in: Proceedings of the Bielefeld Gesture Workshop, September 17-19 1997, Germany. Springer, LNAI 1371.

Robotic Gesture Recognition

Jochen Triesch[1] and Christoph von der Malsburg[1,2]

[1] Institut für Neuroinformatik
Ruhr-Universität Bochum
D-44780 Bochum, Germany
[2] University of Southern California
Dept. of Computer Science and Section for Neurobiology
Los Angeles, CA, USA

Abstract. Robots of the future should communicate with humans in a natural way. We are especially interested in vision-based gesture interfaces. In the context of robotics several constraints exist, which make the task of gesture recognition particularly challenging. We discuss these constraints and report on progress being made in our lab in the development of techniques for building robust gesture interfaces which can handle these constraints. In an example application, the techniques are shown to be easily combined to build a gesture interface for a real robot grasping objects on a table in front of it.

1 Introduction

Robots of the future must interact with humans in a natural way if they are to become part of our everyday lives. They must understand spoken and gestural commands and also articulate themselves by these means. We are especially interested in vision-based gesture recognition for robots operating in real world environments. This poses a number of constraints to *human robot interaction* as a special case of *human computer interaction*:

- The robot must be capable of realtime gesture recognition. If the robot's responses are slow, it is tiresome to use. However, with the increasing speed of standard hardware a system that falls short of realtime perfomance today, will work satisfactorily in a couple of years.
- The system must be person independent. For most applications it is desirable that many potential users can operate the robot, even if the robot has never seen them before.
- The system must not require the user to wear special clothes or cumbersome devices such as coloured markers or data gloves, since this is too tedious for the user.
- The robot will face variable and possibly complex backgrounds against which a user operates it. A system requiring constant background is not flexible enough for real world applications.

– The robot must cope with variable lighting situations. The requirement of constant lighting is too big a restriction for any real world application.

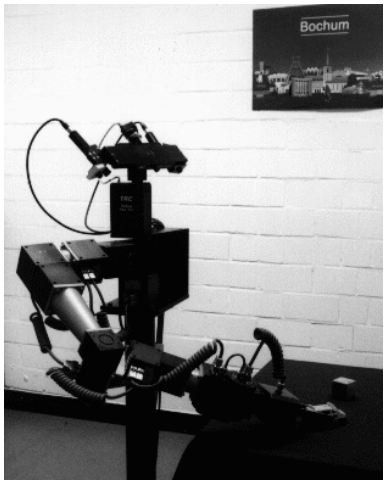

Fig. 1. The robot at our lab has a kinematically redundant arm and a three degrees of freedom stereo camera head.

Automatic visual gesture recognition has received much attention recently, for a review see [10]. However, hardly any published work acknowledges all the requirements stated above.

In the work of Franklin *et al.* [4] an attempt to build a robot waiter is presented, a domain which indeed poses all the above challenges. The robot's gesture analysis, however, so far only distinguishes between an empty hand and a hand holding an object based on how much skin colour is visible. The system presented by Cui and Weng [3] recognizes different hand gestures in front of complex backgrounds. It reaches 93.1% correct recognition for 28 different gestures, but is not person independent and relies on a rather slow segmentation scheme taking 58.3 seconds per image. Heap and Hogg [5] present a method for hand tracking using a deformable model, which also works against complex backgrounds. The deformable model describes one hand posture and certain variations of it and is not aimed at recognizing different postures. The work presented by Campbell *et al.* [2] is an example of a system recognizing two-handed gestures. It allows only for motion based gestures, because it is not analyzing the shape of the user's hands. In Kjeldsen's and Kender's work [7] a realtime gesture system for controlling a window-based computer user interface is presented. The posture analysis is indeed quite fast (2 Hz). A minor drawback of the system is that its hand tracking has to be specifically adapted for each user. The system presented by Maggioni [9] has a similar setup. It requires a constant background to the gesturing hand.

We are working with a real robot and try to acknowledge all the constraints mentioned. The robot at our lab has a kinematically redundant arm with seven degrees of freedom, which allows it to grasp an object from various directions (figure 1). On top of the arm a stereo camera head with three degrees of freedom is mounted; this allows for pan, tilt and vergence motion. The cameras yield images 768 x 572 pixels in size (35° field of view).

The rest of the paper is organized as follows. In section 2 we will present techniques for tracking the head and hands of a person. Section 3 deals with techniques for the recognition of hand postures from camera images. In section 4 we describe how these techniques may be combined to implement an example gesture interface for our robot whose task it is to move around objects scattered on a table in front of it. Finally, section 5 discusses the achievements and gives an outlook.

2 Tracking of Heads and Hands

A prerequisite for the successful recognition of gestures is tracking the head and hands of the gesturing person. We combine motion, colour and stereo cues to reach the robustness demanded by real world applications. Let us first consider a hand pointing to some objects on a table in front of the robot (figure 2). We work

Fig. 2. Tracking scenario. The user's hand is pointing to objects on a table and has to be tracked. This is also the setting of our example application described in section 4.

on low resolution images with a size of 96 x 71 pixels in HSI (hue, saturation, intensity) colour format. The tracking currently runs at a maximal speed of 8 Hz using stereo and 12.5 Hz using only one camera on a standard PentiumPro PC without any special image processing hardware.

2.1 Motion Cue

We compute a thresholded version of the absolute difference images of the intensity (I) components of consecutive images according to

$$M^{l,r}(x,y,t) = \Theta(\mid I^{l,r}(x,y,t) - I^{l,r}(x,y,t-1) \mid - \Gamma),$$

where Γ is a fixed threshold and Θ is the Heavyside function. Afterwards, we apply a local regularization algorithm, which switches on pixels that have a high number of direct neighbours which are on and which switches off isolated pixels being on. The result of such processing is depicted in figure 3. The motion cue responds to all moving image regions, i.e., to all moving objects and to some extent also to their shadows falling over still objects, producing artifacts. Also, it is of course fooled by motions of the camera head itself resulting in perceived motion almost everywhere.

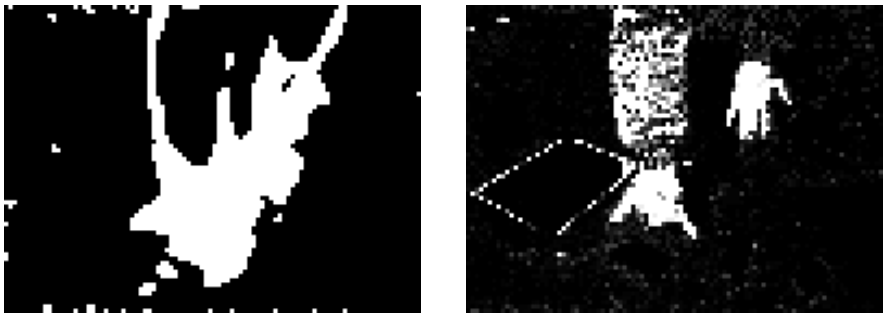

Fig. 3. Results of the motion cue (left) and colour cue (right). Usually, none of the cues alone allows perfect tracking. Only by combining several cues the robustness necessary for real world applications is reached.

2.2 Color Cue

Skin colour detection is based on the hue (H) and saturation (S) components of the image. We distinguish two types of processing. The first uses a very coarse and unrestrictive model of skin colour, defined as a prototypical point in the HS plane. For each pixel of the input image we compute its Euclidean distance to this point, where the axes are appropriately rescaled. The closer the pixel is to the prototype, the higher is its likelyhood of stemming from the head or hands of a person. When the lighting situation changes, e.g., due to the spotlights of a TV team, the prototypical point may no longer be appropriate. In order to deal with such cases we have introduced a second scheme which works with an adapted skin colour table. Before actually using the system its colour table is adapted to the current lighting situation (and the subjects particular skin colour) by

the subject showing his or her hand to the robot for a couple of frames. With this scheme the colour cue can be made much more restrictive. A result for the restrictive colour cue is depicted in figure 3. There is still a lot of noise in the cue and it will react to any approximately skin coloured objetcs such as the subject's jumper in figure 3. For the future, it would be desirable to have an online recalibration taking place while the robot is performing, which detects changes in the lighting situation and automatically adapts to them.

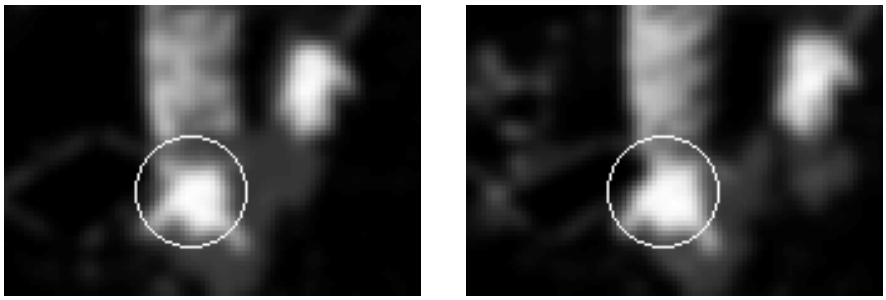

Fig. 4. Attention maps extracted from the two cameras. Each attention map contains two strong blobs of activity referring to the two hands of the subject. The circles are drawn around the hand selected due to the stereo cue (see section 2.4.

2.3 Attention Maps

We compute attention maps for each camera by combining motion and colour cue. For each camera the result of the colour cue and the motion cue are added with appropriate weighting factors. The stronger weight is put on the colour cue since it appears more reliable as head and hands are not necessarily moving all the time. The additive combination of cues ensures that the system will keep working (although in a deteriorated fashion) if one of the cues breaks down. The attention maps are then computed by convolving the summation results with a Gaussian kernel in order to make them smooth and emphasize larger blobs. For the scene of figure 3 the attention maps of the left and right camera are depicted in figure 4. In both attention maps there are two strong blobs of activity belonging to the left and right hand of the subject.

2.4 Stereo Cue

The stereo cue is intended to select only objects that are in the plane of fixation of the robot. For this purpose, the attention maps of left and right image (figure 4) are simply added (figure 5). This highlights only the responses of objects in the plane of fixation because only for these do the contributions of the left and right attention map overlap in image space. The point with the highest response

in the sum of the attention maps is the assumed position of the target hand. Starting from its coordinates, a gradient ascent is performed in the attention maps of the left and right image until the local maxima corresponding to the selected hand are reached. The spatial positon of the hand can now be easily computed by triangulation.

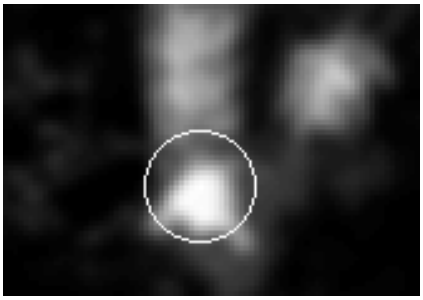

Fig. 5. Stereo cue. When the attenion maps of left and right image are added only the responses of objects in the plane of fixation lead to high responses. In this way, the hand moving in the background could be ruled out.

2.5 Active Tracking

Active tracking means that the camera head uses its three degrees of freedom to keep the target object approximately centered in both camera images. We control the vergence angle independently of the pan and tilt angles.

For control of the vergence angle between the cameras, we add the attention maps of left and right image with three different disparities, i.e., different relative horizontal displacements of a positive, zero and negative number of pixels. The resultant images stress features in front of, in, and behind the plane of fixation, respectively. We compute the global maxima in the three images. If the highest response was for the image focusing in front of the plane of fixation, the vergence angle is increased, i.e., the cameras are converged. Conversely, if it was highest behind the plane of fixation, the vergence angle is decreased.

If the highest of the three global maxima comes close to a vertical or horizontal border of the image, indicating that the hand is about to leave the current field of view, an appropriate saccade with the other two degrees of freedom is made to bring it back to the center. For fast moving objects bigger saccades have to be made in order to compensate for the object's motion during the time necessary to perform the saccade. During saccades the image acquisition is stopped so that the effective frame rate of the tracking depends on the rate and sizes of saccades and is generally slower than for passive tracking.

Fig. 6. Simultaneous tracking of several objects.

2.6 Multiple Objects

For some applications, e.g., sign language recognition, head and both hands of the subject have to be tracked simultaneously. This is done by simply looking for targets in the current frame in the vicinity of targets detected in previous frames (figure 6), i.e., the continuity of the targets' motion is exploited. This simple scheme has of course problems with mutually overlapping targets. If, for instance, both hands overlap strongly and then move apart again, tracking will not be able to tell which one was which. The incorporation of techniques for explicitely modelling the motion of the hands and predicting their future positions should attenuate this problem.

3 Hand Posture Classification

The second important building block of a gesture recognition system is the analysis of hand postures. Our posture recognition is based on *elastic graph matching*, which has already been successfully applied to object and face recognition [8, 13]. For some applications such as sign language recognition, the analysis of facial expressions is important. This can also be done with elastic graph matching (see [6]), but we will not discuss this point any further here.

Processing is done on grey scale images and has been shown to work independent of person and despite varying complex backgrounds. The system is described in more detail in [11]. It is an adaptation of our earlier system [12]. Here, we only give an outline of the system.

In elastic graph matching, objects are represented as labelled graphs, where the nodes carry local image information and the edges contain information about the geometry. One model graph is created for each object. The local image information at each node is often represented by a vector of responses to Gabor-

based kernels called a *jet*. These kernels are DC-free and defined by:

$$\psi_{\boldsymbol{k}}(\boldsymbol{x}) = \frac{\boldsymbol{k}^2}{\sigma^2} \exp\left(-\frac{\boldsymbol{k}^2\boldsymbol{x}^2}{2\sigma^2}\right) \left[\exp(i\boldsymbol{k}\boldsymbol{x}) - \exp\left(\frac{-\sigma^2}{2}\right)\right]$$

They are depicted in figure 7.

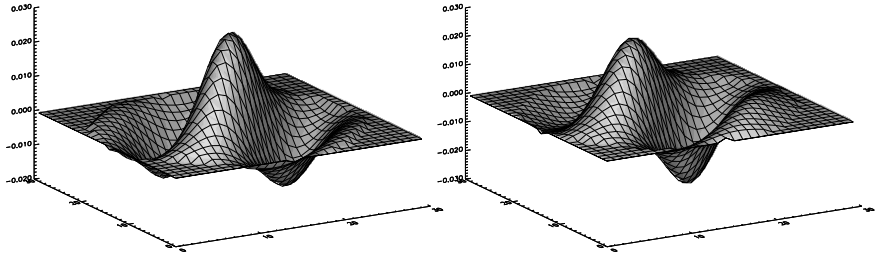

Fig. 7. Nodes of graphs are labeled with the responses to Gabor based kernels. They have the form of a plane wave restricted by a Gaussian envelope function. Left: real part, right: imaginary part.

We fuse the graphs obtained from two persons performing the posture into a single *bunch graph* for each posture; for details on the bunch graph concept see [13]. We currently use a set of six postures taken from a finger spelling alphabet (figure 8).

A graph representing a particular posture is matched onto an image by moving it across the image until the jets at each node fit best to the regions in the image they come to lie on. During the matching process we allow for certain geometrical transformations of the graph such as scaling and rotation in plane. For recognition of a posture, the graphs of all postures are sequentially matched onto the images of left and right camera and the posture whose graph obtains the highest total similarity (sum of similarities in left and right image) is selected as the winner.

The result of a matching process on the image of one camera is depicted in figure 9. As we do not use a local diffusion of nodes but keep the graph rigid, the match does not fit particularly well, but is reliable enough for posture recognition.

Our previous system [12] reached about 86% correct recognition for 10 different postures in front of complex backgrounds, but recognition took about 16 seconds on a Sun Ultra Sparc Workstation. By means of a smaller alphabet of six postures and coarser model graphs, we managed to reduce the recognition time by a factor 3–5. Experiments with novices using the system show a correct classification in four out of five cases. The positions of the graphs in the left and right images after matching are usually precise enough to allow for an estimate

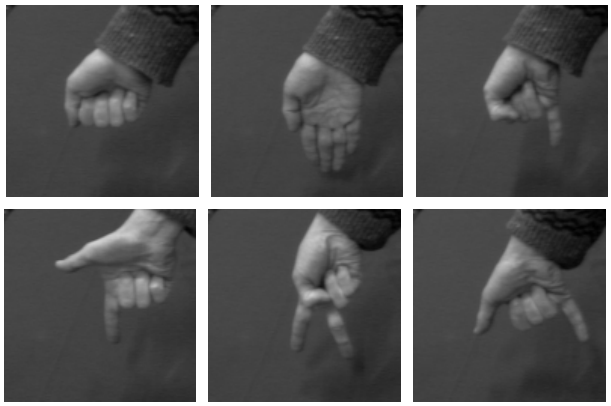

Fig. 8. The six postures tell the robot whether to grasp an object from above, the front, the side, and so on. They are taken from the American Manual Alphabet and were slightly adapted to make them more comfortable and more intuitive.

of the hand's three-dimensional position with 1-2 cm accuracy for a distance of about 1 m from the robot.

3.1 Using Colour

It seems likely that graph matching can be improved by adding colour information to the models. We have done a preliminary study, where we have added also colour information to the nodes of the model graphs. The similarity of such a *compound jet* to a region of the image is defined as a weighted sum of the Gabor responses' similarity and a measure of the local colour similarity. We prepared a model graph for each of the six postures from a single example image of one person signing against uniform background and tested it on a gallery of 108 test images of different persons signing against complex backgrounds (see figure 10), varying the influence of the colour and the Gabor cue. The results are depicted in figure 11. While recognition was relatively poor using only colour (39%) or only Gabor information (54%), the combination of both lead to recognition rates of 70%. The generally low scores are due to the fact that the model graphs were created from only a single example image.

4 An Example Application

We have designed an example application where the robot stands in front of a table with objects on it. The user tells it with gestures which object to grasp and how to grasp it, as well as where to put it (figure 2). This application requires a number of other skills needed by the robot, e.g., recognition of shape and orientation of the object pointed to, grip planning and grip execution, which are discussed elsewhere [1].

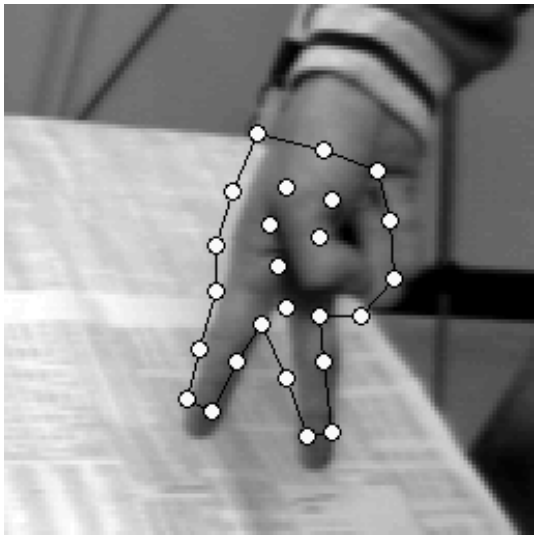

Fig. 9. Example of a graph of the correct posture being matched onto the input image.

The gestures are defined as pointing movements to the objects with a particular hand shape. The user's hand is tracked until it comes to rest using the techniques described in section 2. As the whole table is within the robot's field of view, we can do without active tracking. Also, gestures are single handed so only one hand has to be tracked. After tracking the hand posture is analyzed as described in section 3. It tells the robot whether to grasp the object from, e.g., above, the side, the front, and so on. After the robot has grasped the object the user indicates with a second gesture where the object shall be placed and the robot puts it there.

5 Discussion and Outlook

We have discussed the requirements for gesture interfaces of robots operating in the real world. We have presented vision-based techniques for tracking the head and hands of a person, as well as for the analysis of hand shapes taking these real world requirements into account. Both the tracking and the hand posture recognition are purely vision-based and do not require the user to wear gloves or other devices. They function person independent and despite complex backgrounds. The tracking is capable of realtime performance. For the computationally more extensive posture analysis, there is a complex tradeoff between the allowed number of postures, the spatial accuracy and the speed of the matching process. In its current guise, however, it is not far from realtime performance without any special hardware. In order to increase the robustness with respect to different lighting situations we are currently working on an automatic color constancy module.

Fig. 10. One of the six postures of the preliminary color study performed by nine different subjects against varying complex backgrounds.

We have presented preliminary results indicating that colour information can significantly enhance the hand posture analysis. This point will be further investigated.

For the future, we intend to close the gestural communications loop by letting the robot perform gestures itself, e.g., by pointing to unfamiliar objects whose names or associated gestures the user then supplies to the robot.

Also, we are interested in imitation learning, where the robot learns how to grasp particular objects by observing the human example.

Acknowledgements

This work was supported by a grant from the German Federal Ministry for Science and Technology (01 IN 504 E9).

References

1. M. Becker, E. Kefalea, E. Maël, C. v. d. Malsburg, M. Pagel, J. Triesch, J.C. Vorbrüggen, and S. Zadel. GripSee: a robot for visually-guided grasping. Submitted to: ICRA'98.
2. L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobic, and A. Pentland. Invariant features for 3-d gesture recognition. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.

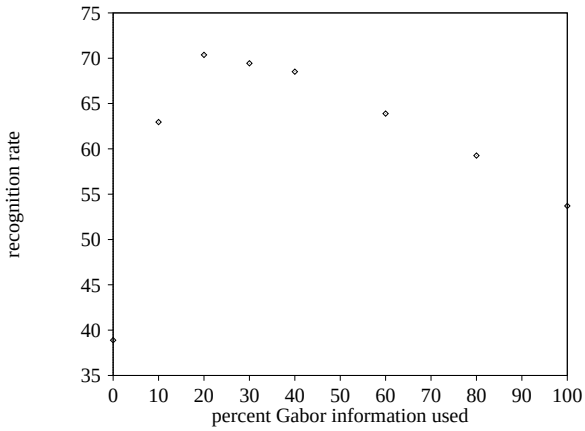

Fig. 11. Results of the preliminary color study: The use of colour information can aid the posture classification. While with Gabor responses to grey images or colour alone recognition performance is rather poor, it may be enhanced significantly by combining the two. In the ordinate is plotted the percentage of Gabor information used for recognition; on abscissa the percentage of correctly recognized postures.

3. Y. Cui and J. J. Weng. Hand sign recognition from intensity image sequences with complex backgrounds. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.
4. D. Franklin, R. E. Kahn, M. J. Swain, and R. J. Firby. Happy patrons make better tippers. Creating a robot waiter using perseus and the animate agent architecture. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.
5. T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.
6. H. Hong, H. Neven, and C. v. d. Malsburg. Online facial expression recognition based on personalized gallery. Accepted for publication at: FG'98, The IEEE Third International Conference on Automatic Face and Gesture Recognition.
7. R. Kjeldsen and J. Kender. Toward the use of gesture in traditional user interfaces. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.
8. M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.
9. C. Maggioni. GestureComputer — new ways of operating a computer. In *Proceedings of the International Workshop on Automatic Face- and Gesture Recognition 1995, Zürich, Switzerland, June 26-28*, 1995.
10. V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. PAMI*, 19 7, 1997.
11. J. Triesch and C. von der Malsburg. A gesture interface for robotics. Accepted for publication at: FG'98, The IEEE Third International Conference on Automatic Face and Gesture Recognition.

12. J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex backgrounds. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14-16*, 1996.
13. L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic graph matching. *IEEE Trans. PAMI*, 19 7, 1997.