Three-Dimensional Shape and Motion Reconstruction for the Analysis of American Sign Language

Liya Ding and Aleix M. Martinez
Dept. of Electrical and Computer Engineering
The Ohio State University
{dingl,aleix}@ece.osu.edu

Abstract

The design of systems that can analyze and recognize American Sign Language (ASL) sentences, require that motion patterns and handshapes be independently represented from one another. In this paper, we define an algorithm that first obtains the 3D shape of each of the linguistically significant fingers (known as "selected" fingers in the linguistic literature), and then represents the motion of the hand as a three-dimensional trajectory in the coordinate system of the camera. In doing so, we address the problems caused by self-occlusions and localization errors. The 3D shape is obtained using an affine structure from motion algorithm that is based on the linear fitting of matrices with missing data. To recover the 3D motion of the hand, we define a robust algorithm that selects the most stable solution from the pool of all the solutions given by the three point resection problem. To be robust to localization errors (of the hand feature points tracked) we define a simple algorithm that searches for that combination of coordinates that gives the largest number of stable solutions in our system of equations. To validate our theory, we provide a set of experimental results using video sequences of ASL signs.

1. Introduction

Current models of American Sign Language (ASL), which are used to study ASL and to build algorithms for its automatic recognition, require that handshapes and motion patterns be described in different representations [2, 17, 3]. Each of these two representations will then be used, for example, to build a classifier for the recognition of ASL words, or to determine which features convey meaning in an ASL sentence. This will ultimately help linguist to better understand the underlying components of the language and provide a large number of applications for human-computer interfaces (HCI) for the deaf and in the classroom.

It is shown, by the above mentioned models of ASL, that only a set of linguistically significant fingers are necessary for the full understanding of each word. For example, Brentari [2] has shown that if the 3D shape of these linguistically significant fingers is known, then it is possible to distinguish among all significant ASL handshapes. These fingers are referred to as "selected fingers" in the literature. Being able to recover the 3D shape of these selected fingers from 2D video sequences is a necessary milestone to develop efficient HCI systems.

Similarly, to properly represent ASL motions, one needs to describe the 3D trajectory of the *dominant hand* [3, 2] with respect to time. This will allow the study of (3D) components of the language, such as pose, velocity and acceleration patterns. This is important because these are also known to contain meaningful information [17].

To date, most computer vision research has focused on the problem of hand tracking and ASL recognition of isolated words [12, 1, 7]. In these algorithms, the discriminant information is generally searched within a feature space constructed with appearance-based features such as images of pre-segmented hands [1], hand binary masks, and hand contours [15]. The other most typically used feature set is motion [18]. Then, for recognition, we can use Hidden Markov Models [15], Neuron Network [18] and Multiple Discriminant Analysis [1].

Opposed to the classical approach defined in the preceding paragraph, we propose to develop a system consistent with the linguistic studies summarized at the beginning of this section. In such a scenario, one needs to first recover the 3D shape and 3D motion trajectory of each sign and then use these to determine the meaning. For example, the two signs shown in Fig.1(a-b) correspond to "never" and "straight". Here, we note that both signs have a common handshape and similar start and end placement. However, they have distinct 3D trajectories, which are what determine the meaning. Another example is given in Fig.1(c-d), where we illustrate the ASL signs for "family" and "class" the motion is common to both signs while the handshape is the one used to distinguish them. In other scenarios, mo-

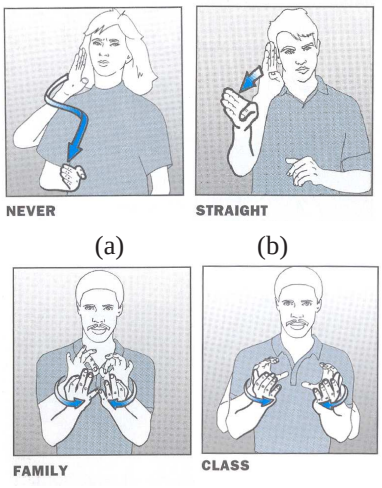


Figure 1. Example signs with same handshape different motions (a-b); and signs with same motion different handshapes (c-d). Illustration figures are from [8], ©Pearson Education 2004, used with permission.

tion patterns can be used to specify the direction of action, duration or verb tense [13, 10, 12]. Hence, recovering the motion trajectory will provide much more information than that recovered by the algorithms summarized in the preceding paragraph where one is limited to the recognition of isolated words.

A first step for the development of a HCI system as defined above is to design a robust algorithm that can recover the 3D handshape and motion trajectory of each sign. This is exactly the goal of this paper. In our study, we allow for self-occlusions and imprecise localization of the fiducial points to occur. Self-occlusions occur when one of the fingers gets occluded by the hand or other fingers. As the hand rotates with respect to the camera, some fingers will become occluded and others (that were previously occluded) will become visible for the first time. Self-occlusions may also be caused by the opposite hand or the arm. Here, we restrict ourselves to the case where the handshape does not change from start to end, because this represents most of the words of the dictionary [14] and allows us to use linear fitting algorithms. Derivations of our method are in Section 2.

To properly recover the 3D motion path of the hand, we define an algorithm that uses Grunert's solution to the three point resection problem [4, 5]. Because Grunert's solution is very sensitive to miss-localized feature points, we introduced a robustified version which searches for the most stable solution within the set of solutions given by those points around the actual fiducial. Derivations for this method are in Section 3. Our results (Section 4) show that the method presented in this paper is robust to self-occlusions and imprecise localizations.

2. Structure From Motion

As argued above, a large number of ASL signs consist of a handshape that does not vary from start to end. In this section we define an algorithm able to extract the relative 3D shape for each of the selected fingers by means of an affine structure from motion algorithms.

We denote the set of all 3D-world hand points of the "selected" fingers (e.g. knuckles) as $\mathbf{P}_e = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$, where $\mathbf{p}_i = (x_i, y_i, z_i)^T$ specifies the three dimensional coordinates of the i^{th} feature point in the Euclidean coordinate system. As it is well-known, the image points in camera j are given by

$$\mathbf{Q}_j = \mathbf{A}_j \mathbf{P}_e + \mathbf{b}_j, \quad j = 1, 2, \cdots, m, \tag{1}$$

where

$$\mathbf{Q}_j = (\mathbf{q}_{j1}, \dots, \mathbf{q}_{jn}) = \begin{pmatrix} u_{j1} \dots u_{jn} \\ v_{j1} \dots v_{jn} \end{pmatrix} \tag{2}$$

are the image points, and $\mathbf{A}_j$ and $\mathbf{b}_j$ are the parameters of the j^{th} affine camera [6].

Since in our application the camera position does not change, the above equation has to be reinterpreted. In our case, $\mathbf{Q}_j$ are the image points in the j^{th} image of our video sequence. Our goal is to recover $\mathbf{P}_e$ with regard to the object (i.e. hand) coordinate system from known $\mathbf{Q}_j, j = 1, 2, ..., m$.

Tomasi and Kanade showed that if we use the center of mass of the object points as the origin of the object coordinate system and the center of the image points in the $j^{th} (j = 1, 2, ..., m)$ image $(\mathbf{q}_c^j)$ as the origin of the image coordinate system in the j^{th} frame, then we can recover both, the camera parameters and 3D shape of the object with a factorization method [16].

In practical applications, there are inevitable occlusion of hand points, e.g., occlusion of one hand by the other hand, and mostly self-occlusion by the same hand. With missing data, one cannot find the center of the image points and, therefore, Tomasi and Kanade's algorithm cannot be applied. For this reason, we will now present an extension of Jacobs' method presented in [9].

Let us first represent the set of equations in (1) in a compact form as

$$\mathbf{D} = \mathbf{A}\mathbf{P}, \tag{3}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \vdots \\ \mathbf{Q}_m \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{A}_2 & \mathbf{b}_2 \\ \vdots & \vdots \\ \mathbf{A}_m & \mathbf{b}_m \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \\ 1 & 1 & \dots & 1 \end{bmatrix}.$$

When there is neither noise nor missing data, **D** is of rank 4 or less, since it is the product of **A** (which has 4 columns) and **P** (4 rows). If we consider a row vector of **D** as a point in the $\mathbb{R}^n$ space, all the points from **D** lie in a 4-dimensional subspace of $\mathbb{R}^n$. This subspace, which is actually the row space of **D**, is denoted as **L**. Any four linear independent rows of **D** should span **L**.

When there is missing data in a row vector $\mathbf{D}_i$ $(i = 1, 2, ..., 2m)$, all possible values (that can occupy this position) have to be considered. The possible points in this row vector create an affine subspace denoted $\mathbf{E}_i$. Assume we have four rows $\mathbf{D}_h, \mathbf{D}_i, \mathbf{D}_j, \mathbf{D}_l$ $(h, i, j, l = 1, 2, ..., 2m$, with $h \neq i \neq j \neq l)$ with or without missing data, and denote the set as

$$\mathbf{F}_k = \{\mathbf{D}_h, \mathbf{D}_i, \mathbf{D}_j, \mathbf{D}_l\}, k \in \mathbb{N}. \tag{4}$$

And, there is a total of n_f $(n_f \in \mathbb{N})$ possible $\mathbf{F}_k$.

If the four affine subspaces $(\mathbf{E}_h, \mathbf{E}_i, \mathbf{E}_j, \mathbf{E}_l)$ corresponding to these four row vectors in $\mathbf{F}_k$ don't intersect, then **L** should be a subset of

$$\mathbf{S}_k = \text{span}(\mathbf{E}_h, \mathbf{E}_i, \mathbf{E}_j, \mathbf{E}_l), k = 1, 2, ..., n_f. \tag{5}$$

And, thus, **L** should be a subset of the intersection of all possible span of this kind. Hence,

$$\mathbf{S} = \bigcap_{k=1,2,...,n_f} \mathbf{S}_k, \text{ and } \mathbf{L} \subseteq \mathbf{S}. \tag{6}$$

Unfortunately, with localization noise and errors caused by inaccurate modelling, this relation of subsets is not retained. Jacobs uses the null-space method to solve this problem [9]. Here, the orthogonal complement of $\mathbf{S}_k$ is denoted as $\mathbf{S}_k^\perp$. If we have the matrix representation of $\mathbf{S}_k^\perp$ as $\mathbf{N}_k$, then

$$\mathbf{N} = [\mathbf{N}_1 \mathbf{N}_2 ... \mathbf{N}_{n_f}] \tag{7}$$

is the matrix representation of $\mathbf{S}^\perp$. And the nullspace of **N** is **S**. Using SVD of $\mathbf{N} = \mathbf{U}\mathbf{W}\mathbf{V}^T$, we can take the four columns in **U** according to the four smallest singular values as the four rows in **P**. Next, we can find the matrix representation **P'** of the subspace **L** that is closest to being its null-space according to the Frobenius norm. In this case, note that one of the vectors spanning **D**'s row space **L** is known to be a vector with all 1s (because in homogeneous form, **P** has a row with 1s).

If an image point is missing (its (u, v) coordinates are missing), taking the two rows corresponding to the same image as in $\mathbf{F}_k$ will prove beneficial for calculating $\mathbf{N}_k$. Then, to calculate $\mathbf{N}_k$ from $\mathbf{S}_k$, we take

$$\mathbf{F}_k = \{\mathbf{D}_{2i-1}, \mathbf{D}_{2i}, \mathbf{D}_{2j-1}, \mathbf{D}_{2j}, \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}\}, i, j = 1, 2, ..., m, i \neq j \tag{8}$$

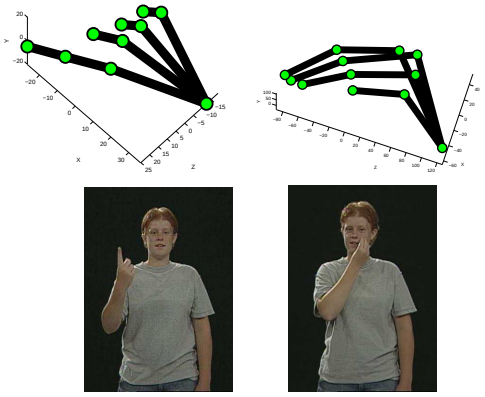


Figure 2. 3D handshape reconstructed from the 2D image points of the knuckles (manually) tracked over a video sequence.

for better stability. And we eliminate the vectors with poor condition in calculation of nullspaces $\mathbf{N}_i, i = 1, .., n_f$ which combine into a more stable solution of **N**. This generally improves the performance of the above defined algorithm in practice.

If a column in **D** has a large number of missing data, we may very well be unable to recover any 3-dimensional information from it. Note, however, that this can only happen when one of the fingers is occluded during the entire sequence, in which case, we will assume that this finger is not linguistically significant. This means we do not need to recover its 3D shape, because this will be irrelevant for the analysis and understanding of the sign.

When **P** has been properly estimated, we can use the non-missing data of each row in **D** to fill in the missing gaps as a linear combination of the rows of **P**. At the same time, we have decomposed the filled $\hat{\mathbf{D}}$ into **P** and $\mathbf{A} = \hat{\mathbf{D}}(\mathbf{P})^+$.

The above result **P** generates what is known as an *affine* shape. Two affine shapes are equivalent if there exists an affine transformation between them. To break this ambiguity, we can include the Euclidean constraints to find that *Euclidean* shape that best approximates the real one. One way to achieve this, is to find a matrix **H** such that $\mathbf{M}_j\mathbf{H}(j = 1, 2, ..., n)$ is orthographic, where $\mathbf{M}_j = (\mathbf{A}_j \ \mathbf{b}_j)$. This is so, because orthographic projections do not carry the shape ambiguity mentioned above. The Euclidean shape can be recovered using the Cholesy decomposition and non-linear optimization [6]. Fig. 2 shows a couple of results of the recovery of the 3D shapes of the selected fingers.

3. Motion Reconstruction

A common way to recover the 3D motion path of an object is by finding the pose difference between each pair of consecutive frames. That means, we need to estimate the translation and rotation made by the object from frame to

frame using the camera coordinate system. A typical solution to this problem is that given by the PNP resection approach [5].

In the three point perspective pose estimation problem, there are three object points, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, with camera coordinates $\mathbf{p}_i = (x_i, y_i, z_i)^T$. Our goal is to recover these values for each of the points. Since the 3D shape of the object has already been recovered, the interpoint distances (i.e., namely $a = \|\mathbf{p}_2 - \mathbf{p}_1\|$, $b = \|\mathbf{p}_1 - \mathbf{p}_3\|$, and $c = \|\mathbf{p}_1 - \mathbf{p}_2\|$) can be easily calculated.

As it is well-known, the perspective model is given by

$$\left\{ \begin{array}{rcl} u_i & = & f\frac{x_i}{z_i} \\ v_i & = & f\frac{y_i}{z_i} \end{array} \right. , \quad i = 1, 2, 3\,, \tag{9}$$

where $\mathbf{q}_i = (u_i, v_i)^T$ is the i^{th} image points. And, of course, the object is in the direction specified by the following unit vector

$$\tilde{j}_i = \frac{1}{\sqrt{u_i^2 + v_i^2 + f^2}} \begin{pmatrix} u_i \\ v_i \\ f \end{pmatrix}, \quad i = 1, 2, 3\,.$$

Now, the task reduces to finding those scalers, s_1, s_2 and s_3, such that

$$\mathbf{p}_i = s_i \tilde{j}_i, \quad i = 1, 2, 3\,.$$

The angles between these unit vectors can be calculated as

$$\cos\alpha = \tilde{j}_2 \cdot \tilde{j}_3,\ \cos\beta = \tilde{j}_1 \cdot \tilde{j}_3,\ \cos\gamma = \tilde{j}_1 \cdot \tilde{j}_2. \tag{10}$$

Grunert's approach is based on the assumption that $s_2 = \mu s_1$ and $s_3 = \nu s_1$, which allows us to reduce the three point resection problem to a fourth order polynomial of ν:

$$A_4\nu^4 + A_3\nu^3 + A_2\nu^2 + A_1\nu + A_0 = 0, \tag{11}$$

where the coefficients A_4,A_3, A_2, A_1 and A_0 are functions of interpoint distances a, b, c and the angles α, β, γ between $\tilde{j}_i$ [4, 5].

Such polynomials are known to have zero, two or four real roots. With each ν's real root, we can calculate μ, s_1, s_2, s_3 and the values of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. To recover the translation and rotation of the hand points, we use the nine equations given by

$$\mathbf{p}_i = \mathbf{R}\ {}^w\mathbf{p}_i + \mathbf{t}, \quad i = 1, 2, 3\,, \tag{12}$$

where ${}^w\mathbf{p}_i$ is the hand point as described in the world coordinate system, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vectors we want to recover. Obviously, the nine entries of the rotation matrix are not independent. To reduce the 12 dependent parameters into 9 and solve from the nine equations, we use a two-step method.

1) Move the triangle formed by the three object points onto the x-y plane of the camera coordinate system: Calculate the norm of the triangle from which we calculate the rotating $\mathbf{R}_1$ – the triangle onto a plane parallel to the x-y plane. By a simple translation, we can move this triangle onto the x-y plane. After this transformation

$$\mathbf{p'}_i = [x'_i, y'_i, z'_i]^T = \mathbf{R_1}\ {}^w\mathbf{p}_i + \mathbf{t_1}, \quad i = 1, 2, 3. \tag{13}$$

Here, we have $z'_i = 0, \quad i = 1, 2, 3.$

2) Use linear methods to solve the system of equations: Substituting ${}^w\mathbf{p}_i$ with $\mathbf{p'}_i$ in Eq. (12), we will need to solve $\mathbf{R}_2$ and $\mathbf{t}_2$ such that:

$$\mathbf{p}_i = \mathbf{R_2}\ \mathbf{p'}_i + \mathbf{t_2}, \quad i = 1, 2, 3. \tag{14}$$

If we denote the columns of $\mathbf{R_2}$ as $\mathbf{r}_1$,$\mathbf{r}_2$ and $\mathbf{r}_3$, then $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. Furthermore, since $z'_i = 0$ for all $i = 1, 2, 3$, the three entries in $\mathbf{r}_3$ are actually not in the equations. Therefore, this becomes an easy to solve linear system of 9 unknowns and 9 equations, followed by a simple cross product to obtain the final solution. Finally, we have

$$\mathbf{p}_i = \mathbf{R_2}\ (\mathbf{R_1}\ {}^w\mathbf{p}_i + \mathbf{t_1}) + \mathbf{t_2}, \quad i = 1, 2, 3, \tag{15}$$

and

$$R = \mathbf{R_2}\ \mathbf{R_1}, \mathbf{t} = [t_x\ t_y\ t_z]^T = \mathbf{R_1 t_1} + \mathbf{t_2}. \tag{16}$$

Furthermore, we can parameterize the rotation matrix into three rotational angles r_x,r_y and r_z satisfying

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos r_x & \sin r_x \\ 0 & -\sin r_x & \cos r_x \end{pmatrix} \begin{pmatrix} \cos r_y & 0 & -\sin r_y \\ 0 & 1 & 0 \\ \sin r_y & 0 & \cos r_y \end{pmatrix} \begin{pmatrix} \cos r_z & \sin r_z & 0 \\ -\sin r_z & \cos r_z & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which means the object is first rotated about the z-axis (in a clockwise direction when looking towards the origin) for angle r_z, then about the y-axis (r_y) and finally about the x-axis (r_x). This means that a solution from each group of three points consists of six parameters,

$$r_x, r_y, r_z, t_x, t_y, t_z.$$

Since we regard the hand as a rigid object during the motion, the rotation and translation of any three-point group are identical. The polynomial defined in (11) may have more than one roots. Unfortunately, in general, it is not known which of these roots corresponds to the solution of our problem. To solve this, we calculate all the solutions given by all possible combinations of three feature points, and describe them in a histogram. This allows us to select the result with highest occurrence (i.e., with most votes) as our solution.

It is also known that the geometric approaches to the three point resection problem are very sensitive to errors of localization. We now define an approach to address this issue.

Here, we assume that the correct localization is close to that given by the user or any automatic tracking algorithm. We generate a set of candidate hand points by moving the original position of the original fiducial about a neighborhood of $p \times p$-pixel window. The solutions for each of the Grunert's polynomials are then used to obtain all possible values for r_x, r_y, r_z, t_x, t_y and t_z. Each of these results is described in a histogram and the result interval I_0 with most votes is checked out. A wider interval centered at I_0 is then chosen. The median of the results within this new interval will correspond to our final solution. Note that voting was first used to eliminate the outliers from the solutions of Grunert's polynomials and, hence, our method is not effected by large deviations of the results. The median is used to select the best result among the correct solutions from different image point localizations.

To show that our method can cope better with localization errors than the classical algorithm proposed by Grunert, we now provide some statistics. For this purpose, we constructed synthetic handshapes with known motion parameters. Noise evenly distributed over $[-n, n]$ (in pixels) with $n = 1, 2, 3, 4$ are added to the image points locations in both horizontal and vertical axis. Error rates of our method and the method of taking the mean of solutions of Gurnerts's algorithm are given in Fig. 3.

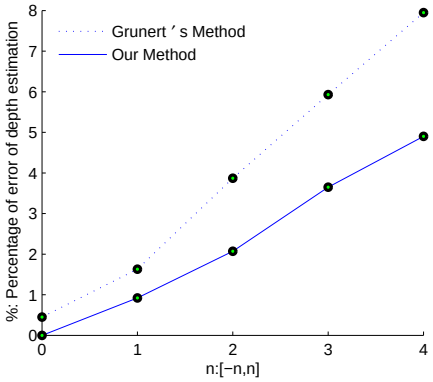


Figure 3. Shown here are the error curves corresponding to the original Grunert's algorithm and the one presented in this paper.

4. Experimental Results

In this section we will test the performance of our algorithm presented above. We are especially interested in testing how our approach addresses the problems caused by self-occlusions and imprecise localizations of the knuckles.

Our first test uses a set of 3D shapes, randomly generated, with associated 3D random motion for a total of twenty frames per shape. Each of these frames are then mapped to a 2D image using perspective projection. To simulate occlusions, twenty percent of the points are randomly selected and assigned a zero value. Moreover, random noise is added to the rest of the points. The reconstruction given by our algorithm is then compared to the original shape. Error rates are in Fig. 4.

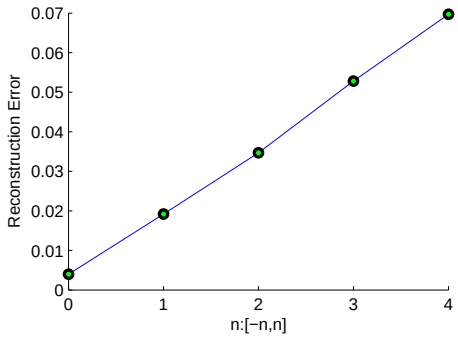


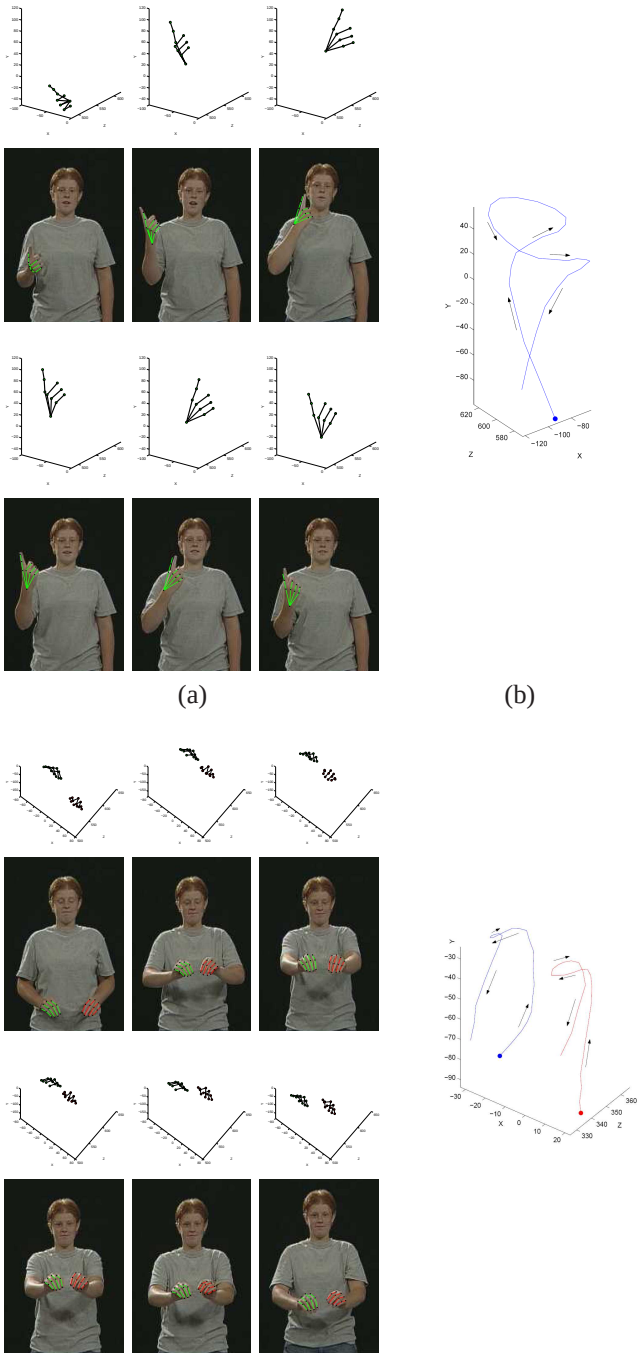
Figure 4. Average error of shape reconstruction as a function of evenly distributed random noise over [-n, n] (in pixels).

We also show some results using video sequences from the Purdue ASL Database [11]. This database contains 2576 video sequences. The first part of this includes a large number of video clips of motion primitives and handshapes (here handshape is invariant during each motion primitive). These are used to test the performance of our algorithm.

In general it is very difficult to track the knuckles of the hand automatically, mainly due to the inevitable self-occlusion and lack of salient characteristics present in the video sequences. Because our goal is to test the robustness of the algorithm presented above, we opted for a manual detection of the image fiducials. In Fig. 5, we show two sequences of images and the corresponding projection of the reconstructed handshapes. The 3D handshapes obtained (in the camera's coordinate system) are shown above each image. In addition, we provide the 3D trajectory recovered by the robust method presented in this paper.

5. Conclusions

This paper describes a set of algorithms that can be used to recover the three-dimensional shape and motion trajectory of the hand from two-dimensional video sequences of American Sign Language words. We showed that one can obtain the shape of the linguistically significant fingers using an affine structure from motion algorithm with linear fitting of low rank matrices with missing data. We have also proposed an algorithm to recover the three-dimensional trajectory of the motion of the hand. This method was robustified to be insensitive to spatial noise. A set of results using synthetic and real data have been presented.



(a) (b) (c) (d)

Figure 5. Reconstruction of the 3D handshape and hand trajectory. (a) and (c) show the 3D handshapes recovered by our method. (b) and (d) show the corresponding 3D trajectories.

6. Acknowledgement

This research was supported in part by the National Institutes of Health under grant R01 DC 005241.

References

[1] Y. Cui and J. Weng, "Appearance-Based Hand Sign Recognition from Intensity Image Sequences," Computer Vision Image Understanding, vol. 78, no. 2, pp. 157-176, 2000. 1

[2] D. Brentari, "A prosodic model of sign language phonology," MIT Press, 2000. 1

[3] K. Emmorey, and J. Reilly (Eds.), "Language, gesture, and space," Hillsdale, N.J.:Lawrence Erlbaum, 1999. 1

[4] J.A. Grunert, "Das Pothenotische Problem in erweiterter Gestalt nebst Über seine Anwendungen in der Geodäsie" *Grunerts Archiv für Mathematik und Physik*, Band 1, pp. 238-248, 1841. 2, 4

[5] R.M. Haralick, C. Lee, K. Ottenberg, M. Nolle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem" International Journal of Computer Vision 13, 3, pp. 331-356, 1994. 2, 4

[6] R. Hartley and A. Zisserman, "Multiple View Geometry in computer vision" Second Edition, Cambridge University Press, 2003. 2, 3

[7] E.-J. Holden and R. Owens, "Visual Sign Language Recognition," Proc. International Workshop Theoretical Foundations of Computer Vision, pp. 270-287, 2000. 1

[8] T. Humphries and C. Padden, "Learning American Sign Language" Second Edition, Pearson Education, 2004. 2

[9] D.W. Jacobs "Linear Fitting with Missing Data for Structure-from-Motion" Proc. IEEE Computer Vision and Pattern Recognition pp. 206-212, 1997. 2, 3

[10] E.S. Klima and U. Bellugi, "The Signs of Language," Harvard Univ. Press, 1979. 2

[11] A.M. Martinez, R.B. Wilbur, R. Shay and A.C. Kak, "The Purdue ASL Database for the Recognition of American Sign Language," In Proc. IEEE Multimodal Interfaces, Pittsburgh (PA), November 2002. 5

[12] S.C.W. Ong, S. Ranganath "Automatic Sign Language Analysis: A survey and the Future beyond Lexical Meaning," IEEE Transactions on Pattern Analysis and Machine Intelligence, VOl 27, No 6, June 2005. 1, 2

[13] H. Poizner, E.S. Klima, U. Bellugi, and R.B. Livingston, "Motion Analysis of Grammatical Processes in a Visual-Gestural Language," Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop, pp. 271-292, 1983. 2

[14] W.C. Stoke, D.C. Casterline, and C.G. Croneberg, "A dictionary of American sign language on linguistic principales," Linstok Press, 1976. 2

[15] N. Tanibata, N. Shimada, and Y. Shirai, "Extraction of Hand Features for Recognition of Sign Language Words," Proc. International Conf. Vision Interface, pp. 391-398, 2002. 1

[16] C. Tomasi and T. Kanade, "Shape and motion from image streams: A factorization method," Proc. National Academy of Scienses, Colloquium paper, Vol.90, pp. 9795-9802, November 1993. 2

[17] R.B. Wilbur, “American Sign Language: Linguistic and applied dimensions” Second Edition, Boston: Little, Brown, 1987. 1

[18] M. Yang, N. Ahuja, and M. Tabb, “Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition,” IEEE Trans. Pattern Analysis Machine Intelligence, vol. 24, no. 8, pp. 1061-1074, Aug. 2002. 1