
# SignSpeak

## Scientific understanding and vision-based technological development for continuous sign language recognition and translation

## Minor Deliverable D.3.3.M24 – Feature Extraction (formerly named 'Collection of Segmented and Glossed Videos')

Release version: V3 (22-09-2011)

Grant Agreement Number 231424

**Small or medium-scale focused research project (STREP)**
**FP7-ICT-2007-3. Cognitive Systems, Interaction, Robotics**

Project start date: 1 April 2009
Project duration: 36 months

| Dissemination Level | | |
|---|---|---|
| PU | Public (can be made available outside of SignSpeak Consortium without restrictions) | X |
| RE | Restricted to SignSpeak Programme participants and a specified group outside of SignSpeak consortium | |
| IN | SignSpeak Internal (only available to (all) SignSpeak programme participants) | |
| LI | SignSpeak Limited (only available to a specified subset of SignSpeak programme participant) | |
| Distribution List (only for RE or LI documents) | | |

# 0 General Information

## 0.1 Document Information

| | |
|---|---|
| Title | Feature Extraction – Report (formerly named 'Collection of Segmented and Glossed Videos') |
| Type | Minor Deliverable |
| Ref | D.3.3.M24 |
| Target Version | V.3 |
| Current Version | V.3 |
| File | SignSpeak.D.3.3.M24.doc |
| Author(s) | Jaume Vergés-Llahí (CRIC), Philippe Dreuw (RWTH) , Jens Forster (RWTH) , Yannick Gweth (RWTH). |
| Reviewer(s) | Gregorio Martínez Ruíz and Jordi Barret (CRIC) |
| Approver(s) | Gregorio Martínez Ruíz (CRIC) |
| Approval date | 22-09-2011 |
| Release date | 22-09-2011 |

## 0.2 Document Scope

This document describes the algorithms developed and implemented as part of Task 3.1, concerning the spatial feature extraction as part of the multi-modal visual analysis in WP3. It accompanies Deliverable D.3.3 on the results of processing and extracting features from the collections of videos employed in this project, and is an extension of previous Deliverable D.3.1 and its report.

## 0.3 Document Content


0 General Information....2
0.1 Document Information....2
0.2 Document Scope....2
0.3 Document Content....2
1 Overview....4
2 Objectives of Task 3.1....4
3 Description of Tasks....5
3.1 Low-level Image and Video Processing (First Year)....5
3.1.1 Contour Detection....5
3.1.2 Colour and Motion Segmentation....6
3.1.2.1 Background/Foreground Segmentation....6
3.1.2.2 Adaptive Skin Detection....6
3.2 Selection and Construction of Features (Second Year)....7
3.2.1 Features Descriptors....8
3.2.2 Point Sampling Methods....9
3.2.3 Selection of Feature Attributes....9
3.2.4 Bag of Visual Words....10
3.2.5 Feature Clustering ....10
3.2.6 Spatial Pyramidal Bag of Visual Words....11
3.2.7 Kernel Functions....12
3.2.8 Feature Extraction in RWTH....12

4 Description of the Software Developed in Task 3.1....13
4.1 Classes Developed in the First Year....13
4.2 Classes Developed in the Second Year....14
5 Description of the Content of the Deliverable....15
6 Work Progress and Conclusions....16
6.1 Progress towards Objectives....16
6.2 Future Work....17
7 Conclusions....17
8 References....18

# 1 Overview

The objective of WP3 is the development of video analysis methods to produce a stream of features which will be used by the sign language recognition and translation methods developed in WP4 and WP5. The present report accompanies Deliverable D.3.3, which is an extension of Deliverable D.3.1. This deliverable focuses on the extraction of spatial features from videos and describes the tasks involved in this process.

Specifically, it provides information on how video sequences are managed, low-level image operations are performed, point-based features extracted and more distinctive spatial feature descriptors constructed to describe image regions. The previous description of the methods to detect and track hand regions has been moved to the report corresponding to Deliverable D.3.4, which deals with the extended prototype for multimodal visual analysis.

The timeline of the work developments so far in Task.3.1 starts at capturing the videos, preprocessing the images, detecting and tracking candidate regions likely corresponding to hands based on motion and skin colour and extracting distinctive descriptions which will be used afterwards in the recognition and translation of the sign language.

In the following sections each of the techniques employed to accomplish these subtasks will be succinctly described in relation to the objectives of Task.3.1. The description will account for the interest in its use and the basics of its functionality, demonstrating the work accomplished and that which is expected in the short future.

# 2 Objectives of Task 3.1

This deliverable is part of Task.3.1: extraction of spatial features for the description of image content, and is an extension of Deliverable D.3.1 corresponding to Month 12.

The final objective is that the spatial feature extraction from raw video data will be robust and self-adapting to changing ambient conditions, and independent with respect to signer, background clutter and illumination. Besides, it is a design requirement that the signers will not require any artificial markers, sensors or gloves, to facilitate a more natural signing scenario.

The goals of the project, however, have been tailored into a more feasible ones. Independence of signer has been reduced in some algorithms to obtain a set of initial settings that adapt better to the corresponding signer. Ambient conditions have also been more controlled. In television image corpus PHOENIX, the illumination, point of view, body pose and clothing of the signer is totally controlled. Other factors adding variability to the data have also been restricted, such as cluttered backgrounds.

In order to carry out the objectives of the Task, the work has been divided into two main tasks described:

- ***Low-level image and video processing*** includes operations such as filtering, keypoint extraction, and tracking. This relies mostly on established techniques, but the work carried out in this Task has been necessary to identify, adapt, and integrate those best suited to the objectives. This is illustrated in more detail in Section 3.

- ***Selection and construction of distinctive features*** for *recognition* (WP4) and *translation* (WP5). Human sign language interpreters rely on parameters that can be explicitly described in everyday terms, such as hand trajectories, finger configurations and facial expressions. However, for machine recognition and translation, there may be other powerful cues that are more easily or robustly

extracted than these explicit parameters, such as spatial image statistics or combinations of other cues. This task focuses on their identification and construction to make them available for WP4 and WP5. More details on feature description in Section 4.

A significant part of the effort devoted to this Deliverable corresponds to the testing of code, different configurations of features and video processing. Also an important quantity of effort has been employed in integrating WP3 with WP6.

# 3 Description of Tasks

This section describes the tasks that have been endeavoured so far (M24) to accomplished the objectives described previously. For sake of unity and clarity, we have included in this description a brief reminding of part of the work that was carried out by the end of M12. This work basically corresponds to the first Subsection 3.1, which copes with the low-level processing of video and images, and some of the first steps into the computation to image features described in Subsection 3.2. The rest of this latter Subsection corresponds to the work done during the duration of the second year of the project.

## *3.1 Low-level Image and Video Processing (First Year)*

The low-level image and video processing operations focus on the preprocessing of frames in order to remove noise and adapt the data so that it can be used by other modules of Task 3.1. this work was carried out during the first year of the project and has been included here for clarity and unity, since it has been extensively used during the second year for the extraction of features.

The operations performed included filtering, colour conversion, removal of regions according to different criteria such as size, position, shape or colour, post-processing of segmentation results by removal of holes, morphological operations, and analysis of the blobs obtained by binarization or segmentation (thresholding, histogram back-projection, or background segmentation) of images. These are basic and standard image processing operations which must be applied to the video data in order to use it in later stages of the visual analysis. Most of them correspond to functions in *OpenCV* and *IVT*, which are completed by providing the necessary auxiliary variables to improve their usage and integration.

### 3.1.1 Contour Detection

This section is about the computation of image contours. The information provided by the contours will be employed in the generation of hand movement descriptions, since it contains structural information of hand configuration that can be helpful in recognition, despite being noisy and variable.

Fig. 1 Frames showing the contour detection on the video sequence

The hand contours are obtained by applying Canny filtering on an image with noise removed. Later, the set of contours obtained are also selected according to their length. So far, a set of basic features corresponding

to each contour can be computed, such as area, length, and bounding box. Additionally, as is mentioned later, the algorithms corresponding to local appearance feature computations, namely *SURF*, *SIFT*, and *HoG*, have been adapted to compute features on the points corresponding to such contours. This way, the structural and appearance description of the hands will be reinforced.

## 3.1.2 Colour and Motion Segmentation

In sign language understanding, the most important part of the meaning is conveyed by hands. Therefore, strong effort must be put in segmenting not only the position of the hands but also their configuration, in order to obtain higher scores in the recognition and translation of the sign language. Nevertheless, the main goal is to isolate the area containing the hands, or at least a set of likely candidates, to be tracked and described for further analysis.

The two main cues which are considered in the first stages to obtain candidate regions are motion and skin colour. For the first cue, a background segmentation algorithm is used to separate pixels corresponding to moving parts of the body from static background pixels. For the second, an algorithm which segments pixels according to their colour is used to detect skin regions.

### *3.1.2.1 Background/Foreground Segmentation*

The method envisaged in Task 3.1 for the isolation of objects with distinctive features of hands and face is performed by a background segmentation algorithm. The reasons for this are first that the camera mostly captures the same static background. The only part of interest to us is the moving part of the signers, that is, their faces and hands.

In this Task, the codebook background segmentation algorithm is employed [11], in which a background model is set up defining learning boxes by means of an interval over each colour axis. These intervals will expand if new background samples fall within the learning limits of the box. If new background samples fall outside of the box, then a new box will be started. In the background difference step, that is, when the image pixels are classified as belonging or not to the background model by using the threshold values defined by these boxes, a pixel can be close to a box boundary and counted as if it were inside the box. The objects left after subtraction are presumably new foreground objects, faces and hands in this case.

Fig.2. Background/foreground segmentation: Areas in the image with motion appear segmented. These regions may belong both to the signer's limbs and to body parts.

### *3.1.2.2 Adaptive Skin Detection*

Available skin detection algorithms are either based on static features such as skin colour or require a significant amount of computation. Such skin detection algorithms are not robust when dealing with real-world conditions, like background noise, change of intensity and lighting effects. This situation can be

improved by using dynamic features of the skin colour in a sequence of images.

The skin detection algorithm here is based on adaptive hue thresholding and adaption using a motion detection technique. The skin classifier is based on the hue histogram of skin pixels, and adapts to the colour of the skin of the persons in the video sequence. This algorithm has demonstrated improvement in comparison to the static skin detection methods. The first part of the algorithm uses a global skin detector that can detect the actual skin pixels with reasonable rate requiring little computational time per pixel. In order to improve the results in a real-world application, there is a second stage with an adaptive threshold that uses motion information to improve the detection of skin colours through the time. The colour histograms obtained applying these two thresholds on the images are combined according to a mixing scheme. The resulting colour information is used to segment the skin regions out by means of histogram back-projection.

Fig.3. Skin segmentation: Areas in the image correspond to moving parts of the body which colour is similar to that of skin, such face, arms, and hands.

## 3.2 Selection and Construction of Features (Second Year)

The second main goal of Task 3.1 is the selection and construction of a set of distinctive features for the tasks of *recognition* (WP4) and *translation* (WP5) of the sign language. For automatic recognition and translation, cues other than hand trajectories and finger configuration can be more easily and robustly extracted, such as spatial image statistics or combinations of other cues. The greatest part of the work described in this section has been accomplished during the second year of the project. Only the Subsections 3.2.1 and 3.2.3 were initiated during the first year. In this case, the corresponding tasks were completed during the second year.

More specifically, the two subtasks which comprise the visual analysis of hands are detection and tracking. Hand detection is difficult due to the great variability in the shape of hands. The fundamental difficulty in extending standard approaches for face detection and tracking is that they do not directly apply to the highly deformable nature of hands, which makes the alignment extremely difficult. The positive training samples trimmed to select a hand region, usually rectangular boxes, include an important amount of background clutter pixels which vary depending on hand posture and cause traditional approaches to fail. Discriminative hand detection is usually carried out by learning a set of positive and negative samples of hands in different configuration and classifying candidate windows from the image by means of an algorithm.

In the current approach, prior to the extraction of a set of features for the recognition and translation of sign language, it is necessary to detect and track the group of image regions which contain the signer's hands, based on adaptive skin colour detector and a tracking algorithm. Once the hand regions are detected and segmented out, the tracking procedure on these regions is carried out with the *Cam-Shift* and *Mean-Shift* algorithms.

The work so far has been focused on the creation of structures and code which permit the integration of some of the most salient features available in several libraries (OpenCV and IVT). These features consist of **SURF**

(*Speeded-Up Robust Features*), **SIFT** (*Scale Invariant Feature Transform*) and **HoG** (*Gradient of Oriented Histograms*). In addition, several operations for the detection of interest points have also been incorporated, including the Harris detector and **GFT** (*Good Features to Track*). Finally, a representation based on these features named **BoVW** (*Bag of Visual Words*) has been implemented as a means of describing the visual content of regions enclosing hands.

A series of new developments in the project are described in this section, including improvements with the addition of several sampling schemes where the set of features are computed and the clustering algorithms now used in the BoVW representation. Finally, a new representation called Pyramid Bag of Visual Words has been implemented to overcome the spatial problems intrinsic to general Bag of Visual Words.

### 3.2.1 Features Descriptors

For any object in an image, interest points can be extracted to provide a feature description of the object. This description, in case of a learning process, can then be used to identify the object when attempting to locate it in a test image containing many other objects. It is important that the set of features extracted from the training image is robust to changes in image scale, noise, illumination and local geometric distortion to perform reliable recognition.

In this Task, the SIFT[8], SURF[9], and HoG[10] feature descriptors are briefly considered. A further consideration of such features is illustrated in Annex 10.3. Their computation has two steps: first, the detection of interest points to compute feature descriptions of the appearance, and second, the generation of the descriptors themselves. In the available libraries which implement such features, both elements are tightly entangled to one another. Nevertheless, in our implementation these steps have been separated *deliberately* so the set of points to compute the features can be selected independently using different sampling methods.

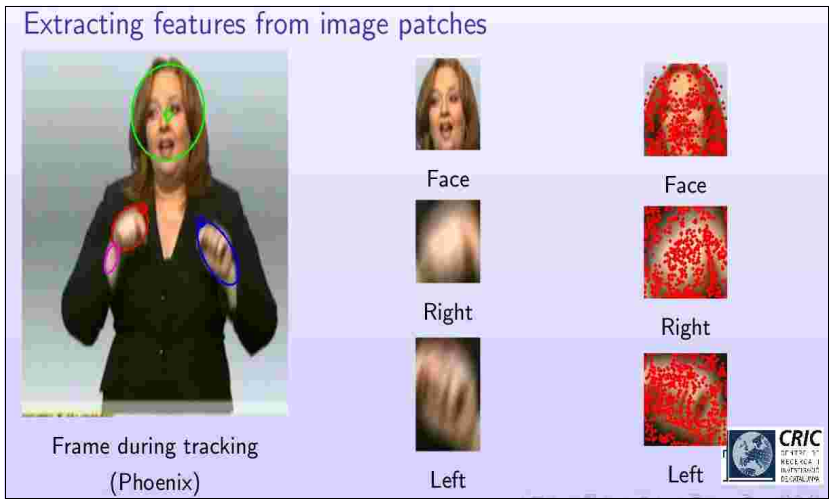


Fig.4. Interest point detection and feature extraction on video sequences

Some clustering algorithms required during the process of spatial feature construction have also been taken into account. Namely, K-Means, PCA, and the Expectation-Minimization algorithms, which can be used for clustering features obtained directly from video frames, or as part of the computations for BoVW and spatial pyramids of bags of features. In Fig 4, the extraction of features from the regions obtained in tracking is depicted.

### 3.2.2 Point Sampling Methods

Different point sampling methods have been included to compensate for the fact that only small patches of the video frames (hands and face) are dealt with in this application and, as a consequence, points usually are too sparse to be useful in other algorithms. The solution to this problem has been to implement different sampling methods which can give as many points as necessary to represent any the image patch.

The use of dense descriptors can help improve the results in the context of categories (object localization), despite the fact that interest points do not necessarily capture all features in an image. Later, the *Pyramidal Bag of Visual Words* will be described, which consists of a dense descriptor that extracts features from smaller sections of the image at multiple scales.

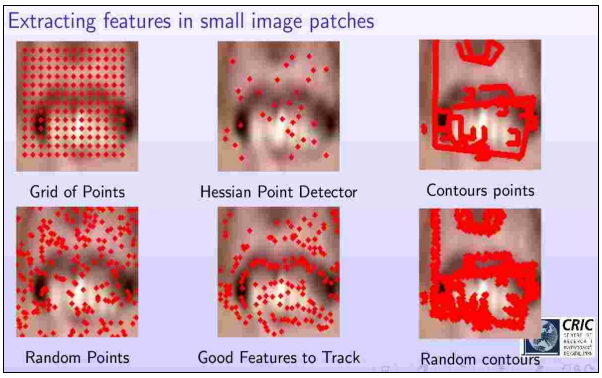


Fig 5. Different types of point sampling methods

The sampling strategies implemented, as depicted in Fig 5, are the following:

- ***Grid of points***: points create a dense grid covering the image patch.
- ***Hessian point detector***: points obtained from a Hessian interest point detector (from SIFT/SURF feature extractor).
- ***Random points***: points generated randomly.
- ***Good Features to Track***: points obtained from the Harris interest point detector in OpenCV's *Good Features to Track* function.
- ***Contour points***: points roughly corresponding to contours.
- ***Random contour points***: points randomly diffused from contours.

### 3.2.3 Selection of Feature Attributes

The capacity to select the attributes used in the computation of features at a selected point has been implemented. This consists in modifying part of existing feature computation code (SIFT/SURF) in order to split the process in two parts, one dedicated to the computation of interest points and the other one to the capture of the features themselves.

In the original routines, in which interest points and features are computed together, the first step is finding the interest points and also the best scale and direction to compute the features. If the points are provided by the user, OpenCV's implementation also allows the computation of features but there is no *optimal* way to

adjust the attributes (eg. scale, direction, or Laplacian) used for this computation. Therefore, we have modified these routines to provide them with the optimal attributes given a set of image points, in the same terms these attributes are obtained in the original routines, since feature computation depends on them.

### 3.2.4 Bag of Visual Words

In this section, a brief exposition of the method to describe images based on *Bag of Visual Words* (BoVW) [4, 7] is shown. This construction provides a representation of hands to perform the tasks of *recognition* and *translation,* based on a set of features as those considered in Section 4.1.

Based on local interest points extracted or a sampled set of image points, an image can be described as a bag of visual words and this representation has been frequently used in the classification of visual data. The sets vary in cardinality and lack meaningful ordering. This creates difficulties for learning methods (classifiers) that require feature vectors of fixed dimension as input. This problem is addressed by using the vector quantization technique which clusters the keypoint descriptors in their feature space into a larger number of clusters using the K-*Means* clustering algorithm and encodes each keypoint by the index of the cluster to which it belongs.

By treating each cluster as a visual word that represents the specific local pattern shared by the keypoints in that cluster, we obtain a visual word vocabulary, which describes all kinds of local image patterns. With its keypoints mapped into visual words, an image (or part of it) can be represented as a *Bag of Visual Words* (BoVW), or specifically, as a vector containing the (weighted) count of each visual word in that image, which is used as feature vector in the classification task. Fig 6 depicts the steps involved in the generation and usage of BoVW.

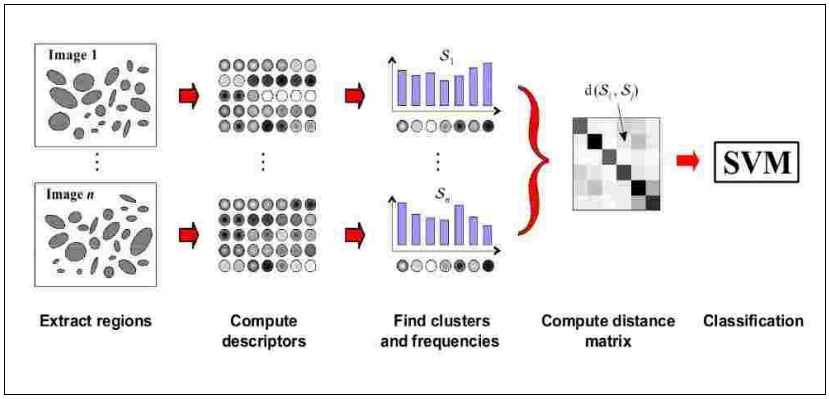


Fig 6. Steps in the construction of Bag of Visual Words

### 3.2.5 Feature Clustering

In order to compute a set of visual words, the set of point features obtained from an image patch as a set of defined representatives, it is necessary to represent them from a vocabulary of visual words. A bag of visual words is a distribution of such elements as a frequency histogram which assigns each feature to a certain representative cluster centre. This procedure provides a fixed dimensional size for the description of a given image patch.

These representatives are obtained by using a clustering algorithm. Several have been tried with different results, advantages and disadvantages.

- ***K-Means***: This is the simplest and most straightforward algorithm. It presents initialization issues as well as being affected by outliers and data distribution. These problems are overcome in our case by

running the algorithm several times and normalizing the data with respect to the covariance matrix. Current implementation can be scaled to huge amount of data and makes this algorithm a feasible candidate to obtain representative cluster centres.

- ***Principal Component Analysis (PCA)***: This algorithm can be used in our context to reduce the dimension of data and fix it to a certain number of dimensions.

- ***Expectation-Maximization (EM)***: This algorithm can be combined with the previous ones to obtain an initialization for cluster centres and minimize data-classes likelihood, which is statistically sound. However, available implementation scales badly with respect to the enormous amount of data and number of clusters employed in our application.

Apart from clustering the feature data, it is also necessary to normalize it so the clusters distribute correctly with respect to the input data distribution to allow the algorithms to work with real data. This process consists in filtering out possible damaged data as well as outliers, and correcting the data so it is better distributed within the feature space. This is accomplished by statistic normalization of data.

### 3.2.6 Spatial Pyramidal Bag of Visual Words

One key advantage of BoVW, apart from the fact that their length (dimension) is fixed irrespectively of the number of detections, is that they are largely unaffected by position and orientation of the object in the image. In many applications, this has proven very successful in the classification of objects. However, in localizing applications it is important to detail an explicit configuration of the visual word position, that is adding spatial information to the bag of features. Not just a rough representation of the distribution of the features corresponding to a certain object, but more explicit information related with how these features distribute spatially.

As can be seen in Fig 7, spatial pyramid representations consist in a *locally orderless* representation at several levels of spatial resolution. It basically consists in concatenating BoVW obtained from the image patch at different scales (levels) and sub-image windows. The resulting representation encompasses all these partials BoVW's into one single vector. Therefore, the spatial information is codified partially into different sections of the vector.

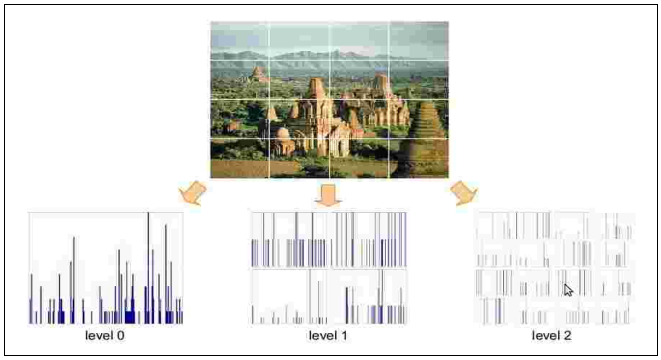


Fig 7. Generation of Spatial Pyramidal Bag of Visual Words from local BoVW

Including spatial information is expected to improve classification and localization results with respect to using bag of features at a single scale. In order to model the spatial information more flexibly, other approaches might also be considered in future developments, such as coarse global position information, flexible object-based models (likelihood object masks) or spatial weighting BoVW.

### 3.2.7 Kernel Functions

The last step in the construction of representations based on bag of features is the computation of a distance matrix. This matrix is necessary for classification as the kernel function in non-linear SVM approaches. Linear SVM finds a linear function to separate positive and negative examples. Nevertheless, classification might not be good enough in case databases are not linearly separable. As an alternative, the *kernel trick* can be employed to operate in a higher dimension feature space.

The original input space can always be mapped to some higher-dimensional feature space where the training set is linearly separable. This is accomplished by means of a kernel function, a distance function defined between features and used to compute a metric matrix. Each element of the matrix is the distance between the corresponding cluster representatives in the vocabulary of visual words.

The distance between BoVW is computed using different functions, taking the bag of features as both a vector and a histogram. Therefore, several options have been implemented, namely, the *Euclidean* distance between vectors, a histogram *correlation* measure, histogram *Chi-square* distance, *histogram intersection, Bhattacharyya, Karnahue-Loewen and Earth Mover's* distances.

### 3.2.8 Feature Extraction in RWTH

In the second period, the sign language recognition and tracking framework by RWTH was extended to allow for direct access to OpenCV data structures and feature files provided by the project partners. RWTH continued work on densely sampled local features for tracking and recognition. SIFT and SURF features have been extracted from the RWTH-BOSTON-104, SIGNUM, RWTH-PHOENIX-v2.0, and Corpus-NGT databases.

RWTH has evaluated densely sampled SIFT and SURF features in dynamic programming tracking (DPT) for the RWTH-BOSTON-104 and Corpus-NGT databases using several different scoring functions. For results and further discussion see Task 3.2. The evaluation of densely sampled SIFT and SURF features for RWTH PHOENIX-v2.0 and SIGNUM is currently on-going in the context of WP4 and is hence not included in this report.

RWTH also extracted geometric features from dominant hand of the signers [14] on the SIGNUM database. In total 34 features were extracted. These features can be classified into four different groups. The first group describes basic properties of the hand including the surface of the hand after segmentation, the length of the border, the coordinate of the centre of gravity and the compactness. The second group is moments-based features. In the third group, the first seven moment invariants as described in [13] are computed and the last group contains combined geometric features such as the distance between the centre-of-gravity of the hand and certain positions in the image. Depending on the group they belong to, the geometric features are invariant against transformations like translation, rotation, flipping and scaling. An example of extracted geometric features is shown in Figure 8.

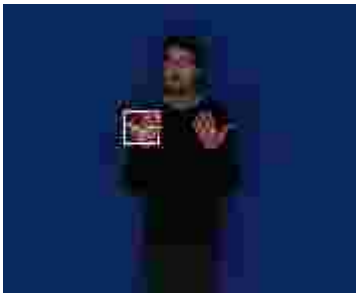

Fig 8. Geometric features extracted from the dominant hand
of one speaker from the SIGNUM database

# 4 Description of the Software Developed in Task 3.1

This section describes the set of general operations that were implemented in the modules corresponding to Task 3.1 in order to process the images and video sequences used throughout the project. All operations related to Task 3.1 were included in a number of classes in C++.

These classes make use of some already existing types and functions in the open-source computer vision libraries OpenCV and IVT (Integrating Vision Toolkit). OpenCV is a computer vision library originally developed by Intel. It is free for use under the open source BSD license. The library is cross-platform and it focuses mainly on real-time image processing. The IVT is an easy-to-use, platform-independent open source C++ computer vision library with an object-oriented architecture developed by the Institute for Anthropometrics of the *Karlsruhe Institute of Technology*. It offers a clean camera interface and a general camera model, as well as many fast implementations of image processing routines and mathematical data structures and functions. The IVT offers its own multi-platform GUI tool kit.

By constructing our own set of classes, we obtain a tailored code that fits our needs such that, on one hand, it makes use of some operations already available in the previous libraries, while on the other some of the operations have been modified to adequate them to our requirements. We want to point out here that this code does not consists just in wrapping of already existing libraries. There has also been modification of code as well as development of new functionalities from scratch. Besides, some utilities were developed before they were eventually provided by OpenCV in their latest versions, like the inclusion of SIFT features.

The advantages of this implementation are the following:

- Easing the combination of different modules and libraries by homogenizing data and functions shared by different classes.
- Facilitating the use of already existing algorithms from other libraries, since these classes enclose all the required variables and auxiliary data necessary for them to run, completing some of the implementations in OpenCV which did not make use of an object-oriented architecture in all functions and libraries in their first versions.
- Helping integrate them with functionalities in the IVT library.

In summary, this implementation increases the reliability of the code as it becomes more and more complex and helps its integration as the number of modules increases.

Despite is difficult to stablish a strict distinction between these classes that have been developed during the first year to those carried out on the second one due to the fact that these classes have been worked on along all the time, we will separate into those that were initialized on the first year and those, on the second year.

## 4.1 Classes Developed in the First Year

The main functionalities of these classes are:

- **CProcessVideoBase**: This class is the basis to perform any process on a video sequence. Besides the initialization parameters, this class uses elements of the class **CProcessDataBase**, which houses all the variables required to process a frame: a pointer to a model and a pointer to the function which actually processes each frame. A model consists in a pointer to an existing class in OpenCV, ITV, or any other library, which is needed to keep the information extracted from the image. Currently, existing models include background models, contour models, and feature extraction models. Instances of this class work with two video streams, that of the input from a file or a camera, and that of the output into a file, besides providing functions for the visualization of the video streams. It is meant to be the basis class for developing further implementations integrating more processing modules.

- **CExtractFeatureBase**: This is the basis class for the extraction of features on video frames, specifically the SURF features. Other derivative classes are **CExtractFeatureSIFT** and **CExtractFeatureHOG**, which implement the extraction of SIFT and HoG features. The main member functions of these classes allow the pre-processing of the image before obtaining its features, applying some basic algorithms to detect interest points, such as corners and star points. The detection and extraction is divided in two steps: the computation of the interest points using the algorithm corresponding to each kind of feature, and the construction of the feature description. Besides these operations, there are other functions which allow the clustering of features using a k-Means clustering algorithm, and matching them to other feature sets by means of a k-dimensional tree search.

Other auxiliary classes implemented were:

- **CProcessParam**: This class is meant to carry parameters that are necessary for the initialization of a class as well as the use of functions. This way, all classes share the same parameter structure and there is no local constant defined and lost within the code, which could later generate problems during the integration of different modules into one single piece of software.

- **OpenVideo**: This class manages the different streams from video files and from camera, opening, closing, reading, writing files, and visualizing frames. It also helps use different formats belonging to the libraries OpenCV and IVT and communicating their image buffers, which are expressed in different types of variables.

- **CProcessDataBase**: This class encompasses required variables needed to process frames, such as function parameters, image buffer pointers, auxiliary images in different types and depths, model pointers and also pointers to source and destination image buffers. The idea is to put together in one single class all the necessary auxiliary structures for the processing of a video frame, to be the basis for any further derivation of similar classes which might be required in the development of this software.

## *4.2 Classes Developed in the Second Year*

The following classed where initiated during the first year, but their development was continued also along the second year.

- **CBagVisualWords**: This class implements the bag of visual words for the description of images. From a list of sets of features (the corpus), the algorithm extracts a vocabulary, a set of words (cluster representatives of all the features in the corpus) which will be later used to compute weights for each of the features representing a certain part of an image, that is, the bag of visual words. Other member functions in the class permit the use of different types of frequencies to count the presence of visual words to be used as vector weights, as well as measuring the distance between BoVWs.

- **CClusterBase**: This is the basis class for implementing clustering algorithms which can be used with the feature descriptors obtained by the member function **CExtractFeatureBase**, or called by functions in other classes of modules. Currently, *k-Means* and *Expectation-Minimization* (EM) algorithms are included in their member functions. The rest of member functions are for the computation of cluster centres and labelling of samples according to the obtained set of clusters.

- **CTrackerBase**: This class was created as a base class for detecting and tracking one hand region at a

time. As the original idea was that of creating an experimenting tool to track hands and obtain useful descriptions of the regions tracked used a posteriori in other modules, this module was not meant to be a final version of a hand tracker, but a working tool with which easily experimenting new approaches and, therefore, developing some improvements out of it. Due to the inclusion in Y2 of PHOENIX database, some of the functions were modified to cope with the new kind characteristics of these images.

Apart form the previous set of classes, in the second year the following new modules were also developed:

- **CMultiTrackerBase**: This is an extension of the previously described class **CTrackerBase** which goal is coping with the tracking of several hand regions at a time and was developed to extract features based on the hand position from videos and frame sequences. An important part of this module is the heuristics to distinguish among several candidate regions and decide which one corresponds to a hand.

- **CProcessCorpus**: Different corpora have been employed in this second year (RWTH's PHOENIX and Nijmegen's CNGT) to extract features. Each corpus has its particularities and needs to be processed in a different way, making it necessary to have special tools to do so.

Other auxiliary classes were:

- **CContourBase** and **CBlobResult** are two auxiliary classes. The first one is used to compute the extract contours from images and sets of features related to them. The second class is used to performed blob analyses on binary images.

- **CSegementFunctions**: A class with functions used to obtain color segmentation of images from videos and sequences. These functions have been employed in those classes that were involved in hand tracking.

- **CProcessFunctions**: This class en compasses a list of functions that share the same structure and were originally developed to process videos in the same way. Each new process was included by adding a new function, while the rest of the structure was kept identical.

- **ReadXmlUtils**: This module is enclosures the set of functions that have been employed to read xml files with information related with corpora and ground truth generated by other members and also by ourselves.

# 5 Description of the Content of the Deliverable

This section describes the files generated in the present deliverable, which corresponds to spatial features extracted from the image frames belonging to the two corpora available in this project, NCGT and PHOENIX. These features are a description of the regions where signer's hands were located by means the tracking system provided by RWTH.

Presently, two corpora of videos have been used, namely corpus NGT from Nijmegen University and corpus PHOENIX from RWTH from Aachen University. The first corpus is composed of videos by numerous signers in a fairly uncontrolled shooting scenario in terms of signer clothing, illumination and situation of the signer in the scene. Despite most of the video scenes apparently being fairly similar, this corpus is still challenging due to such variability.

On the other hand, PHOENIX corpus is a more restricted set of videos where the shooting conditions, light, clothing and posture, are very controlled and stable. The set of features will focus on those obtained from this corpus, although both corpora can be used. Specifically, the release of PHOENIX corpus used consists of 1405 videos with a span of about 212K frames.

Applying feature extraction routines to all frames in the corpus generates an important amount of data that must be managed properly. The number of features corresponding to each image patch is not fixed a priori and, in general, will depend on both the image content itself and the sampling procedure applied. This makes the manipulation difficult in practical terms due to the huge amount of data. However, this problem becomes more manageable when using BoVW representation, which transforms point-based features into a patch-based representation and provides a fixed dimension and a reduction of the total amount of data.

Files generated in such representations consist of point feature histograms for each image patch. A directory was created for each corpus, and inside this directory there are as many subdirectories as sessions or videos exist. For each video file, a single file with all features is generated, which can be used afterwards in WP4 and WP5, and also in any tracking process which is based on learning and recognizing image patches to locate hands in a video.

Data is grouped into folders according to the sampling procedure used and several configuration were provided, depending on the variation in the number of features that were clustered in the computation of the vocabulary employed later to generate the representation into bag of visual words and also the dimension of the final features, which depends on the size of such vocabulary.

# 6 Work Progress and Conclusions

## *6.1 Progress towards Objectives*

With respect to the previous Deliverable concerning the work completed in the first year, there have been substantial advances that will be taken into account in this section. Low-level image and video processing objectives were accomplished so far. Also the generation of spatial features describing hand configuration was continued and completely carried out.

By the use of OpenCV and IVT libraries and the set of classes considered above, it is possible to perform operations such as filtering, interest point extraction, skin colour region detection and segment tracking on video images. As mentioned in the first year Deliverable, despite some of these operations rely on established techniques, this work was dedicated to identifying, adapting, and integrating the best suited techniques to fulfil the needs of the project. In addition, an important amount of time and effort were invested in the integration of code to make its use and integration in WP3 easier, more robust and less prone to errors. Moreover, a significant n of the time was employed in the generation of the sets of features.

The first attempt to determine a feasible and useful construction of the hand description has been accomplished by means of a bag of visual words, which has been established as an effective way of describing general objects. At present, this module is already implemented and tested. The extension to spatial pyramidal bag of visual words has also been implemented and in process of being tested with real data from corpus PHOENIX.

Regarding the selection, construction, and extraction of distinctive spatial features, at present it is possible to extract sets of interest point and associated features from videos and sequences of images. It is also possible to construct the aforementioned features on sets of predetermined points (grids, contours, regions, or lists of points). This functionality required a detailed modification of already existing feature extraction algorithms and was foreseen to be more effective in the description of general hand movements and configurations.

During the second period RWTH focused on two major activities in WP3: the extraction of local features such as SIFT and SURF, as well as geometric features for hand/head tracking and sign language recognition.

## 6.2 Future Work

Further work must be carried out in testing the existing modules, extending spatial features to include the temporal domain, and in the evaluation of these descriptors in hand detection and tracking, as well as in the use of the sets of features generated so far in extending the study on clustering and classification with respect to hand shape configuration.

Existing spatial features will be extended into spatio-temporal features, such as the Harris-type temporal generalizations in [5, 6]. This kind of feature integrates temporal and spatial information so that detection and description are performed at the same time in both spatial and temporal dimensions.

The integration of all modules and code generated in Task 3.1 is almost complete at present, allowing the combination of detection and tracking modules with feature extraction algorithms. Further testing must be performed on the code that generates the spatial pyramids of BoVW and on the integration with the modules employed to cluster and classify hands shapes.

More experiments will be run in order to construct distinctive descriptions of hands with already implemented set of features and different point sampling schemes will be carried out in order to determine the best approach.

Finally, the resulting feature extraction modules will be integrated into the baseline prototype in D.3.5 in order to provide these descriptors for other hand analysis modules to perform a discriminative analysis of hand movements.

# 7 Conclusions

The current state of the tasks developed in Task 3.1 is of wide implementation and testing of all the objectives with respect to low-level image and video processing as well as the selection and construction of distinctive features for analysis of hand movements. This document consists of a self-contained description of the tasks developed so far for sake of unity and clarity. Nevertheless, a clear delineation has been established between the work completed in the first year and that in the second. Given the state of the developments corresponding to this task, there is no significant deviation from the original work programme.

# 8 References

[1] *An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods*. Farhad Dadgostar, Abdolhossein Sarrafzadeh. Pattern Recognition Letters 27 (12), 1342-1352, 2006.

[2] *Discriminative human action recognition in the learned hierarchical manifold space*. Lei Han, Xianxiao Wu, Wei Liang, Guangming Hou, Yunde Jia. Image and Vision Computing 28, 836-849, 2010.

[3] *Vision-based hand pose estimation: A review.* Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, Xander Twombly. Computer Vision and Image Understanding 108, 52-73, 2007.

[4] *Evaluating Bag-of-Visual-Words Representations in Scene Classification*, Jun Yang, yu-Gang Jiang, Alexander G. Hauptmann, Chong-Wah Ngo, International Multimedia Conference, Proceedings of the international workshop on Workshop on multimedia information retrieval, 197-206, Germany 2007.

[5] *Local Velocity-Adapted Motion Events for Spatio-Temporal Recognition*, I. Laptev, B. Caputo, C. Schuldt and T. Lindeberg; in *Computer Vision and Image Understanding*, 108:207-229, 2007.

[6] *Local descriptors for spatio-temporal recognition*, I. Laptev, T. Lindeberg. ECCV'04 Workshop on Spatial Coherence for Visual Motion Analysis, Springer Lecture Notes in Computer Science, Volume 3667. 91–103, 2004.

[7] *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*. S. Lazebnik, C. Schmid, J. Ponce. In Proc. Of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2169-2178, 2006.

[8] *Disctintive Image Features from Scale-Invariant Keypoints*, D.G. Lowe. Int. Journal of Computational Vision, 60 (1), 63-86, 2004.

[9] *SURF: Speeded Up Robust Features*, H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Computer Vision and Image Understanding, vol. 110, No. 3, 346-359, 2008.

[10] *Histograms of Oriented Gradients for Human Detection*, Navneet Dalal, Bill Triggs, Int. Conf. On Computer Vision and Pattern Recognition, vol. 2, 886-893, 2005.

[11] Real-time foreground-background segmentation using codebook model, K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, Real-Time Imaging 11 (2005): 167–256.

[12] *Learning OpenCV: Computer Vision with the OpenCV Library,* Gary Bradsky and Adrian Kaehler. O'Reily, September 2008.

[13] *Visual pattern recognition by moment invariants,* M. Hu. Information Theory, IRE Transactions on, Vol. 8, No. 2, pp. 179–187, 1962.

[14] *Robust Appearance-based Sign Language Recognition,* M. Zahedi. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, Sept. 2007.