Detecting Hand-Head Occlusions in Sign Language Video*

Ville Viitaniemi[1], Matti Karppa[1], Jorma Laaksonen[1], and Tommi Jantunen[2]

[1] Department of Information and Computer Science,
Aalto University School of Science, Espoo, Finland
firstname.lastname@aalto.fi

[2] Sign Language Centre, Department of Languages,
University of Jyväskylä, Finland
tommi.j.jantunen@jyu.fi

Abstract. A large body of current linguistic research on sign language is based on analyzing large corpora of video recordings. This requires either manual or automatic annotation of the videos. In this paper we introduce methods for automatically detecting and classifying hand-head occlusions in sign language videos. Linguistically, hand-head occlusions are an important and interesting subject of study as the head is a structural place of articulation in many signs. Our method combines easily calculable local video properties with more global hand tracking. The experiments carried out with videos of the Suvi on-line dictionary of Finnish Sign Language show that the sensitivity of the proposed local method in detecting occlusion events is 92.6%. When global hand tracking is combined in the method, the specificity can reach the level of 93.7% while still maintaining the detection sensitivity above 90%.

1 Introduction

Statistical corpus-based approaches are common in current natural language research. In the case of sign languages, the corpora consist of videos which are typically annotated at least for signs on the basis of information concerning the locations, shapes, and movements of the hands producing them [1, 7]. Depending on the timetable, available resources and precision requirements, this annotation can be done either manually or with automatic computer-vision-based methods. With ready-made annotations, sign language researchers may exploit the corpora to postulate and empirically verify theories concerning the structure, conventions and frequencies of diverge phenomena in the sign languages.

The head of the signer is an important place of sign articulation — for example, in Finnish Sign Language over 25 percent of lexical signs are produced at or

* This work has been funded by the following grants of the Academy of Finland: 140245, Content-based video analysis and annotation of Finnish Sign Language (CoBaSiL); 251170, Finnish Centre of Excellence in Computational Inference Research (COIN); 134433, Signs, Syllables, and Sentences (3BatS).

near the area of the head. Consequently, knowing the exact or at least the approximate position of the active hand over the head will inevitably give valuable information for the automated annotation and analysis of sign language videos. Hand tracking is an essential part of any operational sign language recognition or analysis system. However, typical approaches based on skin color segmentation have difficulties when the skin blobs are merged because of the hands touching or otherwise occluding each other or the head region. Therefore, attempts at detecting the exact occlusion locations have been rare.

In our CoBaSiL project the aim is to develop new computer vision techniques for automated analysis, annotation and indexing of sign language videos. Our earlier works have concerned, for example, the extraction of hand motion information [5, 8, 9]. In this paper, we present a method for detecting hand-head occlusions. The method is based on local tracking of skin-colored points in the neighborhoods of the head. We also combine the local approach with global tracking of the hand movements to reduce the number of false positive detections. In the experiments, the efficiency of the method is evaluated with videos of the on-line dictionary of Finnish Sign Language, *Suvi*[3], by measuring the sensitivity and specificity of the detections on five linguistically motivated head areas: forehead, cheeks, nose, mouth and neck.

The rest of this paper is organized as follows: Section 2 specifies the problem setting and reviews earlier related works. Section 3 introduces a set of video processing techniques and prepares the reader for Section 4 where we describe the approach we propose for hand-head occlusion detection. Section 5 shows the experiments carried out with *Suvi* videos. Conclusions are drawn in Section 6.

2 Head Occlusions in Signing

From the linguistic point of view, the hand-head occlusion events are very important: in the production of an isolated sign, for example, such occlusions typically signify that one of the main structural parameters of the sign — the place of articulation — is the head. On the area of the head, several sets of linguistically significant sub-locations may be further identified. *Suvi*, the on-line dictionary of Finnish Sign Language, distinguishes five such basic locations: forehead (and the top of the head), the area of cheeks (including ears and eyes), nose, the area of mouth and chin, and neck. These are shown as drawings in Figure 1a and they represent the places of articulation e.g. in signs BLACK, BEG, FAULT, GOOD, and DESIRE, respectively.

Not all visible occlusions are structurally meaningful, or even intentional. Such occlusions occur for two main reasons. First, the signer may raise his or her hand on the level where it occludes the face even though the relatively large distance between the hand and the face would indicate that the head really is not the intended place of articulation for any sign; this type of occlusions may serve some communicative function, e.g. to maximize the perception of signs for the

[3] http://suvi.viittomat.net/

addressee who typically looks at the signers face, or be completely unintentional. Second, the signer may aim e.g. at touching the nose (as in the sign FAULT) but, at the same time, the hand occludes also the mouth and neck areas. This is because it is practically impossible to cover only the nose (or forehead) without simultaneous or preceding coverage of other lower-level head areas.

In real-life signing, it is also typical that places of articulation are not realized in the way they appear in idealized models. This is caused by co-articulation and, from the perspective of articulation places on the head, often corresponds to cases where the signer does not raise the active hand high enough, or bring it close enough from the side of the head, to actually cover any part of the head. The result is that intended places of articulation on the head are not always visible as hand-head occlusions. For a human observer this is not a problem, but for an automatic recognition system it certainly is. However, regardless of such limitations, the automated detection of hand–head occlusions will nevertheless provide sign language linguists valuable phonetic information concerning the actual production of signs, to be integrated into linguistic theories of sign structure.

In many state-of-the-art works on sign language recognition and analysis, the locations of the articulating hands are described with a single (x, y)-coordinate pair each, typically aimed at indicating the positions of the palms [2]. This level of presentation has also been used for annotating publicly available sign language video databases and benchmarks [3]. For much of the work geared towards recognition of sign language (e.g. [11]), the coarse palm-coordinate-based presentation can be quite sufficient. Obtaining even such co-ordinates is challenging in practice, though, when hands occlude each other or the head of the signer. For example, [4] resorts to tracking only the location of the merged skin blob in such cases. Some works regard the palm coordinate estimation unreliable and support multiple hypotheses of hand locations, e.g. [12].

In contrast to the recognition-oriented research, we take a more detailed approach to the analysis of signing and try to delve more exactly into the phonetic constituents of sign language, thereby targeting our work towards the linguistic researchers. Along these lines, in this paper we aim at a detailed analysis on the exact areas where the articulating hand touches or otherwise occludes the head.

3 Video Processing Techniques for SL Analysis

3.1 Video Pre-processing

As pre-processing, we apply three generic sign-language video analysis techniques to our material. The first processing step is the Viola-Jones cascade face detector, as implemented in the OpenCV[4] library. In the second stage, the face detections are fed into the facial landmark detector [10] algorithm that outputs the estimated coordinates of the center of the face, along with the approximate coordinates of seven facial landmarks: the canthi of the eyes, the tip of the

[4] http://opencv.org/

nose, and the left and right corners of the mouth. Figure 1b shows the detected landmarks as white circles in an example video frame. Thirdly, the skin-colored regions of each frame are located with an Extreme Learning Machine (ELM) based classifier on a per-pixel basis. In this paper, we use the word *blob* to denote a connected component in the detected binary skin mask. The performance of these three pre-processing steps is very satisfactory for the video material used in our experiments.

3.2 Tracking of Local Image Neighborhoods

The proposed occlusion detection method builds on tracking local image neighborhoods, in particular, small rectangular patches. We have chosen this elementary matching method because preliminary experiments with the video material at our disposal have indicated that some more advanced descriptors such as SIFT do not remain stable enough between the frames to make it practicable to base the tracking on them. Probably such descriptors suffer too severely from compression artifacts and motion blur in our material. In order to make the tracking more reliable, we tie together a collection of multiple nearby points and track them collectively.

Let us track the set of M points from a reference frame r to a target frame t. Let the coordinates of these tracked points be $\{\mathbf{r}_i\}_{i=1}^{M}$ in the reference frame. We impose a topology $\{N(i)\}_{i=1}^{M}$ upon the points. Here the set $N(i)$ specifies the indices of points that are neighbors of the ith point. There is no necessity of the topology to reflect any specific geometric notion of adjacency in the original image plane. Subsequently, we use the word *gridlet* to denote such a set of points together with the topology defined upon the set. In our formulation, the goal of the tracking is to find the coordinates $\{\mathbf{t}_i\}_{i=1}^{M}$ in the target frame so that the tracking cost function C is minimized:

$$\min_{\{\mathbf{t}_i\}_{i=1}^{M}} C = \sum_{i=1}^{M} \left(A\big(I_r(\mathbf{r}_i), I_t(\mathbf{t}_i)\big) + \alpha \sum_{j \in N(i)} B\big(|\|\mathbf{t}_i - \mathbf{t}_j\| - \|\mathbf{r}_i - \mathbf{r}_j\||\big) \right). \quad (1)$$

Here $I_r(\mathbf{r}_i)$ and $I_t(\mathbf{t}_i)$ are the image neighborhoods of the points $\mathbf{r}_i$ and $\mathbf{t}_i$ in the reference and target frames, respectively, and $A(\cdot,\cdot)$ is the template matching distance. α is a weight parameter of the method and $B(\cdot)$ is a scalar weighting function of distance differences. The cost function thus balances the sum of template matching costs of individual points with a measure how much the inter-point distances in the target frame differ from the corresponding distances in the reference frame. In this paper, the following algorithm has been used for the approximate minimization of the cost function:

1. Initialize tracking, i.e. select initial values for $\{\mathbf{t}_i\}_{i=1}^{M}$ e.g. on the basis of the estimated motion field.
2. Denote the set of indices of the target points requiring update with R. Initialize R by inserting all the indices $1, \ldots, M$ into it.

3. Repeat until R is empty or some external stopping criterion is met (e.g. number of iterations reaches a set maximum)
   (a) Randomly select an index j from R.
   (b) Set $\mathbf{t}_{\text{old}} = \mathbf{t}_j$ and remove j from R.
   (c) Search a new location for the point $\mathbf{t}_j$ that minimizes C of Eq. (1).
   (d) If $\mathbf{t}_j \neq \mathbf{t}_{\text{old}}$, add indices in $N(j)$ into R.

This tracking algorithm will be used as an ingredient in methods of Sections 3.4, 3.5, 3.6 and 4.1 with different choices of the gridlets to track.

3.3 Selection of Points to Track

Given an image area, the OpenCV `GoodFeaturesToTrack` detector is applied to this area to select a list of salient points. The list is augmented with points sampled from the boundary of the area. Furthermore, additional points are selected from a regular grid from areas that are too far away from all the previously selected points.

3.4 Forming of Facial Prohibition Masks

In the material we target at, motion blur and abrupt shape changes are typically much more prevalent in hands than in the area of the face and therefore the tracking of image features in hands is much less reliable than within the face. Because of this, when tracking hand areas over the face, we first track the stable facial areas. Then we constrain hand tracking so that we prohibit the tracker from selecting such locations for the hand area points that would overlap with the previously tracked face. For this, we employ *facial prohibition masks*.

The prohibition masks are binary and formed as follows. A reference frame f_0 is selected so that the face is completely visible in the frame, as well as in $p-1$ preceding frames. From the reference frame, a set of potential points to be tracked is selected using the method of Section 3.3. A set of gridlets is formed out of the points by enumerating all the sets of M nearby points. For the experiments we have used $M = 4$.

We determine an individual tracking cost threshold $T(g)$ for each gridlet g based on tracking the gridlet frame by frame over the p frames until the frame f_0 using the algorithm of Section 3.2. Having determined the thresholds, we form the sets $U(f)$ of unoccluded face points for each frame f as follows:

```
Set the reference frame F(g) of each gridlet g to be f_0.
for each frame f from f_0 + 1 onwards do
    for each gridlet g do
        Track the gridlet g from frame F(g) to frame f.
        if tracking cost C < T(g) then
            Set F(g) = f.
            Add all the points {t_i} of the gridlet g to U(f).
        end
    end
end
```

The facial prohibition mask $X(f)$ is formed from $U(f)$ by using morphological operations for selecting areas where the density of unoccluded face pixels exceeds a threshold.

3.5 Tracking of Hand Blobs

In Section 4.2 we will refine our local image property based occlusion detection method with information about the overall body configuration of a signer. This we obtain by tracking skin blobs that are separate from the head at some moment of time. In practice such blobs are hands in our material. The following procedure collects lists $G(f)$ of gridlets tracked to each frame f:

Initialize the list G of tracked gridlets as empty.
for *each frame f of the video* **do**
Track each gridlet in G from frame $f-1$ to frame f using the facial prohibition mask $X(f)$ of Section 3.4.

Go through all skin blobs in f. If a blob separate from the head is found with no gridlets tracked to it, form a new gridlet g from the blob (see below). If two of the existing gridlets in G have been tracked to the same blob, replace with g the gridlet that would have been more likely to move to the location of g on basis of earlier movement history. Otherwise, add g to G.

Go through all gridlets in G. If the blob where a gridlet has been tracked to is separate from head and contains only one gridlet, the gridlet is re-selected from the points in that blob.

Set $G(f) = G$.
end

As a post-processing step, such gridlets are removed from the head blob that never leave the head again. In the above algorithm, a new gridlet is formed from the points of a blob as follows. $3M$ points are randomly selected from within the blob. M of these points are selected that are farthest from the blob boundaries. A gridlet is defined with all the M points and a topology where each point has J neighbors randomly selected from the remaining $M-1$ points. In the experiments of this paper $M = 250$ and $J = 4$.

3.6 Motion Discontinuity Detection

Discontinuities in the local motion patterns are used as one source of information in the occlusion detection methods we propose in Section 4. The binary motion discontinuity mask $D(f)$ of the face area indicates the spatial discontinuities of the motion between frames $f-1$ and f. Its calculation begins by estimating the facial area motion field between the frames on the basis of tracking gridlets consisting of $M = 4$ points with the method of Section 3.2. In the mask, those pixels are set to non-zero where a discontinuity measure of the motion field multiplied by the final tracking cost function in the area exceeds a threshold.

Fig. 1. (a) The five main head locations of *Suvi* (from left to right): forehead and top of the head; eye, ear, and cheek; nose; mouth and chin; and neck (images from *Suvi*). (b) Corresponding partitioning of the head area based on detected facial landmarks (white circles).

3.7 Head Area Partitioning

We discretize the occlusions of the head into five distinct classes. The classes are derived from the place of articulation labels that have been selected by sign language experts for indexing the *Suvi* dictionary. Figure 1a shows the classes. Our classification method assigns head part labels to the pixels of the head using simple geometric rules that are based on the locations of the facial landmarks detected in Section 3.1. An example of the head partitioning is shown in Figure 1b.

4 Methods for Occlusion Detection

In this section we describe the methods we propose for the detection of facial occlusions. Section 4.1 outlines the basic method that is based on local video properties. The method is then further refined in Section 4.2 by taking information from global hand tracking into account.

4.1 Local-only Method

As preparation, the facial prohibition mask $X(f)$ and set $U(f)$ of unoccluded face points are determined for every frame f of the video as described in Section 3.4. In this section, we use the same reference frame f_0 as in Section 3.4. The sets $O(f)$ of points originating from outside the face region are created as follows:

The list G of tracked gridlets is initialized as empty.
for *each frame* f *from* $f_0 + 1$ *onwards* **do**
  Select list L of outside face skin points with the method of Section 3.3.
  Add all fully-connected gridlets of $M = 4$ nearby points in L to list G.
  for *each gridlet* g *in* G **do**
    Track the gridlet g from frame f to frame $f + 1$ (Section 3.2) using the facial prohibition mask $X(f + 1)$ of Section 3.4.
    Append all the points $\{\mathbf{t}_i\}$ of g to the set $O(f + 1)$.
    Remove g from G if none of its points is in the face area.
  end
end

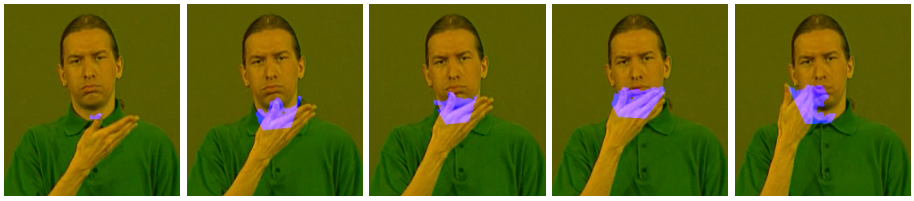

Fig. 2. Example of a detected occlusion. The occlusion pixels are shown in light blue, below the frames are the labels of the occluded head regions (C=cheeks, M=mouth, N=neck).

The algorithm tentatively detects those image regions as occluded that are closer to some point in $O(f)$ than to any point in $U(f)$. These tentative occlusions are filtered based on motion discontinuity masks $D(f)$ of Section 3.6. The final output of the occlusion detection method is the vector of five pixel counts for each frame of the video, indicating the numbers of detected occlusion pixels in each of the five face regions according to the head area partitioning of Section 3.7.

4.2 Globally Refined Method

In the refined version of the method the results of the basic algorithm of Section 3.2 are constrained on basis of hand blob tracking described in Section 3.5, so that some of the false positive detections can be ruled out. Firstly, we constrain the step of the algorithm where the list L is formed. In the refined version, such points are not added to the list whose spatial distance from the tracked hands exceeds a threshold (in our experiments 30 pixels). When forming the list L, the area within the tracked hands is counted as area outside the face even if the hands were tracked to the head blob.

Another way of constraining the results is based on spotting the moments when none of the hands has been tracked to the head blob. In such a case, all the detected occlusions are flagged as erroneous. Also the list G of tracked gridlets maintained by the algorithm of Section 3.2 is emptied on such occasions.

5 Experiments

5.1 Data

The proposed occlusion detection methods were applied to footage from the *Suvi* on-line dictionary of Finnish Sign Language. For these experiments, we chose to use the videos that have been tagged in the dictionary meta data to have the articulation place in the head region as these are very likely to exhibit hand-head occlusions. There are 324 such videos in the dictionary consisting of the total of 27800 frames shot in 25 fps. Figure 2 demonstrates the output of the occlusion detection method in one of these videos.

In order to assess the quality of the results the automatic methods produce, this subset of videos was manually annotated by indicating which of the five *Suvi* head areas shown in Figure 1a appear to be occluded in each frame. The annotations are somewhat subjective, due to the modest image quality and the *Suvi* head regions being open to subjective interpretations. The videos and annotations are available upon request.

5.2 Performance Evaluation

The performance of the automatic methods was evaluated by comparing the counts of detected occlusion events to the manual ground truth annotations. We define an occlusion event to be a temporally contiguous period during which a certain head region is occluded by a signing hand. To make the evaluation robust against short transitory moments of either occlusion or non-occlusion, both the ground truth and the automatic detections are filtered before determining the events. Here we have used majority filtering with the window length of five frames.

We calculate two performance metrics for each facial region by comparing the sets of events in the filtered detections and ground truth. *Sensitivity* counts how large a fraction of the real occlusion events of the ground truth are detected by the automatic method. Here we have interpreted an event as being detected if an occlusion is reported for at least a half of the duration of the event. *Specificity* is calculated symmetrically, only that the roles of the ground truth and the automatic detections are interchanged. That is, the specificity counts how large a fraction of automatically detected events also appear in the ground truth.

5.3 Results

First we look at the proposed methods' capability of detecting hand-head occlusions on the whole. The bottom row in Table 1 shows the performance in the task where we ask the question whether the head is occluded at all, not taking into account the exact part. The performance of both versions of the method can be said to be on a rather satisfactory level in detecting facial occlusions even though not 100% of the occlusion events are detected. In some of the occlusions only tiny parts of the hand momentarily move over the face area and not detecting such occlusions is not too serious for the usefulness of the method. The basic method being a bit more sensitive but less specific than the refined method is quite understandable as the refinement essentially just attempts to block erroneous detections out.

The top rows of Table 1 show more closely how our methods manage to identify the head region that is occluded. We notice that also here the basic method usually is a bit more sensitive, but noticeably less specific than the refined method. Generally the sensitivity of the methods is quite satisfactory compared to the specificity. For some head regions, mostly for the forehead, but also for the cheeks to some degree, the specificity is reasonable as well. However, problematic areas can also be identified. Especially poor is the specificity for the

Table 1. The performance in detection of occluded head areas in the whole material. The GT and n_{d} columns display the number of actual and detected occlusion events.

	GT	Local-only method			Globally refined method		
	events	n_{d}	sensitivity	specificity	n_{d}	sensitivity	specificity
forehead	97	123	85.6%	69.9%	114	82.5%	84.2%
cheeks	237	331	90.7%	52.6%	318	86.5%	68.9%
nose	81	173	74.1%	32.4%	153	74.1%	35.3%
mouth	427	426	85.9%	45.5%	410	83.9%	55.4%
neck	455	455	92.0%	50.3%	451	90.2%	61.2%
overall	**337**	**424**	**92.6%**	**84.7%**	**427**	**90.2%**	**93.7%**

nose. This can be partially attributed to the difficulty of precise determination of the *Suvi* head regions on the basis of the facial landmarks, which mostly affects the nose and mouth regions. The neck area is problematic since our methods are actually targeted at detecting occlusions of the face as opposed to the whole head area, and the neck area only partially coincides with face. Visual inspection of the detection results shows that typically the detections tend to spread both spatially and temporally into neighboring head regions. Although occlusion is mostly reported roughly in the right area, accurately separating occlusions of nearby mouth, nose and cheek regions represents a challenge to our current method.

In the final experiment shown in Table 2, the evaluation was limited to those linguistically distinctive places of articulation that are used in the expert-prepared indexing of the *Suvi* dictionary for on-line searches. In general, we notice that both methods work much better for this restricted subset of occlusion events than for the totality of the occlusions in the video material. The neck area still forms an exception: signs indexed with the neck as articulation place are often such that the hand hardly enters the visual face area, but stays below it.

The occlusions of the areas used for the dictionary indexing apparently are so much more pronounced and visually more distinctive than other sporadic occlusions that their detection with our methods is more accurate. Based on these findings, one might argue that the proposed methods are actually reasonably good in detecting linguistically significant occlusion events that often correspond to structural places of articulation. It is the unintentional occlusions of more coincidental nature where the methods fail the most.

6 Conclusions and Discussion

In this paper, we have presented a method for detecting hand-head occlusions in sign language videos. The accuracy in detecting occlusion events overall is quite good, sensitivity and specificity both exceeding 90%. More detailed classification of the occlusion events according to the covered head regions is more problematic, although some good results are achieved also here, especially in

Table 2. The performance in detection of occlusions of indexing head areas. The GT and n_{d} columns display the number of actual and detected occlusion events.

	GT	Local-only method			Globally refined method		
	events	n_{d}	sensitivity	specificity	n_{d}	sensitivity	specificity
forehead	77	91	88.3%	76.9%	91	85.7%	92.3%
cheeks	88	114	88.6%	57.9%	113	79.5%	79.6%
nose	29	27	62.0%	81.4%	25	55.2%	80.0%
mouth	105	133	82.9%	85.0%	130	80.0%	93.1%
neck	14	8	35.7%	25.0%	8	35.7%	75.0%

the detection of the linguistically most significant occlusions (i.e. the structural places of articulation). The specificity in separating the head regions is greatly increased when semi-global information from hand tracking is combined with the purely local occlusion detection method.

Our methods have been targeted at the readily existing video material that is of modest technical quality. This explains many performance problems, as well as the need to use primitive and robust video analysis methods. It may very well be asked whether one would be better off with collecting higher quality material that would be more suitable to automatic analysis. This approach has been taken in some recent work, employing much better resolution and frame rates than in *Suvi*. Much of the new work on gesture analysis is even based on Kinect recordings that include depth information, facilitating much easier analysis.

However, collecting good quality material is both very tedious and expensive as there is a large gap between the quality that is perceptually sufficient for a natural viewing experience and the quality that would be well suited for automatic analysis. Readily existing collections of perceptual quality videos contain huge amounts of linguistically valuable material. This material has to be analyzed somehow as collecting it again with better technology would be out of the question, both because it would be infeasibly laborious, and perhaps more importantly, because of the impossibility of re-constructing the unique recording situations of the videos while the language and the environment continuously keeps evolving.

We have already used our method for the analysis of longer videos [6]. In the future we will continue along these lines and apply our method to linguistically more interesting larger video collections and longer videos such as the sign usage example videos in the *Suvi* dictionary. We will develop further our way of combining local and global information, possibly looking at model-based approaches of tracking the body parts. One future possibility of research is provided by the question whether we can distinguish touching from other occlusion events.

References

1. O. Crasborn and I. Zwitserlood. Annotation of the video data in the "Corpus NGT". Online publication http://hdl.handle.net/1839/00-0000-0000-000A-3F63-

4, 2008. Dept. of Linguistics, and Centre for Language Studies, Radboud University Nijmegen, The Netherlands.

2. P. Dreuw, J. Forster, T. Deselaers, and H. Ney. Efficient approximations to model-based joint tracking and recognition of continuous sign language. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, Amsterdam, The Netherlands, Sept. 2008.
3. P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff, and H. Ney. Benchmark databases for video-based automatic sign language recognition. In *LREC*. European Language Resources Association, 2008.
4. J. Han, G. Awad, and A. Sutherland. Automatic skin segmentation and tracking in sign language recognition. *IET Computer Vision*, 3(1):24 –35, March 2009.
5. T. Jantunen, M. Koskela, J. Laaksonen, and P. Rainò. Towards automated visualization and analysis of signed language motion: Method and linguistic issues. In *Proceedings of 5th International Conference on Speech Prosody*, Chicago, Ill. (USA), May 2010. Available online at http://speechprosody2010.illinois.edu/papers/100006.pdf.
6. T. Jantunen, V. Viitaniemi, M. Karppa, and J. Laaksonen. The head as a place of articulation: From automated detection to linguistic analysis. July 2013. Poster accepted for presentation at 11th Theoretical Issues in Sign Language Research conference, University College London, July 10-13, 2013.
7. T. Johnston. Guidelines for annotation of the video data in the Auslan corpus. Online publication http://media.auslan.org.au/media/upload/attachments/ Annotation_Guidelines_Auslan_CorpusT5.pdf, 2009. Dept. of Linguistics, Macquarie University, Sydney, Australia.
8. M. Karppa, T. Jantunen, M. Koskela, J. Laaksonen, and V. Viitaniemi. Method for visualisation and analysis of hand and head movements in sign language video. In C. Kirchhof, Z. Malisz, and P. Wagner, editors, *Proceedings of the 2nd Gesture and Speech in Interaction conference (GESPIN 2011)*, Bielefeld, Germany, 2011. Available online as http://coral2.spectrum.uni-bielefeld.de/gespin2011/final/Jantunen.pdf.
9. M. Karppa, T. Jantunen, V. Viitaniemi, J. Laaksonen, B. Burger, and D. De Weerdt. Comparing computer vision analysis of signed language video with motion capture recordings. In *Proceedings of 8th Language Resources and Evaluation Conference (LREC 2012)*, pages 2421–2425, Istanbul, Turkey, May 2012. Available online at http://www.lrec-conf.org/proceedings/lrec2012/pdf/321_Paper.pdf.
10. M. Uřičář, V. Franc, and V. Hlaváč. Detector of facial landmarks learned by the structured output SVM. In G. Csurka and J. Braz, editors, *VISAPP '12: Proceedings of the 7th International Conference on Computer Vision Theory and Applications*, volume 1, pages 547–556, Portugal, February 2012. SciTePress — Science and Technology Publications.
11. H.-D. Yang, S. Sclaroff, and S.-W. Lee. Sign language spotting with a threshold model based on conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1264–1277, 2009.
12. R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):462–477, 2010.