

# Visual Hand Posture Recognition in Monocular Image Sequences

Thorsten Dick<sup>1</sup>, Jörg Zieren<sup>1</sup>, and Karl-Friedrich Kraiss

Institute of Man-Machine-Interaction, RWTH Aachen University, Germany  
{dick,zieren,kraiss}@mmi.rwth-aachen.de <http://www.mmi.rwth-aachen.de>

**Abstract.** We present a model-based method for hand posture recognition in monocular image sequences that measures joint angles, viewing angle, and position in space. Visual markers in form of a colored cotton glove are used to extract descriptive and stable 2D features. Searching a synthetically generated database of 2.6 million entries, each consisting of 3D hand posture parameters and the corresponding 2D features, yields several candidate postures per frame. This ambiguity is resolved by exploiting temporal continuity between successive frames. The method is robust to noise, can be used from any viewing angle, and places no constraints on the hand posture. Self-occlusion of any number of markers is handled. It requires no initialization and retrospectively corrects posture errors when accordant information becomes available. Besides a qualitative evaluation on real images, a quantitative performance measurement using a large amount of synthetic input data featuring various degrees of noise shows the effectiveness of the approach.

## 1 Introduction

Automatic recognition of hand gestures is an intuitive and efficient method for human-computer interaction. Applications for gesture input include virtual reality, motion capture, and sign language recognition. Vision-based recognition methods allow to measure both hand configuration and translational motion. Another important benefit is that a camera also records the user's face – a prerequisite for recognizing sign language.

For many tasks, such as fingertip detection or gesture classification, appearance-based 2D features (i.e. shape and texture) that can be extracted directly from the input image suffice [1, 2]. Reconstruction of the 3D hand posture from 2D images opens up additional applications that require knowledge of individual finger flexion.

This paper describes a model-based approach using visual markers and a matching OpenGL hand model to map an observation, described by 2D features, to a 3D hand posture. A large database of such mappings is generated offline. Efficient algorithms for identifying candidates with similar features form the core of the system. Since ambiguities and uncertainties in individual frames cannot be prevented when using 2D input data, disambiguation is performed by exploiting temporal continuity of the gesturing motion over a period of several seconds. Smoothing in posture space prevents jerkiness that would otherwise result from the finite number of discrete postures in the database.

---

<sup>1</sup> Supported by grant VV-Z50 from the Interdisciplinary Centre for Clinical Research "BIO-MAT:" within the Faculty of Medicine at the RWTH Aachen University.

The system achieves near real-time speed on a standard PC. A qualitative evaluation on signed numbers is presented, as well as an exact measurement of posture error using a total of 37,500 synthetic input images.

## 2 Related Methods and Common Difficulties

Numerous approaches for hand posture recognition have been proposed in recent years. They differ in the number of cameras used, the type of features extracted from the input data, the supported degrees of freedom (DOF), and possible limitations regarding input posture and viewing angle. The mapping of features to postures can be performed by either deriving joint and viewing angles directly through inverse kinematics, or by parameterizing a hand model so that it yields matching features.

Multi-camera approaches restrict translational hand motion at least to the intersection of all cameras' viewfields. Since stereo is less effective for remote objects, the hand is usually recorded from a short distance. Existing publications therefore do not consider significant translational motion [3–5] and further require controlled recording conditions, e.g. placing the hand inside a box containing light source and cameras.

Feature extraction on images of the unmarked hand constitutes a challenging problem, especially in the presence of motion blur and camera noise. A robust feature which can be extracted using a simple skin color model is the hand's contour [6, 7]. However, the contour is not stable because small changes in hand posture may greatly affect it. At the same time, by discarding texture it entails yet another loss of input information in addition to the 3D-to-2D projection. Many different hand postures result in the same contour (for example, a fist and a pointing index finger seen from the pointing direction), rendering this feature problematic for unrestricted posture recognition from arbitrary viewing angles.

Texture features such as edges are more descriptive but computationally demanding due to the high amounts of data and noise involved. Several systems therefore impose restrictions on the allowed input postures. In [8] a set of 26 postures is recognized in perfectly segmented single images of real hands taken from different viewing angles. An accuracy of 13.6% is reported, counting exact matches in posture and a maximum deviation of  $30^\circ$  in viewing angle. After generating a database of 107328 synthetic hand views (26 allowed postures seen from 86 viewing angles at 48 rotation angles), each including edges, lines extracted therefrom, and orientation histograms, the corresponding input features are used as a search key. Processing time per image is 15s on a 1.2GHz PC.

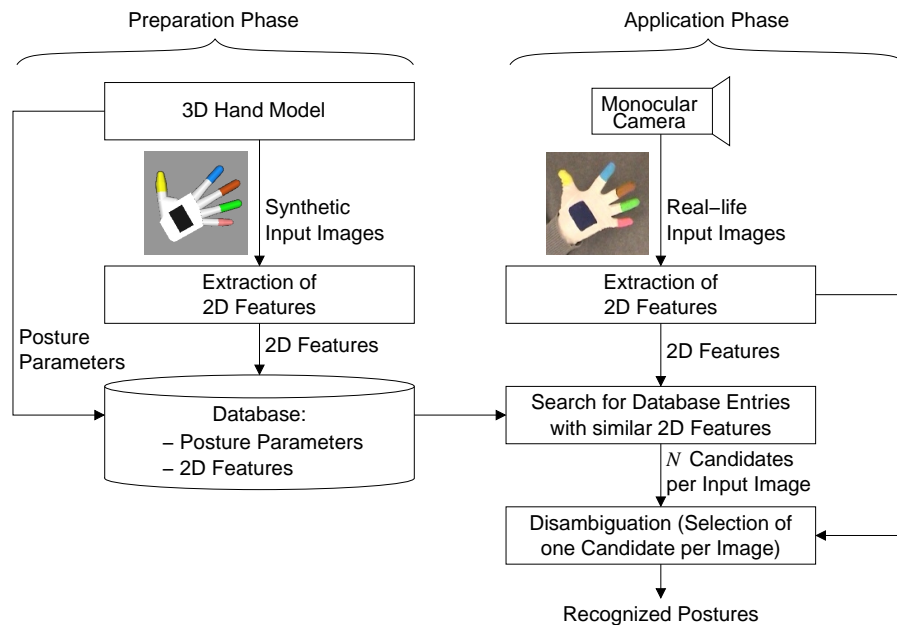
The method presented in [9] processes image sequences and allows hand postures commonly used in sign language, achieving an estimated person-dependent accuracy of 10% in finger flexion and  $15^\circ$  in viewing angle. The space of viewing angles and allowed postures is represented by 60 distinct Active Appearance Models (AAMs) that are extended to track translational motion. To narrow the search space per frame and thereby stabilize the system, transitions are only possible between models corresponding to anatomically similar postures. The AAM training data comprises manually labeled sequences of real images featuring a single person's hand. The hand must be sufficiently close to the camera to yield reliable texture information. Views from the finger

or wrist direction, which naturally exhibit less texture, are not supported. Processing speed is 4 fps on a 1GHz PC.

Several approaches limit the degree of self-occlusion in the input images [4, 5, 10–12], recognizing only a subset of all common hand postures. Methods that iteratively refine a state estimate typically require the hand to assume a specific initialization posture in the first frame [12, 13, 7]. Except for [6], all above systems are susceptible to registration errors since they only pursue a single posture estimate at a time, not accounting for ambiguities in feature space.

### 3 System Overview

Fig. 1 shows an overview of the system. The user wears a cotton glove equipped with six differently colored visual markers, five covering approx. half of each finger and the thumb, and another on the back of the hand. This allows to extract descriptive and stable 2D features from a monocular view. The markers' geometry lends itself to an elliptical approximation in the image plane, resulting in a very compact representation. Hand posture recognition is performed by matching a synthetic hand model featuring identical markers to minimize deviation in feature space.



**Fig. 1.** System overview

In a preparation phase the hand model is used to generate a large number of postures seen from many different view angles. Each posture, together with the corresponding 2D features extracted from the synthetic view, is stored in a database. For evaluation we used a database size of 2.6 million entries. This does not include rotation in the image plane, which is computed online.

Posture recognition is performed by using the 2D features extracted from the input images as a key for querying the database. For each frame a fixed number of  $N$  postures whose features have high similarity to the extracted features are retrieved. This candidate space is then searched for a sequence that maximizes continuity in both posture and feature space. Spline interpolation between successive frames, considering match quality in each, finally yields a smooth posture sequence not restricted to the discretized posture space of the database.

## 4 Hand Model

Regarding possible configurations of fingers and thumb, the human hand has 21 DOF [14]. Each finger possesses one DOF for each of its joints plus a fourth DOF for sidewise abduction. The thumb requires five DOF due to its greater flexibility. Our hand model reduces this to seven DOF by assuming dependencies between a finger’s joints. Fore to little finger are modeled by a single parameter each, ranging from 0.0 (fully outstretched) to 1.0 (maximum bending). The thumb is modeled similarly, using two additional parameters to reflect its flexibility. For a posture  $P$  the seven bending parameters are denoted by  $B^P$ .

Besides dealing with finger bendings the model also handles a posture’s viewing angle, i.e. the hand’s orientation in space. On the surface of an imaginary sphere around the hand (called the view sphere), each point corresponds to a specific view onto the hand. A view point is thus characterized by a latitude  $v_{\text{lat}}$  and a longitude  $v_{\text{lon}}$ . Additionally, for each view point a camera (or hand) rotation  $v_{\text{rot}}$  is possible. For a posture  $P$  these three angles are indicated by  $V^P$ .

In summary, the model parameters for each posture  $P$  comprise ten values and are denoted by  $P = \langle B^P, V^P \rangle$ . For a given posture  $P$  the corresponding synthetic hand image is rendered using OpenGL, modeling finger phalanges as simple cylinders, joints as spheres, and the palm as a combination of several polygons.

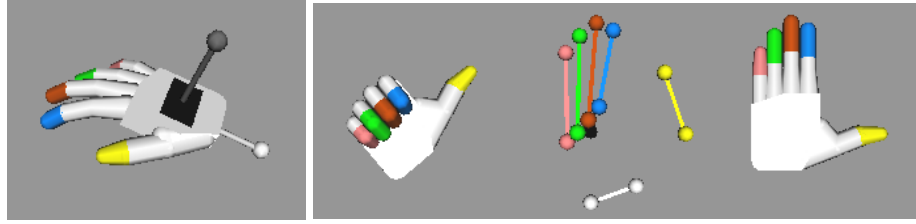
### 4.1 Posture Difference

Besides describing postures and visualizing them, the hand model offers an elegant way to express the difference between two postures. A mapping  $\Psi: \langle B^P, V^P \rangle \mapsto \langle \mathbf{c}_0^P, \dots, \mathbf{c}_6^P \rangle$  transforms the hand model parameters  $P$  to seven coordinates in space relative to the center of the palm.  $\mathbf{c}_0$  to  $\mathbf{c}_4$  represent the positions of the five finger tips.  $\mathbf{c}_5$  and  $\mathbf{c}_6$  are the coordinates of two additional “fingers” above the back of hand and below the wrist as shown in Fig. 2 (a), capturing the orientation of the hand, i.e. the view angles.

The difference  $\Delta_P$  between two postures  $Q$  and  $R$  is then defined as:

$$\Delta_P(Q, R) := \Psi(B^Q, V^Q) - \Psi(B^R, V^R) := \sum_{k=0}^6 |\mathbf{c}_k^Q - \mathbf{c}_k^R|^2 \quad (1)$$

where  $|\cdot|$  denotes the Euclidean distance. Fig. 2 (b) visualizes these distances.



(a) additional coordinates (b) two postures and their pairwise connected coordinates  
**Fig. 2.** Posture difference

## 5 2D Features

In every input frame color-based segmentation is performed to detect the markers. While this is trivial for synthetic images, real-life images may yield several candidates per marker. Disambiguation is performed by pursuing multiple hypotheses over time, computing plausibility scores based on the candidates' geometry and their continuity in feature space. The winner hypothesis is chosen only at the end of the sequence, exploiting all available information. A detailed description of multiple hypotheses tracking can be found in [1].

To reduce the effects of noise and to minimize memory requirements, each detected marker  $k$  is approximated by an ellipse  $E_k$ , specified by area  $f$ , center  $\mathbf{m}$ , radii  $a$  and  $b$ , and orientation  $o$ , i.e.:  $E_k = \langle f_k, \mathbf{m}_k, a_k, b_k, o_k \rangle$ .

For invisible markers  $f$  is zero, all other features are undefined. To achieve translation independence, all centers  $\mathbf{m}_k$  are specified relative to the center of gravity (COG) of all visible marker regions. Furthermore, area  $f_k$  is normalized by  $n_2 = \sum_k f_k$ . Distances and lengths are normalized by  $n_1 = \sqrt{n_2}$ . This provides independence of the distance between hand and camera, as well as camera resolution. Thus, for each input frame  $I$  the feature set  $F^I$  describing the six markers is given by  $F^I = \langle E_0^I, E_1^I, \dots, E_5^I \rangle$ .

## 6 Static Posture Recognition

This stage extracts a set of  $N$  plausible postures from the database for each input image, where  $N$  lies in the range of approx. 100 to 1000. A further high level stage will resolve ambiguities by considering all generated candidates for successive input images.

### 6.1 Appearance Database

A database entry contains the hand model parameters  $\langle B^P, V^P \rangle$  for a posture  $P$  along with the features  $F^P$  that were extracted from the corresponding synthetic image. The set  $B = \{B^P | P \in \text{DB}\}$  of all database postures' finger bending parameters was defined by selecting eight bending values for the thumb, seven for fore and middle finger each, plus six values each for ring and little finger, totaling 14,112 postures per view.

For the set  $V = \{V^P | P \in \text{DB}\}$  of all database postures' view points, angles in steps of  $18^\circ$  have been chosen. Special care has to be taken at the view sphere's poles, where a change of longitude resembles a rotation (an effect called gimbal lock), so  $v_{\text{lon}} = 0$  for

$v_{\text{lat}} = \pm 90$ . Rotation  $v_{\text{rot}}$  is set to zero for all database postures. Thus  $V$  contains these triples of  $\langle v_{\text{lat}}, v_{\text{lon}}, v_{\text{rot}} \rangle$ :

$$V = \{ \langle 18i, 18j, 0 \rangle \mid i, j \in \mathbb{Z} \wedge -5 < i < 5 \wedge 0 \leq j < 20 \} \cup \{ \langle \pm 90, 0, 0 \rangle \} \quad (2)$$

Because a rotation will leave the feature ellipses' areas as well as their relative distances unchanged,  $v_{\text{rot}}$  can be reconstructed before comparing database and input features. With  $|V| = 182$  the database contains  $D = 2,568,384$  entries. Considering rotations in steps of  $18^\circ$  a total of 51,367,680 postures can be recognized by the system.

In order to speed up database retrieval a  $B^*$ -like tree of height six with a fixed branching factor is used. Only  $f_k$  is considered, so the tree's depth equals the number of markers. The branching intervals at each node are non-overlapping, but if a query is within a certain range of an interval border, traversal continues in the neighboring subtree as well. The standard deviation of the interval's elements is used to quantify this range.

## 6.2 Feature Rotation

Let  $F^{\text{ex}}$  be the features extracted from the current input image and  $F^{\text{db}}$  those of a database candidate provided by the search tree. Like for all database entries,  $v_{\text{rot}}^{\text{db}} = 0$ . Let  $\phi(\mathbf{p}, \alpha)$  denote the rotation of point  $\mathbf{p}$  by  $\alpha$ . We define the rotation  $\hat{\alpha}(F^{\text{ex}}, F^{\text{db}})$  that estimates  $v_{\text{rot}}^{\text{db}}$  with respect to  $F^{\text{ex}}$  by

$$\hat{\alpha}(F^{\text{ex}}, F^{\text{db}}) := \underset{\alpha}{\operatorname{argmin}} \left\{ \sum_k f_k^{\text{db}} \cdot f_k^{\text{ex}} \cdot \left| \phi(\mathbf{m}_k^{\text{db}}, \alpha) - \mathbf{m}_k^{\text{ex}} \right|^2 \right\} \quad (3)$$

Due to weighting each summand by the product of the corresponding areas, bigger ellipses have more influence on the result than small ones.

Graphically, the two feature sets are aligned to their COGs and rotated until the sum of squared distances between corresponding ellipses, weighted by the product of their normalized area, is minimal. Since the rotation is actually a camera rotation, it propagates directly from features to postures. In the following,  $F^{\text{db}}$  is assumed to be the database features rotated according to (3).

## 6.3 Feature Difference

Searching the database for the  $N$  feature sets that are most similar to the extracted features  $F^{\text{ex}}$  requires to compute a scalar feature difference  $\Delta_F(F^{\text{ex}}, F^{\text{db}})$  that quantifies the similarity between  $F^{\text{ex}}$  and a set of database features  $F^{\text{db}}$  provided by the search tree. We use eight approximately equispaced points arranged counterclockwise on the ellipse's border (four of which lie at the intersection with the primary and secondary axes) and compute the sum of squared distances between these points for two corresponding ellipses  $E_k^{\text{ex}}$  and  $E_k^{\text{db}}$ . Of the eight possible mappings between both sets of points the one that minimizes this sum is used. This defines a geometric difference measure  $\Delta_E(E_k^{\text{ex}}, E_k^{\text{db}})$ .

For  $f_k^{\text{ex}} = f_k^{\text{db}} = 0$  we define  $\Delta_E = 0$ . If  $f_k^{\text{ex}} > 0 \wedge f_k^{\text{db}} = 0$  the affected marker in the database posture is made visible by not rendering any other component of the hand model (this is done offline). The now visible marker's COG is then used in place of the eight border points to compute  $\Delta_E$ . If  $f_k^{\text{ex}} = 0 \wedge f_k^{\text{db}} > 0$  the database is searched for a posture  $Q$  that differs from  $P^{\text{db}}$  only in the bending of the affected finger, and for which  $f_k$  is minimal or zero (again this happens offline).  $\Delta_E$  for marker  $k$  is then computed between  $Q$  and  $P^{\text{db}}$ . In general, if a marker's visibility differs between  $F^{\text{ex}}$  and  $F^{\text{db}}$ , the visible marker's area will be small since the search tree returns only candidates with  $f_k^{\text{db}} \approx f_k^{\text{ex}} \forall k$ .

In order to favor shape similarity over position congruence a weighting of  $\Delta_E$  by the difference of the ellipses' area is performed. The feature difference is thus defined as

$$\Delta_F(F^{\text{ex}}, F^{\text{db}}) := \sum_k \Delta_E(E_k^{\text{ex}}, E_k^{\text{db}}) \cdot (1 + |f_k^{\text{ex}} - f_k^{\text{db}}|) \quad (4)$$

When querying the database the search tree returns  $M$  entries, where  $N \ll M \ll D$ .  $\Delta_F$  is computed for all  $M$  entries to find the  $N$  that best match  $F^{\text{ex}}$ , which form the set of hypotheses for the considered frame.

## 7 Posture Sequence Recognition

The recognition of posture sequences is based upon the hypotheses provided by the static recognition stage described in the previous section. It computes the actual recognition result and can also be used to optimize system parameters by synthetically generating input data for which ground truth is known. Fig. 3 shows a schematic overview.

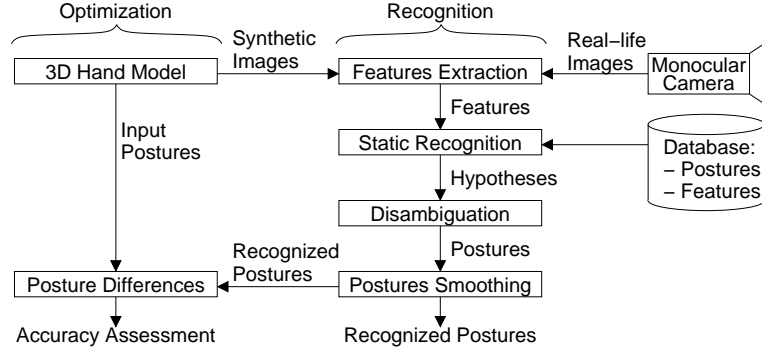


Fig. 3. Posture sequence recognition and parameter optimization

### 7.1 Disambiguation

For a sequence  $I(0), \dots, I(T-1)$  of input images, the corresponding features  $F^{\text{ex}}(0), \dots, F^{\text{ex}}(T-1)$  are extracted, for which the static recognition stage generates the sets of hypotheses  $H(0), \dots, H(T-1)$ , each containing  $N$  possible database features and postures, i.e.  $H(t) = \langle h_0(t), \dots, h_{N-1}(t) \rangle$  with  $h_n(t) = \langle F_n^{\text{db}}(t), P_n^{\text{db}}(t) \rangle$  for  $t = 0, \dots, T-1$  and  $n = 0, \dots, N-1$ .

At 25 fps,  $T = 75$  for a three second input sequence. We use  $N = 200$ , resulting in  $N^T \approx 3.8 \cdot 10^{172}$  posture sequence in hypothesis space, and apply the Viterbi algorithm, which scales with  $O(TN^2)$ . The step metric  $\sigma(h_i(t-1), h_j(t))$ , which represents the cost of choosing hypothesis  $h_j(t)$  given that  $h_i(t-1)$  is its predecessor, is defined as

$$\sigma(h_i(t-1), h_j(t)) := (1 - \gamma) \cdot \max \left\{ \Delta_P \left( P_i^{\text{db}}(t-1), P_j^{\text{db}}(t) \right) - \Delta_*, 0 \right\} + \gamma \cdot \Delta_F \left( F^{\text{ex}}(t), F_j^{\text{db}}(t) \right) \quad \text{for } t = 1, \dots, T-1 \quad (5)$$

The step metric combines the feature difference  $\Delta_F$  between a hypothesis and the corresponding extracted features and the posture difference  $\Delta_P$  between successive hypotheses, thus favoring a high feature similarity as well as temporal continuity of postures.  $\Delta_*$  is a small, positive, empirically determined value subtracted from  $\Delta_P$  in order to allow small posture deviations.  $\gamma$  weights the feature and posture differences and was optimized iteratively using sequences of synthetic images provided by the hand model (cf. left part of Fig. 3).

The path metric  $\pi(h_i(t))$  is the sum of step metrics along the path leading to  $h_i(t)$  and is initialized to

$$\pi(h_i(0)) := \gamma \cdot \Delta_F(F^{\text{ex}}(0), F_i^{\text{db}}(0)) \quad (6)$$

The path that minimizes  $\pi(h)$  for  $h \in H(T-1)$  constitutes the resulting posture sequence  $S(I(0), \dots, I(T-1)) = \langle h(0), \dots, h(T-1) \rangle$ . Since this is only computed after several seconds of input video have been observed, posture errors in individual frames are retrospectively corrected as soon as they become apparent in the light of additional observations.

## 7.2 Smoothing

The sequences provided by the disambiguation stage only contain postures from the database, i.e. from a finite, discrete subset of the infinite space of continuous postures. In order to bridge these jumps and to alleviate small recognition errors, the sequence is smoothed, which happens individually for each bending and view angle parameter.

Recall that  $h(t) = \langle F^{\text{db}}(t), P^{\text{db}}(t) \rangle$  and  $P^{\text{db}}(t) = \langle B^{P^{\text{db}}(t)}, V^{P^{\text{db}}(t)} \rangle$ . Let  $r_0, \dots, r_{R-1}$  be the indices of the most reliable hypotheses from  $\langle h(0), \dots, h(T-1) \rangle$ . Starting with  $r_0 = 0$  the next most reliable hypothesis for  $h(r_i)$  is  $h(r_{i+1}) \in \{h(r_i + \alpha), \dots, h(r_i + \beta)\}$  with minimal  $\Delta_F(F^{\text{ex}}(r_{i+1}), F^{\text{db}}(r_{i+1}))$ . Appropriate values are  $\alpha = 1$  and  $\beta = 4$ .

Let  $\rho(t)$  denote a hand model parameter in  $P^{\text{db}}(t)$ . For  $\{\langle r_i, \rho(r_i) \rangle | i = 0, \dots, R-1\}$  the system computes the interpolating cubic spline  $s(t)$ , i.e.  $s(r_i) = \rho(r_i)$  for  $i = 0, \dots, R-1$ . The smoothed sequence of hand model parameters is then given by  $s(0), \dots, s(T-1)$ . Performing this interpolation individually for each hand model parameter yields the smoothed sequence of postures.

## 8 Evaluation

We evaluated the system's performance on synthetic and real-life images. Using a standard PC with a 1.6 GHz CPU and 1.25 GB RAM, processing speed is approx. 5 fps.

## 8.1 Synthetic Input

Synthetic input images offer the opportunity to measure recognition precision quantitatively. We generated 500 random sequences that evenly cover the posture space, each consisting of 75 consecutive postures featuring continuous changes in both  $B^P$  and  $V^P$ . The corresponding images have been distorted by blanking  $3 \times 3$  tiles overlapping by 1 pixel with a probability  $p_{\text{noise}}$  as illustrated in Fig. 4. The results are listed

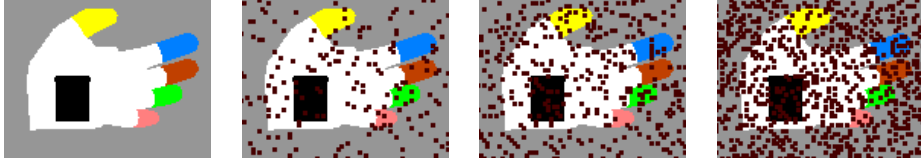


Fig. 4. Noise levels  $p_{\text{noise}} = 0\%, 5\%, 10\%, 20\%$  (left to right; image resolution  $136 \times 105$ )

in Tab. 1, where “Posture Difference” refers to (1), “Fingertip Distance” denotes the average Euclidean distance (in cm) of corresponding fingertips without consideration of view angles, and “Finger Bending Deviation” refers to the posture parameters  $B$  (cf. Sec. 4). These results demonstrate that the system is robust to significant noise.

Table 1. Recognition accuracy for synthetic input

$p_{\text{noise}}$	Posture Difference		Fingertip Distance		Median of Finger Bending Deviation						
	average	median	average	median	th1	th2	th3	ff	mf	rf	lf
0%	111.542	74.642	1.940	1.782	0.227	0.257	0.233	0.075	0.077	0.087	0.099
5%	108.506	75.912	1.952	1.795	0.224	0.255	0.230	0.080	0.077	0.084	0.097
10%	112.841	76.600	1.965	1.815	0.228	0.262	0.248	0.074	0.077	0.087	0.100
20%	144.636	97.338	2.272	2.102	0.241	0.242	0.257	0.105	0.096	0.100	0.134

## 8.2 Real-life Input

Real-life performance has been tested on sequences of signed numbers. Fig. 5 depicts some examples, each showing a magnification ( $220 \times 145$ ) of the actual input image ( $360 \times 288$ ) and the recognized posture. By visual comparison match quality is high. Fig. 5 (d) illustrates the system’s reaction to marker detection failures.

## 9 Discussion and Conclusion

We have presented a method that recognizes hand postures (finger bendings and view angle) from monocular image sequences. It imposes no posture restrictions and requires no initialization or person-dependent training. Performing appearance-based matching by searching a database, coupled with a Viterbi search in posture space, provides an efficient means of handling ambiguities.

Our experiments show promising results. Future work will primarily focus on processing speed, which can be increased by removing anatomically impossible postures and views from the database, and by improving the search tree’s efficiency.

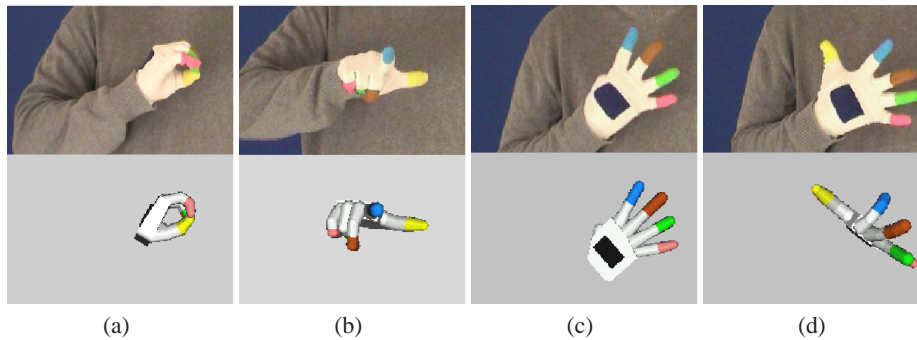


Fig. 5. Real-life examples (for (d) the back-of-hand marker has been removed manually)

## References

1. Zieren, J., Kraiss, K.F.: Robust Person-Independent Visual Sign Language Recognition. In: *IbPRIA 2005. Volume Lecture Notes in Computer Science*. (2005)
2. Zieren, J.: Hand Gesture Commands. In: *Advanced Man-Machine Interaction*. Springer (2006) 7–56
3. Bebis, G., Harris, F., Erol, A., Yi, B., Martinez, J., Hernandez-Usabiaga, J., Fritzinger, S.: Development of a Nationally Competitive Program in Computer Vision Technologies for Effective Human-Computer Interaction in Virtual Environments. Technical report, BioVIS Lab. in BioVIS Technology Center of NASA Ames Research Center (2002)
4. Nölker, C.: Grefit: Ein System zur Visuellen Erkennung von Handposturen. PhD thesis, Technische Fakultät der Universität Bielefeld (2000)
5. Rehg, J.M.: Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking. PhD thesis, School of Computer Science, Carnegie Mellon University (1995)
6. Imai, A., Shimada, N., Shirai, Y.: 3-D Hand Posture Recognition by Training Contour Variation. In: *International Conference on Automatic Face and Gesture Recognition*. (2004)
7. Vittrup, M., Sørensen, M.K.D., McCane, B.: Pose Estimation by Applied Numerical Techniques. In: *Image and Vision Computing, New Zealand*. (2002)
8. Athitsos, V., Sclaroff, S.: Estimating 3D Hand Pose from a Cluttered Image. In: *Proc. IEEE CVPR*. (2003)
9. Fillbrandt, H., Akyol, S., Kraiss, K.F.: Extraction of 3D Hand Shape and Posture from Image Sequences for Sign Language Recognition. In Azada, D., ed.: *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*. (2003)
10. Chua, C.S., Guan, H., Ho, Y.K.: Model-based 3D hand posture estimation from a single 2D image. *Image Vision Comput.* **20**(3) (2002)
11. Lathuilière, F., Hervé, J.Y.: Visual Tracking of Hand Posture in a Robot Control Application. In: *Proceedings of the Vision Interface Conference*. (1999)
12. Heap, T., Hogg, D.: Towards 3D Hand Tracking using a Deformable Model. In: *International Conference on Automatic Face and Gesture Recognition*. (1996)
13. Holden, E.J., Owens, R., Roy, G.G.: 3D Hand Tracker for Visual Sign Recognition. (1999)
14. Sturman, D.J.: Whole-hand Input. PhD thesis, School of Architecture and Planning, Massachusetts Institute of Technology (1992)