Search-By-Example in Multilingual Sign Language Databases*

Ralph Elliott
School of Computing Sciences
University of East Anglia
Norwich, UK
R.Elliott@uea.ac.uk

Helen Cooper
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
Helen.Cooper@surrey.ac.uk

Eng-Jon Ong
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
E.Ong@surrey.ac.uk

John Glauert
School of Computing Sciences
University of East Anglia
Norwich, UK
J.Glauert@uea.ac.uk

Richard Bowden
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
R.Bowden@surrey.ac.uk

François Lefebvre-Albaret
WebSourd
Toulouse, France
francois.lefebvre-albaret@websourd.org

ABSTRACT

We describe a prototype Search-by-Example or look-up tool for signs, based on a newly developed 1000-concept sign lexicon for four national sign languages (GSL, DGS, LSF, BSL), which includes a spoken language gloss, a HamNoSys description, and a video for each sign. The look-up tool combines an interactive sign recognition system, supported by Kinect™ technology, with a real-time sign synthesis system, using a virtual human signer, to present results to the user. The user performs a sign to the system and is presented with animations of signs recognised as similar. The user also has the option to view any of these signs performed in the other three sign languages. We describe the supporting technology and architecture for this system, and present some preliminary evaluation results.

1. INTRODUCTION

Web 2.0 brings many new technologies to internet users but it still revolves around written language. Dicta-Sign[1] is a three-year research project funded by the EU FP7 Programme. It aims to provide Deaf users of the Internet with tools that enable them to use sign language for interaction via Web 2.0 tools. The project therefore develops technologies that enable sign language to be recognised using video input, or devices such as the Microsoft XBox 360 Kinect™. It also develops synthesis technologies for the presentation of sign language through signing avatars. The use of avatars, rather than video, is driven by the requirement to respect anonymity and to enable material from different authors to be edited together in Wiki-style. Users can control their view of an avatar, changing the speed of signing and even the character that performs the signs.

The Hamburg Notation System (HamNoSys) [14, 7] supports detailed description of signs at the phonetic level, covering hand location, shape, and orientation as well as movement direction, distance, and manner. In addition to notation for manual aspects of signs, use of other articulators can be specified, including head and body pose, and facial expressions including mouth, eye, and eyebrow gestures. The notation is not specific to any particular sign language. HamNoSys is used to drive animation of signing avatars through an XML format Sign Gesture Mark-up Language (SiGML) [4, 3]. In addition, SiGML representations are used in training the sign language recognition system.

The project partners work with a range of national sign languages: British Sign Language (BSL), Deutsche Gebärdensprache - German Sign Language (DGS), Greek Sign Language (GSL), and Langue des Signes Française - French Sign Language (LSF). For each language, a corpus of over 10 hours of conversational material has been recorded and is being annotated with sign language glosses that link to lexical databases developed for each language. The Dicta-Sign Basic Lexicon, providing a common core database with signs for over 1000 concepts, has been developed in parallel for all four languages used. Each concept is linked to a video and a HamNoSys transcription in each of the four languages. This enables recognition to be trained for multiple languages, and synthesis to generate corresponding signs in any of the languages.

The Search-By-Example tool presented is a proof-of-concept prototype that has been constructed to show how sign recognition and sign synthesis can be combined in a simple lookup tool. The user signs in front of the Kinect device and is presented with animations of one or more signs recognised as similar to the sign performed. A chosen sign can be pre-

*The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 231135.

[1] http://www.dictasign.eu/

SLTAT 2011, 23 October 2011, Dundee, UK

sented in all of the four supported sign languages so that a user can view and compare signs in different languages.

We present the sign recognition process and the sign synthesis process in some detail. We then describe the architecture and graphical interface of the prototype tool. Finally we discuss results and user evaluation of the system.

2. SIGN RECOGNITION

Previous sign recognition systems have tended towards data driven approaches [6, 21]. However, recent work has shown that using linguistically derived features can offer good performance. [13, 1] One of the more challenging aspects of sign recognition is the tracking of a user's hands. As such the Kinect™ device has offered the sign recognition community a short-cut to real-time performance by exploiting depth information to robustly provide skeleton data. In the relatively short time since its release several proof of concept demonstrations have emerged. Ershaed *et al.* have focussed on Arabic sign language and have created a system which recognises isolated signs. They present a system working for 4 signs and recognise some close up handshape information [5]. At ESIEA they have been using Fast Artificial Neural Networks to train a system which recognises two French signs [20]. This small vocabulary is a proof of concept but it is unlikely to be scalable to larger lexicons. It is for this reason that many sign recognition approaches use variants of Hidden Markov Models (HMMs) [16, 19]. One of the first videos to be uploaded to the web came from Zafrulla *et al.* and was an extension of their previous CopyCat game for deaf children [22]. The original system uses coloured gloves and accelerometers to track the hands, this was replaced by tracking from the Kinect™. They use solely the upper part of the torso and normalise the skeleton according to arm length. They have an internal dataset containing 6 signs; 2 subject signs, 2 prepositions and 2 object signs. The signs are used in 4 sentences (subject, preposition, object) and they have recorded 20 examples of each. They list under further work that signer independence would be desirable which suggests that their dataset is single signer but this is not made clear. By using a cross validated system, they train HMMs (Via the Georgia Tech Gesture Toolkit [9]) to recognise the signs. They perform three types of tests, those with full grammar constraints getting 100%, those where the number of signs is known getting 99.98% and those with no restrictions getting 98.8%.

While these proof of concept works have achieved good results on single signer datasets, there have not been any forays into signer independent recognition. In order to achieve the required generalisation, a set of robust, user-independent features are extracted and sign classifiers are trained across a group of signers. The skeleton of the user is tracked using the OpenNI/Primesense libraries [12, 15]. Following this, a range of features are extracted to describe the sign in terms similar to SiGML or HamNoSys notation. SiGML is an XML-based format which describes the various elements of a sign. HamNoSys is a linguistic notation, also for describing sign via its sub-units. [2] These sub-sign features are then combined into sign level classifiers using a Sequential Pattern Boosting method [11].

[2]Conversion between the two forms is possible for most signs. However while HamNoSys is usually presented via a special font for linguistic use, SiGML is more suited to automatic processing.

2.1 Features

Two types of features are extracted, those encoding the motion of the hands and those encoding the location of the sign being performed. These features are simple and capture general motion which generalises well; achieving excellent results when combined with a suitable learning framework as will be seen in section 5. The tracking returns x,y,z co-ordinates which will be specific not only to an individual but also to the way in which they perform a sign. However, linguistics describes motions in conceptual terms such as 'hands move left' or 'dominant hand moves up' [17, 18]. These generic labels can cover a wide range of signing styles whilst still containing discriminative motion information. Previous work has shown that these types of labels can be successfully applied as geometric features to pre-recorded data. [8, 10]. In this real-time work, linear motion directions are used; specifically, individual hand motions in the x plane (left and right), the y plane (up and down) and the z plane (towards and away from the signer). This is augmented by bi-manual classifiers for 'hands move together', 'hands move apart' and 'hands move in sync'. The approximate size of the head is used as a heuristic to discard ambient motion and the type of motion occurring is derived directly from deterministic rules on the x,y,z co-ordinates of the hand position. Also, locations should not be described as absolute values such as the x,y,z co-ordinates returned by the tracking, but instead related to the signer. A subset of these can be accurately positioned using the skeleton returned by the tracker. As such, the location features are calculated using the distance of the dominant hand from skeletal joints. The 9 joints currently considered are displayed in figure 1. While displayed in 2D, the regions surrounding the joints are actually 3D spheres. When the dominant hand (in this image shown by the smaller red dot) moves into the region around a joint then that feature will fire.

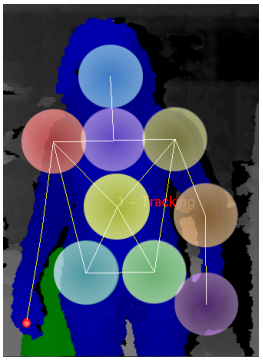
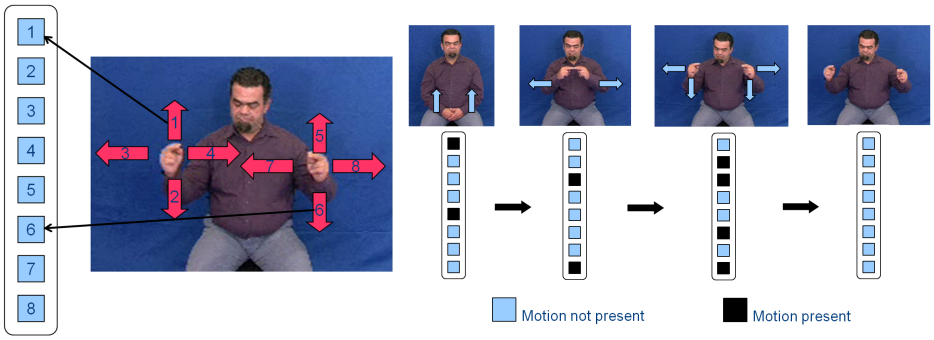


Figure 1: Body joints used to extract sign locations

2.2 Sign Level classification

The motion and location binary feature vectors are concatenated to create a single binary feature vector. This feature vector is then used as the input to a sign level classifier for recognition. By using a binary approach, better generalisation is obtained and far less training data is required than approaches which must generalise over both a continuous input space as well as the variability between signs (e.g. HMMs).



(a) Feature vector (b) Sequential Pattern

Figure 2: Pictorial description of Sequential Patterns.
(a) shows an example feature vector made up of 2D motions of the hands. In this case the first element shows 'right hand moves up', the second 'right hand moves down' etc. (b) shows a plausible pattern that might be found for the GSL sign 'bridge'. In this sign the hands move up to meet each other, they move apart and then curve down as if drawing a hump-back bridge.

Another problem with traditional Markov models is that they encode exact series of transitions over all features rather than relying only on discriminative features. This leads to significant reliance on user dependant feature combinations which, if not replicated in test data, will result in poor recognition performance. Sequential pattern boosting, on the other hand, compares the input data for relevant features and ignores the irrelevant features. A sequential pattern is a sequence of discriminative feature subsets that occur in positive examples of a sign and not negative examples (see Figure 2). Unfortunately, finding SP weak classifiers corresponding to optimal sequential patterns by brute force is not possible due to the immense size of the sequential pattern search space. To this end, the method of Sequential Pattern Boosting is employed. This method poses the learning of discriminative sequential patterns as a tree based search problem. The search is made efficient by employing a set of pruning criteria to find the sequential patterns that provide optimal discrimination between the positive and negative examples. The resulting tree-search method is integrated into a boosting framework; resulting in the SP-Boosting algorithm that combines a set of unique and optimal sequential patterns for a given classification problem.

3. SIGN SYNTHESIS

The final stages of JASigning [2], the realtime animation system used in Dicta-Sign, follow the conventional design of 3D virtual human animation systems based on posing a skeleton for the character in 3D and using facial morph targets to provide animation of facial expressions. The 3D skeleton design is bespoke, but is similar to those used in packages such as Maya and 3ds Max. Indeed, animations from JASigning can be exported to those packages. A textured mesh, conveying the skin and clothing of the character, is linked to the skeleton and moves with it naturally as the skeleton pose is changed. The facial morphs deform key mesh points on the face to create facial expressions.

The novel aspects of JASigning are at the higher level, converting a HamNoSys representation of a sign into a sequence of skeleton positions and morph values to produce a faithful animation of the sign. The HamNoSys is encoded in h-SiGML which is essentially a textual transformation of the sequence of HamNoSys phonetic symbols into XML elements. Some addition information may be added to vary the timing and scale of signing. The h-SiGML form is converted by JASigning into corresponding g-SiGML which retains the HamNoSys information but presents it in a form that captures the syntactic structure of the gesture description, corresponding to the phonetic structure of the gesture itself.

Animgen, the core component of JASigning, transforms a g-SiGML sign, or sequence of signs, into the sequence of skeleton poses and morph values used to render successive frames of an animation. Processing is many times quicker than realtime, allowing realtime animation of signing represented in SiGML using the animation data generated by Animgen. HamNoSys and SiGML abstract away from the physical dimensions of the signer but animation data for a particular avatar must be produced specifically for the character. For example, if data generated for one character is applied to another, fingers that were in contact originally may now be separated, or may collide, due to differences in the lengths of limbs. Animgen therefore uses the definition of the avatar skeleton, along with the location of a range of significant locations on the body, to generate avatar-specific animation data that performs signs in the appropriate fashion for the chosen character.

A HamNoSys description specifies the location, orientation, and shape of the hands at the start of a sign and then specifies zero or more movements that complete the sign. Animgen processes the g-SiGML representation of the sign to plan the series of postures that make up a sign and the movements needed between them. Symbolic HamNoSys values are converted to precise 3D geometric information for the specific avatar. Animation data is then produced by tracking the location of the hands for each frame, using inverse kinematics to pose the skeleton at each time step. The dynamics of movements are controlled by an envelope that represents the pattern of acceleration and deceleration that takes place. When signs are combined in sequences, it is necessary to add movement for the transition between the ending posture of one sign and the starting posture of the next.The nature of movement for inter-sign transitions is somewhat

more relaxed than the intentional intra-sign movements. By generating arbitrary inter-sign transitions, new sequences of signs can be generated automatically, something that is far less practical using video.

JASigning provides Java components that prepares SiGML data, converts it to g-SiGML form if necessary, and processes it via Animgen (which is a native C++ library) to generate animation data. A renderer then animates the animation data in realtime under the control of a range of methods and callback routines. The typical mode of operation is that one or more signs are converted to animation data sequences held in memory. The data sequence may be played in its entirety, or paused and repeated. If desired, the animation data can be saved in a Character Animation Stream (CAS) file, and XML format for animation frame data that can be replayed at a later data. The JASigning components may be used in Java applications or as web applets for embedding signing avatars in web pages. A number of simple applications are provided, processing SiGML data via URLs or acting as servers to which SiGML data can be directed.

4. SEARCH-BY-EXAMPLE SYSTEM

The architecture of the Search-By-Example system involves a server that performs sign recognition controlled by a Java client that uses JASigning components to display animations of signs. The client and server are connected by TCP/IP sockets and use a simple protocol involving exchange of data encoded in XML relating to the recognition process. The client provides a graphical user interface that allows the recognised data to be presented in a number of ways.

Before the user performs a sign to drive a search, the user selects the language that will be used. At present the choice is between GSL and DGS. A message indicating the choice of language is sent from the client to the server to initiate recognition. The user is then able to move into the Kinect^{TM}signing mode. In order to allow the user to transition easily between the keyboard/mouse interface and the Kinect signing interface, motion operated Kinect^{TM}buttons (K-Buttons) have been employed. These K-Buttons are placed outside the signing zone and are used to indicate when the user is ready to sign and when they have finished signing. Once the user has signed their query the result is returned to the client in the form of a ranked list of sign identifiers and confidence levels. The sign identifiers enable the appropriate concepts in the Dicta-Sign Basic Lexicon to be accessed. If the user has performed a known sign, there is a high likelihood that the correct sign will be identified. However, as the recognition process currently only focuses on hand location and movements, signs that differ solely by handshape have the potential to be confused. If the user performs an unknown sign, or a meaningless sign, the system will generally propose signs that share common features with the example.

The animation client provides space to present up to four signs at a time so the user will normally see the top four candidate signs matched by the recognition process. The signs are played continuously until a stop button is pressed. If more than four signs were returned as being significant, they are formed into batches of four, and the user can select any of these batches for display, and switch back and forth between them as desired. The animations use the HamNoSys recorded for the corresponding concept in the Basic Lexicon for the chosen language. A label under the animation panel gives the gloss name and a spoken English word for the concept. If the mouse is placed over the panel, a display of a definition of the concept, extracted from WordNet, is provided. Figure 3 shows the display of the last three results returned for a DGS search.

The user is also able to select any of the concepts individually and view its realisation in all four languages using a Translations button. In this case the HamNoSys data that drives the avatars is extracted from the Basic Lexicon for all four languages for the chosen concept. The label under the animation panel gives the language, concept number, and the spoken word for the concept in the corresponding spoken language. Figure 4 shows the display of the results for the concept Gun. As this concept has a natural iconic representation, it is not surprising that the signs are all similar.

The implementation of the client uses JASigning components. In this case, there will be up to four active instances of the avatar rendering software. The different HamNoSys sequences are converted to SiGML and processed through Animgen to produce up to four sequences of animation data. All the avatar panels are triggered to start playing together and will animate their assigned sequence. Since some signs will take longer to perform, a controlling thread collects signals that the signs have been completed, and triggers a repetition of all the signs when the last has completed. The result is to keep the performances synchronised, which simplifies comparison of signs. The controlling thread also handles broadcasting of Stop and Play signals to the avatar panels. All the usual signing avatar controls are available so that the size and viewing angle of each of the avatars can be varied independently. Although in this case the same virtual character is used for all avatars, it would be a simple matter to allow a choice of avatars, using different characters for each language if desired.

5. RESULTS AND EVALUATION

We present some results and user evaluation for the major components of the Search-By-Example system. For the recognition process it is possible to use controlled data sets to evaluate the accuracy of the system when searching for a known sign. We present a rigorous analysis of recognition accuracy across two languages over a subset of the lexicon. For the animation system there are several aspects to be considered ranging from the accuracy of the HamNoSys representation of the signs, the effectiveness of Animgen in producing an animation, assuming accurate HamNoSys, and compatibility between the signs used by the users and those used for training. In addition to these tests, a version of the prototype has received preliminary external evaluation by Deaf users as a tool, which leads to not only quantitative but also qualitative feedback.

5.1 Recognition Accuracy

While the recognition prototype is intended to work as a live system, quantitative results have been obtained by the standard method of splitting pre-recorded data into training and test sets. The split between test and training data can be done in several ways. This work uses two versions, the first to show results on signer dependent data, as is traditionally used, the second shows performance on un-seen signers, a signer independent test.

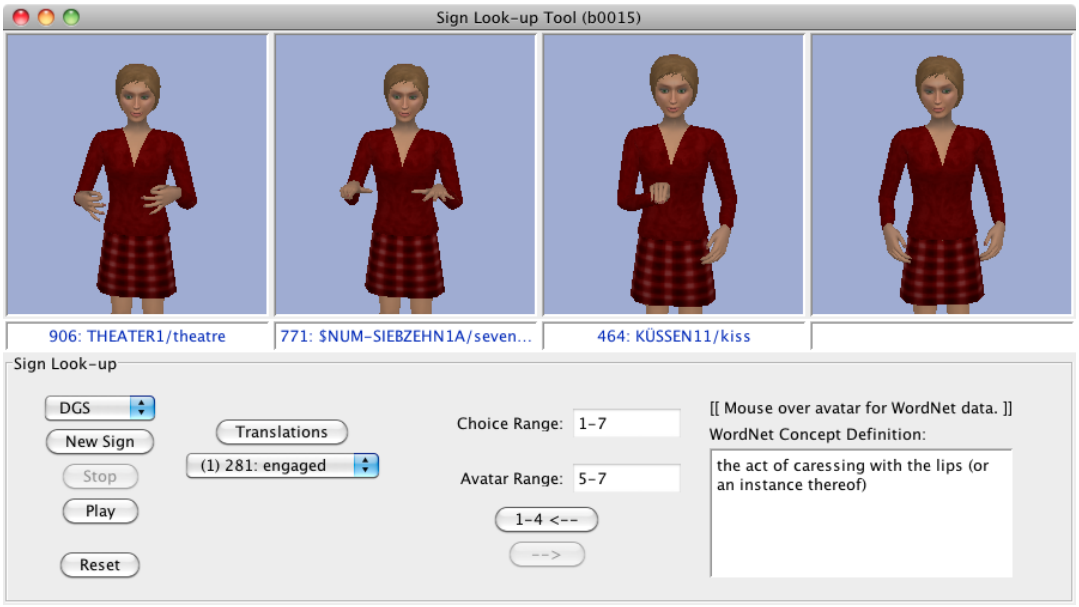


Figure 3: Displaying second batch of matched signs

	GSL 20 Signs		DGS 40 signs	
	D	I	D	I
Top 1	92%	76%	59.8%	49.4%
Top 4	99.9%	95%	91.9%	85.1%

Table 1: Quantitative results of offline tests for D (signer dependent) and I (signer independent) tests.

5.1.1 *Data Sets*

In order to train the Kinect™interface, two data sets were captured for training the dictionary; the first is a data set of 20 GSL signs, randomly chosen from the Dicta-Sign lexicon, containing both similar and dissimilar signs. This data includes six people performing each sign an average of seven times. The signs were all captured in the same environment with the Kinect™and the signer in approximately the same place for each subject. The second data set is larger and more complex. It contains 40 DGS signs from the Dicta-Sign lexicon, chosen to provide a phonetically balanced subset of HamNoSys motion and location phonemes. There are 14 participants each performing all the signs 5 times. The data was captured using a mobile system giving varying view points. All signers in both data sets are non-native giving a wide variety of signing styles for training purposes. Since this application is a dictionary, all signs are captured as root forms. This removes the contextual information added by grammar in the same way as using an infinitive to search in a spoken language dictionary.

5.1.2 *GSL Results*

Two variations of tests were performed; firstly the signer dependent version, where one example from each signer was reserved for testing and the remaining examples were used for training. This variation was cross-validated multiple times by selecting different combinations of train and test data. Of more interest for this application however, is signer independent performance. For this reason the second experiment involves reserving data from a subject for testing, then training on the remaining signers. This process is repeated across all signers in the data set. Since the purpose of this work is as a translation tool, the classification does not return a single response. Instead, like a search engine, it returns a ranked list of possible signs. Ideally the sign would be close to the top of this list. Results are shown for two possibilities; the percentage of signs which are correctly ranked as the first possible sign (Top 1) and the percentage which are ranked in the top four possible signs.

As expected, the Dependant case produces higher results, gaining 92% of first ranked signs and nearly 100% when considering the top four ranked signs. Notably though, the user independent case is equally convincing, dropping to just 76% for the number one ranked slot and 95% within the top four signs.

5.1.3 *DGS Results*

The DGS data set offers a more challenging task as there is a wider range of signers and environments. Experiments were run in the same format as for the GSL data set. With the increased number of signs the percentage accuracy for the first returned result is lower than that of the GSL tests at 59.8% for dependent and 49.4% for independent. However the recall rates within the top four ranked signs (now only 10% of the dataset) are still high at 91.9% for the dependent tests and 85.1% for the independent ones. For comparison, if the system had returned randomly ranked list of signs there would have been a 2.5% chance of the right sign being in the first slot and a 10.4% chance of it being in the top four results.

While most signs are well distinguished, there are some signs which routinely get confused with each other. A good example of this is the three DGS signs ‘already’, ‘Athens’ and ‘Greece’ which share very similar hand motion and location but are distinguishable by handshape which is not currently modelled.

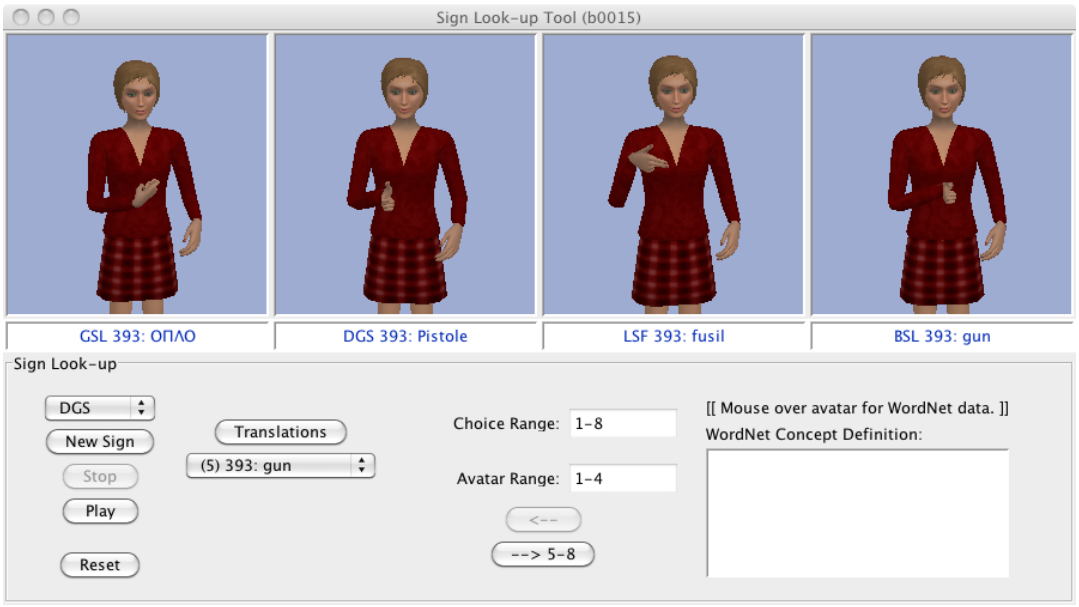


Figure 4: Displaying the signs corresponding to the concept Gun

5.2 Initial User Evaluation

Detailed user evaluation is an ongoing process as the system evolves. Here we present some initial, mainly qualitative, results of evaluation of an early version of the system based on the GSL data. Results of these micro-tests are being used to improve continuing work in this area.

5.2.1 *Familiarity with Signs*

The users performing the tests were familiar with LSF while the tests used exclusively GSL. Hence many of the signs used would be new to the testers, as if they were looking up new signs in a foreign language dictionary. In an initial evaluation, a range of the GSL signs were shown and testers asked to guess the meaning of the signs. About a quarter were highly iconic and were guessed reliably by Deaf testers. About another quarter were guessed correctly by over half the testers.

5.2.2 *Search Effectiveness*

The testers were then able to use the recognition system to search for specific signs. In about 25% of searches, the target sign was the first presented. In 50% of cases the sign was in the top four. The sign was in the top eight in 75% of cases. These results are promising though not as good as for the analysis presented above for the recognition system. A number of factors explain these results. For a proportion of searches, unfamiliarity with using the Kinect^{TM} device meant that the recogniser was not able to track the tester's signing and unreliable results were returned. Even for iconic signs, there will be small differences between languages that will impede recognition. Also, as already noted in section 5.1.1, the original GSL data set was captured under consistent conditions. As the prototype was evaluated under different conditions it became apparent that the system had become dependent on some of the consistency shown in the training data. This informed the method used for capturing future data and a mobile capture system was developed.

5.2.3 *Use in Translation*

Although the Search-By-Example tool is meant as a proof of the concept that sign language input (recognition) and output (synthesis) can be combined in a working tool, it has not been designed to fulfil a specific need for translation between sign languages. Nevertheless, some experiments were done, providing testers with a sentence in GSL to be translated to LSF. When required, the tester could use the translation functionality. An example of one of the sentences for translation was: '*In 2009, a huge fire devastated the area* **around Athens**. *Next* **autumn**, *there were only* **ashes**.' These sentences were recorded as videos by native GSL signers. Highlighted words in the sentence are concepts whose signs that can be recognised by the system. Other parts of the sentences were performed in an iconic way, allowing the LSF tester to directly understand them. GSL syntax was preserved for the test. The testers were asked to watch the videos and to give a translation of the video into LSF using (if needed) the software. Due to the iconic nature of signing, a good proportion of each sentence was correctly translated on the first attempt, although in most cases errors remained. By using the tool to recognise GSL signs and substitute the LSF equivalents, most testers were able to provide an accurate translation in about three attempts. This test also highlighted that significant failures occur when the sign being searched is used in a different context than that trained for. The example highlighted by the evaluation was where the training included the sign 'bend' as in to bend a bar down at the end, while the signers were searching for 'bend' as in bending an object towards them. It is clear that without an obvious root version of a sign, the long term aim should be to incorporate the ability to recognise signs in context.

5.2.4 *User Interface and Avatar Usage*

A number of problems were reported in the ability of testers to recognise the signs shown by the avatar. These included: distraction caused by all animations running together; signing speed appearing to be very high; size of avatar being too small for discrimination of small details. The first of these could be solved by enabling animation to occur only on selected panels. It has been observed before that comprehension of the avatar signing is better at a slower speeds. It is straightforward to slow down the performance of all signs and entirely practical to add a speed control, as is present on a number of existing JASigning applications. Controls already exist for users to zoom in on sections of the avatar, but it is not clear if the testers had been introduced to all the options available in the prototype system. Hence an extended period of training and exploration should be provided in future user evaluations.

6. CONCLUSIONS AND FUTURE WORK

A prototype recognition system has been presented which works well on a subset of signs. It has been shown that signer independent recognition is possible with classification rates comparable to the more simple task of signer dependent recognition. This offers hope for sign recognition systems to be able to work without significant user training in the same way that speech recognition systems currently do. Future work should obviously look at expanding the data set to include more signs. Preliminary tests on video data using a similar recognition architecture resulted in recognition rates above 70% on 984 signs from the Dicta-Sign lexicon. Due to the lack of training data this test was performed solely as a signer dependent test. Since the current architecture is limited by the signs in the Kinect^{TM}training database it is especially desirable that cross-modal features are developed which can allow recognition systems to be trained on existing data sets, whilst still exploiting the real-time benefits of the Kinect^{TM}.

As part of the increase in lexicon it will also be necessary to expand the type of linguistic concepts which are described by the features; of particular use would be the handshape. Several methods are available for appearance based handshape recognition but currently few are sufficiently robust for such an unconstrained environment. The advantages offered by the Kinect^{TM}depth sensors should allow more robust techniques to be developed. However, handshape recognition in a non-constrained environment continues to be a non-trivial task which will require significant effort from the computer vision community. Until consistent handshape recognition can be performed, including results would add noise to the recognition features and be detrimental to the final results.

On the other side of the prototype a synthesis system has been developed to display the results of recognition. User evaluation has revealed a number of areas where the interface could be improved in order to enhance the usefulness of the system. In particular, users should be given the option of greater control over which of the recognised signs are animated and over the speed, size, and view of the avatar.

Dicta-Sign is a research project so independent development of recognition and synthesis systems is an option. However, there is great value to the Deaf user community if working systems are constructed that will prototype the types of networked applications that are made possible by the project. The aim is to pull these technologies together with advanced linguistic processing in a Sign Wiki, a conceptual application that would provide Deaf communities with the same ability to contribute, edit, and view sign language materials developed in a collaborative fashion. The Search-By-Example system is intended as a small step in that direction.

7. REFERENCES

[1] H. Cooper and R. Bowden. Sign language recognition using linguistically derived sub-units. In *Procs. of LRECWorkshop on the Representation and Processing of Sign Languages : Corpora and Sign Languages Technologies*, Valetta, Malta, May17 – 23 2010.

[2] R. Elliott, J. Bueno, R. Kennaway, and J. Glauert. Towards the integrationn of synthetic sl annimation with avatars into corpus annotation tools. In T. Hanke, editor, *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Valletta, Malta, May 2010.

[3] R. Elliott, J. Glauert, V. Jennings, and J. Kennaway. An overview of the sigml notation and sigmlsigning software system. In O. Streiter and C. Vettori, editors, *Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 98–104, Lisbon, Portugal, 2004.

[4] R. Elliott, J. Glauert, J. Kennaway, and K. Parsons. D5-2: SiGML Definition. *ViSiCAST Project working document*, 2001.

[5] H. Ershaed, I. Al-Alali, N. Khasawneh, and M. Fraiwan. An arabic sign language computer interface using the xbox kinect. In *Annual Undergraduate Research Conf. on Applied Computing*, May 2011.

[6] J. Han, G. Awad, and A. Sutherland. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623 – 633, Apr. 2009.

[7] T. Hanke and C. Schmaling. *Sign Language Notation System*. Institute of German Sign Language and Communication of the Deaf, Hamburg, Germany, Jan. 2004.

[8] T. Kadir, R. Bowden, E. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Procs. of BMVC*, volume 2, pages 939 – 948, Kingston, UK, Sept. 7 – 9 2004.

[9] K. Lyons, H. Brashear, T. L. Westeyn, J. S. Kim, and T. Starner. Gart: The gesture and activity recognition toolkit. In *Procs. of Int.Conf.HCI*, pages 718–727, July 2007.

[10] M. Müller, T. RÃűder, and M. Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24(3):677–685, 2005.

[11] E.-J. Ong and R. Bowden. Learning sequential patterns for lipreading. In *Procs. of BMVC To Appear*, Dundee, UK, Aug. 29 – Sept. 10 2011.

[12] OpenNI organization. *OpenNI User Guide*, November 2010. Last viewed 20-04-2011 18:15.

[13] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *Procs. of*

Int.Conf.CVPRWkshp :Gesture Recognition, Colorado Springs, CO, USA, June 21 – 23 2011.

[14] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. *Hamburg Notation System for Sign Languages—An Introductory Guide*. International Studies on Sign Language and the Communication of the Deaf. IDGS, University of Hamburg, 1989.

[15] PrimeSense Inc. *Prime Sensor^{TM}NITE 1.3 Algorithms notes*, 2010. Last viewed 20-04-2011 18:15.

[16] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. *Computational Imaging and Vision*, 9:227 – 244, 1997.

[17] R. Sutton-Spence and B. Woll. *The Linguistics of British Sign Language: An Introduction*. Cambridge University Press, 1999.

[18] C. Valli, C. Lucas, and K. J. Mulrooney. *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press, 2005.

[19] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *Procs. of ICCV*, volume 1, pages 116 – 122, Corfu, Greece, Sept. 21 – 24 1999.

[20] H. Wassner. kinect + reseau de neurone = reconnaissance de gestes. `http://tinyurl.com/5wbteug`, May 2011.

[21] P. Yin, T. Starner, H. Hamilton, I. Essa, and J. M. Rehg. Learning the basic units in american sign language using discriminative segmental feature selection. In *Procs. of ASSP*, pages 4757 – 4760, Taipei, Taiwan, Apr. 19 – 24 2009.

[22] Z. Zafrulla, H. Brashear, P. Presti, H. Hamilton, and T. Starner. Copycat - center for accessible technology in sign. `http://tinyurl.com/3tksn6s`, Dec. 2010.