

Hand Gesture Recognition: Self-Organising Maps as a Graphical User Interface for the Partitioning of Large Training Data Sets

Gunther Heidemann, Holger Bekel, Ingo Bax and Axel Saalbach
AG Neuroinformatics, Bielefeld University
P.O. Box 10 01 31, D-33501 Bielefeld, Germany
gheidema@techfak.uni-bielefeld.de

Abstract

Gesture recognition is a difficult task in computer vision due to the numerous degrees of freedom of a human hand. Fortunately, human gesture covers only a small part of the theoretical “configuration space” of a hand, so an appearance based representation of human gesture becomes tractable. A major problem, however, is the acquisition of appropriate labelled image data from which an appearance based representation can be built. In this paper we apply self-organising maps for a visualisation of large amounts of segmented hands performing pointing gestures. Using a graphical interface, an easy labelling of the data set is facilitated. The labelled set is used to train a neural classification system, which is itself embedded in a larger architecture for the recognition of gestural reference to objects.

1. Introduction

The evaluation of pointing gestures is a major goal in advanced human-machine interaction. Pointing gestures are used to indicate directions as well as for reference to objects or persons and form a fundamental component of communication. The recognition of human hands, however, turned out to be extremely difficult due to the variety of hand postures. While recognition systems for rigid objects have to deal only with three degrees of freedom for pose, approaches for the recognition of hands face an immense amount of additional degrees of freedom. The possible variations of hand postures are not only many but also include the problem of self-occlusion, so varying postures of a hand cannot be considered as minor appearance changes.

To solve problems of that kind, *appearance based* approaches to visual knowledge representation have become popular in computer vision (e.g. [11]). The idea is to capture the signal appearance of an object from sample images,

rather than modelling the object explicitly on a semantic level, e.g., in terms of geometry. Neural networks (NN) seem especially well-suited for appearance based representations, the most frequently named advantages being the following: (i) Easy training from samples, explicit modelling of objects is not required. (ii) Even highly difficult object properties can be captured, e.g., surface reflectance or internal degrees of freedom. (iii) As a consequence, NN allow to go beyond toy problems towards more realistic scenarios.

Looking more closely to the application of NN in practice, however, it turns out that these advantages evaporate, at least in a straight forward approach of NN training. The main reason is that the complexity of objects has to be covered by appropriate and labelled sample images or image patches. To get these, there are two complementary ways:

1. Image acquisition of isolated objects in well-defined poses and in front of uniform background, e.g., using a turntable as in [12]. This method has the following advantages / disadvantages:
 - + After acquisition, no labelling of objects or poses is required.
 - + Perfect object / background separation possible.
 - Only objects which can be isolated and have a certain size can be used.
 - Selection of views is difficult when the object has poses which are not equally likely (like a pencil) or if it has internal degrees of freedom (like a hand) which span a configuration space of which only a small subspace is actually realised.
2. Segmentation of image patches containing the object of interest from real world imagery:
 - + Objects appear under the relevant viewing conditions, only poses actually realised are sampled.
 - Requires extensive hand-labelling of object classes and poses.

- Perfect segmentations are difficult to achieve, e.g., segmentation of a cup in a real world kitchen is comparably hard as recognition itself.

In this contribution, we try to get the best of both these alternatives for the problem of hand gesture recognition. While acquisition conditions can be kept simple (black background, fixed lighting), it is impossible to sample hand postures in the way of a turntable. Instead, natural gestures are used so that only the relevant part of the configuration space is sampled. This leaves the task of labelling a huge dataset. To solve this key problem, we use the Kohonen Self-Organising Map (SOM) [10] to pre-structure and visualise the image data set. A graphical user interface facilitates a fast and easy image labelling. The resulting labelled data set is used to train a neural classifier.

In the following, we will first describe the data acquisition system (section 2) and the neural classifier (section 3) for pointing gesture recognition. Section 5 describes the setup in which the pointing recognition system is embedded.

2. Image data acquisition using SOMs

Fig. 1 shows the data acquisition for pointing gestures. As input, a camera views a table with several objects plus the hand of a subject, who points at one of the objects. The output are labelled patches of the images which can be used to train a classifier.

The subject is asked to point at will at the objects to ensure a natural gesture. The recorded sequence is then segmented for skin colour, such image windows are cut out to form the yet unordered training set $U = \{\vec{x}_1 \dots \vec{x}_{N_U}\}$. Each window is regarded as vector $\vec{x}_i \in \mathbb{R}^D$ of pixel space, where D is the number of pixels of the windows (which is fixed). To visualise U , we use the “Visualisation And Labelling Toolkit” (VALT) which was originally proposed in [15]. VALT trains a SOM on the unordered data set U after the adaptation rule of Kohonen (e.g. [10]). After training, the resulting nodes of the SOM can be visualised in two ways:

1. Weight vectors: The weight vectors are visualised directly in pixel space. This back-projection corresponds only to real images if enough nodes were used, otherwise the nodes represent superpositions of real images.
2. Best match example: In this mode, VALT shows for each node the window which is closest to the weight vector in pixel space.

For examples see section 4. Using the visualisation, the user can now partition U graphically into subsets which contain samples for the desired classes. The partitioning is carried

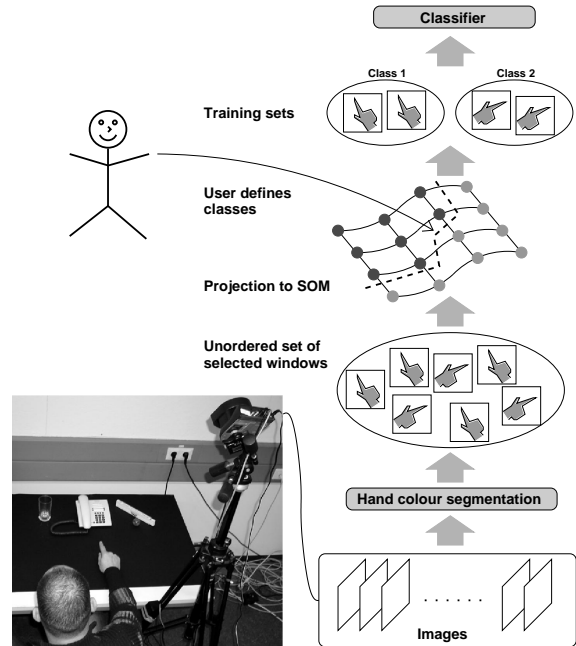


Figure 1. Acquisition of labelled image data: A camera views the pointing hand of the user. Segmentation of skin-coloured blobs yields an unordered set of windows with a pointing hand. This set is projected onto a SOM and visualised, an interactive user interface allows the division into subsets of similar pointing gestures. From these subsets the classifier can be trained.

out by drawing lines around the nodes which are to be assigned to a certain class.

The resulting labelled sets of windows can now be used to train a classifier as described in the next section.

3. The neural classification system

For visual classification, we use the VPL system, which was previously applied to several computer vision tasks (e.g. [5]). “VPL” stands for three processing stages: Vector quantisation, PCA and LLM-network. The VPL classifier combines visual feature extraction and feature classification by means of local principal component analysis (PCA) [9, 2, 17] for dimension reduction followed by a subsequent classification stage. Local PCA can be viewed as a non-linear extension of simple PCA and was applied to various pattern recognition tasks (e.g. [8, 11]). An overview of the processing flow is given in Fig. 2.

Vector quantisation is carried out on the raw image windows to provide a partitioning of the data with N_V reference

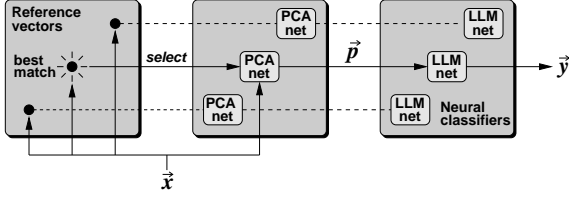


Figure 2. VPL classifier: Features are extracted by local PCA and classified subsequently by neural classifiers.

vectors $\vec{r}_i \in \mathbb{R}^D, i = 1 \dots N_V$, using the *Activity Equalisation Algorithm* proposed in [7]. This algorithm is particularly well suited for the task since it avoids the problem of codeword under-utilization [3] by counting codeword access frequencies in a way related to [1].

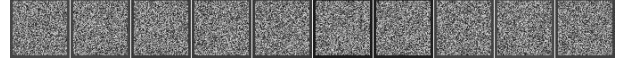
To each reference vector \vec{r}_i a single layer feed forward network is attached for the successive calculation of the principal components (PCs) as proposed by Sanger [16]. After training it projects the input $\vec{x} \in \mathbb{R}^D$ to the $N_P < D$ PCs with the largest eigenvalues: $\vec{x} \rightarrow \vec{p}_i(\vec{x}) \in \mathbb{R}^{N_P}, i = 1 \dots N_V$. On the third processing level, to each PCA-net one “expert” neural classifier of the Local Linear Map – type (LLM network) is attached. The LLM performs the final mapping $\vec{p}_i(\vec{x}) \rightarrow \vec{y}$. The LLM network is related to the self-organising map [10], see e.g. [14] for details. The LLM can be trained to approximate a nonlinear function by a set of locally valid linear mappings.

The three processing stages are trained successively: First vector quantisation and PCA-nets are trained unsupervised on the unordered data set U , then the LLM nets are trained supervised using the labelled data sets. For classification of an input \vec{x} first the best match reference vector $\vec{r}_{n(\vec{x})}$ is found, then \vec{x} is mapped to $\vec{p}_{n(\vec{x})}(\vec{x})$ by the attached PCA-net and finally $\vec{p}_{n(\vec{x})}(\vec{x})$ is mapped to \vec{y} : $\vec{p}_{n(\vec{x})}(\vec{x}) \rightarrow \vec{y}$.

The major advantage of the VPL classifier is its ability to form many highly specific feature detectors (the $N_V \cdot N_P$ local PCs). It could be shown that classification performance and generalisation properties are well-behaved when the main parameters are changed, which are N_V, N_P and the number of nodes in the LLM nets N_L [5].

4. Results

To acquire an image database we asked three subjects to point at random at objects spread out on the table. In total, a set of 3000 image windows showing pointing hands was collected. Due to the fixed lighting conditions hand colour segmentation proved to be reliable. In the real scenario, also other objects are present in the scene (see next section), but



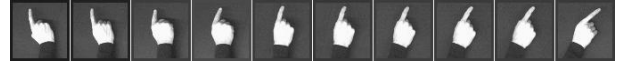
(a) Weight vectors of randomly initialised 10×1 -SOM



(b) Best match examples of randomly initialised SOM



(c) Weight vectors of trained SOM after 10000 steps



(d) Best match examples of trained SOM

Figure 3. Using a 10×1 -SOM for training data visualisation: (a),(b) before training with randomly initialised weights. (c),(d) after 10000 training steps neighbouring nodes represent similar pointing directions. (a),(c) show the weight vectors, (b),(d) the best match example from the training set for each node.

here we concentrate only on the hands for clarity.

The problem to classify a pointing angle in the plane suggests a linear SOM topology. A 10×1 -SOM (see Fig. 3) proved to be a reasonable compromise between resolution and data compression. Fig. 3 shows the SOM in both visualisation modes – *weight vectors* and *best match example* – and both before and after training. Before training, the assigned best match hand gestures point at random directions. Training was performed over 10000 steps with step-size decreasing exponentially from 0.5 to 0.01. After training, the assigned best match samples are well ordered, so angles can be easily assigned to form the labelled training set. For the VPL, 4 reference vectors with 6 local PCs and 20 LLM nodes proved to be sufficient. “Sufficient” means that a good performance can be achieved in the object reference system described in section 5.

The major result is not the accuracy of the classifier, but the almost effortless way in which it was achieved. Image acquisition takes about 10 minutes for each subject. Training of the SOM takes less than one minute on a standard PC and converges with high reliability to a configuration as in

Fig. 3. For 20 repetitions, 18 converged to the well ordered state. Assigning classes (the approximate pointing angle) to each node takes about another five minutes.

5. The embedding system architecture

So far, the system was described for the purpose of pointing direction recognition. Actually, this is only one module of a larger system for visual object reference, the other modules of which are described in [6]. In this system, the camera views both the pointing hands and the objects. On the objects, salient points are detected using low level saliency features like entropy, symmetry [13] or edges and corners using the Plessey detector [4]. These features are integrated by an adaptive weighting to a common saliency map. Maxima of this map indicate locations which are likely to be pointed at, so the “search space” for object reference can be restricted to discrete points. In this scenario, the VPL classifier is trained not only to recognise pointing directions but also to distinguish a pointing hand from (a) a non-pointing gesture and (b) other objects. The training of this VPL requires a larger SOM during the data acquisition phase.

The accuracy of the recognition results can be tested only in terms of the success a user has in referencing objects. This success rate is relatively low if the user has no feedback at all about where he is pointing at, but can be significantly increased using graphical or auditory feedback. A success rate of about 80% for 4° resolution can be realised in a prototypic evaluation scenario, for details see [6].

6. Conclusion

We have presented a system for the fast and easy acquisition of *labelled* image data for the training of neural classifiers. By this approach, both highly artificial acquisition setups and time consuming hand labelling can be avoided. As an example, we used the evaluation of pointing gestures because images of human hands cannot be acquired in the controlled way of a turntable setup. Instead, images of natural movements were used and pre-structured using a SOM for visualisation. The visualisation makes labelling easy because it “suggests” classes to the user, which have the additional advantage of being classifiable on the signal level. In future work, we hope to extend the system to a larger variety of gestures and the simultaneous recognition of a greater selection of objects.

7. Acknowledgement

This work was supported within the project VAMP-IRE (Visual Active Memory Processes and Interactive Retrieval) which is part of the IST programme (IST-2001-34401).

References

- [1] S. C. Ahalt, A. K. Krisnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3:277–290, 1990.
- [2] C. Bregler and S. M. Omohundro. Surface Learning with Applications to Lipreading. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 1993*, volume 6, pages 43–50. Morgan Kaufmann Publishers, 1994.
- [3] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Sci.*, 11:23–63, 1987.
- [4] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conf.*, pages 147–151, 1988.
- [5] G. Heidemann, D. Lücke, and H. Ritter. A System for Various Visual Classification Tasks Based on Neural Networks. In A. S. et al., editor, *Proc. 15th Int'l Conf. on Pattern Recognition ICPR 2000, Barcelona*, volume I, pages 9–12, 2000.
- [6] G. Heidemann, R. Rae, H. Bekel, I. Bax, and H. Ritter. Integrating context-free and context-dependent attentional mechanisms for gestural object reference. In *Proc. ICVS 03*, pages 22–33, Graz, Austria, 2003.
- [7] G. Heidemann and H. Ritter. Efficient Vector Quantization Using the WTA-rule with Activity Equalization. *Neural Processing Letters*, 13(1):17–30, 2001.
- [8] G. E. Hinton, P. Dayan, and M. Revow. Modelling the Manifolds of Images of Handwritten Digits. *IEEE Trans. on Neural Networks*, 8(1):65–74, 1997.
- [9] N. Kambhatla and T. K. Leen. Fast Non-Linear Dimension Reduction. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 1993*, volume 6, pages 152–159. Morgan Kaufmann Publishers, 1994.
- [10] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 1995.
- [11] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.
- [12] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *Int'l J. of Computer Vision*, 14:5–24, 1995.
- [13] D. Reissfeld, H. Wolfson, and Y. Yeshurun. Context-Free Attentional Operators: The Generalized Symmetry Transform. *Int'l J. of Computer Vision*, 14:119–130, 1995.
- [14] H. J. Ritter, T. M. Martinetz, and K. J. Schulten. *Neuronale Netze*. Addison-Wesley, München, 1992.
- [15] A. Saalbach. *Self-Organizing Maps zur halbautomatischen Erzeugung datennaher Klasseneinteilungen*. Master's thesis, Univ. Bielefeld, Technische Fakultät, 2001.
- [16] T. D. Sanger. Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. *Neural Networks*, 2:459–473, 1989.
- [17] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.