
*SEVENTH FRAMEWORK PROGRAMME*
*INFORMATION AND COMMUNICATION TECHNOLOGIES*
*Cognitive Systems, Interaction, and Robotics*

# *Deliverable: D1.3*

| | |
|---|---|
| ***Project acronym:*** | *Dicta-Sign* |
| ***Project full title:*** | *Sign Language Recognition, Generation and Modelling with Application in Deaf Communication* |
| ***Grant agreement no.:*** | 231135 |
| ***Status:*** | *V1.0* |
| ***Date of preparation:*** | *30/11/11* |

# Contents

# 1 Acronyms

**AAM** Active Appearance Model

**BP** Binary Pattern

**DGS** Deutsche Gebärdensprache - German Sign Language

**GSL** Greek Sign Language

**HamNoSys** The Hamburg Notation System

**CART** Classification and Regression Tree

**RDF** Random Decision Forest

**PCP** Percentage of Correctly Matched Parts

**NMF** Non-Manual Features

**PDTS** Postures, Detentions, Transitions, Steady Shifts

**pp** Percentage Point

**SDM** Social Dynamics Model

**SiGML** Sign Gesture Mark-up Language

**SL** Sign Language

**SLR** Sign Language Recognition

**Std Dev** Standard Deviation

**SU** Sub-unit

# 2 Introduction

This deliverable covers the work of WP1 relating to the mapping of features to The Hamburg Notation System (HamNoSys). It relates specifically to T1.3 - Mapping Body Motion to HamNoSys and T1.4 - Mapping Facial Motion to HamNoSys which is an ongoing task until M36. It builds on the work of T1.1 - Visual Tracking and Detection and T1.2 Image/Object Feature extraction which took place in the earlier stages of the project and as such they shall be re-capped and relevant enhancements detailed. Sections of the work presented here feed into D7.3 - A Sign Wiki and will be demonstrated in D8.1 the final project demonstrator.

Among the main reasons that Sign Language Recognition (SLR) has not been yet practical is the lack of phonetic transcriptions and the huge level of effort required for creating detailed annotations. Sign linguists describe sign motions in conceptual terms such as 'hands move left' or 'dominant hand moves up'. They describe locations by relating them to other body parts such as 'by the ear', 'on the right shoulder' or 'on the lips' (Sutton-Spence and Woll, 1999; Hanke and Schmaling, 2004; Valli et al, 2005). These generic labels can cover a wide range of signing styles whilst still containing discriminative motion information. The aim of the work in this report is to convert information contained within the corpora into a similar description to that provided by the linguists. This is useful not only as an annotation aide but also as an input to sign level classifiers and Sub-unit (SU) based recognition. We explore approaches based on appearance data alone as well as those based on tracking data.

Currently there is a lack of appropriate phonetic models in the area of Sign Language (SL) linguistics. There have been several recent successful data-driven methods such as the work of Bowden et al (2004). They employ a linguistic feature vector based on measured visual features, such as relative hand movements. Bauer and Kraiss (2001) cluster independent frames via K-means, and produce 'phenones'. Instead of single frames, (Fang et al, 2004; Han et al, 2009; Yin et al, 2009) cluster sequences of frames on the feature level, such that they exploit the dynamics inherent to sign language. Recently, separate features and modelling for dynamic vs. static segments have been proposed (Pitsikalis et al, 2010).

WP1 has worked on tracking the corpus in both 2D (Section 3.1) and 3D from wide base stereo (Section 3.2) as preparation for learning linguistic features. Given the introduction of the Kinect, research has also continued on creating an alternative pose detection method for real-time applications (Section 3.3). Following this we discuss the work mapping motion to linguistic features using both appearance information and the trajectories. We present options for both learning linguistic features (Sections 4 & 6) as well as those derived deterministically (Sections 5 & 7). Also covered are the appearance based handshapes classifiers used for the corpus data (Section 8.1) and those based on the depth information from the Kinect™ for real-time recognition (Section 8.2). We then include in Section 9 the work of NTUA, for incorporating HamNoSys labels into a SU-based statistical framework for training and alignment so as to map generic visual data onto meaningful phonetic models, as applied for the case of motion and Greek Sign Language (GSL). NTUA then present their facial cues drawn from Active Appearance Models (AAMs) in Section 10 Finally, we cover the ongoing work into turning the tracked facial features into linguistic features (Section 11).

## *2.1 Contributors*


Helen Cooper (UniS)
Brian Holt (UniS)
Dumebi Okwechime (UniS)
Vassilis Pitsikalis (NTUA)
Isidoros Rodomagoulakis (NTUA)
Simon Hadfield (UniS)
Nicolas Pugeault (UniS)
Richard Bowden (UniS)
Stavros Theodorakis (NTUA)
Petros Maragos (NTUA)

# 3 Tracking and Detection

## 3.1 2D Tracking

The 2D tracking uses the work described in *D1.1-Tracking Framework*. It starts by using a skin model to create a skin mask containing the head and hands. This mask is refined using morphological operations and separated into its component body parts. The parts are tracked during non-occluded, key frames using a linear prediction model. During occlusions, a mean shift style tracker is employed and refined using backward/forward prediction. This produces smooth and accurate trajectories as presented by Roussos et al (2010).

## 3.2 3D Tracking

When using the real-time Kinect™, there are several libraries which produce 3D skeletal data for a user. We have chosen the one provided by PrimeSense (PrimeSense Inc., 2010) as found in the OpenNI (OpenNI organization, 2010). This has several advantages over the Microsoft version but does require the user to calibrate the system. While the Dicta-Sign corpus contain a vast number of signs, over a very large vocabulary; the available depth data is inferior to that produced by the Kinect™. It lacks the high fidelity depth maps and 3D skeletal tracking. However, by estimating the 3D trajectories of the head and hands, it is possible to extract general motion features (see section 4.2) of the same form as those extracted from the Kinect data. This provides a great deal of additional training data for learning HamNoSys which can be applied to the real-time system.

The corpus is comprised of videos from 3 viewpoints (front, top and side), with the signer against a blue background in the former 2, as shown in figure 1. This provides a very wide baseline, which makes 3D estimation challenging.

Figure 1: A frame from all 3 viewpoints in the DGS dataset.

### 3.2.1 3D Trajectories Using Scene Flow

Scene Flow is the motion field, defining the 3 dimensional motion at every point in a scene. By projecting the Scene Flow field onto an image plane, you obtain the corresponding optical flow within that image, however scene flow also contains information about out of plane motion.

By clustering the point motions that comprise a Scene Flow field, we are able to perform segmentation of 3D objects. In addition, performing the clustering in both the spatial and motion dimensions allows objects in close proximity to be differentiated, if they have differing motions (such as hands passing each other).

These cluster centres define not only a spatial location of the object, but also the average motion of that object. Thus sequential cluster positions can be combined into continuous

Figure 2: Example trajectories created over 15 frames by combining sequential sets of cluster centres based on a constant velocity motion model.

trajectories, by using a simple constant velocity assumption and assigning the closest new cluster to the object, example trajectories are shown in figure 2.

An adaptive skin colour model is used to segment all viewpoints. Scene Flow Estimation is then performed only within the located skin-like regions, significantly improving runtime. Currently roughly 25 hours of trajectories have been estimated, split between the GSL, and DGS datasets.

### 3.2.2 Scene Particles

A novel algorithm, termed the "Scene Particles Algorithm" Hadfield and Bowden (2011), has been developed for estimating the scene flow field. Conventional scene flow estimation techniques are framed as optimisation problems, with regularisation used to resolve ambiguities by favouring smoothness of the motion field. The Scene Particles algorithm instead maintains all valid hypotheses, with weights, updating them with each new set of observations. By avoiding smoothness constraints, the algorithm avoids over-smoothing artefacts at object boundaries, which are common in alternative approaches, and make object segmentation much more difficult.

One useful feature of the Scene Particles algorithm in comparison to alternative techniques, is it's ability to operate with any number of viewpoints, and any combination of input modalities. This allows the algorithm to operate on the Dicta-Sign corpus using multi-view appearance only data, and the Kinect front end, using single viewpoint, appearance & depth data. Table 1 shows the performance of the Scene Particle algorithm, compared to alternative approaches, on a scene flow benchmark dataset.

The scene particles algorithm, operating on both sets of inputs, outperforms alternative techniques in all performance measures except directional error. The reason for this is that the Scene Particles algorithm is a stochastic process, and so there is small amount of inherent noise. On moving objects this noise floor is negligible compared to the objects motion and so accuracy is very high. However on background objects with little to no motion, a small amount of noise translates to a large change in direction.

Another useful feature of the Scene Particles algorithm is that the number of motion hypotheses maintained per ray can be tuned to trade off accuracy against speed. Figure 3 demonstrates that errors experience an exponential decay as runtime is increased.

| Algorithm | Data | Views | Op. Flow NRMSE | Stereo Flow NRMSE | Struct. NRMSE | AAE (Deg) |
|---|---|---|---|---|---|---|
| **Scene Particle (depth) (Hadfield and Bowden, 2011)** | Venus | 1 | 0.09 | **0.00** | - | 5.44 |
| **Scene Particles (multiview)** | Venus | 4 | **0.03** | **0.00** | **0.23** | 3.88 |
| Basha et al (2010) | Venus | 4 | 1.55 | **0.00** | 5.39 | 1.09 |
| Huguet and Devernay (2007) | Venus | 2 | 3.70 | 3.05 | 8.84 | **0.98** |
| **Scene Particles (depth) (Hadfield and Bowden, 2011)** | Cones | 1 | 0.09 | **0.00** | - | 5.02 |
| **Scene Particles (multiview)** | Cones | 4 | **0.05** | **0.00** | **0.26** | 3.81 |
| Basha et al (2010) | Cones | 4 | 1.32 | 0.01 | 6.22 | **0.12** |
| Huguet and Devernay (2007) | Cones | 2 | 5.79 | 8.24 | 5.55 | 0.69 |
| **Scene Particle (depth) (Hadfield and Bowden, 2011)** | Teddy | 1 | 0.11 | 0.00 | - | 5.04 |
| **Scene Particles (multiview)** | Teddy | 4 | **0.04** | **0.00** | 0.19 | 3.77 |
| Basha et al (2010) | Teddy | 4 | 2.53 | 0.02 | 6.13 | **0.22** |
| Huguet and Devernay (2007) | Teddy | 2 | 6.21 | 11.58 | 5.64 | 0.51 |

Table 1: Performance on Middlebury dataset, measuring estimation accuracy in terms of motion magnitude in the image plane, motion magnitude out of the image plane, directional accuracy and structural accuracy. The Scene Particles algorithm is examined in hybrid (appearance & depth) and multi-view modes.

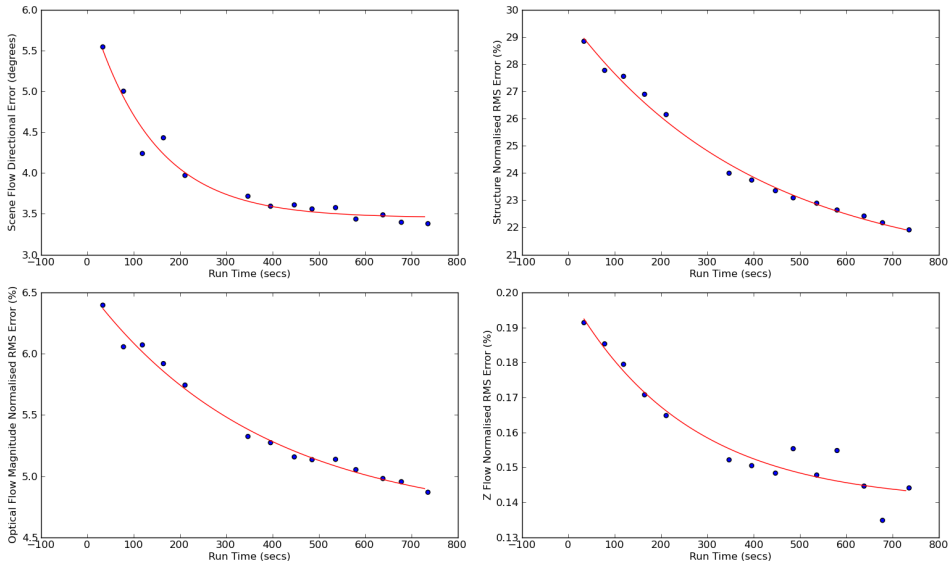



Figure 3: Accuracy of the multi-view system on the cones dataset, as the number of motion hypotheses (and hence the runtime) is increased.

| Object | Agreement | X RMS error | Y RMS error |
|---|---|---|---|
| Head | 100% | 0.057 | 0.097 |
| Right Hand | 93.535% | 0.191 | 0.100 |
| Left Hand | 88.054% | 0.277 | 0.091 |

Table 2: Agreement between projection of estimated 3D trajectories and 2D trajectories from an alternative system (values in palm widths).

To evaluate the quality of the 3D tracks obtained from the scene flow clustering, they were projected onto the frontal image plane and compared to a high accuracy 2D tracker (Roussos et al, 2010). Table 2 shows this comparison over a 30,000 frame sequence. Trajectories were assumed to be in agreement if their x and y positions were within $\frac{1}{3}$ of a palm width. The head and right hand objects are extremely accurate, while the performance of the left hand is lower, due to it's almost constant occlusion in the side view, providing less supporting information.

### *3.3 Pose Estimation from Depth*

While the previous sections have dealt with video tracking and depth tracking, work has continued into identifying poses in depth images so as to avoid the need for calibration when using the Kinect™. In this section we describe a novel algorithm developed to estimate the articulated body pose from depth information (Holt et al, 2011).

Using depth images is a natural way to classify pose as, compared to appearance, it directly contains information on the object in the third dimension. Depth information provides some significant advantages over appearance data alone. The depth data can be calibrated to the real world, providing information about the scale and size of the subject, it can can be used directly to disambiguate occluding parts and it provides illumination invariance since the depth information remains largely constant under variable conditions (depending on the capture method).

Pose estimation has been a topic of interest to the Computer Vision community for many years. There are two main approaches to the problem. The first is a generative approach where, given a model of a person, the model is rendered and the output is compared to the evidence in the image iteratively until a best fit is found if possible (see (Moeslund et al, 2006) for a survey). This approach has the advantage that it is possible to render any pose described by the model, but the optimisation process is not guaranteed to converge to a global solution, and can take a long time to estimate the pose. The second, discriminative approach, attempts to train a learning algorithm to predict the pose directly based on the evidence in the image. The two main ways of doing so are estimating the global pose directly using a classifier or by detecting and assembling parts. Example-based approaches benefit from being fast, yet the many degrees of freedom in the human body means that a large training set containing all the expected poses is required, making it infeasible to cover the entire pose space. Part based approaches have shown to be very effective. These approaches have two stages: firstly the detection, and secondly the assembly of detected parts into a global configuration.

We describe a novel algorithm developed for articulated pose estimation that combines the best aspects of part-based and example-based estimation. First detecting poselets extracted from the training data; our method then applies a modified Random Decision Forest to identify poselet activations. By combining key point predictions from poselet activations within a graphical model, we can infer the marginal distribution over each key point without any kinematic constraints.

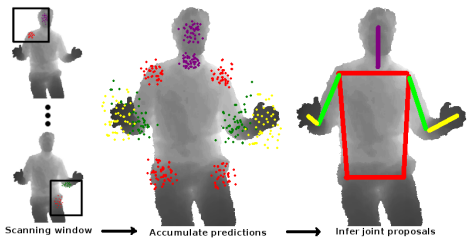



Figure 4: **Overview**. For an input single depth image, run a multi-scale scanning window. Each window is evaluated by a Random Forest classifier to detect poselet activations. Each activated poselet makes local predictions for the body parts it was trained on. The overall configuration is inferred by combining key point predictions within a graphical model, and finding the maximum over the marginal distributions.

### 3.3.1 Poselet Representation

Our objective is to estimate the configuration of a person in the 2D image plane parametrised by $B$ body parts by making use of a small training set. We define the set of body parts $\mathbb{B} = \{\mathbf{b}_i\}_{i=1}^{B}$ where $\mathbf{b}_i \in \Re^2$. The labels of corresponding to $\mathbb{B}$ comprise $\mathbb{Q} = \{$head, neck, shoulder$_L$, shoulder$_R$, hip$_L$, hip$_R$, elbow$_L$, elbow$_R$, hand$_L$, hand$_R\}$ where $|\mathbb{Q}| = B$.

A poselet, by the definition of Bourdev et al (2010); Bourdev and Malik (2009), is a set of parts that are "tightly clustered in configuration space and appearance space". The name *poselet* reflects the fact that it is related to the concept of a pose (a specific configuration of the human body), but is only a subset of the overall configuration. The basic assumption through part-based pose estimation literature is that a "part" should correspond closely to an anatomical subdivision of the body such as "hand" or "forearm", but is not necessarily the most salient feature for visual recognition especially if the part is itself is highly deformable, making it susceptible to high levels of false positive detections. In contrast, a description such as "half a frontal face and shoulder" or "legs in a scissor shape" may be far easier to detect reliably. While Bourdev et al (2010) proposed poselets for human detection, this approach proposes to apply the concept of a poselet to pose estimation.

Given that the definition of poselet covers any subset of configurations of the human body, the term could be applied to a part described by single joint or to the configuration of the entire body and anywhere in between. Using just the smallest poselets (anatomically defined parts) corresponds to the part-based pose estimation approach, whereas using the entire configuration corresponds to the example-based approach to pose estimation. Therefore the use of poselets can be seen as a hybrid between these two major approaches.

We formally define poselets as follows. Let $\mathbb{P} = \{\mathbf{p}_i\}_{i=1}^{P}$ be the set of $P$ poselets. Each poselet $\mathbf{p}_i = (r_i, q_i, \mathbb{I}_i)$. The number of *instances* of poselet $\mathbf{p}_i$ is $|Pi|$. $r_i = (w_i, h_i)$ where $w_i$ and $h_i$ are the width and height of poselet $\mathbf{p}_i$, $q_i = (q_i^1, q_i^2)$ are the labels of the two body parts on which the poselet was extracted, where $q_i^1 \neq q_i^2$, $q_i^{1,2} \in \mathbb{Q}$ and the set

Figure 5: Example Poselets. Each row contains representative examples of a poselet.

of poselet instances $\mathbb{I}_i = \{\mathbf{i}_{ij}\}_{j=1}^{|Pi|}$. We define $\mathbf{i}_{ij} = (A_{ij}, c_{ij}^1, c_{ij}^2)$ where $A_{ij} \in \Re^D$ is the pixel intensities, sub-sampled to a window of dimension 24x18 and vectorised. $c_{ij}^1$ is a hypothesis of the location for body part $q_i^1$. $c_{ij}^2$ is the potential location in rectangle $r_i$ for body part $q_i^2$.

### 3.3.2 Extracting poselets

Each poselet $\mathbf{p}_i$ is treated as a semantic class. The goal is to find a set of poselets $\mathbb{P}$ from the training data that maximally span the configuration space. Our algorithm for selecting poselets is very similar to that of Bourdev et al (2010). A seed window $r_i$ for poselet $\mathbf{p}_i$ is randomly chosen within a randomly selected training image, and other examples are found by searching each of the other training images for a patch where the local configuration of key points is similar to that in the seed. However, since our goal is pose estimation and not person detection, we propose to use a different distance measure, having found empirically that poselets generated using the algorithm of Bourdev et al (2010) tend to concentrate around the torso at the expense of arm-centric poselets. By applying a euclidean distance over the seed poselet body part locations $c_{i0}^{1,2}$ and new candidate poselet $c_{ij}^{1,2}$, we found that the extracted poselets to be widely distributed over the pose space. A probability distribution over the key points for each poselet is computed, which is used at test time to make predictions of the locations of key points.

Poselet extraction is essentially an unsupervised clustering step in which the data is preprocessed for the classification task to follow.

### 3.3.3 Detecting poselets

Given that our images are sourced from a depth camera, we believe that the pixel relationships within a local window contain sufficient information to efficiently discriminate between poselets, especially with a tree based learner. We use Random Decision Forest (RDF) classifiers are an ensemble classifier, consisting of $T$ individual binary decision trees, usually Classification and Regression Tree (CART) (Breiman et al, 1984). Breiman (2001) demonstrated that using multiple trees trained on random partitions of the dataset are superior classifiers than individual trees.

**Training** Given a supervised training set consisting of $p$ $F$-dimensional vector and label pairs $(S_i, l)$ where $S_i \in R^F, \quad i = 1, ..., p$ and $l \in R^1$, a decision tree recursively partitions the data such that impurity in the node is minimised, or equivalently the information gain is maximised through the partition.

Let the data at node $m$ be represented by $Q$. For each candidate split $\theta = (j, \tau_m)$ consisting of a feature $j$ and threshold $\tau_m$, partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets. The standard decision tree formulation Breiman et al (1984), applies axis-aligned splits to recursively partition the space

$$Q_{left}(\theta) = (x, l)|x_j \leq \tau_m \tag{1}$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta) \tag{2}$$

The impurity over the data $Q$ at node $m$ is computed using an impurity function $H()$, the choice of which depends on the task being solved (classification or regression). The impurity $G(Q, \theta)$ is computed as

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)) \tag{3}$$

Select for each node $m$ the splitting parameters $\theta$ that minimise

$$\theta^* = \arg\min_{\theta} G(Q, \theta) \tag{4}$$

This algorithm is applied recursively until all the class distribution at a node is pure ($Q_n(\theta) = 0$) or until the maximum allowable depth of the tree is reached.

**Test** For a given window position and scale **x**, re-sample and vectorise to yield $X \in \Re^D$. Evaluating the RDF on $X$ with $P$ poselets, each tree returns a probability distribution over the poselet set $\alpha_{ti} = (\alpha_{t1}, \ldots, \alpha_{tP})$, where $\sum_{i=1}^{P} \alpha_t i = 1$. $\alpha_{ti}$ is therefore probability that the $i^{\text{th}}$ is activated at the given window **x**.

The output of the decision is averaged to yield the final output for the RDF.

$$P(\alpha|X) = \frac{1}{T} \sum_{t=1}^{T} \alpha_{ti} \tag{5}$$

A poselet $\mathbf{p}_{i^*}$ is considered as detected if

$$i^* = \arg\max_i \alpha_i \quad \text{and} \quad \alpha_{i^*} > c \tag{6}$$

where $c$ is an empirically derived threshold.

### 3.3.4 Predictions and Inference

Given an input image $I$ of resolution $I_w \times I_h$ pixels, for each body part $q_j \in \mathbb{Q}$ we define a probability distribution $\{\mathbb{Y}_q\}, \forall q \in \mathbb{Q}$ where $\mathbb{Y} \in \Re^{I_w} \times \Re^{I_h}, \mathbb{Y} = 0 \quad$ for all pixels.

We find that the approach described here achieves very high accuracy on the head, shoulders and waist. This makes sense intuitively because these parts maintain a consistent rigid spatial relationship, whereas the elbows and hands are less constrained. We improve overall predictive quality by introducing a graphical model to model the relationship between upper body anatomical parts. This is done by defining a hierarchy of levels inspired by Pictorial Structures, where the torso is the root, connecting to the elbows and hands. An iterative process to predict body parts at each of these levels is followed. On the first pass, probability distributions of the body-parts at the first level are computed. On subsequent passes, the information from all the previous levels is used to constrain predictions.

The most likely pose configuration $\hat{\mathbb{B}}$ is derived by applying non-maximal suppression to the final probability distributions.

### 3.3.5 Experiments

| | Head | Shoulders | Side | Waist | Upper arm | Forearm | Total |
|---|---|---|---|---|---|---|---|
| Without inference | 0.99 | 0.78 | 0.93 0.73 | 0.65 | 0.52 0.58 | 0.14 0.21 | 0.61 |
| With inference | 0.99 | 0.78 | 0.93 0.73 | 0.65 | 0.69 0.66 | 0.22 0.33 | **0.67** |

Table 3: Percentage of Correctly Matched Parts. Where two number are present in a cell, they refer to left/right respectively.

For each body part $q_i \in \mathbb{Q}$, a probability distribution is computed for that body part. Figure 7a shows the raw probability maps of the left/right elbow/hand respectively as insets as well as the maximum a posterior body part configuration overlaid on the depth image. We show 4 of the 10 possible distributions, that are affected by our graphical model. Figure 7b shows the same body part distributions with the inclusion of the graphical model. Notice the reduced ambiguity in the distributions that leads to a better final estimate of the pose.

We use the evaluation metric proposed by Ferrari et al (2008): "A body part is considered as correctly matched if its segment endpoints lie within $r = 50\%$ of the length of the ground-truth segment from their annotated location." The Percentage of Correctly Matched Parts (PCP) is then the percentage of these correct matches. Figure 6 shows the effect of varying $r$ in the PCP calculation. A low value for $r$ expects a very high level of accuracy in the estimation of both endpoints and relaxes this requirement as $r$ increases to its highest value. We report our results at $r = 0.5$ in Table 3. Our method is 67% accurate overall, showing very high accuracy on the torso, with accuracy decreasing as we move from torso to hand where the benefits of the inference process can clearly be seen. These results are comparable in terms of overall accuracy to the results of other pose estimation approaches working on appearance data. However, as the underlying datasets are different further comparisons are not possible. Importantly these results are achieved without any kinematic constraints on the estimate. We also plot the PCP curve in Figure 6. Inference doesn't provide fine corrections to posture, but is a strong prior that reduces false positive detections in favour of more likely poses.

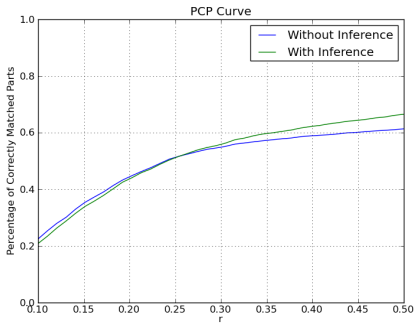
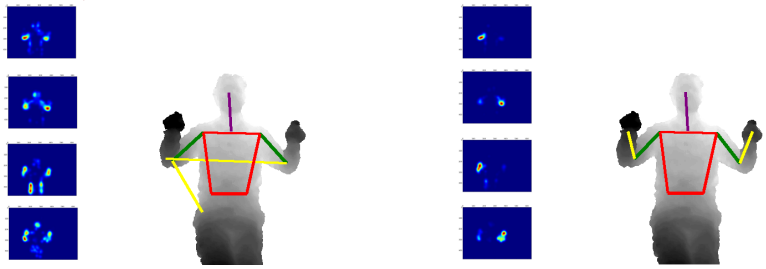



Figure 6: Pixel errors over test set.



(a) Without inference (b) With inference

Figure 7: Qualitative examples showing how inference improves predictions.

# 4 Learnt Motion Features

In this section we discuss the procedure for learning linguistically derived motion SUs from data with accurately labelled boundaries. This is done by creating weak classifiers based on the temporal feature vectors and using boosting to combine them together. Boosting is an iterative learning method which requires positive and negative examples of each class. On each iteration, examples which were not correctly classified in the previous iteration are increased in importance. Thus on each successive iteration boosting works harder to

Figure 8: *Left side:* Example predictions using the proposed method with inference. *Right side:* Predictions without inference. This shows where the proposed method works, and also includes a few failure cases.

correctly classify the more difficult examples. For a detailed description of boosting refer to Schapire and Singer (1999). The work here covers two approaches, the first based on appearance data and the second using tracked data. Due to boosting requiring uncontaminated examples for training, these classifiers have been quantitatively tested on a dataset which has been hand labelled at the SU level. This dataset contains a single signer performing 164 signs 10 times each.

## *4.1 Appearance based*

With appearance based methods, there are no exact positions for the hands. However, *Hand Arrangement* describes where the hands are in relation to each other, essentially, the shape they make in the image space. *Motion* looks at similar properties, how the hands move in relation to each other, or how the shape they make in the image space changes over time. Moments offer a way of encoding the shapes in an image; if vectors of moment values per frame are concatenated, then they can encode the change in shape of an image too. By applying moments to a skin mask, the motion and arrangement of the hands can be learnt for a set of linguistic features.

There are several different types of moments which can be calculated, each of them displaying different properties. Four of the basic types were chosen to form a feature vector, $\mathbf{m}$: spatial, $m_{ab}$, central, $\mu_{ab}$, normalised central, $\bar{\mu}_{ab}$ and the Hu set of invariant moments (Hu, 1962) $\mathcal{H}_1$-$\mathcal{H}_7$. Moments are a description of where in the image pixels are located, based on the distance of a pixel from the origin and the pixel luminance. The order of a moment is defined as $a + b$. This work uses all the moments up to the 3rd order, 10 in total. Central moments are based on a similar principle, but these are translation invariant as they are taken relative to the centroid of the pixels in the image. Again these are taken to the 3rd order, so there are 10 of these moments, 2 of which are always 0, (01, 10). As well as central moments, there are also normalised central moments, which are invariant to scale and translation, there are 10 of these also, 2 of which are always 0 due to their being based on the central moments. Finally, the Hu set of invariant moments are considered, there are 7 of these moments and they are created by combining the normalised central moments, see (Hu, 1962) for full details, they offer invariance to scale, translation, rotation and skew. This gives a 37 dimensional feature vector, with a wide range of different properties. There are several other varieties of moments which could be included in the feature vector, yet the current set covers a wide range of properties whilst maintaining manageability in terms of vector size. Since spatial moments are not invariant to translation and scale, there needs to be a common point of origin and similar scale across examples. To this end, the image is centred and scaled about the face of the signer before computation. The video clips are described by a stack of these vectors, $\mathbf{M}$, like a series of 2D arrays, as shown in figure 9(a) where the horizontal vectors of moments are concatenated vertically, the lighter the colour, the higher the value of the moment on that frame. As has been previously discussed, the *Motion* classifiers are looking for changes in the moments over time. Binary Patterns (BPs)

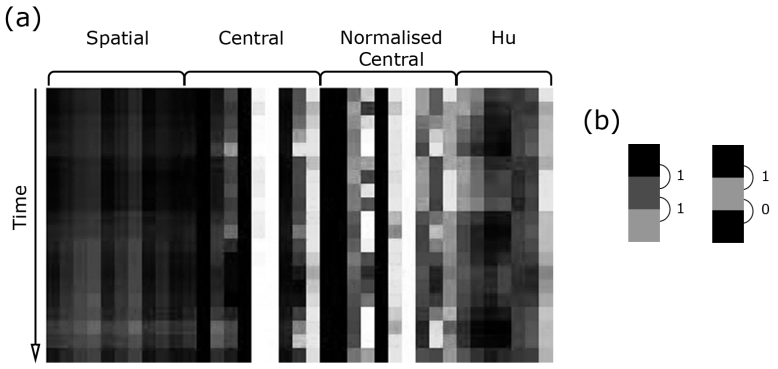



Figure 9: Moment vectors and local binary patterns for two stage classification. (a) A pictorial description of moment vectors (normalised along each moment type for a selection of examples), the lighter the colour the larger the moment value. (b) BP, working from top to bottom an increase in gradient is depicted by a 1 and a decrease or no change by a 0.

can be used to encode whether a moment is increasing or decreasing with time, giving the relative information about the motion. A temporal vector is said to match the given BP if every '1' accompanies an increase in a value between concurrent frames and every '0' a decrease/'no change'. This is shown in equation 7 where $\mathbf{O}_{i,t}$ is the value of the component, $\mathbf{o}_i$, at time $t$ and $\mathbf{bp}_t$ is the value of the BP at frame $t$. See figure 10 for an example where feature vector A makes the weak classifier fire, whereas feature vector B fails, due to the ringed gradients being incompatible.

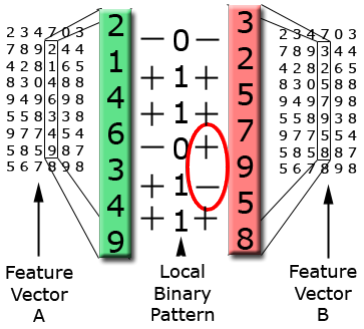



Figure 10: An example of a BP being used to classify two examples. A comparison is made between the elements of the weak classifiers BP and the temporal vector of the component being assessed. If every '1' in the BP aligns with an increase in the component and every '0' aligns with a decrease or 'no change' then the component vector is said to match (*e.g.* case A). However if there are inconsistencies as ringed in case B then the weak classifier will not fire.

$$
\begin{aligned}
R_{wc} &= |\arg\max_{\forall t}(BP(\mathbf{O}_{i,t})) - 1| \\
BP(\mathbf{O}_{i,t}) &= \mathbf{bp}_t - d(\mathbf{O}_{i,t}, \mathbf{O}_{i,t+1}) \\
d(\mathbf{O}_{i,t}, \mathbf{O}_{i,t+1}) &= \left\{ \begin{array}{c} 0 \; if \; \mathbf{O}_{i,t} \leq \mathbf{O}_{i,t+1} \\ 1 \; otherwise \end{array} \right\}
\end{aligned} \tag{7}
$$

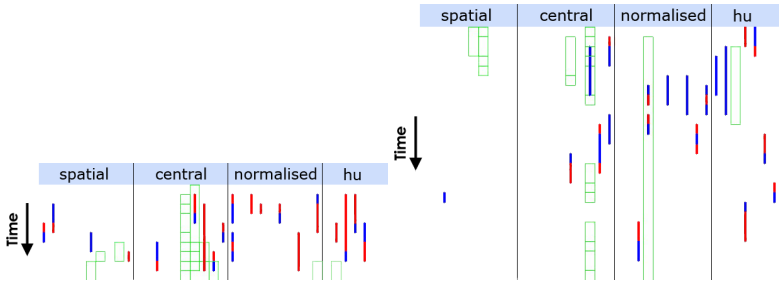
The BPs work on the gradient of a feature over time, they vary in size from 2 bits to 5 bits and there are therefore 60 different classifier patterns ($2^2 + 2^3 + 2^4 + 2^5$). The BPs are

run in parallel with the time axis, so that they are always operating on one type of value. For a BP to return a 1, every gradient must match its corresponding value in the patten, 1 for an increase or 0 for a decrease/no-change as is expressed in equation 7, this time acting on $\mathbf{M}_{i,t}$, a moment component, rather than a tracking one. Shown in figure 9(b) is a pictorial representation of this approach, working from the top, an increase in value is encoded with a 1 and a decrease in value is encoded with a 0.

Discarding all magnitude information would mean that salient information might be removed. To retain this information additive classifiers are also employed. These look at the average magnitude of a component over time. The weak classifiers are created by applying a threshold, $\mathcal{T}_{wc}$, to the summation of a given moment type, over several frames. This threshold is optimised across the training data during the learning phase. For an additive classifier of size $T$, over component $\mathbf{o}_i$, the response of the classifier, $R_{wc}$, can be described as in equation 8.

$$R_{wc} = \left\{ \begin{array}{c} 1 \; if \; \mathcal{T}_{wc} \leq \sum_{t=0}^{T} \mathbf{O}_{i,t} \\ 0 \; otherwise \end{array} \right\} \tag{8}$$

They can be as short as a single value, or as large as the maximum classifier size allowed. They therefore contain information about the magnitude of values across a given SU which complements the BPs gradient data.




(a) hands_move_apart (b) Hands_move_together

Figure 11: Boosted temporal moments BP and additive *Motion* classifiers. The moment vectors are stacked one frame ahead of another. The green boxes show where an additive classifier has been chosen, a blue line shows a decreasing moment value and a red line an increasing value.

Examples of the classifiers learnt are shown in figure 11, additive classifiers are shown by green boxes, increasing BPs are shown in red and decreasing ones in blue. When looking at a SU such as 'hands move apart' (figure 11a), the majority of the BP classifiers show increasing moments, which is what would be expected, as the eccentricity of the moments is likely to increase as the hands move apart. Conversely, for 'hands move together' (figure 11b), most of the BPs are decreasing.

## 4.2 Tracking Based

The tracking provides a frame by frame set of co-ordinates for the hands so motions can be described by changes in these values. The linguistic motion SUs do not encode magnitude information in their base form. Therefore the classifiers used to describe them need

to encode non-magnitude dependent information. If the values from the tracking are concatenated temporally into 2D vectors, then it is possible to examine individual components across time. In this way, a weak classifier can look for changes in, for example, the x co-ordinate of the dominant hand. This would encode left and right motion of the dominant hand. Component values can either increase, decrease or remain the same, from one frame to the next. If an increase is described as a 1 and a decrease or no change is described as a 0 then a BP can be used to encode a series of increases/decreases.

Boosting is given all possible combinations of BPs, acting on each of the possible tracking components. The BPs are limited in size to being between 2 and 5 changes (3 - 6 frames) long. Similar to before, additive features are also applied to all the possible components, the lengths permitted are between 1 and 26 frames. Both sets of weak classifiers can be temporally offset from the beginning of an example, by any distance up to the maximum distance of 26 frames.

### *4.3 Results*

Classifiers are trained on SUs from four out of ten available signs, then tested on the SUs from the remaining six. The chosen SUs are defined by the labelled data as such it is not a complete set. It was noted during training that some classifiers were having difficulty due to the overlap between classes. For instance, during 'hands move down' there is exactly the same information as with 'dominant hand moves down' so a test was performed where labels were adjusted to account for this. In this case the labels which were full supersets of other labels (as above) were removed from the negative examples.

The results of these tests are shown in table 4. As can be seen the results vary between the different SU. Overall the tracked classifiers perform better than the appearance based ones. The max result is 4% higher at 98% and the average is 5% higher at 59%. By adjusting the labels to account for the label overlaps the overall recognition rate drops slightly as there is now less negative training data. However, whereas some classes were failing (the dual handed circles for instance) they are now classifiable 41-49% of the time. This work was completed before the 3D tracked data became available and work is now on-going into continuing this work in 3D.

## 5 Deterministic Motion Features

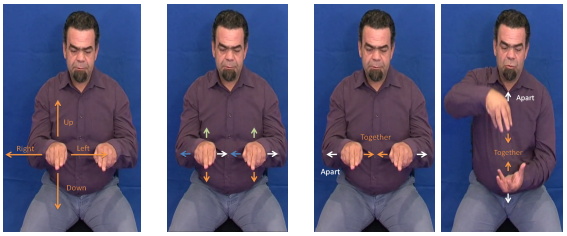
It has been shown that deterministic features perform well for sign language recognition (Kadir et al, 2004; Cooper et al, 2011; Elliott et al, 2011). They have also proven to be extremely useful in the developed prototypes due to the speed that they can be computed on live data and their ability to generalise between subjects.

### *5.1 2D Tracking*

In order to link the 2D x,y co-ordinates obtained from the tracking to the concepts used by sign linguists, rules are employed to extract HamNoSys based information from the trajectories. The approximate size of the head is used as a heuristic to discard ambient motion (that less than 0.25 the head size) and the type of motion occurring is derived directly from deterministic rules on the x and y co-ordinates of the hand position. The types of motions encoded are shown in figure 12, the single handed motions are available for both hands and the dual handed motions are orientation independent so as to match linguistic concepts.

| | Linguistic Unit | Appearance (%) | Tracked (%) | Adjusted Labels(%) |
|---|---|---|---|---|
| **Dominant Hand** | up | 71 | 90 | 74 |
| | down | 50 | 74 | 40 |
| | left | 52 | 61 | 48 |
| | right | 55 | 28 | 27 |
| | towards | 26 | 51 | 26 |
| | away | 55 | 46 | 36 |
| | circle_left_down | 94 | 95 | 84 |
| | circle_left_towards | 36 | 98 | 98 |
| **Dual Hand** | up | 72 | 72 | 71 |
| | down | 72 | 67 | 53 |
| | together | 47 | 66 | 52 |
| | apart | 42 | 81 | 74 |
| | circle_together_down | 22 | 0 | 41 |
| | circle_together_towards | 57 | 0 | 49 |
| | **Max** | 94 | 98 | 98 |
| | **Min** | 22 | 0 | 26 |
| | **Mean** | 54 | 59 | 55 |
| | **Std Dev** | 19 | 32 | 22 |

Table 4: Results of *Motion* appearance based classifiers for appearance and tracked data both with and without the label adjustments. This is a subset of the classifiers trained, limited to those which appear in most linguistic descriptions. A full discussion and details are available in (Cooper, 2010)




(a) Single handed (b) Bimanual: Synchronous (c) Bimanual: Together/Apart

Figure 12: Motions detected from 2D tracking

## 5.2 3D Tracking

In 3D we also currently focus on linear motion directions. Specifically, individual hand motions in the x plane (left and right), the y plane (up and down) and the z plane (towards and away from the signer). This is augmented by bi-manual classifiers for 'hands move together', 'hands move apart' and 'hands move in sync'. The approximate size of the head is used as a heuristic to discard ambient motion (that less than 0.25 the head size) and

| Locations | Motions | | |
|---|---|---|---|
| | Right or Left Hand | | Bi-manual |
| head | left | $\Delta x > \lambda$ | in sync |
| neck | right | $\Delta x < -\lambda$ | $\lvert\delta(L,R)\rvert < \lambda$ |
| torso | up | $\Delta y > \lambda$ | and |
| L shoulder | down | $\Delta y < -\lambda$ | $F^R = F^L$ |
| L elbow | towards | $\Delta z > \lambda$ | together |
| L hand | away | $\Delta z < -\lambda$ | $\Delta(\delta(L,R)) < -\lambda$ |
| L hip | none | $\Delta L < \lambda$ | apart |
| R shoulder | | $\Delta R < \lambda$ | $\Delta(\delta(L,R)) > \lambda$ |
| R hip | | | |

Table 5: Table listing the locations and hand motions included in the feature vectors. The conditions for motion are shown with the label. Where $x, y, z$ is the position of the hand, either left ($L$) or right ($R$), $\Delta$ indicates a change from one frame to the next and $\delta(L,R)$ is the Euclidean distance between the left and right hands. $\lambda$ is the threshold value to reduce noise and increase generalisation, this is set to be a quarter the head height. $F^R$ and $F^L$ are the motion feature vectors relating to the right and left hand respectively.

the type of motion occurring is derived directly from deterministic rules on the x,y,z co-ordinates of the hand position. The resulting feature vector is a binary representation of the found linguistic values. The list of 17 motion features extracted is shown in table 5.

# 6 Learnt Location Features

As with the motion information, the location classifiers are built using boosting from features based on first appearance and then tracked data. These are also tested on the same, hand labelled, dataset as used in Section 4.

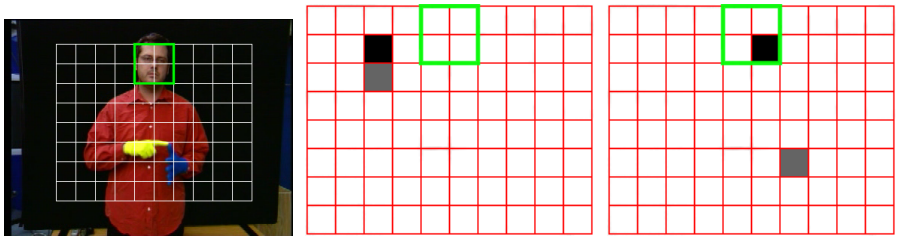
## *6.1 Appearance based*

In order that the sign can be localised in relation to the signer, a grid is applied to the image, dependent upon the position and scale of the face detection. Each cell in the grid is a quarter of the face size and the grid is 10 rectangles wide by 8 deep, as shown in figure 13a. Each skin segmented frame is quantised into this grid and a cell fires if over 50% of its pixels are made up of skin. This is shown in equation 9 where $R_{wc}$ is the weak classifier response and $\Lambda_{skin}(x,y)$ is the likelihood that a pixel contains skin. $f$ is the face height and all the grid values are relative to this dimension.

$$
\begin{aligned}
R_{wc} &= \left\{ \begin{array}{c} 1 \; if \; \frac{f^2}{8} < \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} (\Lambda_{skin}(i,j) > 0) \\ 0 \; otherwise \end{array} \right\} \\
\forall G_x, \forall G_y & \left\{ \begin{array}{c} x_1 = G_x f, x_2 = (G_x + 0.5) f, \\ y_1 = G_y f, y_2 = (G_y + 0.5) f \end{array} \right\} \\
G_x &= \{-2.5, -2, -1.5 \ldots 2\} \\
G_y &= \{-4, -3.5, -3 \ldots 0\}
\end{aligned} \tag{9}
$$

For each of the *Location* sub-units, a classifier was built via AdaBoost to combine cells which fire for each particular SU, examples of these classifiers are shown in figures 13b and (c). Note how the first cell to be picked by the boosting (shown in black) is the one directly related to the area indicated by the SU label. The second cell chosen by boosting

either adds to this location information, as in figure 13b, or comments on the stationary, non-dominant hand, as in figure 13c.



(a) The grid applied over the signer (b) On Right Shoulder (c) Lower Face/Chin

Figure 13: Grid features for two stage classification. (a) shows an example of the grid produced from the face dimensions while (b) and (c) show grid features chosen by boosting for two of the 18 *Location* SUs. The thick green box shows the face location and the first and second features chosen, are shown in black and grey respectively.

## 6.2 Tracking Based

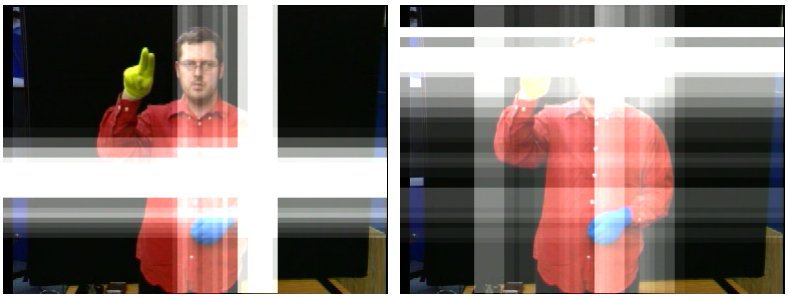
The positions of the hands are given by the tracking and the position of the face can be found using the Viola Jones face detector (Viola and Jones, 2001). Classifiers are then built which consider relational distances. Each classifier operates on an x or y feature, $i$, within the tracking vector, $\mathbf{o}$, comparing it to an upper and lower limit, $\mathcal{T}_U$ and $\mathcal{T}_L$ respectively. If the value falls within this range, then the classifier fires. The upper and lower limits are individual to each classifier and calculated relative to the size ($f$) and position ($f_{xy}$) of the signer's face, see equation 10.

$$\begin{aligned} \mathcal{T}_L &= f_{xy} + nf \\ \mathcal{T}_U &= \mathcal{T}_L + sf \\ n &\in \{-3, -2.9, -2.8 \ldots 3\} \\ s &\in \{0.1, 0.2, 0.3 \ldots 1\} \\ R_{wc} &= \left\{ \begin{array}{c} 1 \; if \; \mathcal{T}_L < \mathbf{o}_i \leq \mathcal{T}_U \\ 0 \; otherwise \end{array} \right\} \end{aligned} \tag{10}$$

Classifiers can work on either the x or y co-ordinates of either the dominant or non-dominant hand. Each classifier covers a strip of a given constant width, either in the x or y plane. Boosting is used to combine these weak classifier strips, to create areas relative to the signer as shown in figure 14. The strips are shown by increasing the luminosity of the pixels. When many weak classifiers overlap, the area turns white. As can be seen, the white areas coincide with the area being learnt, *i.e.* 14a 'face' and 14b 'upper arm'.

## 6.3 Hierarchical Labels

Some of the SU types contain values which are not mutually exclusive, this needs to be taken into account when labelling and using SU data. Using boosting to train classifiers requires positive and negative examples. For best results, examples should not be contaminated, *i.e.* the positive set should not contain negatives and the negative set should not contain positives. Trying to distinguish between an area and its sub-areas can prove futile, *e.g.* the mouth is also on the face and therefore there are likely to be false negatives in the training set when training face against mouth. A hierarchy can be created of areas and their sub-areas as shown in figure 15; a classifier is trained for each node of the tree, using examples which belong to it, or its children, as positive data. Examples which do not belong to



(a) Upper Arm (b) Face

Figure 14: Examples of tracked *Location* classifiers for the areas 'upper arm' and 'face'. Boosting combines strips in the x and y planes to show where the hand is expected to be for each *Location* label. The lighter the area in the picture the more strips are overlaying it.

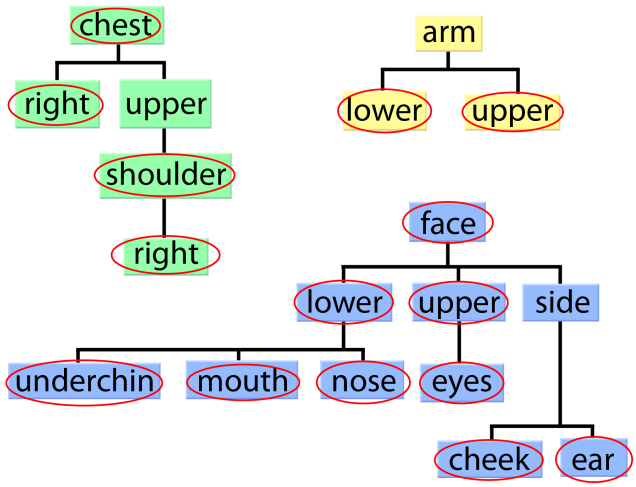



Figure 15: The three *Location* SU trees used for classification
There are three separate trees, based around areas of the body which do not overlap. Areas on the leaves of the tree are sub-areas of their parent nodes. The red rings indicate that there are exact examples of that type in the data set.

it, its parent or its child nodes provide negative data. This eliminates false negatives from the data set and avoids confusion. In figure 15 the ringed nodes show the SUs for which there exist examples.

## 6.4 Results

Using the hierarchical label system, see table 7, the first thing to note is that confusions are only considered between labels of the same level (*e.g.* 'face' is compared to 'arm' but not to 'face_lower' or 'arm_upper'). This is because the data for some of the lower levels is used as positive training data for the higher labels, so a direct comparison cannot be made with the non-hierarchical labels due to the changes in the way the confusion matrices need to be constructed.

The appearance based classifiers again result in lower classification rates, table 6, giving a true-positive average of 48.86%, this is poor in comparison to the 79.84% achieved by

the classifiers trained on the tracked data. The main cause of this is the difficulty the classifiers tend to have in distinguishing between different parts of the face. If the face regions are removed, then the true positive rate increases to 70.27%. This result is explained by considering the method of extracting the positions, looking at the grid in figure 13a, it can be seen that there are only four boxes covering the face and as such there is not enough resolution to be able to distinguish between the different regions of the face. It is also notable that with the appearance, there is little ability to be able to distinguish between dominant and non-dominant hand. The tracking based classifiers separate the two hands and have much finer granularity, allowing the classifiers to better distinguish between the labels. For comparison, when attempting to learn tracking based classifiers without the hierarchical labels, the average classification rate is 46.95%.

Table 6: Confusion matrix of *Location* appearance based classifiers. Confusion is only considered between labels on the same level. All values are percentages. Diagonal Mean : 48.86% (70.27% for non face related), Max : 100%, Min : 0%, Std Dev : 37.95pp

| | arm | chest | face | arm_lower | arm_upper | chest_right | chest_upper | face_lower | face_side | face_upper | chest_upper_shoulder | face_lower_mouth | face_lower_nose | face_lower_underchin | face_side_cheek | face_side_ear | face_upper_eyes | chest_upper_shoulder_right |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| arm | **42** | 2 | 0 | | | | | | | | | | | | | | | |
| chest | 58 | **74** | 0 | | | | | | | | | | | | | | | |
| face | 0 | 25 | **100** | | | | | | | | | | | | | | | |
| arm_lower | | | | **84** | 0 | 2 | 0 | 7 | 4 | 0 | | | | | | | | |
| arm_upper | | | | 16 | **100** | 0 | 10 | 6 | 0 | 0 | | | | | | | | |
| chest_right | | | | 0 | 0 | **0** | 0 | 0 | 0 | 0 | | | | | | | | |
| chest_upper | | | | 0 | 0 | 9 | **73** | 18 | 3 | 0 | | | | | | | | |
| face_lower | | | | 0 | 0 | 23 | 1 | **34** | 15 | 3 | | | | | | | | |
| face_side | | | | 0 | 0 | 38 | 8 | 12 | **5** | 0 | | | | | | | | |
| face_upper | | | | 0 | 0 | 28 | 7 | 23 | 73 | **97** | | | | | | | | |
| chest_upper_shoulder | | | | | | | | | | | **79** | 8 | 8 | 8 | 0 | 18 | 16 | |
| face_lower_mouth | | | | | | | | | | | 0 | **23** | 23 | 25 | 100 | 7 | 9 | 0 |
| face_lower_nose | | | | | | | | | | | 0 | 24 | **24** | 21 | 0 | 0 | 5 | 0 |
| face_lower_underchin | | | | | | | | | | | 0 | 3 | 3 | **1** | 0 | 0 | 0 | 0 |
| face_side_cheek | | | | | | | | | | | 0 | 10 | 10 | 11 | **0** | 0 | 4 | 0 |
| face_side_ear | | | | | | | | | | | 21 | 11 | 11 | 12 | 0 | **57** | 45 | 19 |
| face_upper_eyes | | | | | | | | | | | 0 | 13 | 13 | 13 | 0 | 0 | **5** | 0 |
| chest_upper_shoulder_right | | | | | | | | | | | | 8 | 8 | 8 | 0 | 18 | 16 | **81** |

Table 7: Confusion matrix of *Location* hierarchical tracking based classifiers. Confusion is only considered between labels on the same level. All values are percentages. Diagonal Mean : 79.84%, Max : 98.32%, Min : 34.70%, Std Dev : 17.73pp

| | arm | chest | face | arm_lower | arm_upper | chest_right | chest_upper | face_lower | face_side | face_upper | chest_upper_shoulder | face_lower_mouth | face_lower_nose | face_lower_underchin | face_side_cheek | face_side_ear | face_upper_eyes | chest_upper_shoulder_right |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| arm | **97** | 7 | 1 | | | | | | | | | | | | | | | |
| chest | 4 | **35** | 4 | | | | | | | | | | | | | | | |
| face | 0 | 57 | **95** | | | | | | | | | | | | | | | |
| arm_lower | | | | **71** | 37 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| arm_upper | | | | 19 | **54** | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| chest_right | | | | 0 | 3 | **88** | 0 | 0 | 0 | 0 | | | | | | | | |
| chest_upper | | | | 0 | 1 | 12 | **97** | 0 | 0 | 0 | | | | | | | | |
| face_lower | | | | 9 | 4 | 0 | 1 | **78** | 12 | 4 | | | | | | | | |
| face_side | | | | 0 | 0 | 0 | 1.6 | 14 | **75** | 15 | | | | | | | | |
| face_upper | | | | 1 | 1 | 0 | 1 | 8 | 13 | **81** | | | | | | | | |
| chest_upper_shoulder | | | | | | | | | | | **91** | 0 | 0 | 0 | 0 | 0 | 0 | |
| face_lower_mouth | | | | | | | | | | | 1 | **59** | 17 | 4 | 23 | 6 | 8 | 0 |
| face_lower_nose | | | | | | | | | | | 0 | 12 | **83** | 0 | 0 | 0 | 12 | 0 |
| face_lower_underchin | | | | | | | | | | | 0 | 4 | 0 | **95** | 0 | 0 | 0 | 0 |
| face_side_cheek | | | | | | | | | | | 6 | 11 | 0 | 1 | **67** | 13 | 1 | 0 |
| face_side_ear | | | | | | | | | | | 2 | 2 | 0 | 0 | 9 | **81** | 5 | 2 |
| face_upper_eyes | | | | | | | | | | | 0 | 13 | 0 | 1 | 1 | 0 | **75** | 0 |
| chest_upper_shoulder_right | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | **98** |

# 7 Deterministic Location Features

Again, here we consider deterministic features for location. Where the full skeleton is available, multiple joints are used to identify the positions. In the case of 2D tracking results where only the head and hands are available, other solutions must be used.

## *7.1 2D Tracking*

The x and y co-ordinates are quantised into a codebook based on the signer's head position and scale in the image. For any given hand position $(x_h, y_h)$ the quantised version $(x'_h, y'_h)$ is achieved using the quantisation rules shown in equation 11, where $(x_f, y_f)$ is the face position and $(w_f, h_f)$ is the face size.

$$\begin{aligned} x' &= (x_h - x_f)/w_f \\ y' &= (y_h - y_f)/h_f \end{aligned} \tag{11}$$

This gives values in the range of $y' \in \{0..10\}$ and $x' \in \{0..8\}$ (for a standard signing space) which can be expressed as a binary feature vector of size 40.

## *7.2 3D Tracking*

A subset of linguistic *Location* features can be accurately positioned using the skeleton returned by the Kinect™ tracker. As such, the location features are calculated using the distance of the dominant hand from skeletal joints. A feature will fire if the dominant hand is closer than $H^{head}/2$ of the joint in question. A list of the 9 joints considered is shown in table 5 and displayed to scale in figure 16. While displayed in 2D, the regions surrounding the joints are actually 3D spheres. When the dominant hand (in this image shown by the smaller red dot) moves into the region around a joint then that feature will fire. In the example shown it would be difficult for two features to fire at once. When in motion, the left hand and elbow regions may overlap with other body regions meaning that more than one feature can fire at a time. However while these are in some part related to HamNoSys

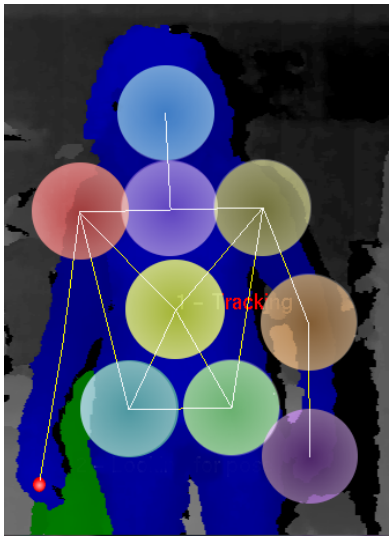


Figure 16: Body joints used to extract sign locations

they are not all directly related. By using the HamNoSys concept of signing space the position of the hands can be described by 2 pieces of information; the first uses the $x, z$

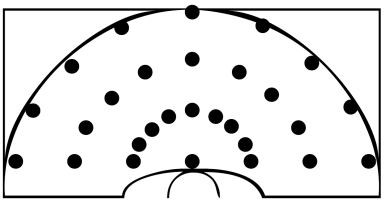


Figure 17: The signing space positions mapped from polar co-ordinates

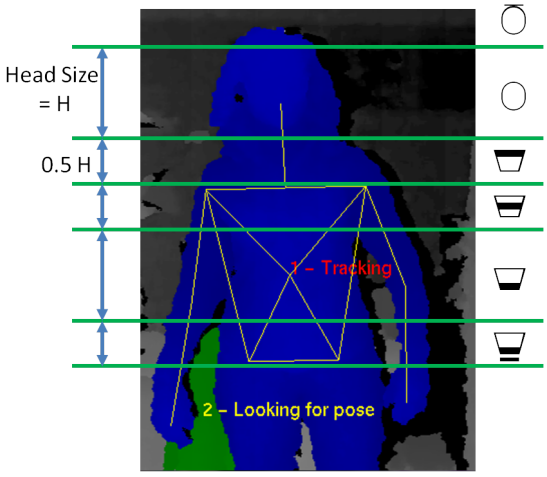



Figure 18: The signing space positions mapped from heights

combination in a system based on polar co-ordinates. The second uses the $y$ information in combination with the skeleton (for the Kinect$^{TM}$) and the shoulders (for the 3D video data).

Using polar co-ordinates the signing space is split into nine angular sections emanating from the centre of the signer. The sections are centred around 0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5 and 180 degrees and each contain 3 possible locations as shown in figure 17. The radial distance between locations is calculated using the size of the head.

The signing space heights are located using the skeleton where available, see figure 18. Where the skeleton is not available, such as for the 3D tracked corpus, approximate values are derived from manually labelled shoulder positions. This approach works well when the signer is relatively stable, as during the lemmata, but is less effective during animated signing sessions. The heights are all calculated using the head height. This is calculated as the distance between the neck and the head for skeletal data and between the average shoulder height and the head for video data.

# 8 Learnt HandShape Features

SiGML lists 12 basic handshapes which can be augmented using finger bending, thumb position and openeness characteristics as well as position using palm and finger orientation. Currently this work focusses on just the basic handshapes, building multi-modal classifiers to account for the different orientations. A list of these handshapes is shown in figure 19.

Unfortunately, linguists annotating sign do so only at the *sign* level while most SUs occur for only *part* of a sign. Also, not only do handshapes change throughout the sign, they are also made more difficult to recognise due to motion blur. Using the motion of the hands, the sign can be split into its component parts (as in Liddell *et al*. Liddell and Johnson (1989)), that are then aligned with the sign annotations. This is done using the work of Vogler (2011) and is described in more detail in Section 9.3. Then the frames which are most likely to contain a static handshape are extracted for training.

Note that, as shown in figure. 20, one single SiGML/HamNoSys class (in this case 'finger2') may contain examples very varied in appearance, making visual classification an extremely difficult task.

## *8.1 Appearance*

The extracted handshapes are classified using a multi-class random forest. Random forests were proposed by Amit and Geman (1997) and Breiman (2001). They have been shown to yield good performance on a variety of classification and regression problems, and can be trained efficiently in a parallel manner, allowing training on large feature vectors and datasets. In our system, the forest is trained from automatically extracted samples of all 12 handshapes in the dataset shown in figure 19. Since signs may have multiple handshapes or several instances of the same handshape, the total occurrences are greater than the number of signs, however they are not equally distributed between the handshape classes. The large

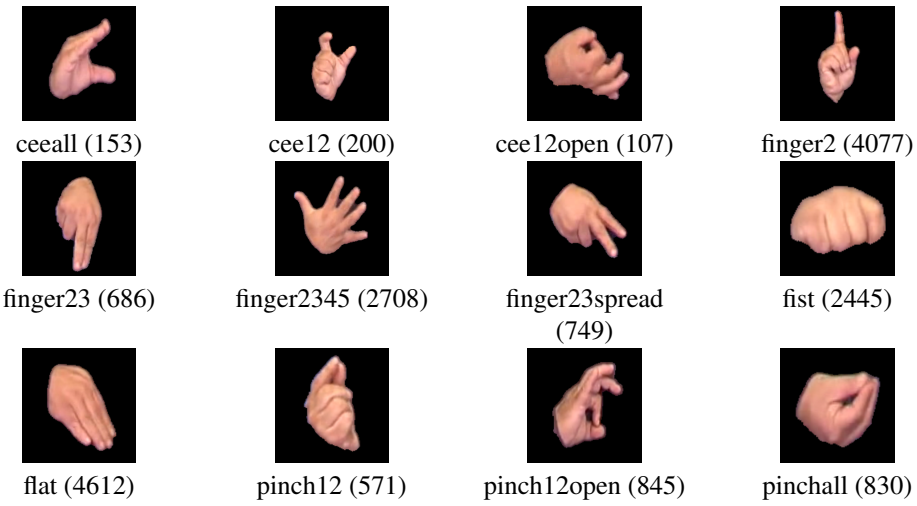



Figure 19: The base handshapes (Number of occurrences in the dataset)

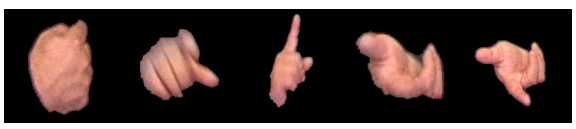


Figure 20: A variety of examples for the HamNoSys/SiGML class 'finger2'.

| handshape | predictions | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| flat | **0.35** | 0.19 | 0.09 | 0.03 | 0.08 | 0.06 | 0.03 | 0.06 | 0.06 | 0.01 | 0.03 | 0.01 |
| fist | 0.03 | **0.69** | 0.02 | 0.04 | 0.11 | 0.05 | | 0.02 | 0.03 | | | 0.02 |
| finger2345 | 0.16 | 0.19 | **0.36** | 0.02 | 0.03 | 0.05 | | 0.06 | 0.02 | 0.03 | 0.06 | 0.01 |
| finger2 | 0.02 | 0.33 | 0.07 | **0.31** | 0.11 | 0.05 | 0.02 | 0.03 | 0.02 | | 0.04 | |
| pinchall | 0.03 | 0.09 | 0.04 | 0.01 | **0.65** | 0.11 | 0.01 | 0.01 | | | | 0.04 |
| pinch12 | 0.02 | 0.20 | 0.01 | 0.02 | 0.13 | **0.56** | 0.01 | | 0.01 | | 0.01 | 0.02 |
| finger23 | 0.05 | 0.17 | 0.04 | 0.02 | 0.05 | 0.04 | **0.54** | 0.01 | | | 0.07 | 0.01 |
| pinch12open | 0.03 | 0.12 | 0.07 | 0.01 | 0.15 | 0.04 | 0.01 | **0.56** | | | | 0.01 |
| cee12 | 0.01 | 0.05 | 0.01 | 0.03 | 0.04 | | | 0.01 | **0.82** | | 0.01 | |
| cee12open | | | | | 0.01 | | | | | **0.99** | | |
| finger23spread | 0.01 | 0.15 | 0.02 | | 0.06 | 0.01 | 0.05 | 0.02 | | | **0.65** | |
| ceeall | 0.01 | 0.08 | 0.03 | | 0.08 | 0.01 | 0.02 | 0.01 | | | 0.01 | **0.77** |

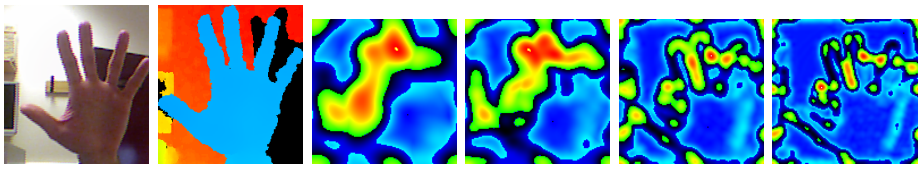
Table 8: Confusion matrix of the handshape recognition, for all 12 classes.

disparities in the number of examples between classes (see figure 19) may bias the learning, therefore the training set is rebalanced before learning by selecting 1,000 random samples for each class, forming a new balanced dataset. The forest used consists of $N = 100$ multi-class decision trees $T_i$, each of which is trained on a random subset of the training data. Each tree node splits the feature space in two by applying a threshold on one dimension of the feature vector. This dimension (chosen from a random subset) and the threshold value are chosen to yield the largest reduction in entropy in the class distribution. This recursive partitioning of the dataset continues until a node contains a subset of examples that belong to one single class, or if the tree reaches a maximal depth (set to 10). Each leaf is then labelled according to the mode of the contained samples. As a result, the forest yields a probability distribution over all classes, where the likelihood for each class is the proportion of trees that voted for this class. Formally, the confidence that feature vector $x$ describes the handshape $c$ is given by:

$$p[c] = \frac{1}{N} \sum_{i<N} \delta_c(T_i(x)), \tag{12}$$

where $N$ is the number of trees in the forest, $T_i(x)$ is the leaf of the $i$th tree $T_i$ into which $x$ falls, and $\delta_c(a)$ is the Kronecker delta function ($\delta_c(a) = 1$ iff. $c = a$, $\delta_c(a) = 0$ otherwise).

The performance of this handshape classification on the test set is recorded on table 8, where each row corresponds to a shape, and each column corresponds to a predicted class (empty cells signify zero). Lower performance is achieved for classes that were more frequent in the dataset. This is likely to be due to the large amount of variability in these classes; see, *e.g.*, figure 20 for the case of 'finger2'—the worst performing case.



(a) color (b) depth (c) scale 0 (d) scale 1 (e) scale 2 (f) scale 3

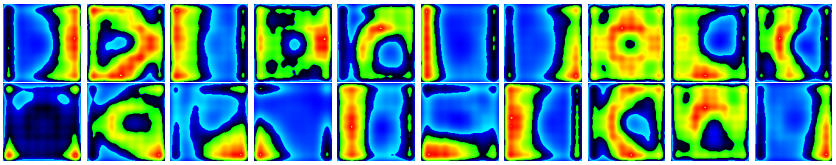
Figure 21: Illustration of the feature extraction process. a) colour patch extracted around the hand; b) depth patch extracted around the hand; c-f) responses of the Gabor convolution for each scale, cumulated over all orientations.

## *8.2 Depth*

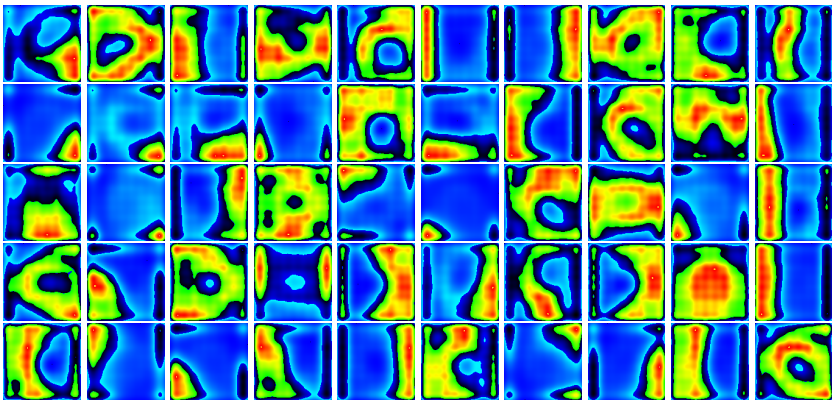
In the case of the live demo, we can make use of depth information provided by the Microsoft Kinect device. The feature extracted is similar as in the visual domain. The depth image is resized to $128 \times 128$ and filtered using a bank of Gabor filters at 6 scales and 8 orientations. The resulting response maps are averaged over $8 \times 8$ regularly spaced and overlapping radial basis functions, resulting in a vector of dimension 3072. This type of depth features were used successfully for ASL finger spelling recognition in Pugeault and Bowden (2011)—this is illustrated in Figure 21.

For the more challenging purpose of encoding the 12 generic SiGML handshapes, we recorded a dataset of 14 persons doing each handshape in a variety of arm poses and orientation, and clustered the resulting depth features in 20, 50 and 100 clusters using K-means—see Figure 22. The distances from a given depth feature to each cluster provides us with a 170-dimension feature vector that describes each handshape.

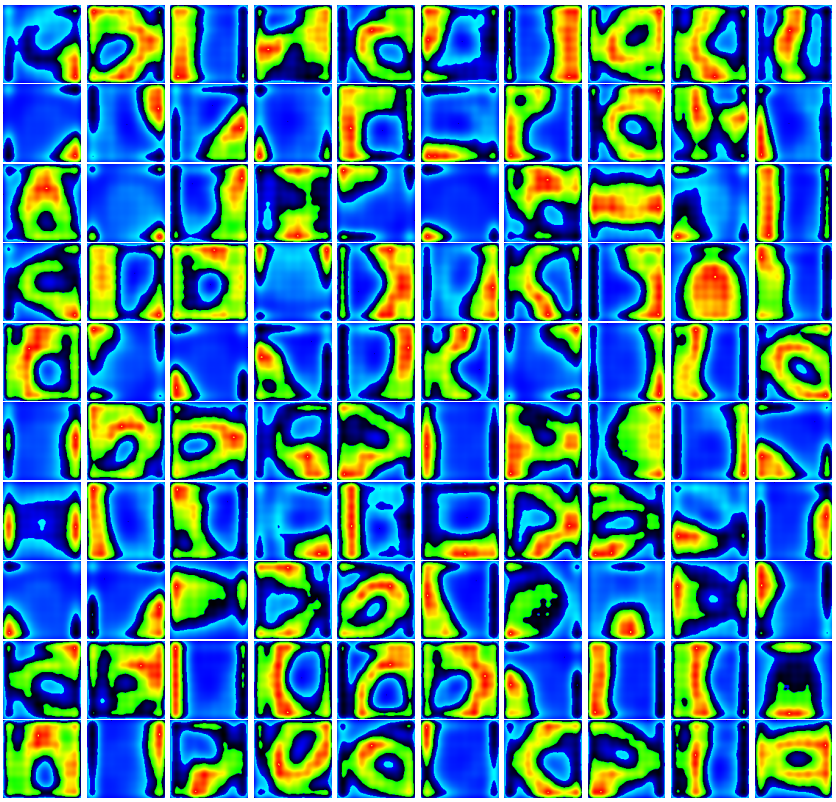
Alternatively, we generated a binary vector of the same dimension by using soft assignment: a data point is associated to any cluster if its distance to the centre is less than twice the cluster's isotropic standard deviation (95% bound).



20 clusters



50 clusters



100 clusters

Figure 22: Illustration of the feature cluster centres extracted from the depth patches.

# 9 Phonetic-Based Sub-Unit Modelling (NTUA)

In this section we report a novel approach to address the issues outlined in the introduction (Pitsikalis et al, 2011). This is based on two aspects: (1) converting SL annotations into structured sequential phonetic labels which is covered by C. Vogler in Tech.Rep. of WP4 (Vogler, 2011) and (2) incorporating these labels into a SU-based statistical framework for training and alignment so as to map the visual data onto meaningful phonetic models.

## *9.1 Phonetic Based Sub-units, Training, Alignment and Mapping to Phonetic Transcriptions*

From the Postures, Detentions, Transitions, Steady Shifts (PDTS) system we obtain a sequence of HamNoSyssymbols that we incorporate as phonetic labels into a statistical system that maps the visual data onto these phonetic labels. This can be further evaluated via incorporation into a recognition system. Data-driven-only SUs, without any phonetic information, adapt well to specific feature spaces. However, they produce phonetically meaningless SU labels.

We call the process of incorporating the phonetic information "Phonetic-based Sub-unit Construction". This has been presented in (Pitsikalis et al, 2011) being the first time that this specific mapping is addressed in an automatic, statistical, and systematic way. The procedure includes the following inputs: (1) *Phonetic transcriptions* of SL, provided by the PDTS system (Vogler, 2011), and (2) the corresponding underlying *visual data and features* from processing the video data and the feature extraction. The procedures involve: (1) phonetic SU construction and training, (2) phonetic label alignment and segmentation, and (3) lexicon construction.

## *9.2 Data and Visual Processing*

**Data:** The *GSL Lemmas Corpus* consists of 1046 isolated signs, 5 repetitions each, from two native signers (male and female). The videos have a uniform background and a resolution of 1440x1080 pixels, recorded at 25 frames per second interlaced.
**Visual Processing:** For the segmentation and detection of the signer's hands and head in the GSL Lemmas Corpus, we employed a skin colour model utilizing a Gaussian Markov Model (GMM), accompanied by morphological processing to enhance skin detection. Moreover, for tracking we employed forward-backward linear prediction, and template matching, in order to disambiguate occlusions. The adopted approach is described in (Roussos et al, 2010). The extracted feature vector has five components, and consists of 1) the planar coordinates of the dominant hand, 2) its instantaneous direction, and 3) its corresponding velocity.

## *9.3 Phonetic Sub-Unit Model Training Alignment and Time Segmentation*

For each phonetic label we train one SU HMM employing the features from the visual processing. These SUs have both phonetic labels from the PDTS structure, and statistical parameters stemming from the data-driven models, as a result of the training step. We use different HMM parameters for each type of SU. Distinguishing between movements (T/E) and postures/detentions (P/D) corresponds to making a distinction between dynamic and static segments. This is also consistent with the concepts in the old Movement-Hold model (Liddell and Johnson, 1989) and on our previous work on data-driven dynamic-static SUs (Pitsikalis et al, 2010).

After Sub-Unit training we construct a recognition network with the trained HMMs and decode each feature sequence via the Viterbi algorithm. This results in a sequence of

(a) E (b) P (c) T (d) P (e) T (f) P (g) E

Figure 23: Sign for PILE: Segments after incorporation of PDTS phonetic labels into Phonetic Sub-unit Construction, Training and Alignment. Superimposed start and end frames of each sequence of segments, accompanied with an arrow for transitions and epenthesis. Each segment corresponds to a single phonetic label. PDTS segments labels are of type Epenthesis (E), Posture (P), Transition (T)

phonetic PDTS labels, together with their respective starting and ending frames. Doing this for all sequences results in a lexicon with segmentation boundaries for each PDTS label.

Fig. 23 shows an example of the segmentation acquired during the decoding, which illustrates the sequence of phonetic SUs for the above-mentioned sign for "PILE". Each image corresponds to a phonetic PDTS segment produced by the decoding. For visualization, we adopt the following conventions: (1) For T and E segments, we superimpose their respective initial and final frames. We also highlight specific movement trajectories with an arrow from the initial to the final hand position in the respective segment. (2) For P and D segments, we show only the first frame of the segment, as the hand does not move within them.

## *9.4 Conclusions*

In this section we have presented work on gesture analysis and the mapping of visual data on phonetic models. It explores new ways for the incorporation of linguistic evidence, in the form of sequences of phonetic labels, which are extracted from sign language annotations. This incorporation is based on the visual evidence from simple features, such as the tracking of the dominant hand. We construct phonetic SUs that carry phonetic information, unlike previous data-driven approaches. The phonetic modelling also gives us the time alignment information that the phonetic labels initially lack. So far, we have used only motion related features, however the concepts can be extended to other cues, such as handshape-features.

# 10 Global-Local Active Appearance Modelling for Facial Cues (NTUA)

This section is contributed by the NTUA partner. In this work, we tackle the problem of face tracking and modelling in sign language videos with head pose variations and partial head occlusions. We also measure and analyse low level facial cues in order to correlate them with higher level of linguistic information. The proposed framework for facial expression modelling and feature extraction in SL videos is based on a Global and Local AAM (*GLAAM*) (Rodomagoulakis et al, 2011) providing local features for places of interest and tracking robustness (e.g. against occlusions, pose variation) at the same time. The overall system incorporates prior information in terms of static shape and texture and deals acceptable in most cases with extreme pose variation. In cases with rapid SL performing, we improve the robustness of the Global AAM tracking by applying an accurate initialization on its parameters. The extracted local features offer a compact and descriptive representation of the specific facial regions of interest. The proposed Global+Local Active Appearance facial Modelling (GLAAM) is evaluated for GSL data.

Non-manual cues are starting to receive more attention in the SLR literature (Michael et al, 2011; Agris et al, 2008; Vogler and Goldenstein, 2008; Erdem and Sclaroff, 2002). Model-based approaches for facial features, like those seen in Active Appearance Models (AAM)(Cootes et al, 1998), offer an inherent benefit over holistic or local, purely appearance-based ones (raw pixels, optical flow, eigenfaces) in the sense they can account for both shape and texture and also treat the problems of face detection, tracking and coding in a unified framework (Lanitis et al, 1997). However, such approaches have not been systematically applied in sign language analysis and recognition.

## *10.1 Global AAM Tracking*

In the task of face tracking in a video sequence using AAMs, the model recovers the parametric description for each face instance through an optimization process. We take advantage of adaptive and constrained inverse compositional methods (Papandreou and Maragos, 2008) for improved accuracy and performance during the AAM fitting process. We denote the vector $\tilde{\mathbf{p}} = [\mathbf{t}_{1:4}, \mathbf{p}_{1:N_s}]^T$ as the AAM shape parameters, where $\mathbf{t}$ are the similarity transform parameters necessary for the face's global scaling, rotation and translation. The AAM texture parameters are $\boldsymbol{\lambda}_{1:N_t}$ defined as the projections over the $N_t$ most important eigentextures of the aligned facial database images.

We trained subject-specific AAMs on one signer from the GSL database. The PCA projection error is minimized and the representation of the facial variance becomes nearly lossless when we keep 90% of the total variance. Table 9 contains information about the size of the training set, the number of landmark points $N$, the mean shape's resolution and the shape and texture models PCA modes of variation $N_s$ and $N_t$ accordingly for each AAM model. The training set have been selected so that there is balance between the extreme pose and mouth instances. Even though the AAMs are very effective on the detection

| *# frames* | *# train images* | $N$ | $N_s$ | $N_t$ | *Mean shape resolution* |
|---|---|---|---|---|---|
| 8082 | 70 | 46 | 41 | 63 | 2385 |

Table 9: AAM training on GSL

and tracking of high-variation local movements of facial parts, they are not suitable for robust face tracking within a large pose variation, which is intense in sign language tasks. We deal with the *initialization problem* by using modules for skin colour and face detection in order to initialize the similarity transform parameters $\mathbf{t}_{1:4}$. Based on the orientation of the head ellipsis we assign an initial value in initial rotation. Then, we estimate scaling by

measuring the ellipsis width and finally we initialize translation by aligning the centroids of AAM mean shape and facial skin mask. In cases with insufficient fitting result, we apply a minor binary searching task within a small window around the ellipsis' centroid targeting on the initial MSE error's minimization.

## 10.2 Fitting and Tracking Results

The fitting and tracking result is sufficiently accurate on all the non-occlusion frames. Especially on the facial regions of interest such as eyes and mouth the fitting is very effective, due to intense texture differences. The accuracy of tracking and the importance of an effective initialization framework are even more highlighted by the fact that there are extreme mouthings and pose variations on a SL video. Figure 24 shows some fitting results on GSL database.

Figure 24: AAM fitting results on GSL video frames with pose variation and mouthings.

## 10.3 Global and Local AAM Visual Features

The *GAAM* is trained to model the appearance of the whole face in facial expression analysis and the 3D pose estimation. Nevertheless, it is uncertain and inaccurate in response with the displayed variability in small scales dealing with certain face micro-structures. The articulation of non-manual markers in signing combines distinct facial components. In this direction, the efficiency and usability of the global fitting is extended by exploring the existence of a local-global, local-local hierarchical relationship. Once the global face graph is fitted, we extend with supplementary *LAAMs* given the geometrical global-local relations on the labelled AAM face graph. Local AAM fitting is initialized on the corresponding sub-graphs and takes a few iterations to fine-tune on the local structures. Depending on the local shape and texture variability, each model has different complexity (see table 10) due to changes of the corresponding facial regions.

| AAM Type | # Eigen-Shapes | # Eigen-Textures |
|---|---|---|
| Global | 6 | 6 |
| LBrow + Rbrow | 3+3 | 9+7 |
| LEye + REye | 2+4 | 5+4 |
| Mouth | 4 | 8 |

Table 10: Global-Local AAM parameters

## 10.4 Facial Features

Using the trained local models, we can derive features from the signer's face in specific local areas of linguistic interest. The parameter selection in table 11 is based on the classification results. As a consequence, shape parameters are sufficient to measure the state of the eye brows without texture information. On the other hand, eyes and mouth varying in appearance also. The GLAAM configuration also offers insight of the hand position in

| Locality | Shape | Texture |
|---|---|---|
| brows | $[t_{1-4}, p_{1-2}]$ | - |
| eyes | $[t_{1-4}, p_{1-3}]$ | $\lambda_{1-9}$ |
| mouth | $[t_{1-4}, p_{1-2}]$ | $\lambda_{1-8}$ |

Table 11: Local AAM features

the face region in terms of uncertainty measurements in local AAMs as a by product result from the GLAAM framework.

Linear Discriminant Analysis (LDA) on the appearance feature space, which is the concatenation of the shape and texture ones, results in compactness and discrimination among the classes we consider. The final feature vector contains the $N_f = (N_q)/2$ most discriminant LDA components where $N_q$ is the number of the input PCA components $p_i, \lambda_i$ plus the similarity parameters $t_i$ (see Sec. 10.1). LDA features discriminate the examined classes for facial cues. Our work focuses on the Dicta-Sign Greek sign language corpus

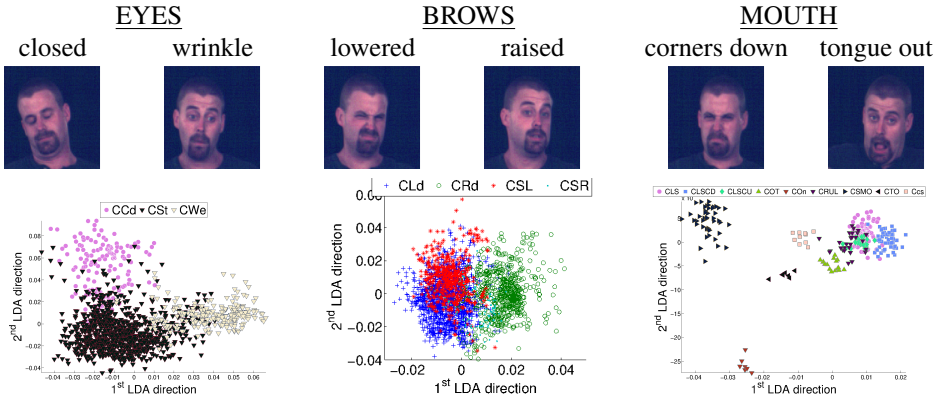



Figure 25: (top row) Realizations of the considered dominant classes in each classification task (bottom row) Their corresponding distributions projected on the 1st and 2nd dimensions of the LDA feature space

while initial development work has been done on the BU-400 American SL corpus. The classification experiments on isolated facial actions are based on local and global features extracted by the proposed framework (Rodomagoulakis et al, 2011). The facial parameters we classify are ($T_1$) the eye brows position (raised "CRd", lowered "CLd"), ($T_2$) the aperture of the eyes (opened "COd", wrinkle "CWe") and ($T_3$) the state of the mouth region (closed "CCd", tongue out "CTo" etc.). The selected regions are considered as the most informative in the SLR literature. For the GSL data, due to the lack of facial annotations, our results are either qualitative or they are based on unsupervised techniques, in order to further assist facial cue feature extraction.

# 11 Mapping Facial Features to NonManuals

Although there is significant facial information on signers during sign language conversations, sign language recognition systems generally focus on manual components of signing (i.e. hand signals). The purpose of this work is to investigate the use of non-manual features (i.e. facial expressions) to aid sign language recognition systems. To this end, this work comprises of two main stages: Non-Manual Features (NMF) detection, and *conversation context* recognition. By conversation context, we include; questions, negation, yes/no answers, *etc*. A similar application was proposed by Michael et al (2011). However, we intend on extending the multiple instance learning approach by using a Social Dynamics Model (SDM) which also incorporates temporal information and graphical representation.

## *11.1 Tracking Facial Features*

In order to track facial features, the method of linear predictor flocks is used. Each Linear Predictor (LP) provides a mapping from sparse template differences to the displacement vector of a tracked facial feature. Multiple LPs can then be grouped into rigid flocks to track a single feature point with greater robustness and accuracy. The linear predictors are extended into a full linear regression function by introducing a bias factor, resulting in *biased-LPs* . To learn the required LPs and the locations of the visual information used by them, an LP selection method based on probabilistic selection is used. Here, the proposed selection method migrates member LPs into optimal positions, essentially identifying the visual context for tracking a particular feature point. The tracking robustness is further increased by integrating the selected LP flocks into a hierarchical multi-resolution framework. Additional improvement in performance is obtained by employing a simple shape model to correct gross tracking errors of individual feature points.

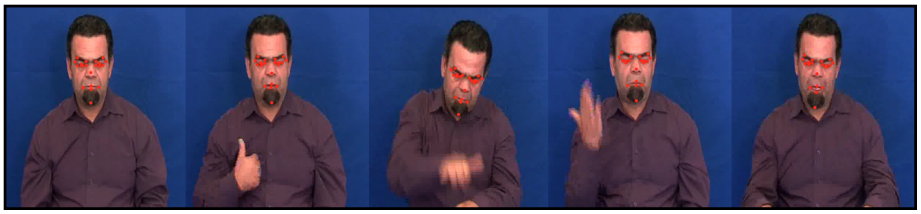


Figure 26: Kostas non-manual feature tracking using an LP tracker (Ong et al, 2009)

The non-manual features for the GSL lemmata subject Kostas are tracked using this Linear Predictor Tracker. As shown in Figure 26, this incorporates the main facial features; eyes, lips, eyebrows, eye pupils, nose, and chin. Pixel intensity and body contour information will be used to detect body pose orientation, hence, body tracking was not necessary.

By linking the NMF-list to the lemmata, the ability to search and extract specific video segments, signs, and NMF features was possible. Figure 27 shows the tracker results for a Kostas video segment. The text written in black at the top right corner of Figure 27 are the labelled NMF features for the given sign, and the text in red are preliminary NMF detection results. As such, we are able to observe the signer's behaviour against their expected NMF labels.

## *11.2 Geometric Features*

The next step is to encode facial information using geometric features as used by Sheerman-Chase et al (2009). Formally used for detecting complex emotions (Kaliouby and Robinson, 2005), encoding geometric features provide a simple way to extract static facial features that are independent of pose. This includes head yaw, head pitch, head roll, eyebrow

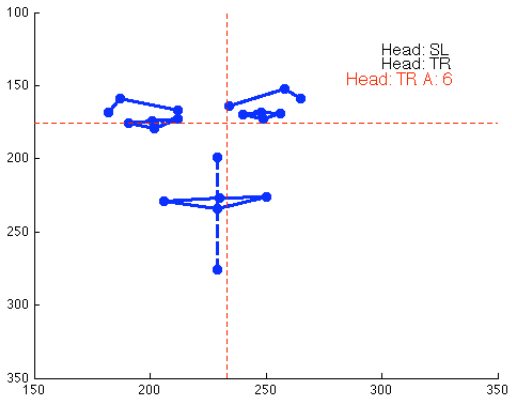



Figure 27: Kostas non-manual feature tracker marker results

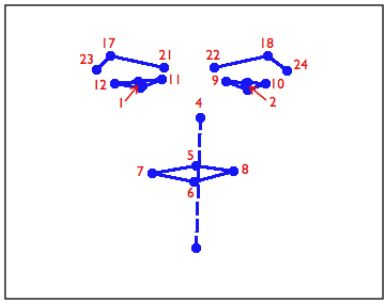



Figure 28: Marker points used for geometric feature generation

raise, lip pull/pucker, *etc*. Currently, we use 11 geometric features which are extracted as shown in Table 12. These correlate to marker points shown on Figure 28.

The heuristic features shown in Table 12 have also be used to detect non-verbal communication in natural conversation (Sheerman-Chase et al, 2009). In this work, we intend on applying these features to generate a SDM. SDM is a data driven approach to generate distinct rules that are predominate to a given conversational context. This is achieved using Association Rule data mining, whereby association rule is a relationship of the form $\{R_i^A\} \Rightarrow R_i^C$ where $R_i^A$ is a set of antecedents, and $R_i^C$ is the consequence. The belief of each rule is measured by a *support* and *confidence* value. The *support* measures the statistical significance of a rule, it is the probability that a transaction contains itemset $R_i^A$.

$$sup(\{R_i^A\} \Rightarrow R_i^C) = sup(\{R_i^A\} \cup R_i^C) \tag{13}$$

The *confidence* is the number of occurrences in which the rule is correct, relative to the number of cases in which it is applicable.

$$conf = \frac{sup(\{R_i^A\} \cup R_i^C)}{sup(R_i^A)} * 100 \tag{14}$$

*Temporal bagging* within a set temporal window will be used to enforce a temporal coherence between features.

The idea is to identify distinct rules for a given conversation context and use these rules to generate SDM classifiers. These classifier will be able to recognise specific conversation contexts, which can then be applied to further assist sign language recognition. This work is ongoing and will continue until the end of the project

| | |
|---|---|
| Head yaw | $\frac{\overline{P_9P_{10}}}{\overline{P_{11}P_{12}}}$ |
| Head pitch | $P_4[t] - P_4[t-1]$ |
| Head roll | $\angle P_9P_{11}$ |
| Eyebrow raise | $\frac{(\overline{P_{11}P_{21}}+\overline{P_1P_{17}}+\overline{P_{12}P_{23}})_t}{(\overline{P_{11}P_{21}}+\overline{P_1P_{17}}+\overline{P_{12}P_{23}})_0}$ |
| Lip pull/pucker | $\frac{(\overline{AP_7}+\overline{AP_8})_t-(\overline{AP_7}+\overline{AP_8})_0}{(\overline{AP_7}+\overline{AP_8})_0}$ |
| Right eye horizontal | $\frac{P_{11}P_{12}\cdot P_{11}P_1}{\lvert P_{11}P_{12}\rvert}$ |
| Left eye horizontal | $\frac{P_9P_{10}\cdot P_9P_2}{\lvert P_9P_{10}\rvert}$ |
| Mean eye horizontal | $\frac{(P_{11}P_{12}\cdot P_{11}P_1)(P_9P_{10}\cdot P_9P_2)}{2\lvert P_{11}P_{12}\rvert\lvert P_9P_{10}\rvert}$ |
| Right eye vertical | $\frac{\lvert P_{11}P_{12}\times P_{11}P_1\rvert}{\lvert P_{11}P_{12}\rvert}$ |
| Left eye vertical | $\frac{\lvert P_{11}P_{12}\times P_{11}P_1\rvert}{\lvert P_{11}P_{12}\rvert}$ |
| Mean eye vertical | $\frac{\lvert P_{11}P_{12}\times P_{11}P_1\rvert\lvert P_{11}P_{12}\times P_{11}P_1\rvert}{2\lvert P_{11}P_{12}\rvert\lvert P_{11}P_{12}\rvert}$ |

Table 12: Geometric features used to detect NMF during signing

# 12 Summary

In summary, this document gives a comprehensive overview of work performed in WP1, covering tracking and feature extraction methods developed within the Dicta-Sign project and how these can be mapped to linguistic annotation, such as HamNoSys and it's animation counterpart Sign Gesture Mark-up Language (SiGML). These features are employed in WP2 for recognition, in WP4 for linguistic modelling, in WP6 for corpus processing and in WP7 for demonstration systems, although in each case different subsets of the work are used dependent upon the nature and requirements.

# References


Agris U, Zieren J, Canzler U, Bauer B, Kraiss KF (2008) Recent developments in visual sign language recognition. Universal Access in the Information Society 6:323–362

Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Computation 9:1545–1588

Basha T, Moses Y, Kiryati N (2010) Multi-view scene flow estimation: A view centered variational approach. In: Proceedings of the Proc. Conf. on Computer Vision & Pattern Recognition, San Francisco, CA, USA, pp 1506–1513, DOI 10.1109/CVPR.2010.5539791

Bauer B, Kraiss KF (2001) Towards an automatic sign language recognition system using subunits. In: Proc. of Int'l Gesture Workshop, vol 2298, pp 64–75

Bourdev L, Malik J (2009) Poselets: Body part detectors trained using 3d human pose annotations. In: Proceedings of the Proc. Int'l Conf. on Computer Vision, Kyoto, Japan, URL `http://www.eecs.berkeley.edu/~lbourdev/poselets`

Bourdev L, Maji S, Brox T, Malik J (2010) Detecting people using mutually consistent poselet activations. In: Proceedings of the Proc. Europ. Conf. on Computer Vision, Springer, Heraklion, Crete, pp 168 – 181, DOI 10.1007/978-3-642-15567-3_13

Bowden R, Windridge D, Kadir T, Zisserman A, Brady M (2004) A linguistic feature vector for the visual interpretation of sign language. In: Proc. Europ. Conf. on Computer Vision

Breiman L (2001) Random forests. Machine Learning 45:5–32, URL http://dx.doi.org/10.1023/A:1010933404324

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Chapman and Hall

Cooper H (2010) Sign language recognition : Generalising to more complex corpora. PhD thesis, Centre For Vision Speech and Signal Processing, University Of Surrey

Cooper H, Pugeault N, Bowden R (2011) Reading the signs: A video based sign dictionary. In: Proceedings of the Workshop : ARTEMIS workshopIEEE International Conference on Computer Vision, Barcelona, Spain

Cootes T, Edwards G, Taylor C (1998) Active appearance models. In: Computer Vision ECCV98, Lecture Notes in Computer Science, vol 1407, Springer Berlin / Heidelberg, pp 484–498

Elliott R, Cooper H, Glauert J, Bowden R, Lefebvre-Albaret F (2011) Search-by-example in multilingual sign language databases. In: Proceedings of the Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT), Dundee, Scotland

Erdem U, Sclaroff S (2002) Automatic detection of relevant head gestures in american sign language communication. In: Proc. Int'l Conf. on Pattern Recognition, vol 1, pp 460 – 463 vol.1

Fang G, Gao X, Gao W, Chen Y (2004) A novel approach to automatically extracting basic units from chinese sign language. In: Proc. Int'l Conf. on Pattern Recognition, vol 4, pp 454–457

Ferrari V, Marin-Jimenez M, Zisserman A (2008) Progressive search space reduction for human pose estimation. In: Proceedings of the Proc. Conf. on Computer Vision & Pattern Recognition, Anchorage, AK, USA, pp 1 – 8, DOI 10.1109/CVPR.2008.4587468

Hadfield S, Bowden R (2011) Kinecting the dots: Particle based scene flow from depth sensors. In: Proceedings of the Proc. Int'l Conf. on Computer Vision, Barcelona, Spain

Han J, Awad G, Sutherland A (2009) Modelling and segmenting subunits for sign language recognition based on hand motion analysis. Pattern Recognition Letters 30:623–633

Hanke T, Schmaling C (2004) Sign Language Notation System. Institute of German Sign Language and Communication of the Deaf, Hamburg, Germany, URL http://www.sign-lang.uni-hamburg.de/projects/hamnosys.html

Holt B, Ong EJ, Cooper H, Bowden R (2011) Putting the pieces together: Connected poselets for human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision Workshop : Consumer Depth Cameras for Computer Vision, Barcelona, Spain

Hu MK (1962) Visual pattern recognition by moment invariants. IRE Transactions on Information Theory IT-8:179–187

Huguet F, Devernay F (2007) A variational method for scene flow estimation from stereo sequences. In: Proceedings of the Proc. Int'l Conf. on Computer Vision, Rio de Janario, Brazil, pp 1–7, DOI 10.1109/ICCV.2007.4409000

Kadir T, Bowden R, Ong E, Zisserman A (2004) Minimal training, large lexicon, unconstrained sign language recognition. In: Proceedings of the Proc. British Machine Vision Conference, Kingston, UK, vol 2, pp 939 – 948

Kaliouby R, Robinson P (2005) Real-time inference of complex mental states from facial expressions and head gestures. Real-time vision for human-computer interaction pp 181–200

Lanitis A, Taylor CJ, Cootes T (1997) Automatic interpretation and coding of face images using flexible models. IEEE Trans on Pattern Analysis and Machine Intelligence 19:743–756

Liddell SK, Johnson RE (1989) American sign language: The phonological base. Sign Language Studies 64:195 – 278

Michael N, Yang P, Liu Q, Metaxas D, Neidle C (2011) A framework for the recognition of nonmanual markers in segmented sequences of american sign language. In: Proceedings of the British Machine Vision Conference, BMVA Press, pp 124.1–124.12, http://dx.doi.org/10.5244/C.25.124

Moeslund T, Hilton A, Krüger V (2006) A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104(2-3):90 – 126, DOI 10.1016/j.cviu.2006.08.002

Ong EJ, Lan Y, Theobald BJ, Harvey R, Bowden R (2009) Robust facial feature tracking using selected multi-resolution linear predictors. In: Int. Conf. Computer Vision ICCV 2009

OpenNI organization (2010) OpenNI User Guide. OpenNI organization, last viewed 20-04-2011 18:15

Papandreou G, Maragos P (2008) Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In: Proc. Conf. on Computer Vision & Pattern Recognition, DOI 10.1109/CVPR.2008.4587540

Pitsikalis V, Theodorakis S, Maragos P (2010) Data-driven sub-units and modeling structure for continuous sign language recognition with multiple cues. In: LREC Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies

Pitsikalis V, Theodorakis S, Vogler C, Maragos P (2011) Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In: Proceedings of the International Conference IEEE Computer Society Conference on Computer Vision and Pattern RecognitionWorkshop : Gesture Recognition, Colorado Springs, CO, USA

PrimeSense Inc (2010) Prime Sensor$^{TM}$NITE 1.3 Algorithms notes. PrimeSense Inc., last viewed 20-04-2011 18:15

Pugeault N, Bowden R (2011) Spelling it out: Real–time ASL fingerspelling recognition. In: Proceedings of the Proc. Int'l Conf. on Computer VisionWorkshop : Consumer Depth Cameras for Computer Vision, Barcelona, Spain

Rodomagoulakis I, Theodorakis S, Pitsikalis V, Maragos P (2011) Experiments on global and local active appearance models for analysis of sign language facial expressions. In: Proc. of Int'l Gesture Workshop

Roussos A, Theodorakis S, Pitsikalis V, Maragos P (2010) Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In: Proceedings of the International Conference European Conference on Computer VisionWorkshop : SGA, Heraklion, Crete

Schapire RE, Singer Y (1999) Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3):297–336, DOI 10.1023/A:1007614523901

Sheerman-Chase T, Ong E, Bowden R (2009) Feature selection of facial displays for detection of non verbal communication in natural conversation. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, pp 1985–1992

Sutton-Spence R, Woll B (1999) The Linguistics of British Sign Language: An Introduction. Cambridge University Press

Valli C, Lucas C, Mulrooney KJ (2005) Linguistics of American Sign Language: An Introduction. Gallaudet University Press

Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the Proc. Conf. on Computer Vision & Pattern Recognition, Kauai, HI, USA, vol 1, pp 511 – 518

Vogler C (2011) Extraction of segmental phonetic structures from hamnosys annotations. Tech. Rep. D4.2, Institute for Language and Speech Processing, Greece

Vogler C, Goldenstein S (2008) Facial movement analysis in asl. Universal Access in the Information Society 6:363–374

Yin P, Starner T, Hamilton H, Essa I, Rehg J (2009) Learning the basic units in american sign language using discriminative segmental feature selection. In: Proc. ICASSP, pp 4757–4760