3D Hand Model Fitting for Virtual Keyboard System

Paper ID :: xxxx

Abstract

In this paper, a 3D hand model fitting method is presented which can recover the accurate finger positions for a virtual keyboard system. The 3D hand model consists of a detailed polygonal skin driven by an underlying skeleton system. The system uses a structured light sensor to generate dense range measurements of user's hand motion. We exploit depth information and match it against the model to estimate the pose of the hand. The parameters for model deformation are optimized with the guide of the applied forces between model points and range measurements. To speed up the optimization, we simplify the physical model and apply hash table-based fast point pair matching. The system can be used in any application requiring zero formfactor and requires no contact with a medium. Examples of applications include virtual reality, gaming, design, etc.

1. Introduction

A virtual keyboard system is known as a touch-typing device that does not have a physical manifestation of the sensing area, that is, the sensing area which acts as a button is not per se a button but instead is programmed to act as one [1]. It has many applications in human-computer interaction, virtual reality, game control, 3D designs, etc. Wearable sensors can capture hand motion accurately, but they are expensive and inconvenient. Vision-based devices, on the other hand, are less intrusive to human users, and provide fairly high flexibility and accuracy for both implementation and application.

However, tracking articulated structures, like hands, is a nonlinear search problem in a high dimensional space. A realistic 3D model of the human hand has at least 26 degrees of freedom (DOFs). The arsenal of tracking approaches that can track such structures quickly and reliably is still very small.

Two general techniques have been proposed for visual hand tracking. One choice is the appearance-based approach, which estimates hand posture directly from the images after establishing the mapping between the image feature space and the hand motion space. Rosales [2] mapped the low level visual features to hand joint configuration, with a supervised learning framework for training the mapping function. Wu and Huang [3] combined the supervised and the unsupervised learning framework and thus incorporate a large set of unlabeled training data. The major advantage of using appearance based methods is the simplicity of their parameter computation for the temporal gesture. However, the mapping may not be one-to-one, and the loss of precise spatial information makes them especially less suited for hand position reconstruction.

Another approach is the 3D model-based approach. By projecting a 3D hand model to the image space and by matching it with the observed image features, Rehg and Kanade [4] introduced a highly articulated 3D hand model in their *DigitEyes* hand tracking system. The assumption in this work was that the closest available feature is the correct match. Lee and Kunii [5] employed the skeletal model to simulate the human hand in the real image. The constraints on the joints are used to reduce the dimension of the search space. Wu and Huang [6] decoupled the articulated hand motion into global hand motion and local finger motion, and the hand pose determination is formulated as a least median of squares (LMS) problem.

We propose a 3D model based fitting method for accurate hand tracking, i.e. for recovering the 3D world location of the occluded fingers. The system employs dense 3D data generated by a structured light system. We use a static reference image to subtract the background. In each frame, the 3D hand model is sampled and matched against the depth map, and the deformation parameters of the hand model are optimized to minimize the Euclidean distance between the model surface and the depth map surface. The optimization is carried out with a physical based model fitting technique (PMF) [7]. PMF assigns 3D virtual forces between the model-data point pairs and directly maps the virtual force to the parameter space to guide the optimization of the parameters.

The paper is organized as follows. Section 2 describes our 3D hand model and its kinematic transformation chain. Section 3 gives a brief introduction of the physical based optimization model and its simplification. In section 4 our model fitting method is given for reconstructing the accurate hand pose. Section 5 and 6 present the results and a discussion.

2. 3D Hand Model

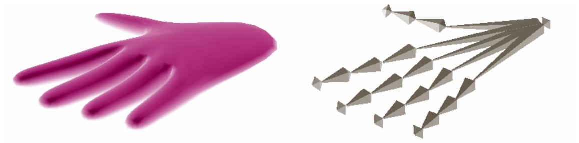


Figure 1. Polygonal skin and the underlying skeleton system of the hand model

Different hand models have been used to represent the hand posture, as simple as a 2D binary silhouette and contour [8], and as complex as a 3D textured volumetric model [9]. However most of them consist only of simple primitives, such as cylinders or truncated cones. Our approach is based on a more detailed, skeleton-driven deformable model. This model consists of a polygonal skin, driven by an underlying skeleton system, which supports both the global hand motion as the translation and rotation of the palm, and the local finger motion as the bending and twisting of all the joints. The 3D illustration of the hand model is shown in Figure 1.

2.1. Hand Model Deformation

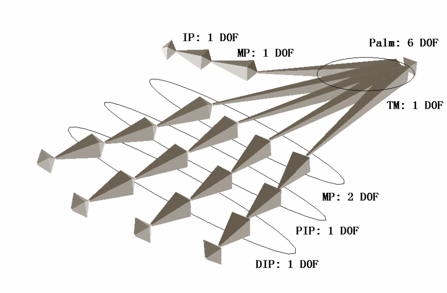


Figure 2. Skeleton system of the hand model

Each skin vertex is rigidly coupled to a subset of the skeleton joints, and when we deform the skeleton system with global or local motion, the shape of the skin is thereafter deformed by linearly blending the motions of each vertex [9]. The new position of a vertex v, influenced by n joints, is computed as:

$$v = \sum_{i=1}^{n} \gamma_i M_i M_{i0}^{-1} v_0 \qquad (1)$$

where γ_i is the blending weight and v_0 is the position in the initial pose of the vertex v. M_i is the global transformation matrix of i-th joint in current pose and M_{i0} is the global transformation matrix of i-th joint in the initial pose.

The structure of the skeleton system is based on the anatomical respects, which provides a hierarchical control for animating the deformation of the polygonal skin. It is defined as a sequence of rigid links and joints. Figure 2 illustrates the skeleton system we used. The constraints of the human hand motion reduce the model to 30 DOFs: one DOF (extension / flexion) for each distal interphalangeal (DIP), interphalangeal (IP) and proximal interphalangeal (PIP) joints, two DOFs (extension / flexion and adduction / abduction) for each metacapophalangeal (MCP) joints except for the thumb (one DOF for extension / flexion), and one DOF (twist) for each trapeziometacarpal (TM) joints except for the thumb (two DOFs for adduction / abduction and twist). The palm has six DOFs for the wrist's translation and rotation movement.

2.2. Kinematics of Hand Model

To describe the translational and rotational relationships between adjacent links of the open kinematic chain, the Denavit-Hartenberg notation (D-H notation) [10] has been used because of its strength in handling a large number of degrees of freedom and its ability to systematically enable kinematic and dynamic analysis. D-H notation uses a minimum number of parameters to completely describe the kinematic relationship, thus establishing a coordinate system to each link of articulated chain in robotics.

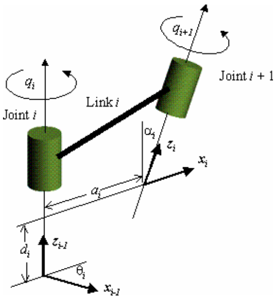


Figure 3. Parameters describing the kinematic relationship

The overall D-H coordinate transformation matrix from frame i coordinate system relative to frame i-1 coordinate system is given as:

$$T_{i-1}^{ik} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\alpha_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where a_i, d_i, α_i and θ_i are the translational and rotational parameters illustrated in Figure 3. And for an *n*-joint manipulator, given *n*-homogeneous transformation matrices $T_0^1, T_1^2, T_2^3, ..., T_{n-1}^n$, the transformation matrix from the end-effector frame *n* to the global frame *0* is:

$$T_0^n = T_0^1 T_1^2 T_2^3 T_3^4 ... T_{n-1}^n \quad (3)$$

Using the D-H notation, the local coordinate systems for each DOF of the joint are illustrated as Figure 4. The skeleton system of the 3D hand model can be viewed as a set of six serial kinematic chains (phalange links). All are attached to a base frame which is defined at the end of the palm.

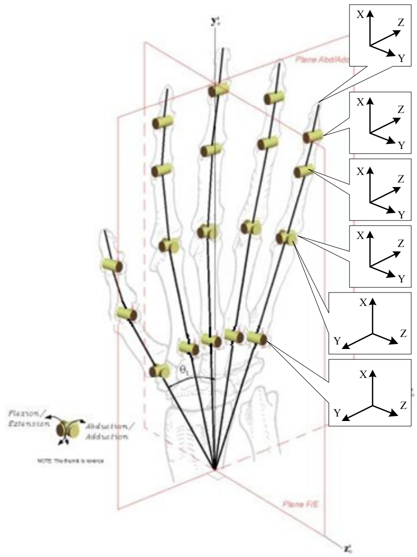


Figure 4. Coordinate frames for D-H notation

3. Physical Based Optimization Model

Physical based deformable models are shapes specified by a set of parameters that deform based on a physical model due to forces, which are determined from visual cues such as edges in an image [11]. Metaxas and Terzopoulos [7] developed this technique for 3D shape and non-rigid motion estimation. Their framework applies dynamic models that incorporate the mechanical principals of rigid and non-rigid bodies into conventional geometric primitives.

Given the shape *s* of a deformable model parameterized by a vector *q*, which includes both for the global translational and rotational movement, and for the local finger motions, each vertex on the model can be expressed as:

$$x(u) = s(u;q) \quad (4)$$

where *u* is used to identify specific points on the model and to provide its topological structure.

Estimation of the model parameters is based on first and second order Lagrangian dynamics. As the shape changes, velocities of points on the model are given by:

$$\dot{x}(u) = L(u)\dot{q} \quad (5)$$

where *L(u)=∂x(u)/q* is the model Jacobian.

From Lagrangian mechanics, the second order equation of motion is in the form of:

$$M\ddot{q} + D\dot{q} + Kq = g_q + f_q \quad (6)$$

where $M = \int \mu L^T L du$ is the mass matrix if we assume the model has a mass distribution *μ(u)* and it is subject to frictional damping, the stiffness matrix *K* may be obtained from a deformation strain energy, and the Raleigh damping matrix $D = \alpha M + \beta K$. The generalized inertial forces $g_q = -\int \mu L^T L\dot{q} du$. And the generalized external forces $f_q = \int L^T f du$ are associated with the component of *q*, where *f(u)* is the force distribution applied to the model.

For single frame model fitting problem, as in the case of this paper, it makes sense to ignore the inertia and set the mass density to zero, which get *M* and g_q to zero. Lacking inertia, the model will come to rest as soon as all the internal and applied forces equilibrate. If we assume the adjacent points on the model barely change their relative positions, which suits well for our skeleton-driven deformation, the stiffness matrix *K* would be very small and is ignored. Then the dynamic equation of motion of the model can be simplified as:

$$\dot{q} = f_q, \qquad f_q = \int L(u)^T f(u) du, \quad (7)$$

where we use a unit damping matrix *D* in this paper and our experiment.

Using *L(u)*, the 3D applied forces *f* are converted to forces acting on *q* and are integrated over the model to find the total parameter force f_q. The distribution of forces on the model is based on forces computed from some measure of the Euclidean distance of the depth map and the 3D hand model.

4. 3D Hand Fitting

The 3D depth map is generated by the structured light system from Vialux^{TM}. The background subtraction process is then applied to the depth map with the help of a static reference frame to mask out the background. To remove the spurious outliers near the rim of the hand and to reduce the 3D measurement noise, a 3×3 Gaussian window filtering is applied to the range data.

4.1. Stochastic Sub-Sampling

In our experiments, a fully expanded hand region in the depth map contains roughly 20,000 points. To speed up the fitting, the hand is sub-sampled at 500 stochastically determined points, which is a good tradeoff for our application keeping both the fitting accuracy and the optimization speed. Our parameter optimization problem is a highly non-rigid searching problem, and there exists in the whole searching space many local minima introduced by the discrete nature of the samplings and the noise in the measurements. By randomly changing the set of samplings where the objective function is evaluated at each iteration step, we lower the chance for the optimizer to get stuck in local minima. Note that the stochastic sub-sampling need not be executed at every iteration of the fitting process. It can be executed every l iterations, where l is dependent on some measure of the fitting error and the parameter variation.

4.2. 3D Forces Assignment

The physical based model fitting uses 3D applied forces to guide the parameter optimization. These forces are calculated as the Euclidean distance between the model samplings and the depth measurements. In our experiments we couple the model samplings to the range data, which is indicated by the dark dot in Figure 5, by looking for the nearest neighbor among the measurements to each vertex on the hand model, referred to as 'point pairs'.

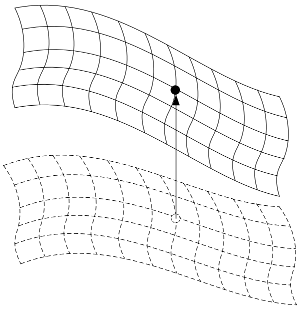


Figure 5. Applied 3D forces to point pair

The brute-force nearest neighbor searching method would work in this case, however too inefficiency with the large amount of measurements. For simplicity, we assume the pseudo-orthographic projection in camera coordinate system and map the 3D measurements to x-y plane. The hand region can be divided into small grids each containing 30 to 50 measurements. The 3D hand model is also projected into the same x-y plane and classified into a different grid. The nearest neighbor searching is only carried out among the measurements in the same grid with the sampled vertex. Thus the grid space acts like a hash table and speeds up the searching tremendously.

If the projected sampling falls out of the hand region in x-y plane, its nearest grid is first searched among all the boundary grids of the hand region. Then, the nearest neighbor searching is carried out among the measurements in that grid.

4.3. Objective Function

For a given set of samplings on the hand model, we seek to minimize the total Euclidean distance between the samplings and their nearest neighbors in the measurements. The Euclidean distance, thereby the 3D applied forces can be expressed as a squared function:

$$f(s) = \|s - s'\|^2 \tag{8}$$

where || • || denote the L_2-norm, and the nearest neighbor s' is found based on the fast point pair matching method in section 4.2.

4.4. Optimization Process

Using the Jacobian matrix L, the 3D applied forces $f(s)$ are then converted to the parameter space acting on q and summed up to estimate the gradient of the parameter vector q as:

$$\hat{\dot{q}} = \sum_{s \in S} L^T f(s) \tag{9}$$

where S is the set of model samplings.

Note that we set the damping matrix D to unit matrix during our model simplification, so there exists a scale factor λ between the estimated gradient $\hat{\dot{q}}$ and the actual $\dot{q}$.

To find an appropriate λ, one can use linear programming, which is quite inefficient if used in every iteration of the optimization process. We use an adaptive variable scale factor λ to circumvent this issue. The step size is adjusted based on the optimization result of every iteration: if the total Euclidean distance is decreased with the current λ, the parameters will be updated with the corresponding scaled gradient and the scale factor will be doubled. Otherwise the update for the parameters will be rejected and the scale factor will be halved to re-calculate the change.

We solve the dynamic system by integrating over time, using the standard differential equation integration techniques:

$$q(t+1)=q(t)+\dot{q}\Delta t \qquad (10)$$

For a series of frames fitting, the *q(0)* of the first frame is taken from the initial pose of the hand model, and for the sequential frames, the *q(0)* is taken from the fitting result of the previous frame as the initial guess.

Figure 6 shows the whole fitting process of our method, with the inner-loop for the parameter optimization based on a certain set of samplings, and the outer loop for stochastically changing the sampling set and control the fitting accuracy.

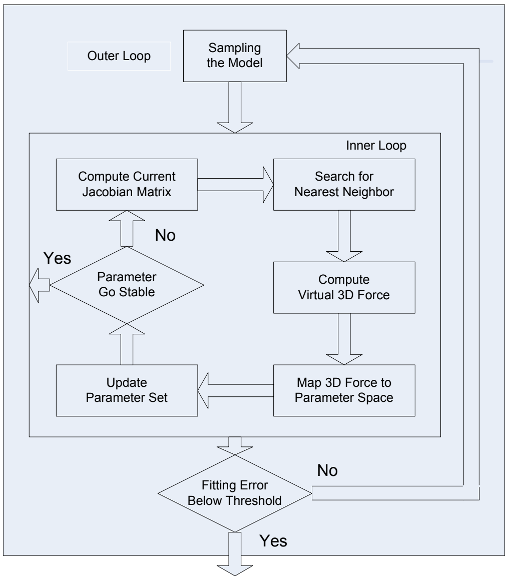
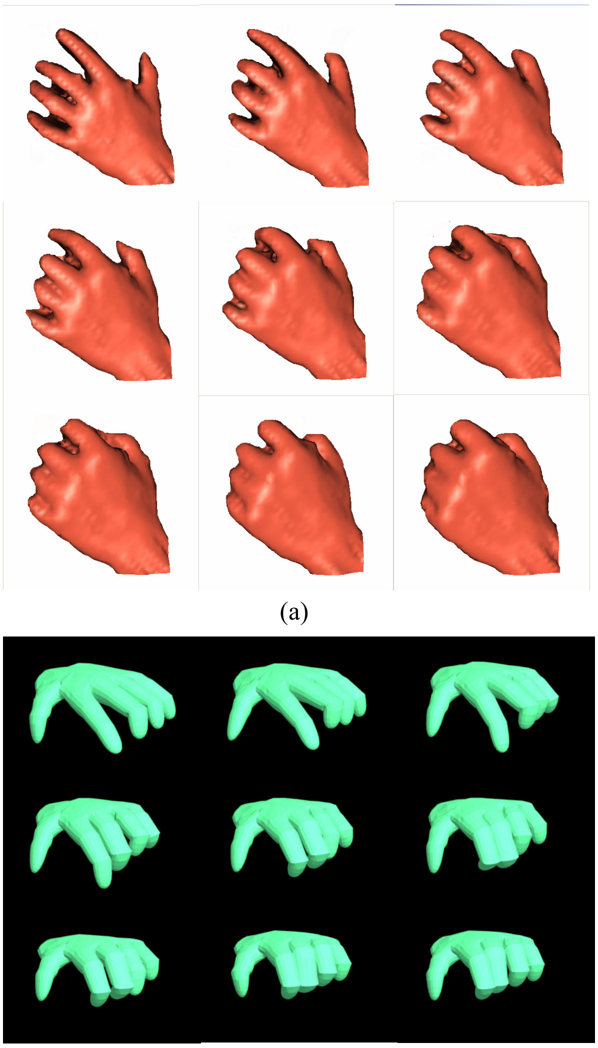

Outer Loop
Sampling the Model
Inner Loop
Compute Current Jacobian Matrix
Search for Nearest Neighbor
No
Yes
Parameter Go Stable
Compute Virtual 3D Force
Update Parameter Set
Map 3D Force to Parameter Space
No
Fitting Error Below Threshold
Yes

Figure 6. Flowchart of model fitting process

5. Results

We have tested our method on real human hand motion sequences. The polygonal skin of the 3D hand model totally has 1114 vertices and 2224 triangular faces. The 3D data are acquired at 16 frames per second with the image resolution of 640×480 pixels. The following experiments were executed on a Pentium 3 1.6GHz PC. The inner-loop in Figure 6 takes at most 30 iterations. The fitting speed is about 0.6 second per frame with 300 iterations in total for our method. Figure 7 shows the raw depth maps (Figure 7.a) and the reconstructed hand poses (Figure 7.b). Note that for the latter two frames, although most fingertips have been self-occluded from the camera's views, we still recovered very accurate and natural positions for all the fingertips. Figure 8 shows the polygonal skin surface projected back to the gray-scale images of the same sequence.



(a)

(b)

Figure 7. (a) Raw depth maps sequence (b) Reconstructed 3D hand sequence

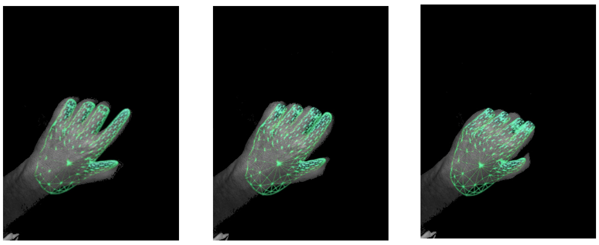


Figure 8. The polygonal surface projected back to the gray-scale images

To measure the fitting accuracy, we define the average fitting error in our case as:

$$\varepsilon_{\mathrm{Avg}} = \frac{1}{N} \sum_{m \in H, n \in M, n = \arg\min_{c \in M} dist(m,c)} dist(m,n) \qquad (11)$$

where H is the point set of the hand model and M is the point set of the range measurements, and N is the total number of points on the hand model. The function *dist(m, n)* compute the Euclidean distance between two 3D points m and n. Figure 9 shows how the average fitting error changes as the iterations go on. The final average fitting error is about 0.2 cm after 150 iterations, which is sufficient for the virtual keyboard applications.

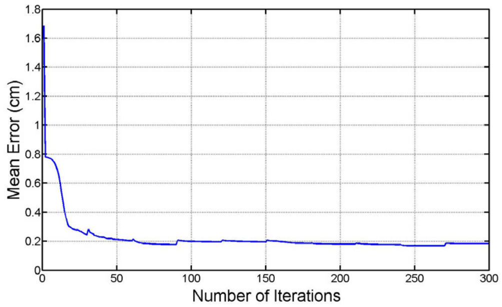


Figure 9. Average fitting error changes with the number of iterations

6. Conclusion

Recovering finger position is a difficult problem due to the large number of DOFs. We have described a 3D hand model fitting method that can recover the accurate locations of fingertips. A detailed 3D hand model is used to match against the range measurements generated by the structured light system. To speed up the fitting process, a simplified physical based model is employed to measure the Euclidean distance between the hand model and the range measurements. Our experiments show that this fitting method performs robustly and efficiently even for some complex self-occlusion cases.

Further research will be carried out in several directions: a pre-calibration process to adjust the 3D hand model to any user with different hand shape / size is a must for a practical virtual keyboard system. The hand motion tracking with the training and analysis of normal human typing movement [12] is needed to estimate / predict the model deformation parameters and will enhance the optimization performance ultimately. Using an implicit expression [13] of the skin surface instead of the explicit polygonal one, thus simplifying our non-rigid discontinuous optimization problem to a well-functioned differentiable solution, is also in our near future consideration.

References

[1] M. Kölsch and M. Turk, "Keyboards without keyboards: a survey of virtual keyboards", *Workshop on Sensing and Input for Media-centric Systems (SIMS 02)*, Santa Barbara, CA, July 2002.

[2] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff, "3D hand pose reconstruction using specialized mappings", *Proc. ICCV*, Vancouver, Canada, July 2001, Vol. 1, pp. 378-385.

[3] Y. Wu and T.S. Huang, "View-independent recognition of hand postures", *Proc. CVPR,* Hilton Head Island, South Carolina, June 2000, Vol. 2, pp. 88-94.

[4] J. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking", *Proc. ECCV94,* Stockholm, Sweden, Vol. 2, pp. 35-46.

[5] J. Lee and T.L. Kunii, "Constraint-based hand animation", *Models and Techniques in Computer Animation,* Springer, Tokyo, June 1993, pp. 110-127.

[6] Y. Wu and T.S. Huang, "Capturing articulated human hand motion: A divide-and-conquer approach", *Proc. ICCV,* Kerkyra, Sept. 1999, Vol. 1, pp. 606-611.

[7] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis", *IEEE Trans. on PAMI,* June 1993, Vol. 15, No. 6, pp. 580-591.

[8] U. Bröckl-Fox, "Real-time 3D interaction with up to 16 degrees of freedom from monocular image flows", *Intl. Workshop on Automatic Face and Gesture Recognition,* Zurich, Switzerland, June 1995, pp. 172-178.

[9] M. Bray, E. Koller-Meier, P. Müller, L. Van Gool, and N.N. Schraudolph, "3D hand tracking by rapid stochastic gradient descent using a skinning model", *Proc. of the 1st European Conf. on Visual Media Production*, London, England, March 2004, pp. 59-68.

[10] M.W. Spong and M. Vidyasagar, *Robot dynamics and control*, John Wiley and Sons, New York, 1989.

[11] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking", *IJCV,* July 2000, Vol. 38, No. 2, pp. 99-127.

[12] R. Urtasun, D.J. Fleet, and P. Fua, "3D people tracking with Gaussian process dynamical models", *Proc. CVPR,* New York, June 2006, Vol. 1, pp. 238-245.

[13] R. Plankers and P. Fua, "Articulated soft objects for multiview shape and motion capture", *IEEE Trans. on PAMI,* Sept. 2003, Vol. 25, No. 9, pp. 1182-1187.