Dynamic Affine-Invariant Shape-Appearance Handshape Features and Classification in Sign Language Videos

Anastasios Roussos TROUSSOS@EECS.QMUL.AC.UK
Queen Mary, University of London
School of Electronic Engineering and Computer Science
Mile End Road, London E1 4NS, UK

Stavros Theodorakis STH@CS.NTUA.GR
Vassilis Pitsikalis VPITSIK@CS.NTUA.GR
Petros Maragos MARAGOS@CS.NTUA.GR
National Technical University of Athens
School of Electrical and Computer Engineering
Zografou Campus, Athens 15773, Greece

Editor: Isabelle Guyon and Vassilis Athitsos

Abstract

We propose the novel approach of dynamic affine-invariant shape-appearance model (Aff-SAM) and employ it for handshape classification and sign recognition in sign language (SL) videos. Aff-SAM offers a compact and descriptive representation of hand configurations as well as regularized model-fitting, assisting hand tracking and extracting handshape features. We construct SA images representing the hand's shape and appearance *without* landmark points. We model the variation of the images by linear combinations of eigenimages followed by affine transformations, accounting for 3D hand pose changes and improving model's compactness. We also incorporate static and dynamic handshape priors, offering robustness in occlusions, which occur often in signing. The approach includes an *affine signer adaptation* component at the visual level, without requiring training from scratch a new singer-specific model. We rather employ a short development data set to adapt the models for a new signer. Experiments on the Boston-University-400 continuous SL corpus demonstrate improvements on handshape classification when compared to other feature extraction approaches. Supplementary evaluations of sign recognition experiments, are conducted on a multi-signer, 100-sign data set, from the Greek sign language lemmas corpus. These explore the fusion with movement cues as well as signer adaptation of Aff-SAM to multiple signers providing promising results.

Keywords: affine-invariant shape-appearance model, landmarks-free shape representation, static and dynamic priors, feature extraction, handshape classification

1. Introduction

Sign languages (SL), that is, languages that convey information via visual patterns, commonly serve as an alternative or complementary mode of human communication. The visual patterns of SL are formed mainly by handshapes and manual motion, as well as by non-manual patterns. The hand localization and tracking in a sign video as well as the derivation of features that reliably describe the configuration of the signer's hand are crucial for successful handshape classification. All the above are essential components for automatic sign language recognition systems or for gesture based human-computer interaction. Nevertheless, these tasks still pose several challenges, which are mainly due to the fast movement and the great variation of the hand's 3D shape and pose.

In this article, we propose a novel modeling of the shape and dynamics of the hands during signing that leads to efficient handshape features, employed to train statistical handshape models and finally for handshape classification and sign recognition. Based on 2D images acquired by a monocular camera, we employ a video processing approach that outputs reliable and accurate masks for the signer's hands and head. We construct *Shape-Appearance (SA) images* of the hand by combining 1) the hand's shape, as determined by its 2D hand mask, with 2) the hand's appearance, as determined by a normalized mapping of the colors inside the hand mask. The proposed modeling does not employ any landmark points and bypasses the point correspondence problem. In order to design a model of the variation of the SA images, which we call *Affine Shape-Appearance Model* (Aff-SAM), we modify the classic linear combination of eigenimages by incorporating *2D affine transformations*. These effectively account for various changes in the 3D hand pose and improve the model's compactness. After developing a procedure for the training of the Aff-SAM, we design a robust hand tracking system by adopting regularized model fitting that exploits prior information about the handshape and its dynamics. Furthermore, we propose to use as handshape features the Aff-SAM's eigenimage weights estimated by the fitting process.

The extracted features are fed into statistical classifiers based on Gaussian mixture models (GMM), via a supervised training scheme. The overall framework is evaluated and compared to other methods in extensive handshape classification experiments. The SL data are from the Boston University BU400 corpus (Neidle and Vogler, 2012). The experiments are based on manual annotation of handshapes that contain 3D pose parameters and the American Sign Language (ASL) handshape configuration. Next, we define classes that account for varying dependency of the handshapes w.r.t. the orientation parameters. The experimental evaluation addresses first, in a qualitative analysis the feature spaces via a cluster quality index. Second, we evaluate via supervised training a variety of classification tasks accounting for dependency w.r.t. orientation/pose parameters, with/without occlusions. In all cases we also provide comparisons with other baseline approaches or more competitive ones. The experiments demonstrate improved feature quality indices as well as classification accuracies when compared with other approaches. Improvements in classification accuracy for the non-occlusion cases are on average of 35% over baseline methods and 3% over more competitive ones. Improvements by taking into account the occlusion cases are on average of 9.7% over the more competitive methods.

In addition to the above, we explore the impact of Aff-SAM features in a sign recognition task based on statistical data-driven subunits and hidden Markov models. These experiments are applied on data from the Greek Sign Language (GSL) lemmas corpus (DictaSign, 2012), for two different signers, providing a test-bed for the fusion with movement-position cues, and as evaluation of the affine-adapted SA model to a new signer, for which there has been no Aff-SAM training. These experiments show that the proposed approach can be practically applied to multiple signers without requiring training from scratch for the Aff-SAM models.

2. Background and Related Work

The first step of a hand gesture analysis system is the localization of the hands. This is usually implemented using several types of visual features, as skin color, edge information, shape and motion. Color cues are applicable because of the characteristic colors of the human skin. Many methods, including the one presented here, use skin color segmentation for hand detection (Argyros and Lourakis, 2004; Yang et al., 2002; Sherrah and Gong, 2000). Some degree of robustness to illumination changes can be achieved by selecting color spaces, as the *HSV*, *YCbCr* or the *CIE-Lab*, that separate the chromaticity from the luminance components (Terrillon et al., 2000; Kakumanu et al., 2007). In our approach, we adopt the *CIE-Lab* color space, due to its property of being perceptually uniform. Cui and Weng (2000) and Huang and Jeng (2001) employ motion cues assuming the hand is the only moving object on a stationary background, and that the signer is relatively still.

The next visual processing step is the hand tracking. This is usually based on blobs (Starner et al., 1998; Tanibata et al., 2002; Argyros and Lourakis, 2004), hand appearance (Huang and Jeng, 2001), or hand boundary (Chen et al., 2003; Cui and Weng, 2000). The frequent occlusions during signing make this problem quite challenging. In order to achieve robustness against occlusions and fast movements, Zieren et al. (2002), Sherrah and Gong (2000) and Buehler et al. (2009) apply probabilistic or heuristic reasoning for simultaneous assignment of labels to the possible hand/face regions. Our strategy for detecting and labeling the body-parts shares similarities with the above. Nevertheless, we have developed a more elaborate preprocessing of the skin mask, which is based on the mathematical morphology and helps us separate the masks of different body parts even in cases of overlaps.

Furthermore, a crucial issue to address in a SL recognition system is hand feature extraction, which is the focus of this paper. A commonly extracted positional feature is the 2D or 3D center-of-gravity of the hand blob (Starner et al., 1998; Bauer and Kraiss, 2001; Tanibata et al., 2002; Cui and Weng, 2000), as well as motion features (e.g., Yang et al., 2002; Chen et al., 2003). Several works use geometric measures related to the hand, such as shape moments (Hu, 1962; Starner et al., 1998) or sizes and distances between fingers, palm, and back of the hand (Bauer and Kraiss, 2001), though the latter employs color gloves. In other cases, the contour that surrounds the hand is used to extract translation, scale, and/or in-plane rotation invariant features, such as Fourier descriptors (Chen et al., 2003; Conseil et al., 2007).

Segmented hand images are usually normalized for size, in-plane orientation, and/or illumination and afterwards principal component analysis (PCA) is often applied for dimensionality reduction and descriptive representation of handshape (Sweeney and Downton, 1996; Birk et al., 1997; Cui and Weng, 2000; Wu and Huang, 2000; Deng and Tsui, 2002; Dreuw et al., 2008; Du and Piater, 2010). Our model uses a similar framework but differs from these methods mainly in the following aspects. First, we employ a more general class of transforms to align the hand images, namely affine transforms that extend both similarity transforms, used, for example, by Birk et al. (1997) and translation-scale transforms as in the works of Cui and Weng (2000), Wu and Huang (2000) and Du and Piater (2010). In this way, we can effectively approximate a wider range of changes in the 3D hand pose. Second, the estimation of the optimum transforms is done simultaneously with the estimation of the PCA weights, instead of using a pipeline to make these two sets of estimations. Finally, unlike all the above methods, we incorporate combined static and dynamic priors, which make these estimations robust and allow us to adapt an existing model on a new signer.

Closely related to PCA approaches, active shape and active appearance models (Cootes and Taylor, 2004; Matthews and Baker, 2004) are employed for handshape feature extraction and recognition (Ahmad et al., 1997; Huang and Jeng, 2001; Bowden and Sarhadi, 2002; Fillbrandt et al., 2003). Our proposed shape-appearance model follows the same paradigm with these methods but differs: the modeled images are Shape-Appearance images and the image warps are not controlled by the shape landmarks but more simply by the 6 parameters of the affine transformation. In this

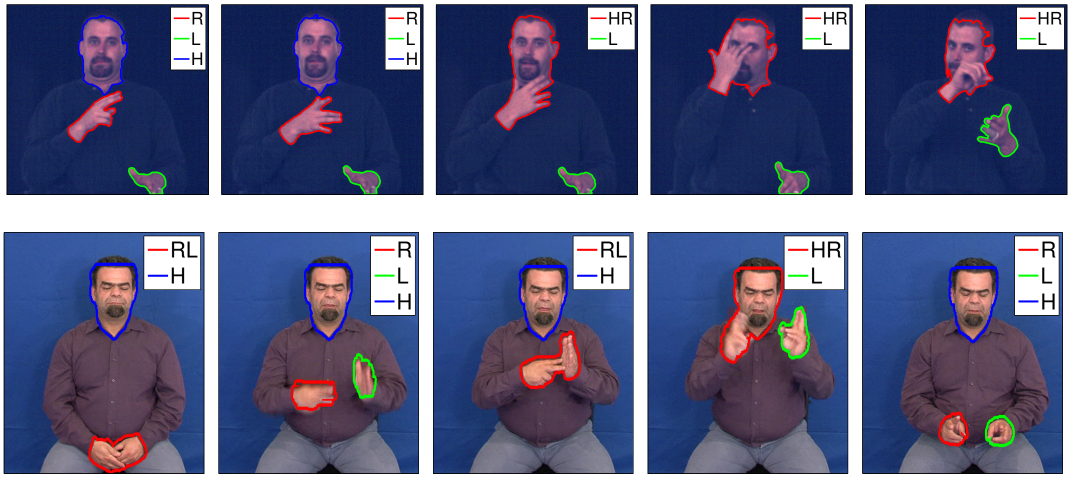


Figure 1: Output of the initial hands and head tracking in two videos of two different signers, from different databases. Example frames with extracted skin region masks and assigned body-part labels *H* (head), *L* (left hand), *R* (right hand).

way, it avoids shape representation through landmarks and the cumbersome manual annotation related to that.

Other more general purpose approaches have also been seen in the literature. A method earlier employed for action-type features is the histogram of oriented gradients (HOG): these descriptors are used for the handshapes of a signer (Buehler et al., 2009; Liwicki and Everingham, 2009; Ong et al., 2012). Farhadi et al. (2007) employ the scale invariant feature transform (SIFT) descriptors. Finally, Thangali et al. (2011) take advantage of linguistic constraints and exploit them via a Bayesian network to improve handshape recognition accuracy. Apart from the methods that process 2D hand images, there are methods built on a 3D hand model, in order to estimate the finger joint angles and the 3D hand pose (Athitsos and Sclaroff, 2002; Fillbrandt et al., 2003; Stenger et al., 2006; Ding and Martinez, 2009; Agris et al., 2008). These methods have the advantage that they can potentially achieve view-independent tracking and feature extraction; however, their model fitting process might be computationally slow.

Finally, regarding our related work, Roussos et al. (2010b) have included a short description of an initial tracking system similar to the one we adopt here. A preliminary version of the Aff-SAM method was presented by Roussos et al. (2010a). This is substantially extended here in many aspects, the main of which are the following: 1) We incorporate dynamic and static handshape priors offering robustness in cases of occlusions, 2) We develop an affine signer adaptation component, exploring the adaptation of Aff-SAM to multiple signers, 3) Extensive handshape classification experiments are presented, 4) Sign recognition experiments are conducted on a multi-signer database. In the sign recognition experiments of Section 8, we employ the handshape subunits construction presented by Roussos et al. (2010b). Finally, Theodorakis et al. (2012) and Theodorakis et al. (2011) present preliminary results on movement-handshape integration for continuous sign recognition.

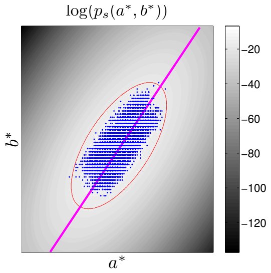


Figure 2: Skin color modeling. Training samples in the a^*-b^* space and fitted pdf $p_s(a^*,b^*)$. The ellipse bounds the colors that are classified to skin, according to the thresholding of $p_s(a^*(x),b^*(x))$. The straight line corresponds to the first PCA eigendirection on the skin samples and determines the projection that defines the mapping $g(I)$ used in the Shape-Appearance images formation.

3. Visual Front-End Preprocessing

The initial step of the visual processing is not the main focus of our method, nevertheless we describe it for completeness and reproducibility. The output of this subsystem at every frame is a set of skin region masks together with one or multiple *labels* assigned to every region, Figure 1. These labels correspond to the *body-parts of interest* for sign language recognition: head (H), left hand (L) and right hand (R). The case that a mask has multiple labels reflects an *overlap* of the 2D regions of the corresponding body-parts, that is, there is an *occlusion* of some body-parts. Referring for example to the right hand, there are the following cases: 1) The system outputs a mask that contains the right hand only, therefore there is *no occlusion* related to that hand, and 2) The output mask includes the right hand as well as other body-part region(s), therefore there is an *occlusion*. As presented in Section 4, the framework of SA refines this tracking while extracting handshape features.

3.1 Probabilistic Skin Color Modeling

We are based on the color cue for body-parts detection. We consider a Gaussian model of the signer's skin color in the perceptually uniform color space *CIE-Lab*, after keeping the two chromaticity components a^*, b^*, to obtain robustness to illumination (Cai and Goshtasby, 1999). We assume that the (a^*,b^*) values of skin pixels follow a bivariate Gaussian distribution $p_s(a^*,b^*)$, which is fitted using a training set of color samples (Figure 2). These samples are automatically extracted from pixels of the signer's face, detected using a face detector (Viola and Jones, 2003).

3.2 Morphological Processing of Skin Masks

In each frame, a first estimation of the skin mask S_0 is derived by thresholding at every pixel x the value $p_s(a^*(x),b^*(x))$ of the learned skin color distribution, see Figures 2, 3(b). The corresponding threshold is determined so that a percentage of the training skin color samples are classified to skin. This percentage is set to 99% to cope with training samples outliers. The skin mask S_0 may contain spurious regions or holes inside the head area due to parts with different color, as for instance eyes, mouth. For this, we regularize S_0 with tools from mathematical morphology (Soille, 2004; Maragos, 2005): First, we use the concept of *holes* $\mathcal{H}(S)$ of a binary image S, that is, the set of background

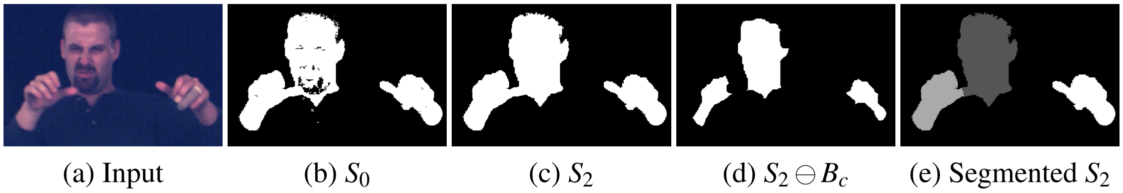

(a) Input (b) S_0 (c) S_2 (d) $S_2 \ominus B_c$ (e) Segmented S_2

Figure 3: Results of skin mask extraction and morphological segmentation. (a) Input. (b) Initial skin mask estimation S_0. (c) Final skin mask S_2 (morphological refinement). (d) Erosion $S_2 \ominus B_c$ of S_2 and separation of overlapped regions. (e) Segmentation of S_2 based on competitive reconstruction opening.

components, not connected to the border of the image. In order to fill also some background regions that are not holes in the strict sense but are connected to the image border passing from a small "canal", we designed a filter that we call *generalized hole filling*. This filter yields a refined skin mask estimation $S_1 = S_0 \cup \mathcal{H}(S_0) \cup \left\{\mathcal{H}(S_0 \bullet B) \oplus B\right\}$ where B is a structuring element with size 5×5 pixels, and $\oplus$ and $\bullet$ denotes Minkowski dilation, closing respectively. The connected components (CCs) of relevant skin regions can be at most three (corresponding to the head and the two hands) and cannot have an area smaller than a threshold A_{min}, which corresponds to the smallest possible area of a hand region for the current signer and video acquisition conditions. Therefore, we apply an *area opening* with a varying threshold value: we find all CCs of S_1, compute their areas and finally discard all the components whose area is not on the top 3 or is less than A_{min}. This yields the final skin mask S_2, see Figure 3(c).

3.3 Morphological Segmentation of the Skin Masks

In the frames where S_2 contains three CCs, these yield an adequate segmentation. On the contrary, when S_2 contains less than three CCs, the skin regions of interest occlude each other. In such cases though, the occlusions are not always essential: different skin regions in S_2 may be connected via a thin connection, Figure 3(c). Therefore we further segment the skin masks of some frames by separating occluded skin regions with thin connections: If S_2 contains $N_{cc} < 3$ connected components, we find the CCs of $S_2 \ominus B_c$, Figure 3(d), for a structuring element B_c of small radius, for example, 3 pixels and discard those CCs whose area is smaller than A_{min}. A number of remaining CCs not greater than N_{cc} implies the absence of any thin connection, thus does not provide any occlusion separation. Otherwise, we use each one of these CCs as the seed of a different segment and expand it to cover S_2. For this we propose a *competitive reconstruction opening*, see Figure 3(e), described by the following iterative algorithm: In every iteration 1) each evolving segment expands using its conditional dilation by the 3×3 cross, relative to S_2, 2) pixels belonging to more than one segment are excluded from all segments. This means that segments are expanded inside S_2 but their expansion stops wherever they meet other segments. The above two steps are repeated until all segments remain unchanged.

3.4 Body-part Label Assignment

This algorithm yields 1) an assignment of one or multiple body-part labels, *head, left* and *right hand*, to all the segments and 2) an estimation of ellipses at segments with multiple labels (occluded).

Note that these ellipses yield a rough estimate of the shapes of the occluded regions and contribute to the correct assignment of labels after each occlusion. A detailed presentation of this algorithm falls beyond the scope of this article. A brief description follows. *Non-occlusions*: For the hands' labels, given their values in the previous frames, we employ a prediction of the centroid position of each hand region taking into account three preceding frames and using a constant acceleration model. Then, we assign the labels based on minimum distances between the predicted positions and the segments' centroids. We also fit one ellipse on each segment since an ellipse can coarsely approximate the hand or head contour. *Occlusions*: Using the parameters of the body-part ellipses already computed from the three preceding frames, we employ similarly forward prediction for all ellipses parameters, assuming constant acceleration. We face non-disambiguated cases by obtaining an auxiliary centroid estimation of each body-part via template matching of the corresponding image region between consecutive frames. Then, we repeat the estimations backwards in time. Forward and backward predictions, are fused yielding a final estimation of the ellipses' parameters for the signer's head and hands. Figure 1 depicts the output of the initial tracking in sequences of frames with non-occlusion and occlusion cases. We observe that the system yields accurate skin extraction and labels assignment.

4. Affine Shape-Appearance Modeling

In this section, we describe the proposed framework of dynamic affine-invariant shape-appearance model which offers a descriptive representation of the hand configurations as well as a simultaneous hand tracking and feature extraction process.

4.1 Representation by Shape-Appearance images

We aim to model all possible configurations of the dominant hand during signing, using directly the 2D hand images. These images exhibit a high diversity due to the variations on the configuration and 3D hand pose. Further, the set of the visible points of the hand is significantly varying. Therefore, it is more effective to represent the 2D handshape without using any landmarks. We thus represent the handshape by implicitly using its binary mask M, while incorporating also the *appearance* of the hand, that is, the color values inside this mask. These values depend on the hand texture and shading, and offer crucial 3D information.

If $I(x)$ is a cropped part of the current color frame around the hand mask M, then the hand is represented by the following *Shape-Appearance (SA) image* (see Figure 4):

$$f(x) = \begin{cases} g(I(x)), & \text{if } x \in M \\ -c_b, & \text{otherwise} \end{cases},$$

where $g : \mathbb{R}^3 \to \mathbb{R}$ maps the color values of the skin pixels to a color parameter that is appropriate for the hand appearance representation. This mapping is more descriptive for hand representation than a common color-to-gray transform. In addition, g is normalized so that the mapped values $g(I)$ of skin colors I have zero mean and unit variance. $c_b > 1$ is a background constant that controls the balance between shape and appearance. As c_b gets larger, the appearance variation gets relatively less weighted and more emphasis is given to the shape part. In the experiments, we have used $c_b = 3$ (that is three times the standard deviation of the foreground values $g(I)$).

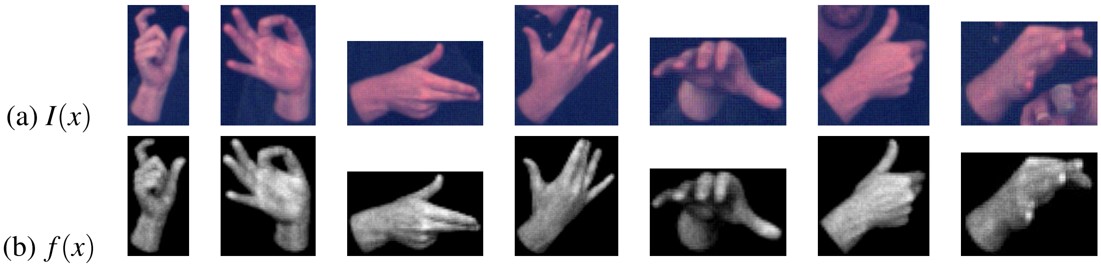


Figure 4: Construction of Shape-Appearance images. (a) Cropped hand images $I(x)$. (b) Corresponding Shape-Appearance images $f(x)$. For the foreground of $f(x)$ we use the most descriptive feature of the skin chromaticity. The background has been replaced by a constant value that is out of the range of the foreground values.

The mapping $g(I)$ is constructed as follows. First we transform each color value I to the *CIE-Lab* color space, then keep only the chromaticity components a^*,b^*. Finally, we output the normalized weight of the first principal eigendirection of the PCA on the skin samples, that is the major axis of the Gaussian $p_s(a^*,b^*)$, see Section 3.1 and Figure 2(c). The output $g(I)$ is the most descriptive value for the skin pixels' chromaticity. Furthermore, if considered together with the training of $p_s(a^*,b^*)$, the mapping $g(I)$ is invariant to global similarity transforms of the values (a^*,b^*). Therefore, the SA images are invariant not only to changes of the luminance component L but also to a wide set of global transforms of the chromaticity pair (a^*,b^*). As it will be described in Section 5, this facilitates the signer adaptation.

4.2 Modeling the Variation of Hand Shape-Appearance Images

Following Matthews and Baker (2004), the SA images of the hand, $f(x)$, are modeled by a linear combination of predefined variation images followed by an affine transformation:

$$f(W_p(x)) \approx A_0(x) + \sum_{i=1}^{N_c} \lambda_i A_i(x), \, x \in \Omega_M \, . \tag{1}$$

$A_0(x)$ is the mean image, $A_i(x)$ are N_c eigenimages that model the linear variation. These images can be considered as affine-transformation-free images. In addition, $\lambda = (\lambda_1 \cdots \lambda_{N_c})$ are the weights of the linear combination and W_p is an affine transformation with parameters $p = (p_1 \cdots p_6)$ that is defined as follows:

$$W_p(x,y) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} .$$

The affine transformation models similarity transforms of the image as well as a significant range of changes in the 3D hand pose. It has a non-linear impact on the SA images and reduces the variation that is to be explained by the linear combination part, as compared to other appearance-based approaches that use linear models directly in the domain of the original images, (e.g., Cui and Weng, 2000). The linear combination of (1) models the changes in the configuration of the hand and the changes in the 3D orientation that cannot be modeled by the affine transform.

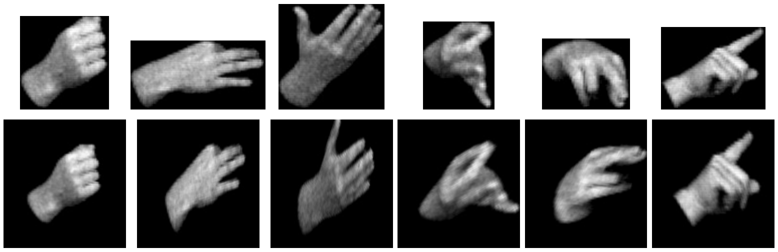


Figure 5: Semi-automatic affine alignment of a training set of Shape-Appearance images. (*Top row*) 6 out of 500 SA images of the training set. (*Bottom row*) Corresponding transformed images, after affine alignment of the training set. A video that demonstrates this affine alignment is available online (see text).

We will hereafter refer to the proposed model as *Shape-Appearance Model (SAM)*. A specific model of hand SA images is defined from the base image $A_0(x)$ and the eigenimages $A_i(x)$, which are statistically learned from training data. The vectors p and λ are the model parameters that fit the model to the hand SA image of every frame. These parameters are considered as features of hand pose and shape respectively.

4.3 Training of the SAM Linear Combination

In order to train the hand SA images model, we employ a representative set of handshape images from frames where the modeled hand is fully visible and non-occluded. Currently, this set is constructed by a random selection of approximately 500 such images. To exclude the variation that can be explained by the affine transformations of the model, we apply a semi-automatic affine alignment of the training SA images. For this, we use the framework of *procrustes analysis* (Cootes and Taylor, 2004; Dryden and Mardia, 1998), which is an iterative process that is repeatedly applying 1-1 alignments between pairs of training samples. In our case, the 1-1 alignments are affine alignments, implemented by applying the inverse-compositional (IC) algorithm (Gross et al., 2005) on pairs of SA images.

The IC algorithm result depends on the initialization of the affine warp, since the algorithm converges to a local optimum. Therefore, in each 1-1 alignment we test two different initializations: Using the binary masks M of foreground pixels of the two SA images, these initializations correspond to the two similarity transforms that make the two masks have the same centroid, area and orientation.[1] Among the two alignment results, the plausible one is kept, according to manual feedback from a user.

It must be stressed that the manual annotation of plausible alignment results is needed only during the training of the SA model, not during the fitting of the model. Also, compared to methods that use landmarks to model the shape (e.g., Cootes and Taylor, 2004; Matthews and Baker, 2004; Ahmad et al., 1997; Bowden and Sarhadi, 2002), the amount of manual annotation during training is substantially decreased: The user here is not required to annotate points but just make a binary decision by choosing the plausible result of 1-1 alignments. Other related methods for aligning sets of images are described by Learned-Miller (2005) and Peng et al. (2010). However, the adopted Pro-

[1] The existence of two such transforms is due to the modulo-π ambiguity of the orientation.

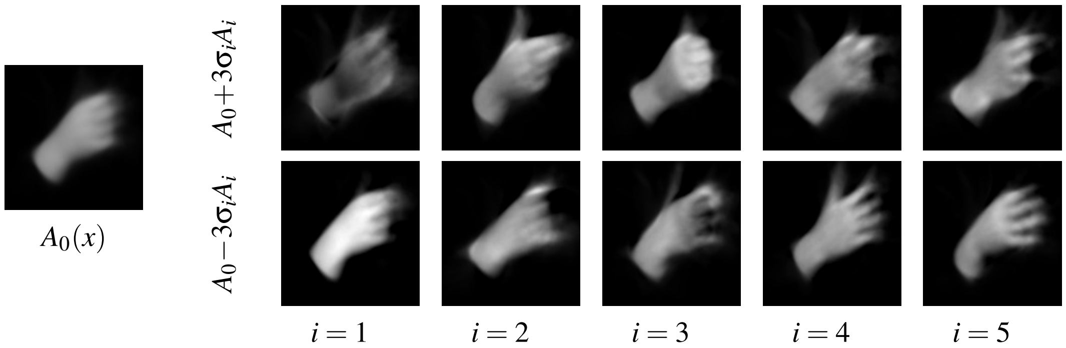


Figure 6: Result of the PCA-based learning of the linear variation images of Equation (1): Mean image $A_0(x)$ and principal modes of variation that demonstrate the first 5 eigenimages. The top (bottom) row corresponds to deviating from A_0 in the direction of the corresponding eigenimage, with a weight of $3\sigma_i$ ($-3\sigma_i$), where σ_i is the standard deviation of the corresponding component.

crustes analysis framework facilitates the incorporation of the manual annotation in the alignment procedure. Figure 5 shows some results from the affine alignment of the training set. For more details, please refer to the following URL that contains a video demonstration of the training set alignment: `http://cvsp.cs.ntua.gr/research/sign/aff_SAM`. We observe that the alignment produces satisfactory results, despite the large variability of the images of the training set. Note that the resolution of the aligned images is 127×133 pixels.

Then, the images A_i of the linear combination of the SA model are statistically learned using principal component analysis (PCA) on the aligned training SA images. The number N_c of eigenimages kept is a basic parameter of the SA model. Using a larger N_c, the model can better discriminate different hand configurations. On the other hand, if N_c gets too large, the model may not generalize well, in the sense that it will be consumed on explaining variation due to noise or indifferent information. In the setup of our experiments, we have practically concluded that the value $N_c = 35$ is quite effective. With this choice, the eigenimages kept explain 78% of the total variance of the aligned images.

Figure 6 demonstrates results of the application of PCA. Even though the modes of principal variation do not correspond to real handshapes, there is some intuition behind the influence of each eigenimage at the modeled hand SA image. For example, the first eigenimage A_1 has mainly to do with the foreground appearance: as its weight gets larger, the foreground intensities get darker and vice-versa. As another example, we see that by increasing the weight of the second eigenimage A_2, the thumb is extended. Note also that when we decrease the weight of A_4 all fingers extend and start detaching from each other.

4.4 Regularized SAM Fitting with Static and Dynamic Priors

After having built the shape-appearance model, we fit it in the frames of an input sign language video, in order to track the hand and extract handshape features. Precisely, we aim to find in every frame n the parameters $\lambda = \lambda[n]$ and $p = p[n]$ that generate a model-based synthesized image that is sufficiently "close" to the current input SA image $f(x)$. In parallel, to achieve robustness against occlusions, we exploit prior information about the handshape and its dynamics. Therefore, we minimize the following energy:

$$E(\lambda, p) = E_{rec}(\lambda, p) + w_S E_S(\lambda, p) + w_D E_D(\lambda, p) \;, \tag{2}$$

where E_{rec} is a reconstruction error term. The terms $E_S(\lambda, p)$ and $E_D(\lambda, p)$ correspond to static and dynamic priors on the SAM parameters λ and p. The values w_S,w_D are positive weights that control the balance between the 3 terms.

The reconstruction error term E_{rec} is a mean square difference defined by:

$$E_{rec}(\lambda, p) = \frac{1}{N_M} \sum_x \left\{ A_0(x) + \sum_{i=1}^{N_c} \lambda_i A_i(x) - f(W_p(x)) \right\}^2 ,$$

where the above summation is done over all the N_M pixels x of the domain of the images $A_i(x)$.

The static priors term $E_S(\lambda, p)$ ensures that the solution stays relatively close to the parameters mean values λ_0,p_0 :

$$E_S(\lambda, p) = \frac{1}{N_c} \|\lambda - \lambda_0\|^2_{\Sigma_\lambda} + \frac{1}{N_p} \|p - p_0\|^2_{\Sigma_p} \;,$$

where N_c and N_p are the dimensions of λ and p respectively (since we model affine transforms, N_p=6). These numbers act as normalization constants, since they correspond to the expected values of the quadratic terms that they divide. Also, Σ_λ and Σ_p are the covariance matrices of λ and p respectively,[2] which are estimated during the training of the priors (Section 4.4.2). We denote by $\|y\|_A$, with A being a $N \times N$ symmetric positive-definite matrix and $y \in \mathbb{R}^N$, the following Mahalanobis distance from y to 0:

$$\|y\|_A \triangleq \sqrt{y^T A^{-1} y} \,.$$

Using such a distance, the term $E_S(\lambda, p)$ penalizes the deviation from the mean values but in a weighted way, according to the appropriate covariance matrices.

The dynamic priors term $E_D(\lambda, p)$ makes the solution stay close to the parameters estimations $\lambda^e = \lambda^e[n]$, $p^e = p^e[n]$ based on already fitted values on adjacent frames (for how these estimations are derived, see Section 4.4.1):

$$E_D(\lambda, p) = \frac{1}{N_c} \|\lambda - \lambda^e\|^2_{\Sigma_{\varepsilon_\lambda}} + \frac{1}{N_p} \|p - p^e\|^2_{\Sigma_{\varepsilon_p}} \;, \tag{3}$$

where $\Sigma_{\varepsilon_\lambda}$ and Σ_{ε_p} are the covariance matrices of the estimation errors of λ and p respectively, see Section 4.4.2 for the training of these quantities too. The numbers N_c and N_p act again as normalization constants. Similarly to $E_S(\lambda, p)$, the term $E_D(\lambda, p)$ penalizes the deviation from the predicted values in a weighted way, by taking into account the corresponding covariance matrices. Since the parameters λ are the weights of the eigenimages $A_i(x)$ derived from PCA, we assume that their mean $\lambda_0 = 0$ and their covariance matrix Σ_λ is diagonal, which means that each component of λ is independent from all the rest.

It is worth mentioning that the energy-balancing weights w_S,w_D are not constant through time, but depend on whether the modeled hand in the current frame is occluded or not (this information is provided by the initial tracking preprocessing step of Section 3). In the occlusion cases, we are

[2] We have assumed that the parameters λ and p are statistically independent.

less confident than in the non-occlusion cases about the input SA image $f(x)$, which is involved in the term $E_{rec}(\lambda, p)$. Therefore, in these cases we obtain more robustness by increasing the weights w_S,w_D. In parallel, we decrease the relative weight of the dynamic priors term $\frac{w_D}{w_S+w_D}$, in order to prevent error accumulation that could be propagated in long occlusions via the predictions λ^e, p^e. After parameters tuning, we have concluded that the following choices are effective for the setting of our experiments: 1) w_S=0.07, w_D=0.07 for the non-occluded cases and 2) w_S=0.98, w_D=0.42 for the occluded cases.

An input video is split into much smaller temporal segments, so that the SAM fitting is *sequential* inside every segment as well *independent* from the fittings in all the rest segments: All the video segments of consecutive non-occluded and occluded frames are found and the middle frame of each segment is specified. For each non-occluded segment, we start from its middle frame and we get 1) a segment with forward direction by ending to the middle frame of the next occluded segment and 2) a segment with backward direction by ending after the middle frame of the previous occluded segment. With this splitting, we increase the confidence of the beginning of each sequential fitting, since in a non-occluded frame the fitting can be accurate even without dynamic priors. In the same time, we also get the most out of the dynamic priors, which are mainly useful in the occluded frames. Finally, this splitting strategy allows a high degree of parallelization.

4.4.1 DYNAMICAL MODELS FOR PARAMETER PREDICTION

In order to extract the parameter estimations λ^e, p^e that are used in the dynamic prior term E_D (3), we use linear prediction models (Rabiner and Schafer, 2007). At each frame n, a varying number $K = K(n)$ of already fitted frames is used for the parameter prediction. If the frame is far enough from the beginning of the current sequential fitting, K takes its maximum value, K_{max}. This maximum length of a prediction window is a parameter of our system (in our experiments, we used $K_{max} = 8$ frames). If on the other hand, the frame is close to the beginning of the corresponding segment, then K varies from 0 to K_{max}, depending on the number of frames of the segment that have been already fitted.

If $K = 0$, we are at the starting frame of the sequential fitting, therefore no prediction from other available frames can be made. In this case, which is degenerate for the linear prediction, we consider that the estimations are derived from the prior means $\lambda^e = \lambda_0$, $p^e = p_0$ and also that $\Sigma_{\varepsilon_\lambda} = \Sigma_\varepsilon$, $\Sigma_{\varepsilon_p} = \Sigma_p$, which results to $E_D(\lambda, p) = E_S(\lambda, p)$. In all the rest cases, we apply the framework that is described next.

Given the prediction window value K, the parameters λ are predicted using the following autoregressive model:

$$\lambda^e[n] = \sum_{\nu=1}^{K} \mathrm{A}_\nu \, \lambda[n \mp \nu] \,,$$

where the $-$ sign ($+$ sign) corresponds to the case of forward (backward) prediction. Also, A_ν are $N_c \times N_c$ weight matrices that are learned during training (see Section 4.4.2). Note that for every prediction direction and for every K, we use a different set of weight matrices A_ν that is derived from a separate training. This is done to optimize the prediction accuracy for the specific case of every prediction window. Since the components of λ are assumed independent to each other, it is reasonable to consider that all weight matrices A_ν are diagonal, which means that each component has an independent prediction model.

As far as the parameters p are concerned, they do not have zero mean and we cannot consider them as independent since, in contrast to λ, they are not derived from a PCA. Therefore, in order to apply the same framework as above, we consider the following re-parametrization:

$$\widetilde{p} = U_p^T (p - p_0) \Leftrightarrow p = p_0 + U_p \widetilde{p} \ ,$$

where the matrix U_p contains column-wise the eigenvectors of Σ_p. The new parameters $\widetilde{p}$ have zero mean and diagonal covariance matrix. Similarly to λ, the normalized parameters $\widetilde{p}$ are predicted using the following model:

$$\widetilde{p}^e[n] = \sum_{\nu=1}^{K} \mathrm{B}_\nu \widetilde{p}[n \mp \nu] \ ,$$

where B_ν are the corresponding weight matrices which again are all considered diagonal.

4.4.2 AUTOMATIC TRAINING OF THE STATIC AND DYNAMIC PRIORS

In order to apply the regularized SAM fitting, we first learn the priors on the parameters λ and p and their dynamics. This is done by training subsequences of frames where the modeled hand is not occluded. This training does not require any manual annotation. We first apply a random selection of such subsequences from videos of the same signer. Currently, the randomly selected subsequences used in the experiments are 120 containing totally 2882 non-occluded frames and coming from 3 videos. In all the training subsequences, we fit the SAM in each frame independently by minimizing the energy in Equation (2) with w_S=w_D=0 (that is without prior terms). In this way, we extract fitted parameters λ, p for all the training frames. These are used to train the static and dynamic priors.

4.4.3 STATIC PRIORS

In this case, for both cases of λ and p, the extracted parameters from all the frames are used as samples of the same multivariate distribution, without any consideration of their successiveness in the training subsequences. In this way, we form the training sets T_λ and T_p that correspond to λ and p respectively. Concerning the parameter vector λ, we have assumed that its mean $\lambda_0 = 0$ and its covariance matrix Σ_λ is diagonal. Therefore, only the diagonal elements of Σ_λ, that is the variances $\sigma^2_{\lambda_i}$ of the components of λ, are to be specified. This could be done using the result of the PCA (Section 4.2), but we employ the training parameters of T_λ that come from the direct SAM fitting, since they are derived from a process that is closer to the regularized SAM fitting. Therefore, we estimate each $\sigma^2_{\lambda_i}$ from the empirical variance of the corresponding component λ_i in the training set T_λ. Concerning the parameters p, we estimate p_0 and Σ_p from the empirical mean and covariance matrix of the training set T_p.

4.4.4 DYNAMIC PRIORS

As already mentioned, for each prediction direction (forward, backward) and for each length K of the prediction window, we consider a different prediction model. The $(K+1)$-plets[3] of samples for each one of these models are derived by sliding the appropriate window in the training sequences. In order to have as good accuracy as possible, we do not make any zero (or other) padding in unknown parameter values. Therefore, the samples are picked only when the window fits entirely inside the

[3] The $(K+1)$-plets follow from the fact that we need K neighbouring samples + the current sample.

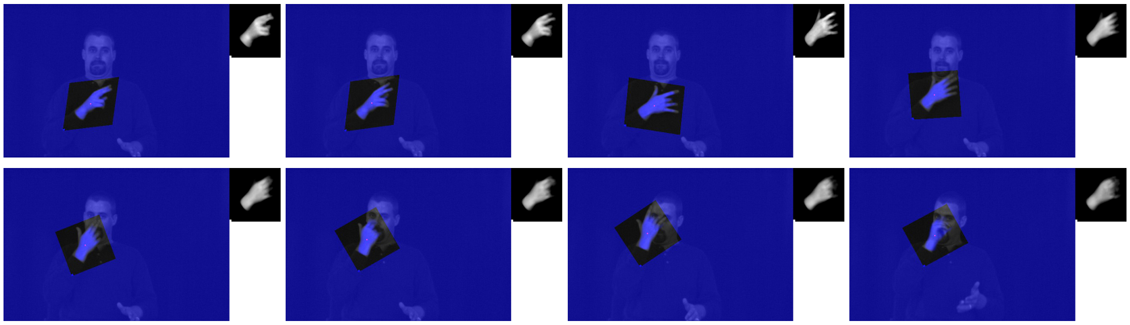


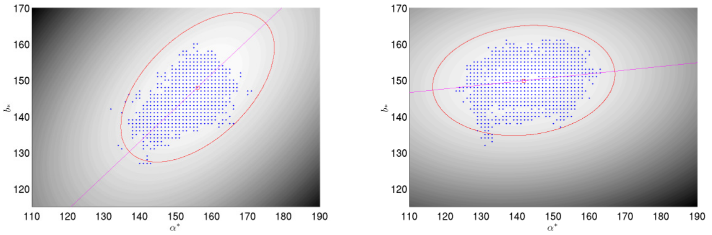
Figure 7: Regularized Shape-Appearance Model fitting in a sign language video. In every input frame, we superimpose the model-based reconstruction of the hand in the frame domain, $A_0(W_p^{-1}(x))+\sum\lambda_i A_i(W_p^{-1}(x))$. In the upper-right corner, we display the reconstruction in the model domain, $A_0(x)+\sum\lambda_i A_i(x)$, which determines the optimum weights λ. A demo video is available online (see text).

training sequence. Similarly to linear predictive analysis (Rabiner and Schafer, 2007) and other tracking methods that use dynamics (e.g., Blake and Isard, 1998) we learn the weight matrices A_v, B_v by minimizing the mean square estimation error over all the prediction-testing frames. Since we have assumed that A_v and B_v are diagonal, this optimization is done independently for each component of λ and $\widetilde{p}$, which is treated as 1D signal. The predictive weights for each component are thus derived from the solution of an ordinary least squares problem. The optimum values of the mean squared errors yield the diagonal elements of the prediction errors' covariance matrices $\Sigma_{\varepsilon_\lambda}$ and $\Sigma_{\varepsilon_{\widetilde{p}}}$, which are diagonal.

4.4.5 Implementation and Results of SAM Fitting

The energy $E(\lambda,p)$ (2) of the proposed regularized SAM fitting is a special case of the general objective function that is minimized by the *simultaneous inverse compositional with a prior* (SICP) algorithm of Baker et al. (2004). Therefore, in order to minimize $E(\lambda,p)$, we specialize this algorithm for the specific types of our prior terms. Details are given in the Appendix A. At each frame n of a video segment, the fitting algorithm is initialized as follows. If the current frame is not the starting frame of the sequential fitting (that is $K(n)\neq 0$), then the parameters λ, p are initialized from the predictions λ^e, p^e. Otherwise, if $K(n)=0$, we test as initializations the two similarity transforms that, when applied to the SAM mean image A_0, make its mask have the same centroid, area and orientation as the mask of the current frame's SA image. We twice apply the SICP algorithm using these two initializations, and finally choose the initialization that yields the smallest regularized energy $E(\lambda,p)$.

Figure 7 demonstrates indicative results of the regularized fitting of the dominant hand's SAM in a sign language video. For more details, please refer to the following URL that contains a video of these results: `http://cvsp.cs.ntua.gr/research/sign/aff_SAM`. We observe that in non-occlusion cases, this regularized method is effective and accurately tracks the handshape. Further, in occlusion cases, even after a lot of occluded frames, the result is especially robust. Nevertheless, the accuracy of the extracted handshape is smaller in cases of occlusions, compared to the non-



Signer A

Signer B

Figure 8: Skin color modeling for the two signers of the GSL lemmas corpus, where we test the signer adaptation. Training samples in the a^*-b^* chromaticity space and fitted pdf's $p_s(a^*,b^*)$. In each case, the straight line defines the normalized mapping $g(I)$ used in the Shape-Appearance images formation.

occlusion cases, since the prior terms keep the result closer to the SAM mean image A_0. In addition, extensive handshape classification experiments were performed in order to evaluate the extracted handshape features employing the proposed Aff-SAM method (see Section 7).

5. Signer Adaptation

We develop a method for adapting a trained Aff-SAM model to a new signer. This adaptation is facilitated by the characteristics of the Aff-SAM framework. Let us consider an Aff-SAM model trained to a signer, using the procedure described in Section 4.3. We aim to reliably adapt and fit the existing Aff-SAM model on videos from a *new signer*.

5.1 Skin Color and Normalization

The employed skin color modeling adapts on the characteristics of the skin color of a new signer. Figure 8 illustrates the skin color modeling for the two signers of the GSL lemmas corpus, where we test the adaptation. For each new signer, the color model is built from skin samples of a face tracker (Section 3.1, Section 4.1). Even though there is an intersection, the domain of colors classified as skin is different between the two. In addition, the mapping $g(I)$ of skin color values, used to create the SA images, is normalized according to the skin color distribution of each signer. The differences in the lines of projection reveal that the normalized mapping $g(I)$ is different in these two cases. This skin color adaptation makes the body-parts label extraction of the visual front-end preprocessing to behave robustly over different signers. In addition, the extracted SA images have the same range of values and are directly comparable across signers.

5.2 Hand Shape and Affine Transforms

Affine transforms can reliably compensate for the anatomical differences of the hands of different signers. Figure 9 demonstrates some examples. In each case, the right hands of the signers are in a similar configuration and viewpoint. We observe that there exist pairs of affine transformations that successfully align the handshapes of both signers to the common model domain. For instance,

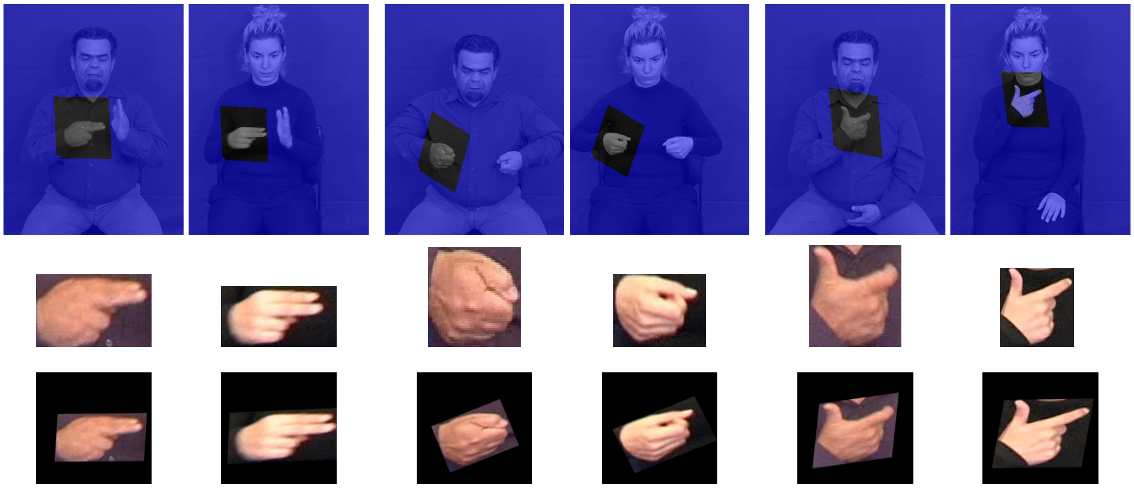


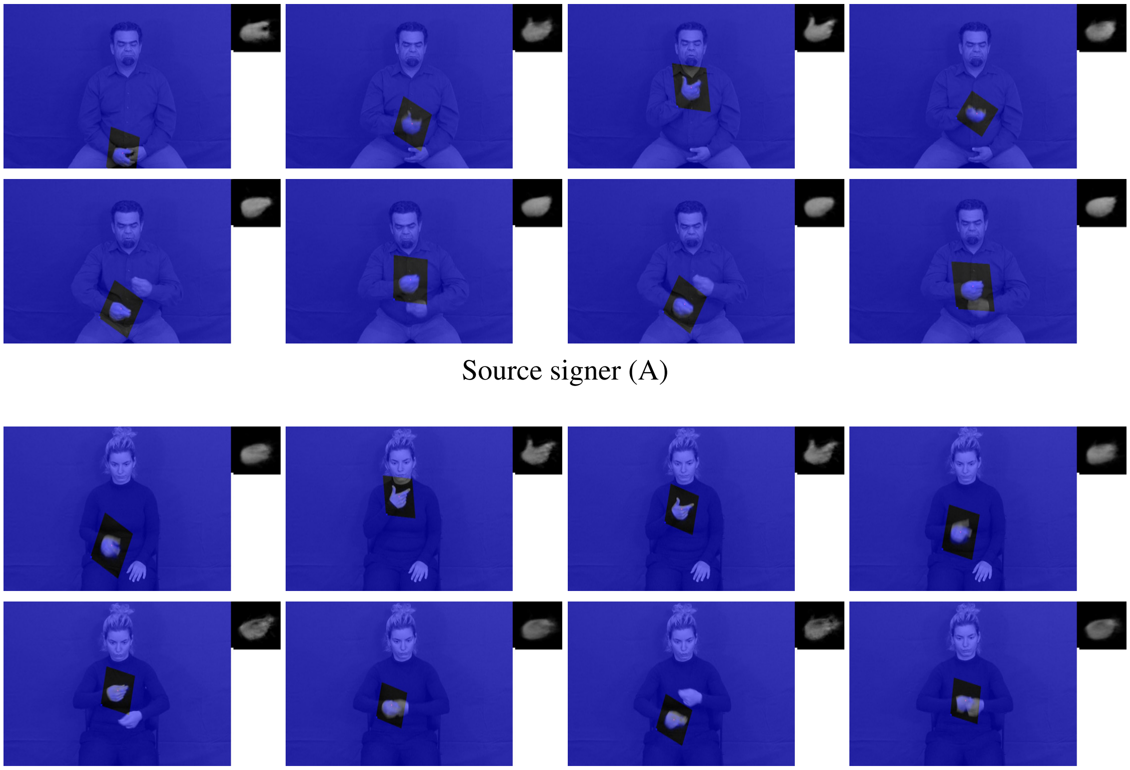
Figure 9: Alignment of the hands of two different signers, using affine transformations. *First row*: Input frames with superimposed rectangles that visualize the affine transformations. *Second row*: Cropped images around the hand. *Third row*: Alignment of the cropped images in a common model domain, using the affine transformations.

the affine transforms have the ability to stretch or shrink the hand images over the major hand axis. They thus automatically compensate for the fact that the second signer has thinner hands and longer fingers. In general, the class of affine transforms can effectively approximate the transformation needed to align the 2D hand shapes of different signers.

5.3 New Signer Fitting

To process a new signer the visual front-end is applied as in Section 3. Then, we only need to retrain the static and dynamic priors on the new signer. For this, we randomly select frames where the hand is not occluded. Then, for the purposes of this training, the existing SAM is fitted on them by minimizing the energy in Equation (2) with w_S=w_D=0, namely the reconstruction error term without prior terms. Since the SAM is trained on another signer, this fitting is not always successful, at this step. At that point, the user annotates the frames where this fitting has succeeded. This feedback is binary and is only needed during training and for a relatively small number of frames. For example, in the case of the GSL lemmas corpus, we sampled frames from approximately 1.2% of all corpus videos of this signer. In 15% of the sampled frames, this fitting with no priors was annotated as successful. Using the samples from these frames, we learn the static and dynamic priors of λ and p, as described in Section 4.4.2 for the new signer. The regularized SAM fitting is implemented as in Section 4.4.5.

Figure 10 demonstrates results of the SAM fitting, in the case of signer adaptation. The SAM eigenimages are learned using solely Signer A. The SAM is then fitted on the signer B, as above. For comparison, we also visualize the result of the SAM fitting to the signer A, for the same sign. Demo videos for these fittings also are included in the following URL: `http://cvsp.cs.ntua.gr/research/sign/aff_SAM`. We observe that, despite the anatomical differences of the two signers,



Source signer (A)

New signer (B)

Figure 10: Regularized Shape-Appearance Model fitting on 2 signers. The SA model was trained on Signer A and adapted for Signer B. Demo videos are available online (see text).

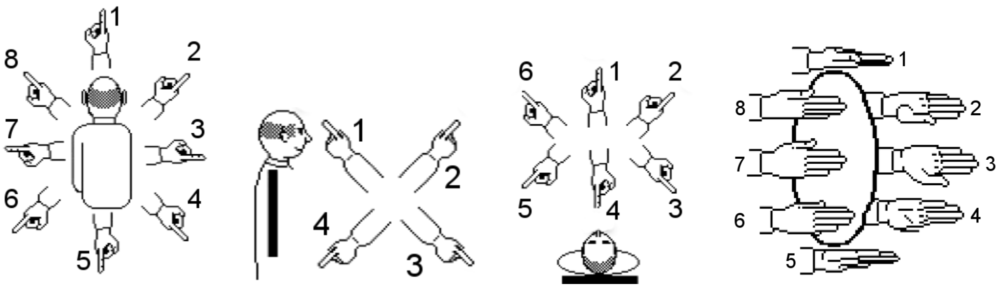
the performance of the SAM fitting is satisfactory after the adaptation. In both signers, the fitting yields accurate shape estimation in non-occlusion cases.

6. Data Set and Handshape Annotation for Handshape Classification

The *SL Corpus BU400* (Neidle and Vogler, 2012) is a continuous American sign language database. The background is uniform and the images have a resolution of 648x484 pixels, recorded at 60 frames per second. In the classification experiments we employ the front camera video, data from a single signer, and the story ‘Accident’. We next describe the annotation parameters required to produce the ground-truth labels. These concern the pose and handshape configurations and are essential for the supervised classification experiments.

6.1 Handshape Parameters and Annotation

The parameters that need to be specified for the annotation of the data are the (pose-independent) handshape configuration and the 3D hand pose, that is the orientation of the hand in the 3D space. For the annotation of the handshape configurations we followed the *SignStream annotation conventions* (Neidle, 2007). For the 3D hand pose we parametrized the 3D hand orientations inspired



(*a*) Front (F) (*b*) Side (S) (*c*) Bird's (B) (*d*) Palm (P)

Figure 11: 3D Hand Orientation parameters: (*a-c*) Extended Finger Direction Parameters: (*a*) Signer's front view (F), (*b*) Side view (S), (*c*) Birds' view (B); (*d*) Palm orientation (P). Note that we have modified the corresponding figures of Hanke (2004) with numerical parameters.

by the HamNoSys description (Hanke, 2004). The adopted annotation parameters are as follows: 1) *Handshape identity (HSId)* which defines the handshape configuration, that is, ('*A*', '*B*', '*1*', '*C*' etc.), see Table 1 for examples. 2) *3D Hand Orientation* (hand pose) consisting of the following parameters (see Figure 11): i) *Extended Finger Direction* parameters that define the orientation of the hand axis. These correspond to the hand orientation relatively to the three planes that are defined relatively to: the Signer's Front view (referred to as F), the Bird's view (B) and the Side view (S). ii) *Palm Orientation* parameter (referred to as P) for a given extended finger direction. This parameter is defined w.r.t. the bird's view, as shown in Figure 11(d).

6.2 Data Selection and Classes

We select and annotate a set of occluded and non-occluded handshapes so that 1) they cover substantial handshape and pose variation as they are observed in the data and 2) they are quite frequent. More specifically we have employed three different data sets (DS): 1) *DS-1*: 1430 non-occluded handshape instances with 18 different HSIds. 2) *DS-1-extend*: 3000 non-occluded handshape instances with 24 different HSIds. 3) *DS-2*: 4962 occluded and non-occluded handshape instances with 42 different HSIds. Table 1 presents an indicative list of annotated handshape configurations and 3D hand orientation parameters.

7. Handshape Classification Experiments

In this section we present the experimental framework consisting of the statistical system for handshape classification. This is based 1) on the handshape features extracted as described in Section 4; 2) on the annotations as described in Section 6.1 as well as 3) on the data selection and classes (Section 6.2). Next, we describe the experimental protocol containing the main experimental variations of the data sets, of the class dependency, and of the feature extraction method.

HSId		*1*	*1*	*4*	*4*	*5C*	*5*	*5*	*5*	*A*	*A*	*BL*	*BL*	*BL*	*BL*
3D hand pose	F	8	1	7	6	1	7	8	1	8	8	8	7	8	8
	S	0	0	0	3	1	0	2	2	0	2	0	0	0	0
	B	0	0	0	6	4	0	1	1	0	6	0	0	0	0
	P	1	8	3	1	3	3	1	5	3	2	2	3	3	4
# insts.		14	24	10	12	27	38	14	19	14	31	10	15	23	30
exmpls.															

HSId		*BL*	*CUL*	*F*	*F*	*U*	*UL*	*V*	*Y*	*b1*	*c5*	*c5*	*cS*	*cS*	*fO2*
3D hand pose	F	8	7	7	1	7	7	8	8	7	8	8	7	8	8
	S	2	0	0	2	0	0	0	0	0	0	0	0	2	0
	B	6	0	0	1	0	0	0	0	0	0	6	6	6	0
	P	4	3	3	3	2	3	2	2	3	3	1	3	3	1
# insts.		20	13	23	13	10	60	16	16	10	17	18	10	34	12
exmpls.															

Table 1: Samples of annotated handshape identities (HSId) and corresponding 3D hand orientation (pose) parameters for the D-HFSBP class dependency and the corresponding experiment; in this case each model is fully dependent on all of the orientation parameters. '# insts.' corresponds to the number of instances in the dataset. In each case, we show an example handshape image that is randomly selected among the corresponding handshape instances of the same class.

7.1 Experimental Protocol and Other Approaches

The experiments are conducted by employing cross-validation by selecting five different random partitions of the dataset into train-test sets. We employ 60% of the data for training and 40% for testing. This partitioning samples data, among all realizations per handshape class in order to equalize class occurrence. The number of realizations per handshape class are on average 50, with a minimum and maximum number of realizations in the range of 10 to 300 depending on the experiment and the handshape class definition. We assign to each experiment's training set one GMM per handshape class; each has one mixture and diagonal covariance matrix. The GMMs are uniformly initialized and are afterwards trained employing Baum-Welch re-estimation (Young et al., 1999). Note that we are not employing other classifiers since we are interested in the evaluation of the handshape features and not the classifier. Moreover this framework fits with common hidden Markov model (HMM)-based SL recognition frameworks (Vogler and Metaxas, 1999), as in Section 8.

7.1.1 Experimental Parameters

The experiments are characterized by the dataset employed, the class dependency and the feature extraction method as follows:

Class	Annotation Parameters				
Dependency label	HSId(H)	Front(F)	Side(S)	Bird's(B)	Palm(P)
D-HFSBP	D	D	D	D	D
D-HSBP	D	*	D	D	D
D-HBP	D	*	*	D	D
D-HP	D	*	*	*	D
D-H	D	*	*	*	*

Table 2: Class dependency on orientation parameters. One row for each model dependency w.r.t. the annotation parameters. The dependency or non-dependency state to a particular parameter for the handshape trained models is noted as 'D' or '*' respectively. For instance the D-HBP model is dependent on the HSId and Bird's view and Palm orientation parameters.

Data Set (DS): We have experimented employing three different data sets DS-1, DS-1-extend and DS-2 (Section 6.2 for details).

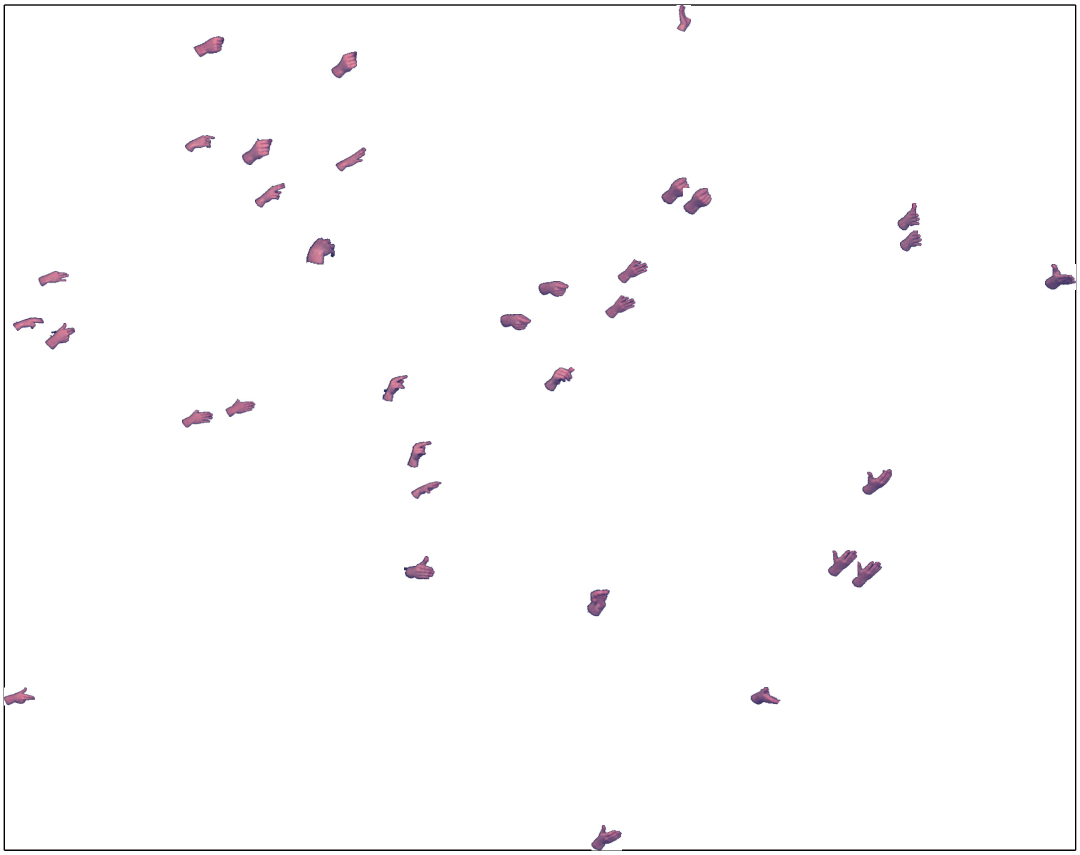
Class dependency (CD): The class dependency defines the orientation parameters in which our trained models are dependent to (Table 2). Take for instance the orientation parameter 'Front' (F). There are two choices, either 1) construct handshape models independent to this parameter or 2) construct different handshape models for each value of the parameter. In other words, at one extent CD restricts the models generalization by making each handshape model specific to the annotation parameters, thus highly discriminable, see for instance in Table 2 the experiment corresponding to D-HFSBP. At the other extent CD extends the handshape models generalization w.r.t. to the annotation parameters, by letting the handshape models account for pose variability (that is depend only on the HSId; same HSId's with different pose parameters are tied), see for instance experiment corresponding to the case D-H (Table 2). The CD field takes the values shown in Table 2.

7.1.2 FEATURE EXTRACTION METHOD

Apart from the proposed Aff-SAM method, the methods employed for handshape feature extraction are the following:

Direct Similarity Shape-Appearance Modeling (DS-SAM): Main differences of this method with Aff-SAM are as follows: *1)* we replace the affine transformations that are incorporated in the SA model (1) by simpler *similarity* transforms and *2)* we replace the regularized model fitting by direct estimation (*without* optimization) of the similarity transform parameters using the centroid, area and major axis orientation of the hand region followed by projection into the PCA subspace to find the eigenimage weights. Note that in the occlusion cases, this simplified fitting is done directly on the SA image of the region that contains the modeled hand as well as the other occluded body-part(s) (that is the other hand and/or the head), without using any static or dynamic priors as those of Section 4.4. This approach is similar to Birk et al. (1997) and is adapted to fit our framework.

Direct Translation Scale Shape-Appearance Modeling (DTS-SAM): The main differences of this method with Aff-SAM are the following: *1)* we replace the affine transformations that are incorporated in the Shape-Appearance model (1) by simpler *translation-scale* transforms and *2)* we replace the regularized model fitting by direct estimation of the translation and scale parameters



(a)

Figure 12: Feature space for the Aff-SAM features and the D-HFSBP experiment case (see text). The trained models are visualized via projections on the $\lambda_1 - \lambda_2$ plane that is formed from the weights of the two principal Aff-SAM eigenimages. Cropped handshape images are placed at the models' centroids.

using the square that tightly surrounds the hand mask, followed again by projection into the PCA subspace to find the eigenimage weights. In this simplified version too, the hand occlusion cases are treated by simply fitting the model to the Shape-Appearance image that contains the occlusion, without static or dynamic priors. This approach is similar to Cui and Weng (2000), Wu and Huang (2000) and Du and Piater (2010) and is adapted so as to fit our proposed framework.

Other tested methods from the literature contain the *Fourier Descriptors* (FD): These are derived from the Fourier coefficients of the contour that surrounds the hand, after appropriate normalizations for scale and rotation invariance (Chen et al., 2003; Conseil et al., 2007). For dimensionality reduction, we keep the descriptors that correspond to the first N_{FD} frequencies. We tested different values for the parameter N_{FD} and finally kept $N_{FD} = 50$ that yield the best performance. *Moments* (M): These consist of the seven Hu moment invariants of the hand region (Hu, 1962). These depend only on the central moment of the binary shape of the hand region and are invariant to similarity transforms of the hand region. *Region Based* (RB): These consist of the area, eccentricity, com-

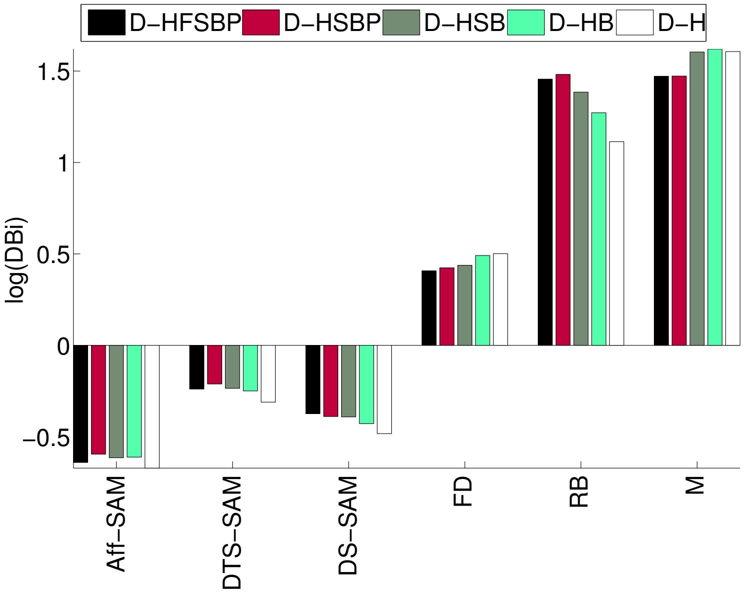


Figure 13: Davies-Bouldin index (DBi) in logarithmic scale (y-axis) for multiple feature spaces and varying models class dependency to the orientation parameters. Lower values of DBi indicate better compactness and separability of classes.

pactness and minor and major axis lengths of the hand region (Agris et al., 2008). Compared to the proposed Aff-SAM features we consider the rest five sets of features belonging to either *baseline features* or *more advanced features*. First, the baseline features contain the FD, M and RB approaches. Second, the more advanced features contain the DS-SAM and DTS-SAM methods which we have implemented as simplified versions of the proposed Aff-SAM. As it will be revealed by the evaluations, the more advanced features are more competitive than the baseline features and the comparisons with them are more challenging.

7.2 Feature Space Evaluation Results

Herein we evaluate the feature space of the Aff-SAM method. In order to approximately visualize it, we employ the weights λ_1, λ_2 of the two principal eigenimages of Aff-SAM. Figure 12(a) provides a visualization of the trained models per class, for the experiment corresponding to D-HFSBP class dependency (that is each class is fully dependent on orientation parameters). It presents a single indicative cropped handshape image per class to add intuition on the presentation: these images correspond to the points in the feature space that are closest to the specific classes' centroids. We observe that similar handshape models share close positions in the space. The presented feature space is indicative and it seems clear when compared to feature spaces of other methods. To support this we compare the feature spaces with the Davies-Boulding index (DBi), which quantifies their quality. In brief, the DBi is the average over all n clusters, of the ratio of intra-cluster distances σ_i versus the inter-cluster distance $d_{i,j}$ of i, j clusters, as a measure of their separation: $DBi = \frac{1}{n}\sum_{i=1}^{n} \max_{i \neq j}(\frac{\sigma_i + \sigma_j}{d_{i,j}})$ (Davies and Bouldin, 1979). Figure 13 presents the results. The reported

Data Set	# HSIds	CD	Occ.	Feat. Method	Avg.Acc.%	Std.
DS-1	18	Table. 2	✗	Aff-SAM	**93.7**	1.5
				DS-SAM	93.4	1.6
				DTS-SAM	89.2	1.9
DS-1-extend	24	‘D-H’	✗	Aff-SAM	**77.2**	1.6
				DS-SAM	74	2.3
				DTS-SAM	67	1.4
DS-2	42	Table. 2	✓	Aff-SAM	**74.9**	0.9
				DS-SAM	66.1	1.1
				DTS-SAM	62.7	1.4

Table 3: Experiments overview with selected average overall results over different main feature extraction methods and experimental cases of DS and CD experiments, with occlusion or not (see Section 7.1). CD: class dependency. Occ.: indicates whether the dataset includes occlusion cases. # HSIds: the number of HSId employed, Avg.Acc.: average classification accuracy, Std.: standard deviation of the classification accuracy.

indices are for varying CD field, that is the orientation parameters on which the handshape models are dependent or not (as discussed in Section 7.1) and are referred in Table 2. We observe that the DBi’s for the Aff-SAM features are lower that is the classes are more compact and more separable, compared to the other cases. The closest DBi’s are these of DS-SAM. In addition, the proposed features show stable performance over experiments w.r.t. class-dependency, indicating robustness to some amount of pose variation.

7.3 Results of Classification Experiments

We next show average classification accuracy results after 5-fold cross-validation for each experiment. together with the standard deviation of the accuracies. The experiments consist of 1) Class dependency and Feature variation for non-occlusion cases and 2) Class dependency and Feature variation for both occlusion and non-occlusion cases. Table 3 presents averages as well as comparisons with other features for the three main experimental data sets discussed. The averages are over all cross-validation cases, and over the multiple experiments w.r.t. class dependency, where applicable. For instance, in the first block for the case ‘DS-1’, that is non-occluded data from the dataset DS-1, the average is taken over all cases of class dependency experiments as described in Table 2. For the ‘DS-1-extend’ case, the average is taken over the D-H class dependency experiment, since we want to increase the variability within each class.

7.3.1 FEATURE COMPARISONS FOR NON-OCCLUDED CASES

Next, follow comparisons by employing the referred feature extraction approaches, for two cases of data sets, while accounting for non-occluded cases.

7.3.2 DATA SET DS-1

In Figure 14 we compare the employed methods, while varying the models’ dependency w.r.t. the annotation parameters (x axis). We employ the DS-1 data set, consisting of 18 handshape types

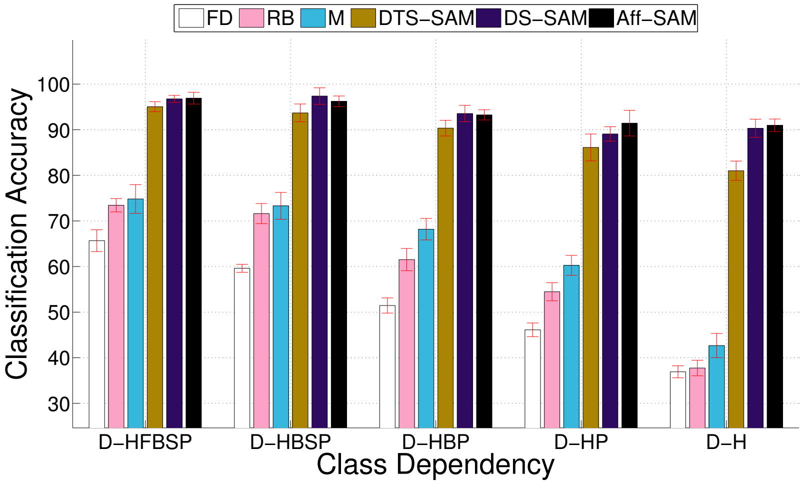


Figure 14: Classification experiments for non-occlusion cases, dataset DS-1. Classification Accuracy for varying experiments (x-axis) that is the dependency of each class w.r.t. the annotation parameters [H,F,B,S,P] and the feature employed (legend). For the numbers of classes per experiment see Table 4.

Class dependency Parameters	D-HFSBP	D-HSBP	D-HBP	D-HP	D-H
# Classes	34	33	33	31	18

Table 4: Number of classes for each type of class dependency (classification experiments for Non-Occlusion cases).

from non-occlusion cases. The number of classes are shown in Table 4. In Figure 14 we depict the performance over the different methods and models' dependency. At the one extent (that is 'D-HFBSP') we trained one GMM model for each different combination of the handshape configuration parameters (H,F,B,S,P). Thus, the trained models were dependent on the 3D handshape pose and so are the classes for the classification (34 different classes). In the other extent ('D-H') we trained one GMM model for each HSId thus the trained models were independent to the 3D handshape pose and so are the classes for the classification (18 different classes). Furthermore we observe that the proposed method outperforms the baseline methods (FD, RB, M) and DTS-SAM. However the classification performance of Aff-SAM and DS-SAM methods is quite close in some cases. This is due to the easy classification task (small number of HSIds and 3D pose variability and non-occlusion cases). The classification performance of the proposed method is slightly affected from the decrease of the dependency on the annotation parameters. This strengthens our previous observation that the proposed method can handle small pose variations. For a results' overview see Table 3 (DS-1 block). The averages are across all pose-dependency cases.

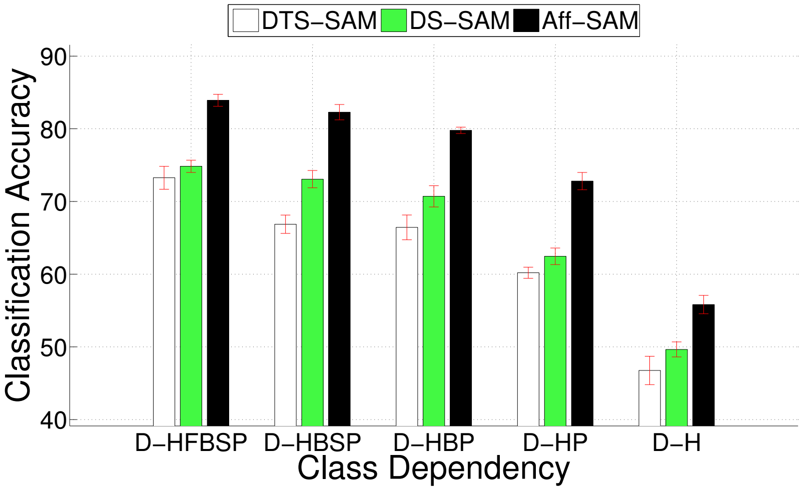


Figure 15: Classification experiments for both occluded and non-occluded cases. Classification Accuracy by varying the dependency of each class w.r.t. to the annotation parameters [H,F,B,S,P] (x-axis) and the feature employed (legend). For the numbers of classes per experiment see Table 5.

7.3.3 DATA SET DS-1-EXTEND

This is an extension of DS-1 and consists of 24 different HSIds with much more 3D handshape pose variability. We trained models independent to the 3D handshape pose. Thus, these experiments refer to the D-H case. Table 3 (DS-1-extend block) shows average results for the three competitive methods. We observe that Aff-SAM outperforms both DS-SAM and DTS-SAM achieving average improvements of 3.2% and 10.2% respectively. This indicates the advancement of the Aff-SAM over the other two competitive methods (DS-SAM and DTS-SAM) in more difficult tasks. It also shows that, by incorporating more data with extended variability w.r.t. pose parameters, there is an increase in the average improvements.

Class dependency Parameters	D-HFSBP	D-HSBP	D-HBP	D-HP	D-H
# Classes	100	88	83	72	42

Table 5: Number of classes for each type of class dependency (classification experiments for Occlusion and Non-Occlusion cases).

7.3.4 FEATURE COMPARISONS FOR OCCLUDED AND NON-OCCLUDED CASES

In Figure 15 we vary the models' dependency w.r.t. the annotation parameters similar to Section 7.3.1. However, DS-2 data set consists of 42 handshape HSIds for *both* occlusion and non-occlusion cases. For the number of classes per experiment see Table 5. Aff-SAM outperforms both DS-SAM and DST-SAM obtaining on average 10% performance increase in all cases (Figure 15).

This indicates that Aff-SAM handles handshape classification obtaining decent results even during occlusions. The performance for the other baseline methods is not shown since they cannot handle occlusions and the results are lower. The comparisons with the two more competitive methods show the differential gain due to the *claimed* contributions of the Aff-SAM. By making our models independent to 3D pose orientation, that is,-H, the classification performance decreases. This makes sense since by taking into consideration the occlusion cases the variability of the handshapes' 3D pose increases; as a consequence the classification task is more difficult. Moreover, the classification during occlusions may already include errors at the visual modeling level concerning the estimated occluded handshape. In this experiment, the range of 3D pose variations is larger than the amount handled by the affine transforms of the Aff-SAM.

8. Sign Recognition

Next, we evaluate the Aff-SAM approach, on automatic sign recognition experiments, while fusing with movement/position cues, as well as concerning its application on multiple signers. The experiments are applied on data from the GSL lexicon corpus (DictaSign, 2012). By employing the presented framework for tracking and feature extraction (Section 3) we extract the Aff-SAM features (Section 4). These are then employed to construct data-driven subunits as in Roussos et al. (2010b) and Theodorakis et al. (2012), which are further statistically trained. The lexicon corpus contains data from two different signers, A and B. Given the Aff-SAM based models from signer A these are then adapted and fitted to another signer (B) as in Section 5 for which no Aff-SAM models have been trained. The features resulting as a product of the visual level adaptation, are employed next in the recognition experiment. For signer A, the features are extracted from the signer's own model. Note that, there are other aspects concerning signer adaptation during SL recognition, as for instance the manner of signing or the different pronunciations, which are not within the focus of this article.

GSL Lemmas: We employ 100 signs from the *GSL lemmas corpus*. These are articulated in isolation with five repetitions each, from two native signers (male and female). The videos have a uniform background and a resolution of 1440x1080 pixels, recorded at 25 fps.

8.1 Sub-unit Modeling and Sign Recognition

The SL recognition framework consists of the following: 1) First by employing the movement-position cue we construct dynamic/static SUs based on dynamic and static discrimination (Pitsikalis et al., 2010; Theodorakis et al., 2012). 2) Second we employ the handshape features and the sub-unit construction via clustering of the handshape features (Roussos et al., 2010b). 3) We then create one lexicon for each information cue, that is, movement-position and handshape. For the movement-position lexicon we recompose the constructed dynamic/static SUs, whereas for the Handshape lexicon we recompose the handshape subunits (HSU) to form each sign realization. 4) Next, for the training of the SUs we employ a GMM for the static and handshape subunits and an 5-state HMM for the dynamic subunits. Concerning the training, we employ four realizations for each sign for training and one for testing. 5) Finally, we fuse the movement-position and handshape cues via one possible late integration scheme, that is Parallel HMMs (PaHMMs) (Vogler and Metaxas, 1999).

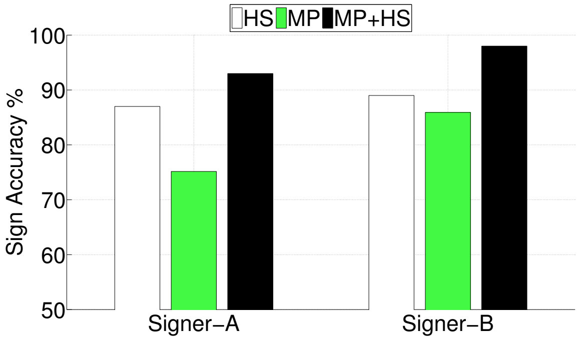


Figure 16: Sign recognition in GSL lemmas corpus employing 100 signs for each signer A and B, and multiple cues: Hanshape (HS), Movement-Position (MP) cue and MP+HS fusion between both via Parallel HMMs.

8.2 Sign Recognition Results

In Figure 16 we present the sign recognition performance on the GSL lemmas corpus employing 100 signs from two signers, A and B, while varying the cues employed: movement-position (MP), handshape (HS) recognition performance and the fusion of both MP+HS cues via PaHMMs. For both signers A and B, handshape-based recognition outperforms the one of movement-position cue. This is expected, and indicates that handshape cue is crucial for sign recognition. Nevertheless, the main result we focus is the following: The sign recognition performance in Signer-B is similar to Signer-A, where the Aff-SAM model has been trained. Thus by applying the affine adaptation procedure and employing only a small development set, as presented in Section 5 we can extract reliable handshape features for multiple signers. As a result, when both cues are employed, and for both signers, the recognition performance increases, leading to a 15% and 7.5% absolute improvement w.r.t. the single cues respectively.

9. Conclusions

In this paper, we propose a new framework that incorporates dynamic affine-invariant Shape - Appearance modeling and feature extraction for handshape classification. The proposed framework leads to the extraction of effective features for hand configurations. The main contributions of this work are the following: 1) We employ Shape-Appearance hand images for the representation of the hand configurations. These images are modeled with a linear combination of eigenimages followed by an affine transformation, which effectively accounts for some 3D hand pose variations. 2) In order to achieve robustness w.r.t. occlusions, we employ a regularized fitting of the SAM that exploits prior information on the handshape and its dynamics. This process outputs an accurate tracking of the hand as well as descriptive handshape features. 3) We introduce an affine-adaptation for different signers than the signer that was used to train the model. 4) All the above features are integrated in a statistical handshape classification GMM and a sign recognition HMM-based system.

The overall visual feature extraction and classification framework is evaluated on classification experiments as well as on sign recognition experiments. These explore multiple tasks of gradual difficulty in relation to the orientation parameters, as well as both occlusion and non-occlusion cases. We compare with existing baseline features as well as with more competitive features, which are implemented as simplifications of the proposed SAM method. We investigate the quality of the feature spaces and evaluate the compactness-separation of the different features in which the proposed features show superiority. The Aff-SAM features yield improvements in classification accuracy too. For the non-occlusion cases, these are on average 35% over the baseline methods (FD, RB, M) and 3% over the most competitive SAM methods (DS-SAM, DST-SAM). Furthermore, when we also consider the occlusion cases, the improvements in classification accuracy are on average 9.7% over the most competitive SAM methods (DS-SAM, DST-SAM). Although DS-SAM yields similar performance in some cases, it under-performs in the more difficult and extended data set classification tasks. On the task of sign recognition for a 100-sign lexicon of GSL lemmas, the approach is evaluated via handshape subunits and also fused with movement-position cues, leading to promising results. Moreover, it is shown to have similar results, even if we do not train an explicit signer dependent Aff-SA model, given the introduction of the affine-signer adaptation component. In this way, the approach can be easily applicable to multiple signers.

To conclude with, given that handshape is among the main sign language phonetic parameters, we address issues that are indispensable for automatic sign language recognition. Even though the framework is applied on SL data, its application is extendable on other gesture-like data. The quantitative evaluation and the intuitive results presented show the perspective of the proposed framework for further research.

Acknowledgments

This research work was supported by the EU under the research program Dictasign with grant FP7-ICT-3-231135. A. Roussos was also supported by the ERC Starting Grant 204871-HUMANIS.

Appendix A. Details about the Regularized Fitting Algorithm

We provide here details about the algorithm of the regularized fitting of the shape-appearance model. The total energy $E(\lambda, p)$ that is to be minimized can be written as (after a multiplication with N_M that does not affect the optimum parameters):

$$\begin{aligned} J(\lambda, p) = & \sum_x \left\{ A_0(x) + \sum_{i=1}^{N_c} \lambda_i A_i(x) - f(W_p(x)) \right\}^2 + \\ & \frac{N_M}{N_c} \left(w_S \left\| \lambda - \lambda_0 \right\|^2_{\Sigma_\lambda} + w_D \left\| \lambda - \lambda^e \right\|^2_{\Sigma_{\varepsilon_\lambda}} \right) + \\ & \frac{N_M}{N_p} \left(w_S \left\| p - p_0 \right\|^2_{\Sigma_p} + w_D \left\| p - p^e \right\|^2_{\Sigma_{\varepsilon_p}} \right) . \end{aligned} \tag{4}$$

If σ_{λ_i}, $\sigma_{\tilde{p}_i}$ are the standard deviations of the components of the parameters λ, $\tilde{p}$ respectively and $\sigma_{\varepsilon_{\lambda,i}}$, $\sigma_{\varepsilon_{\tilde{p},i}}$ are the standard deviations of the components of the parameters' prediction errors ε_λ, $\varepsilon_{\tilde{p}}$, then the corresponding covariance matrices Σ_λ, $\Sigma_{\tilde{p}}$, $\Sigma_{\varepsilon_\lambda}$, $\Sigma_{\varepsilon_{\tilde{p}}}$, which are diagonal, can be written as:

$$\begin{aligned} & \Sigma_\lambda = \mathrm{diag}(\sigma^2_{\lambda_1}, \ldots, \sigma^2_{\lambda_{N_c}}), \Sigma_{\tilde{p}} = \mathrm{diag}(\sigma^2_{\tilde{p}_1}, \ldots, \sigma^2_{\tilde{p}_{N_c}}), \\ & \Sigma_{\varepsilon_\lambda} = \mathrm{diag}(\sigma^2_{\varepsilon_{\lambda,1}}, \ldots, \sigma^2_{\varepsilon_{\lambda,N_c}}), \Sigma_{\varepsilon_{\tilde{p}}} = \mathrm{diag}(\sigma^2_{\varepsilon_{\tilde{p},1}}, \ldots, \sigma^2_{\varepsilon_{\tilde{p},N_p}}). \end{aligned}$$

The squared norms of the prior terms in Equation (4) are thus given by:

$$\|\lambda-\lambda_0\|^2_{\Sigma_\lambda} = \sum_{i=1}^{N_c}\left(\frac{\lambda_i}{\sigma_{\lambda_i}}\right)^2 ,$$

$$\|\lambda-\lambda^e\|^2_{\Sigma_{\varepsilon_\lambda}} = \sum_{i=1}^{N_c}\left(\frac{\lambda_i-\lambda_i^e}{\sigma_{\varepsilon_{\lambda,i}}}\right)^2 ,$$

$$\|p-p_0\|^2_{\Sigma_p} = (p-p_0)^T U_p \Sigma_{\widetilde{p}}^{-1} U_p^T (p-p_0) = \|\widetilde{p}\|^2_{\Sigma_{\widetilde{p}}} = \sum_{i=1}^{N_p}\left(\frac{\tilde{p}_i}{\sigma_{\tilde{p}_i}}\right)^2 ,$$

$$\|p-p^e\|^2_{\Sigma_{\varepsilon_p}} = \|\widetilde{p}-\widetilde{p}^e\|^2_{\Sigma_{\varepsilon_{\widetilde{p}}}} = \sum_{i=1}^{N_p}\left(\frac{\tilde{p}_i-\tilde{p}_i^e}{\sigma_{\varepsilon_{\tilde{p},i}}}\right)^2 .$$

Therefore, if we set:

$$m_1 = \sqrt{w_S N_M/N_c}\,, m_2 = \sqrt{w_D N_M/N_c}\,,$$
$$m_3 = \sqrt{w_S N_M/N_p}\,, m_4 = \sqrt{w_D N_M/N_p}\,,$$

the energy in Equation (4) takes the form:

$$J(\lambda,p) = \sum_x \left\{A_0(x) + \sum_{i=1}^{N_c}\lambda_i A_i(x) - f(W_p(x))\right\}^2 + \sum_{i=1}^{N_G} G_i^2(\lambda,p)\,, \tag{5}$$

with $G_i(\lambda,p)$ being $N_G = 2N_c + 2N_p$ prior functions defined by:

$$G_i(\lambda,p) = \begin{cases} m_1\frac{\lambda_i}{\sigma_{\lambda_i}}, & 1\le i\le N_c \\ m_2\frac{\lambda_j-\lambda_j^e}{\sigma_{\varepsilon_{\lambda,j}}}, j=i-N_c, & N_c+1\le i\le 2N_c \\ m_3\frac{\tilde{p}_j}{\sigma_{\tilde{p}_j}}, j=i-2N_c, & 2N_c+1\le i\le 2N_c+N_p \\ m_4\frac{\tilde{p}_j-\tilde{p}_j^e}{\sigma_{\varepsilon_{\tilde{p},j}}}, j=i-2N_c-N_p, & 2N_c{+}N_p{+}1{\le}i{\le}2N_c{+}2N_p \end{cases}. \tag{6}$$

Each component $\tilde{p}_j$, $j=1,\ldots,N_p$, of the re-parametrization of p can be written as:

$$\tilde{p}_j = v_{\tilde{p}_j}^T (p-p_0)\,, \tag{7}$$

where $v_{\tilde{p}_j}$ is the j-th column of U_p, that is the eigenvector of the covariance matrix Σ_p that corresponds to the j-th principal component $\tilde{p}_j$.

In fact, the energy $J(\lambda,p)$, Equation (5), for general prior functions $G_i(\lambda,p)$, has exactly the same form as the energy that is minimized by the algorithm of Baker et al. (2004). Next, we describe this algorithm and then we specialize it in the specific case of our framework.

A.1 Simultaneous Inverse Compositional Algorithm with a Prior

We briefly present here the algorithm *simultaneous inverse compositional with a prior* (SICP) (Baker et al., 2004). This is a *Gauss-Newton* algorithm that finds a local minimum of the energy $J(\lambda, p)$ (5) for general cases of prior functions $G_i(\lambda, p)$ and warps $W_p(x)$ that are controlled by some parameters p.

The algorithm starts from some initial estimates of λ and p. Afterwards, in every iteration, the previous estimates of λ and p are updated to λ' and p' as follows. It is considered that a vector $\Delta\lambda$ is added to λ:

$$\lambda' = \lambda + \Delta\lambda \tag{8}$$

and a warp with parameters Δp is applied to the synthesized image $A_0(x) + \sum \lambda_i A_i(x)$. As an approximation, the latter is taken as equivalent to updating the warp parameters from p to p' by composing $W_p(x)$ with the inverse of $W_{\Delta p}(x)$:

$$W_{p'} = W_p \circ W_{\Delta p}^{-1} \ . \tag{9}$$

From the above relation, given that p is constant, p' can be expressed as a $\mathbb{R}^{N_p} \to \mathbb{R}^{N_p}$ function of Δp, $p' = p'(\Delta p)$, with $p'(\Delta p = 0) = p$. Further, $p'(\Delta p)$ is approximated with a first order Taylor expansion around $\Delta p = 0$:

$$p'(\Delta p) = p + \frac{\partial p'}{\partial \Delta p}\Delta p \ . \tag{10}$$

where $\frac{\partial p'}{\partial \Delta p}$ is the Jacobian of the function $p'(\Delta p)$, which generally depends on Δp.

Based on the aforementioned type of updates of λ and p as well as the considered approximations, the values $\Delta\lambda$ and Δp are specified by minimizing the following energy:

$$\begin{aligned} F(\Delta\lambda, \Delta p) = \sum_x \Big\{ & A_0\big(W_{\Delta p}(x)\big) + \sum_{i=1}^{N_c} (\lambda_i + \Delta\lambda_i) A_i\big(W_{\Delta p}(x)\big) \\ & - f\big(W_p(x)\big) \Big\}^2 + \sum_{i=1}^{N_G} G_i^2 \left(\lambda + \Delta\lambda, p + \frac{\partial p'}{\partial \Delta p}\Delta p \right) \ , \end{aligned}$$

simultaneously with respect to $\Delta\lambda$ and Δp. By applying first order Taylor approximations on the two terms of the above energy $F(\lambda, p)$, one gets:

$$\begin{aligned} F(\Delta\lambda, \Delta p) \approx & \sum_x \left\{ E_{sim}(x) + SD_{sim}(x) \begin{pmatrix} \Delta\lambda \\ \Delta p \end{pmatrix} \right\}^2 + \\ & \sum_{i=1}^{N_G} \left\{ G_i(\lambda, p) + SD_{G_i} \begin{pmatrix} \Delta\lambda \\ \Delta p \end{pmatrix} \right\}^2 , \end{aligned} \tag{11}$$

where $E_{sim}(x)$ is the image of reconstruction error evaluated at the model domain:

$$E_{sim}(x) = A_0(x) + \sum_{i=1}^{N_c} \lambda_i A_i(x) - f\big(W_p(x)\big)$$

and $SD_{sim}(x)$ is a vector-valued "steepest descent" image with $N_c + N_p$ channels, each one of them corresponding to a specific component of the parameter vectors λ and p:

$$SD_{sim}(x) = \left[A_1(x), \ldots, A_{N_c}(x), \left(\nabla A_0(x) + \sum_{i=1}^{N_c} \lambda_i \nabla A_i(x) \right) \frac{\partial W_p(x)}{\partial p} \right] , \tag{12}$$

where the gradients $\nabla A_i(x) = \left[\frac{\partial A_i}{\partial x_1}, \frac{\partial A_i}{\partial x_2}\right]$ are considered as row vector functions. Also SD_{G_i}, for each $i = 1,..,N_G$, is a row vector with dimension $N_c + N_p$ that corresponds to the steepest descent direction of the prior term $G_i(\lambda, p)$:

$$SD_{G_i} = \left(\frac{\partial G_i}{\partial \lambda}, \frac{\partial G_i}{\partial p}\frac{\partial p'}{\partial \Delta p}\right) . \tag{13}$$

The approximated energy $F(\lambda, p)$ (11) is quadratic with respect to both $\Delta\lambda$ and Δp, therefore the minimization can be done analytically and leads to the following solution:

$$\left(\begin{array}{c}\Delta\lambda \\ \Delta p\end{array}\right) = -H^{-1}\left[\sum_x SD_{sim}^T(x)E_{sim}(x) + \sum_{i=1}^{N_G} SD_{G_i}^T G_i(\lambda, p)\right], \tag{14}$$

where H is the matrix (which approximates the Hessian of F):

$$H = \sum_x SD_{sim}^T(x)SD_{sim}(x) + \sum_{i=1}^{N_G} SD_{G_i}^T SD_{G_i} .$$

In conclusion, in every iteration of the SICP algorithm, the Equation (14) is applied and the parameters λ and p are updated using Equations (8) and (10). This process terminates when a norm of the update vector $\left(\begin{array}{c}\Delta\lambda \\ \Delta p\end{array}\right)$ falls below a relatively small threshold and then it is considered that the process has converged.

A.1.1 COMBINATION WITH LEVENBERG-MARQUARDT ALGORITHM

In the algorithm described above, there is no guarantee that the original energy (5), that is the objective function before any approximation, decreases in every iteration; it might increase if the involved approximations are not accurate. Therefore, following Baker and Matthews (2002), we use a modification of this algorithm by combining it with the *Levenberg-Marquardt* algorithm: In Equation (14) that specifies the updates, we replace the Hessian approximation H by $H + \delta\,\text{diag}(H)$, where δ is a positive weight and $\text{diag}(H)$ is the diagonal matrix that contains the diagonal elements of H. This corresponds to an interpolation between the updates given by the Gauss-Newton algorithm and weighted gradient descent. As δ increases, the algorithm has a behavior closer to gradient descent, which means that from the one hand is slower but from the other hand yields updates that are more reliable, in the sense that the energy will eventually decrease for sufficiently large δ.

In every iteration, we specify the appropriate weight δ as follows. Starting from setting δ to $1/10$ of its value in the previous iteration (or from $\delta = 0.01$ if this is the first iteration), we compute the updates $\Delta\lambda$ and Δp using the Hessian approximation $H + \delta\,\text{diag}(H)$ and then evaluate the original energy (5). If the energy has decreased we keep the updates and finish the iteration. If the energy has increased, we set $\delta \to 10\,\delta$ and try again. We repeat that step until the energy decreases.

A.2 Specialization in the Current Framework

In this section, we derive the SICP algorithm for the special case that concerns our method. This case arises when 1) the general warps $W_p(x)$ are specialized to affine transforms and 2) the general prior functions $G_i(\lambda, p)$ are given by Equation (6).

A.2.1 THE CASE OF AFFINE TRANSFORMS

In our framework, the general warps $W_p(x)$ of the SICP algorithm are specialized to affine transforms with parameters $p = (p_1 \cdots p_6)$ that are defined by:

$$W_p(x,y) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

In this special case, which is analyzed also in Baker et al. (2004), the Jacobian $\frac{\partial W_p(x)}{\partial p}$ that is used in Equation (12) is given by:

$$\frac{\partial W_p(x)}{\partial p} = \begin{pmatrix} x_1 & 0 & x_2 & 0 & 1 & 0 \\ 0 & x_1 & 0 & x_2 & 0 & 1 \end{pmatrix}.$$

The restriction to affine transforms implies also a special form for the Jacobian $\frac{\partial p'}{\partial \Delta p}$ that is used in Equation (13). More precisely, as described in Baker et al. (2004), a first order Taylor approximation is first applied to the inverse warp $W_{\Delta p}^{-1}$ and yields $W_{\Delta p}^{-1} \approx W_{-\Delta p}$. Afterwards, based on Equation (9) and the fact that the parameters of a composition $W_r = W_p \circ W_q$ of two affine transforms are given by:

$$r = \begin{pmatrix} p_1 + q_1 + p_1 q_1 + p_3 q_2 \\ p_2 + q_2 + p_2 q_1 + p_4 q_2 \\ p_3 + q_3 + p_1 q_3 + p_3 q_4 \\ p_4 + q_4 + p_2 q_3 + p_4 q_4 \\ p_5 + q_5 + p_1 q_5 + p_3 q_6 \\ p_6 + q_6 + p_2 q_5 + p_4 q_6 \end{pmatrix},$$

the function $p'(\Delta p)$ (10) is approximated as:

$$p'(\Delta p) = \begin{pmatrix} p_1 - \Delta p_1 - p_1 \Delta p_1 - p_3 \Delta p_2 \\ p_2 - \Delta p_2 - p_2 \Delta p_1 - p_4 \Delta p_2 \\ p_3 - \Delta p_3 - p_1 \Delta p_3 - p_3 \Delta p_4 \\ p_4 - \Delta p_4 - p_2 \Delta p_3 - p_4 \Delta p_4 \\ p_5 - \Delta p_5 - p_1 \Delta p_5 - p_3 \Delta p_6 \\ p_6 - \Delta p_6 - p_2 \Delta p_5 - p_4 \Delta p_6 \end{pmatrix}.$$

Therefore, its Jacobian is given by:

$$\frac{\partial p'}{\partial \Delta p} = - \begin{pmatrix} 1+p_1 & p_3 & 0 & 0 & 0 & 0 \\ p_2 & 1+p_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+p_1 & p_3 & 0 & 0 \\ 0 & 0 & p_2 & 1+p_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+p_1 & p_3 \\ 0 & 0 & 0 & 0 & p_2 & 1+p_4 \end{pmatrix}.$$

A.2.2 SPECIFIC TYPE OF PRIOR FUNCTIONS

Apart from the restriction to affine transforms, in the proposed framework of the regularized shape-appearance model fitting, we have derived the specific formulas of Equation (6) for the prior functions $G_i(\lambda, p)$ of the energy $J(\lambda, p)$ in Equation (5). Therefore, in our case, their partial derivatives, which are involved in the above described SICP algorithm (see Equation (13)), are specialized as follows:

$$\frac{\partial G_i}{\partial p} \overset{(7)}{=} \begin{cases} 0, & 1 \leq i \leq 2N_c \\ \frac{m_3}{\sigma_{\tilde{p}_j}} v_{\tilde{p}_j}^T \, , j = i - 2N_c \, , & 2N_c + 1 \leq i \leq 2N_c + N_p \\ \frac{m_4}{\sigma_{\varepsilon_{\tilde{p},j}}} v_{\tilde{p}_j}^T \, , j = i - 2N_c - N_p \, , & 2N_c{+}N_p{+}1 \leq i \leq 2N_c{+}2N_p \end{cases} ,$$

$$\frac{\partial G_i}{\partial \lambda} = \begin{cases} \frac{m_1}{\sigma_{\lambda_i}} e_i^T \, , & 1 \leq i \leq N_c \\ \frac{m_2}{\sigma_{\varepsilon_{\lambda,j}}} e_j^T \, , j = i - N_c \, , & N_c + 1 \leq i \leq 2N_c \\ 0, & 2N_c + 1 \leq i \leq 2N_c + 2N_p \end{cases} ,$$

where e_i, $1 \leq i \leq N_c$, is the i-th column of the $N_c \times N_c$ identity matrix.

References

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

T. Ahmad, C.J. Taylor, and T.F. Lanitis, A. Cootes. Tracking and recognising hand gestures, using statistical shape models. *Image and Visual Computing*, 15(5):345–352, 1997.

A. Argyros and M. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *Proceedings of the European Conference on Computer Vision*, 2004.

V. Athitsos and S. Sclaroff. An appearance-based framework for 3d hand shape classification and camera viewpoint estimation. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 45–52, 2002.

S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Technical report, Carnegie Mellon University, 2002.

S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 4. Technical report, Carnegie Mellon University, 2004.

B. Bauer and K. F. Kraiss. Towards an automatic sign language recognition system using subunits. In *Proceedings of the International Gesture Workshop*, volume 2298, pages 64–75, 2001.

H. Birk, T.B. Moeslund, and C.B. Madsen. Real-time recognition of hand alphabet gestures using principal component analysis. In *Proceedings of the Scandinavian Conference Image Analysis*, 1997.

A. Blake and M. Isard. *Active Contours*. Springer, 1998.

R. Bowden and M. Sarhadi. A nonlinear model of shape and motion for tracking fingerspelt american sign language. *Image and Visual Computing*, 20:597–607, 2002.

P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

J. Cai and A. Goshtasby. Detecting human faces in color images. *Image and Visual Computing*, 18: 63–75, 1999.

F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Visual Computing*, 21(8):745–758, 2003.

S. Conseil, S. Bourennane, and L. Martin. Comparison of Fourier descriptors and Hu moments for hand posture recognition. In *Proceedings of the European Conference on Signal Processing*, 2007.

T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, 2004.

Y. Cui and J. Weng. Appearance-based hand sign recognition from intensity image sequences. *Computer Vision and Image Understanding*, 78(2):157–176, 2000.

L. Davies, David and W. Bouldin, Donald. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224 – 227, April 1979.

J.-W. Deng and H.T. Tsui. A novel two-layer PCA/MDA scheme for hand posture recognition. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 283–286, 2002.

DictaSign. Greek sign language corpus. http://www.sign-lang.uni-hamburg.de/dicta-sign/portal, 2012.

L. Ding and A. M. Martinez. Modelling and recognition of the linguistic components in american sign language. *Image and Visual Computing*, 27(12):1826 – 1844, 2009.

P. Dreuw, J. Forster, T. Deselaers, and H. Ney. Efficient approximations to model-based joint tracking and recognition of continuous sign language. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Sep. 2008.

I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.

W. Du and J. Piater. Hand modeling and tracking for video-based sign language recognition by robust principal component analysis. In *Proceedings of the ECCV Workshop on Sign, Gesture and Activity*, September 2010.

A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

H. Fillbrandt, S. Akyol, and K.-F. Kraiss. Extraction of 3D hand shape and posture from images sequences from sign language recognition. In *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pages 181–186, 2003.

R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Visual Computing*, 23(12):1080–1093, 2005.

T. Hanke. HamNoSys Representing sign language data in language resources and language processing contexts. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2004.

M.-K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187, February 1962.

C.-L. Huang and S.-H. Jeng. A model-based hand gesture recognition system. *Machine Vision and Application*, 12(5):243–258, 2001.

P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, Mar. 2007.

E. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2005.

S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in British sign language. In *Proceedings of the CVPR Workshop on Human Communicative Behavior Analysis*, 2009.

P. Maragos. *Morphological Filtering for Image Enhancement and Feature Detection*, chapter The Image and Video Processing Handbook. Elsevier, 2005.

I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

C. Neidle. Signstream annotation: Addendum to conventions used for the american sign language linguistic research project. Technical report, 2007.

C. Neidle and C. Vogler. A new web interface to facilitate access to corpora: development of the ASLLRP data access interface. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2012.

E.J. Ong, H. Cooper, N. Pugeault, and R. Bowden. Sign language recognition using sequential pattern trees. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2200–2207. IEEE, 2012.

Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2010.

V. Pitsikalis, S. Theodorakis, and P. Maragos. Data-driven sub-units and modeling structure for continuous sign language recognition with multiple cues. In *LREC Workshop Repr. & Proc. SL: Corpora and SL Technologies*, 2010.

L. R. Rabiner and R.W. Schafer. Introduction to digital speech processing. *Foundations and Trends in Signal Processing*, 1(1-2):1–194, 2007.

A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Affine-invariant modeling of shape-appearance images applied on sign language handshape classification. In *Proceedings of the International Conference on Image Processing*, Sep. 2010a.

A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *Proceedings of the ECCV Workshop on Sign, Gesture and Activity*, September 2010b.

J. Sherrah and S. Gong. Resolving visual uncertainty and occlusion through probabilistic reasoning. In *Proceedings of the British Machine Vision Conference*, pages 252–261, 2000.

P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 2004.

T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, Dec. 1998.

B. Stenger, A. Thayananthan, P.H.S Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9): 1372–1384, Sep. 2006.

G.J. Sweeney and A.C. Downton. Towards appearance-based multi-channel gesture recognition. In *Proceedings of the International Gesture Workshop*, pages 7–16, 1996.

N. Tanibata, N. Shimada, and Y. Shirai. Extraction of hand features for recognition of sign language words. In *Proceedings of the International Conference on Vision Interface*, pages 391–398, 2002.

J. Terrillon, M. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 54–61, 2000.

A. Thangali, J.P. Nash, S. Sclaroff, and C. Neidle. Exploiting phonological constraints for handshape inference in asl video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 521–528. IEEE, 2011.

S. Theodorakis, V. Pitsikalis, and P. Maragos. Advances in dynamic-static integration of movement and handshape cues for sign language recognition. In *Proceedings of the International Gesture Workshop*, 2011.

S. Theodorakis, V. Pitsikalis, I. Rodomagoulakis, and P. Maragos. Recognition with raw canonical phonetic movement and handshape subunits on videos of continuous sign language. In *Proceedings of the International Conference on Image Processing*, 2012.

M. Viola and M. J. Jones. Fast multi-view face detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2003.

C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 116–122, 1999.

Y. Wu and T.S. Huang. View-independent recognition of hand postures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 88–94, 2000.

M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (8):1061–1074, Aug. 2002.

S. Young, D. Kershaw, J. Odell, D. Ollason, V. Woodland, and P. Valtchevand. *The HTK Book*. Entropic Ltd., 1999.

J. Zieren, N. Unger, and S. Akyol. Hands tracking from frontal view for vision-based gesture recognition. In *Pattern Recognition*, LNCS, pages 531–539, 2002.