classifier.

### *4.1. Hand Gesture Behavior Recognition Theory*

Sequences of events or features, over time and space, compose a behavior. The events/features are the low-level classifications; in this paper, they are the detected hand postures (as described in Sec. 3). The detected hand postures are concatenated in a temporal sequence to form a behavior, which is a hand gesture. As sequences of hand postures are detected, a method is needed to read these sequences and classify which hand gesture the sequence is most similar to.

In order to read and classify sequences, we use a syntactical grammar-based approach [5,4]. Before classifying sequences, the various hand gesture behavior structures need to be defined a priori. Each hand gesture behavior can be seen as an infinite set of sequences of hand postures of similar temporal structure; this infinite set is known as a language. Each hand gesture behavior will have different temporal structures of the hand postures. These sequence structures are defined with syntax rules, specifically regular grammars [18], where the alphabet are the various hand postures (also known as symbols); the syntax rules then define ways these postures can be combined together to form the temporal structures of interest. A set of syntax rules, defining a hand gesture behavior, is also called a grammar. The grammar is implemented through a finite state machine (FSM). In other words, the FSM reads the sequence of hand postures. If a sequence of hand postures matches a certain hand gesture behavior, its corresponding FSM will accept this sequence. Therefore, the sequence of hand postures is classified into the hand gesture behavior whose corresponding FSM accepts the sequence. Since systems are not one-hundred percent predictable and reliable, its likely that a sequence will not be accepted by any of the predefined hand gesture behaviors. This could be due to errors in the low-level classification of the hand postures, or a user error by making the hand gesture with an incorrect hand posture in the sequence of postures. Therefore, the sequence of hand postures must be classified as the hand gesture behavior to which it is most similar. In order to do so, a distance metric between a sequence of hand postures and a hand gesture behavior is defined. To continue in defining the hand gesture behavior recognition classification, preliminary definitions are shown.

Let an alphabet $\Sigma$ be the set of predefined hand postures. An example alphabet is $\Sigma = a, b, c, d, e, f$ where each letter represents a detected hand posture. More details on the hand posture symbols is shown in Sec. 5. A hand gesture behavior is then a set of syntax rules combining the elements of $\Sigma$.

If an infinite set of sequences of postures is the following $k^{th}$ language:

$$\left[L(B_k)\right] = \left[\left| ab \| aab \| \cdots \| a \cdots ab \right|\right] \tag{1}$$

then the hand gesture behavior (grammar) that generated this language is:

$$B_k = \begin{bmatrix} S \to aQ_1 \\ Q_1 \to aQ_1 \\ Q_1 \to bF \end{bmatrix} \tag{2}$$

Let a hand gesture, a temporal sequence of detected hand postures be denoted by $\mathbf{s} = s_1 s_2 \ldots s_n$, where each $s_j$ is a hand posture from $\Sigma$. If $\mathbf{s}$ matches one of the sequences in $L(B_k)$, it is the $k^{th}$ hand gesture behavior, and its corresponding FSM $M_k$ will accept this sequence. Let there be $K$ predefined hand gesture behaviors $B_1, B_2, \ldots B_K$, with $K$ corresponding finite state machines $M_1, M_2, \ldots M_K$ that implement each hand gesture behavior. The sequence $\mathbf{s}$ will be classified into the hand gesture behavior to whose corresponding FSM it is accepted. If $\mathbf{s}$ is not accepted by any $M_l$, sequence $\mathbf{s}$ will then be classified as the hand gesture behavior to which it is most similar. Therefore, as $M_l$ is parsing sequence $\mathbf{s}$, it will edit the sequence so that the $M_l$ accepts it, but with a cost per edit, and then a total cost. Therefore, the distance between a sequence $\mathbf{s}$ and a hand gesture behavior $B_l$ , denoted by $d(\mathbf{s}, B_l)$, is determined by the cost-weighted number of editions required to transform $\mathbf{s}$ into $B_l$. The possible posture symbol editions are substitution and deletion, where each edition has an a priori cost assigned. As $\mathbf{s}$ is being parsed, $M_l$, will carry out the minimum number of edits required to transform $\mathbf{s}$ into a sequence in $B_l$. In order to allow edits with an associated cost in a hand gesture behavior, the original set of syntax rules per behavior and corresponding FSM must be augmented. Let the augmented $k^{th}$ hand gesture behavior and corresponding FSM be denoted by $B_k^{'}$ and $M_k^{'}$. With the example, $B_k$, let the augmented set of syntax rules be denoted by

$$B_k^{'} = \left\{ \begin{matrix} S \to aQ_1, 0 \\ Q_1 \to aQ_1, 0 \\ Q_1 \to bF, 0 \\ S \to bQ_1, C_{S(b,a)} \\ Q_1 \to bQ_1, C_{S(b,a)} \\ Q_1 \to aF, C_{S(a,b)} \\ S \to \varepsilon Q_1, C_{D(b)} \\ Q_1 \to \varepsilon Q_1, C_{D(b)} \\ Q_1 \to \varepsilon F, C_{D(a)} \end{matrix} \right\} \tag{3}$$

Let $S(a, b)$ denote substituting the true posture $b$ for the mislabeled posture $a$, and the associated cost $C_{S(b,a)}$, and $D(a)$ denote deleting a mislabeled posture $a$ with a cost $C_{D(a)}$. The corresponding modified FSM $M_k^{'}$ is shown in Fig. 5.

In order to calculate the distance between a sequence of hand postures and a hand gesture behavior, let the possible hand postures be $\Sigma = r_1, r_2, \cdots, r_N$, where $N$ is the total number of hand postures. With this, the distance from a sequence of hand postures $\mathbf{s}$ and a hand gesture behavior $B_l$ is given by,

$$d(\mathbf{s}, B_l) = \sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} C_{S(r_i,r_j))} n_{S(r_i,r_j)} + \sum_{i=1}^{|\Sigma|} C_{D(r_i)} n_{D(r_i)} \tag{4}$$

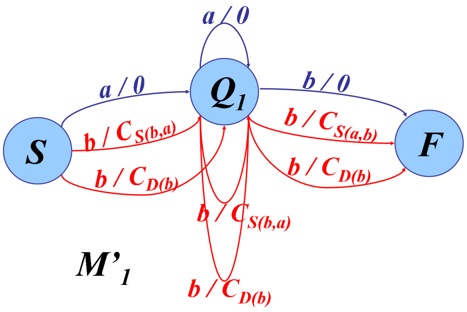

**Figure 5.** Augmented FSM $M_1^{'}$ of hand gesture behavior $B_k^{'}$. The augmented syntax rules are in red, and original syntax rules in blue with zero cost.

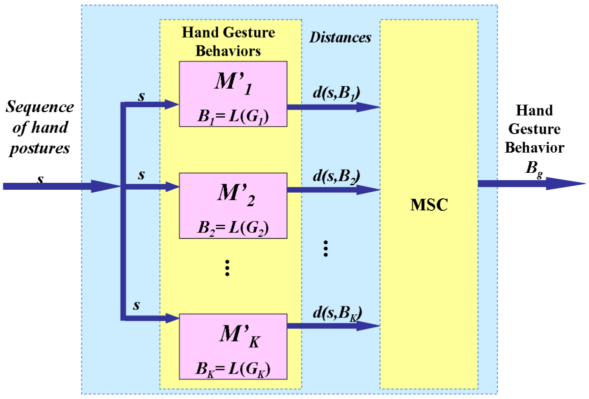

**Figure 6.** Maximum similarity classification (MSC), with the input sequences of hand postures and output the hand gesture behavior.

where $n_{S(r_i,r_j)}$ is the number of substitutions of true hand posture $r_j$ for mislabeled hand posture $r_i$, and $n_{D(r_i)}$ number of deletions of the mislabeled hand posture $r_i$.

With a distance metric between a sequence of hand postures and an a priori defined hand gesture behavior, the classification definition can be elaborated upon. Assuming $K$ hand gesture behaviors, each behavior and its associated FSM are augmented a priori so that any sequence of hand postures is accepted by each hand gesture behavior, but with a total cost. The augmented hand gesture behaviors are then denoted by $B_1^{'}, B_2^{'}, \ldots B_K^{'}$, and their $K$ corresponding augmented finite state machines are denoted by $M_1^{'}, M_2^{'}, \ldots M_K^{'}$. An unknown sequence $\mathbf{s}$ of hand postures is then parsed by each $M_l^{'}$, with a cost $d(\mathbf{s}, B_l^{'})$. The sequence $\mathbf{s}$ is then classified as the hand gesture $B_g$, where $B_g = \min d(\mathbf{s}, B_l^{'}, l = 1, 2, \cdots, K)$, and is the hand gesture behavior to which it is most similar. Therefore, sequences of hand postures are classifed based upon Maximum Similarity Classification (MSC) as seen in Fig. 6.

The hand gesture behavior classifier is innovatively implemented. The implementation is generalized so that the overall hand gesture behavior classification structure stays

**Table 1.** Confusion Matrix for Posture Recognition Classifier

| | Labeled: $a$ | Labeled: $b$ | Labeled: $c$ |
|---|---|---|---|
| True: $a$ | $Count(a,a)$ | $Count(a,b)$ | $Count(a,c)$ |
| True: $b$ | $Count(b,a)$ | $Count(b,b)$ | $Count(b,c)$ |
| True: $c$ | $Count(c,a)$ | $Count(c,b)$ | $Count(c,c)$ |

the same, and can operate on various number of hand gesture behaviors, various number of syntax rules per hand gesture behavior, various number of sequences of hand postures to classify, and various number of hand postures per sequence to classify as a hand gesture. The hand gesture classifier is also easily scalable to additional hand postures and hand gesture behaviors.

### *4.2. Hand Gesture Recognition Cost Automation*

In low-level classification of hand postures, hand postures can be mislabeled at times. The probability of misclassifying certain hand postures as other hand postures, is known a priori. For example, the probability of mislabeling the true hand posture $b$ as hand posture $a$ is known a priori. This knowledge is used in classifying sequences of hand postures; in other words, these probabilities are used in automating the costs per syntax rule in a hand gesture behavior. The probabilities of mislabeling certain hand postures as other hand postures is extracted from the confusion matrix, also known as the recognition summary in section Sec .3.

The probability that the low-level hand posture recognition classifier mislabeled hand postures is then calculated from the confusion matrix for the posture recognition classifier, as seen in Table 1. Let $N$ be the total number of hand postures classified by the low-level hand posture recognition classifier (also is the size $|\Sigma|$).

$$P(labeledposture = a|trueposture = b) = \tag{5}$$

$$= \frac{P(labeledposture = a, trueposture = b)}{P(trueposture = b)} \tag{6}$$

$$= \frac{Count(b,a)/N}{(Count(b,a) + Count(b,b) + Count(b,c))/N} \tag{7}$$

$$= \frac{Count(b,a)}{Count(b,a) + Count(b,b) + Count(b,c)} \tag{8}$$

The conditional probability estimate for all possible mislabeled hand postures can now detected. The substitution costs are defined as the inversion of the conditional probability. This can be seen in an example; let the cost for substituting the mislabeled posture $a$ with the true posture $b$ be the inversion of the probability that the low-level hand posture classifier mislabeled posture $b$ as posture $a$. By denoting the conditional probabilities $P(labeledposture = a|trueposture = b)$ as as $P(a|b)$, then the cost for substituting a mislabeled the true posture $a$ with the true posture $b$ is:

$$C_{S(a,b)} = 10log_{10}(\frac{1}{P(a|b)}) \tag{9}$$

**Table 2.** Costs - Confusion Matrix for Posture Recognition Classifier

| Labeled: $x$ | $1/x$ | $10log_{10}(1/x)$ |
|---|---|---|
| 0+eps | 4.5x$10^{15}$ | 156.5356 |
| 0.1 | 10 | 10 |
| 0.2 | 5 | 6.9897 |
| 0.3 | 3.3333 | 5.2288 |
| 0.4 | 2.5 | 3.9794 |
| 0.5 | 2 | 3.0103 |
| 0.6 | 1.6667 | 2.2185 |
| 0.7 | 1.4286 | 1.5490 |
| 0.8 | 1.25 | 0.9691 |
| 0.9 | 1.1111 | 0.4576 |
| 1.0 | 1 | 0 |

Thinking about this intuitively, if there is a high probability that the low-level hand posture recognition classifier mislabels the true posture $b$ with posture $a$, where $P(a|b)$ is high, then the cost for substituting posture $a$ for $b$, $C_{S(b,a)}$, is low.

Additionally, if this probability $P(a|b)$ is nonzero, then $P(b|b)$ is less than one, since

$$P(b|b) = 1 - P(a|b) - P(c|b). \tag{10}$$

In order to get an understanding of the range of potential costs per edit, let $x$ be an entry in the confusion matrix, the cost is normalized in a range by taking the $10log_{10}(1/x)$, where $x$ is the probability such as $P(a|b)$. In addition, to avoid infinite costs, such as when the probability is zero, probabilities of zero have $\epsilon$ added, $0 + \epsilon$, where $\epsilon = 2x10^{-16}$. To get insight into the costs from the probabilities of misclassifying hand postures, see the costs in Table 2 lists the range.

For this data set, we set the cost for deleting a posture instance $b$, for example, $D(b)$ be $20log_{10}(1/(0+eps))$, with this, constraining the hand gesture classifier to chose substitutions for the minimum cost edits possible, and scale deletions to other applications in future work. The infrastructure is in place. Deleting could be used for cases where edit is not a result of low-level classification, but the user used the wrong hand posture by accident.

## 5. Experimental Design and Results

This section will demonstrate hand gesture behaviors enhancing low-level classifications of hand postures and scaling the possible number of computer commands available for AAL. Since low-level hand posture classifications can have errors, sequencing the hand postures together over time can enhance the low-level classifications for ambient assisted living environments. In addition to enhancing the low-level classification, if you limit the computer commands to the fixed set of hand postures, the number of computer commands possible is the number of hand postures possible. If you sequence various hand postures together over time, additional computer commands are possible. Therefore, the experimental results of this paper will show that sequencing hand postures together will enhance low-level classifications and sequencing various hand postures together allows for more possible computer commands. For each classification, the hand gesture behavior classification performance accuracy results will also be shown. The overall experiment design focus can be seen in Fig. 4.

## *5.1. Experimental Design*

This section describes the experimental design.

As discussed in section Sec .3, the possible hand postures are Closed, Open, Lback, Lpalm, Victory, and Sidepoint. In order to define the various hand gesture behaviors, a symbol is assigned to each hand posture as seen in Table 3, with the alphabet of hand postures, $\Sigma = a, b, c, d, e, f$.

**Table 3.** Hand Postures and Corresponding Symbols

| **Hand Posture** | **Symbol** |
|---|---|
| *Closed* | $a$ |
| *Open* | $b$ |
| *Lback* | $c$ |
| *Lpalm* | $d$ |
| *Victory* | $e$ |
| *Sidepoint* | $f$ |

Various data sets were created, where each data set has six hand gesture behaviors, in order to compare six hand gesture behaviors with six individual hand postures, as seen in Table 4. Data Set 1 sequences the same hand posture together, and therefore classifying this hand gesture is similar to a weighted average over the detected hand posture over time (e.g. a low-pass filter to remove and smooth out the noise of posture mislabeling). In addition, combining various hand postures, extends the number of potential computer commands usable in an AAL environment. Therefore, various combinations of the hand postures were defined per hand gesture behavior, such as Data Set 2 uses the most reliable pairs of hand postures, Data Set 3 uses the least reliable pairs of hand postures, Data Set 4 uses a combination of the most reliable and least reliable hand posture label.

**Table 4.** Hand Gesture Behaviors (Six Per Data Set), where $n, m, k, > 1$ for all Hand Gesture Behaviors.

| **Hand Gesture Behavior** | **Data Set 1** | **Data Set 2** | **Data Set 3** | **Data Set 4** |
|---|---|---|---|---|
| Behavior 1 | $a^n$ | $a^n b^m$ | $c^n d^m$ | $a^n f^m$ |
| Behavior 2 | $b^n$ | $a^n c^m$ | $c^n e^m$ | $a^n e^m$ |
| Behavior 3 | $c^n$ | $b^n c^m$ | $c^n f^m$ | $b^n f^m$ |
| Behavior 4 | $d^n$ | $a^n d^m$ | $d^n e^m$ | $b^n e^m$ |
| Behavior 5 | $e^n$ | $b^n d^m$ | $d^n f^m$ | $c^n f^m$ |
| Behavior 6 | $f^n$ | $c^n d^m$ | $e^n f^m$ | $c^n e^m$ |

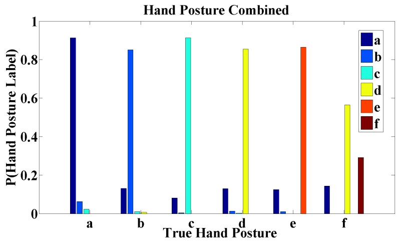

**Figure 7.** Probabilities of hand posture classification (y-axis) given the true hand posture (x-axis).

### *5.2. Experimental Results*

This section shows the results of hand gesture behavior classification, based upon sequences of lower-level classifications of hand postures.

The low-level hand posture detection is shown in Fig. 7. In this figure, probability of classifying a certain hand posture, given the true hand posture is shown. The x-axis is true posture label, the y-axis is probability of classifying an observed label (shown in the various colors, represented in the legend) given the true label. The six hand postures, from Table 3 are shown. Note, the inherent high probability of the low-level posture classifier mislabeling the true *Sidepoint*, $f$, hand posture as the *Lpalm*, $d$, hand posture.

The hand gesture behavior detection is shown in Fig. 8. In this figure, probability of classifying a certain hand gesture behavior, given the true hand gesture behavior is shown. The x-axis is true hand gesture behavior label, the y-axis is probability of classifying an observed label (shown in the various colors, represented in the legend) given the true label. Six hand gesture behaviors per data sets, from Table 4 are shown. The robustness of the high-level hand gesture classifier is portrayed in these results. The first dataset, Data Set 1, the gestures made up of single postures, was designed to show the error-correcting capability of the high-level behavior classifier. Notice the 100% accuracy of classifying the sequences of *Sidepoint* postures, even though the underlying raw posture symbols were mostly mislabeled. In addition, the other Data Sets show various hand gesture behavior classifications, enhancing the low-level hand posture recognitions and increasing the vocabulary size, therefore increasing the number of computer commands that can be interpreted for AAL.

## 6. Future Work

Future work can be broken into improved algorithm development for usability and scaled to additional applications. For improved algorithm development, we plan to incorporate usability factors into the costs (fuse these factors with the costs from the low-level classifier). For example, “how often does a user misuse a hand posture in a hand gesture?” should be additionally incorporated into the hand gesture behavior costs. In addition, this paper can scale to various applications, such as increased awareness and environment enabled AAL, where a house would have a network of cameras throughout the house that can interpret hand gestures as commands. Another potential application is a enabling a smart environments and communications using a surveillance system infrastructure. In addition to surveillance, people with certain features (e.g. airport security, etc.) can

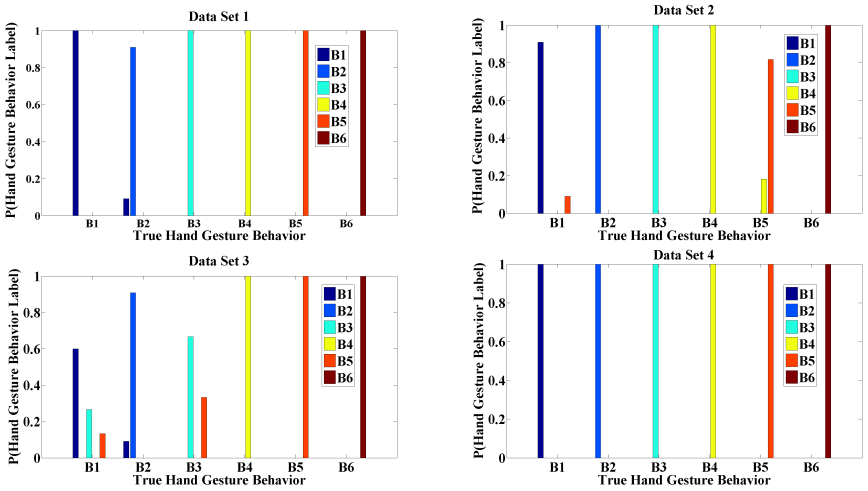

**Figure 8.** Probabilities of hand gesture behavior classification (y-axis) given the true hand gesture behaviors (x-axis).

communicate signals to a central control station, and communicate information such as unusual behavior they observed or reporting a status on certain suspicious persons. In this case, the start and end of a hand gesture would be defined, e.g. specific hand postures would be defined to initiate and terminate a hand gesture communication. In addition, the structure of hand gesture behavior classification can be scaled to classifying sequences of body postures for human gesture behavior classification for surveillance applications.

## 7. Summary

We built a vision-based user interface to support intentional interaction with a smart assisted-living environment. Our experiments show that the use of temporal hand postures, creating hand gesture behaviors, improves the recognition rates and increases the available vocabulary, two very important considerations for a user interface. The low-level classifications of hand postures are therefore enhanced through hand gesture behaviors for more robust human computer interaction; in addition, through hand gesture behaviors, with an increased vocabulary possible, the number of computer commands interpreted from hand gesture behaviors also increases. Overall, ambient assisted living is enabled through computer command interpretations from hand gesture behaviors, as a result of low-level posture recognitions over time. The vision-based intentional interface is robust in that it can be scaled to a range of hand posture types and hand gesture behavior types, and therefore additional applications as discussed in Sec. 6 can be carried out.

## References

[1] R. A. Bolt. Put-That-There: Voice and Gesture in the Graphics Interface. *Computer Graphics, ACM SIGGRAPH*, 14(3):262–270, 1980.

[2] F. Bremond and G. Medioni. Scenario recognition in airborne video imagery. In *DARPA98*, pages 211–216, 1998.

[3] W. T. Freeman, D. B. Anderson, P. A. Beardsley, C. N. Dodge, M. Roth, C. D. Weissman, and W. S. Yerazunis. Computer Vision for Interactive Computer Graphics. *IEEE Computer Graphics and Applications*, pages 42–53, May-June 1998.

[4] R. Goshorn. *Sequential Behavior Classification Using Augmented Grammars*. Master's thesis, University of California, San Diego, June 2001.

[5] R. Goshorn. *Syntactical Classification of Extracted Sequential Spectral Features Adapted to Priming Selected Interference Cancelers*. PhD thesis, University of California, San Diego, June 2005.

[6] R. Goshorn and D. Goshorn. Vision-based syntactical classification of hand gestures to enable robust human computer interaction. In *3rd Workshop on AI Techniques for Ambient Intelligence, co-located with European Conference on Ambient Intelligence (ECAI08).*, pages 211–216, 1998.

[7] Y. Ivanov and A. Bobick. Probabilistic parsing in action recognition, 1997.

[8] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.

[9] M. J. Jones and J. M. Rehg. Statistical Color Models with Application to Skin Detection. *Int. Journal of Computer Vision*, 46(1):81–96, Jan 2002.

[10] N. Kohtake, J. Rekimoto, and Y. Anzai. InfoPoint: A Device that Provides a Uniform User Interface to Allow Appliances to Work Together over a Network. *Personal and Ubiquitous Computing*, 5(4):264–274, 2001.

[11] M. Kölsch. *Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments*. PhD thesis, Computer Science Department, University of California, Santa Barbara, September 2004.

[12] M. Kölsch and M. Turk. Robust Hand Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2004.

[13] M. Kölsch and M. Turk. Hand Tracking with Flocks of Features. In *Video Proc. CVPR IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[14] M. Kölsch, M. Turk, and T. Höllerer. Vision-Based Interfaces for Mobility. In *Intl. Conference on Mobile and Ubiquitous Systems (MobiQuitous)*, August 2004.

[15] K. Nickel, E. Seemann, and R. Stiefelhagen. 3D-tracking of Head and Hands for Pointing Gesture Recognition in a Human-Robot Interaction Scenario. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2004.

[16] E. J. Ong and R. Bowden. A Boosted Classifier Tree for Hand Shape Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 889–894, 2004.

[17] R. Pausch and R. D. Williams. Tailor: creating custom user interfaces based on gesture. In *Proceedings of the the third annual ACM SIGGRAPH symposium on User interface software and technology*, 1990.

[18] M. Sipser. *Thoery of Computation*. PWS Publishing Company, Massachusetts, 1997.

[19] T. E. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. In *AFGR, Zurich*, 1995.

[20] M. Turk. Gesture recognition. In K. Stanney, editor, *Handbook of Virtual Environments: Design, Implementation and Applications*. Lawrence Erlbaum Associates Inc., December 2001.

[21] P. Viola and M. Jones. Robust Real-time Object Detection. *Int. Journal of Computer Vision*, May 2004.

[22] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler. Gestix: a doctor-computer sterile gesture interface for dynamic environments. In *Soft Computing in Industrial Applications: Recent and Emerging Methods and Techniques*, pages 30–39, 2007.

[23] J. P. Wachs, H. Stern, and Y. Edan. Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(6):932–944, 2005.

[24] A. Wilson and S. Shafer. XWand: UI for Intelligent Spaces. In *ACM CHI*, 2003.