Visual-Motor Jacobian based Image Modelling

Martin Jägersand
Computing Science, University of Alberta
Edmonton, Alberta, T6G 2E8, CANADA
http://www.cs.ualberta.cs/˜ jag

Abstract

Linear viewing geometry models such as affine are appealing because of their simple structure, and despite being approximations they are often applicable in real situations. The visual-motor (VM), or image, Jacobian relates image information not to other visual fiducial points, but to another external (normally motor) frame. These are used in robotics, particularly visual servoing. We show how the Jacobian models image structure, and contrast it to the closest well known linear model, the affine model. Transforms that expand an affine description to a VM and reduce a VM to the best least square fit affine are provided. A composite model which retains the generality of the VM model, while being almost as compact as an affine model, is proposed, and experiments illustrating its use are shown.

1 Introduction

Image or viewing geometry models provide means for organizing and representing image structure, such as the points on moving objects. In computer vision many models, both linear and non-linear, have been considered, such as orthographic, affine and projective[4, 1]. These models represent image structure in image based frames, ie frames which are defined purely visually. Another less explored possibility is to represent image structure in some other frame. For motion the motor frame comes to mind since even if we don't control the motion something must be causing it. In some cases the motor state is directly readable (e.g. from robots or other machines); in others an observation process or other kinds of indirect measurements can be used. Intuitively the key idea is to take advantage of the knowledge of how the object or object-camera system can move since that also affects how the images can change. This kind of knowledge is often the easiest to describe in the motor coordinate frame that directly parameterizes the motion. Such knowledge is of course also task dependent. Some examples are: Jointed structures such as arms and hands where the links restrict motion; non-rigid elastic materials which deform in complex ways, but the deformation is caused by a (relatively) simpler external force. The basic design question is: Can motor space be parameterized so that there exists a function between motor state and visual state? (Where visual state is defined as the image projections of points.) This is the case if you choose joint angles for the linked structure mentioned above or forces for the elastic material. In many cases it is hard or undesirable to have to find this function globally or analytically. Instead we propose to estimate on-line locally linear segments from tracking visual and motor states. Each such linear segment is represented by a Jacobian matrix with a function similar to the basis matrices in linear viewing geometry models. The combination of several Jacobians or segments make up a sparse piecewise approximation of the underlying non-linear function.

In the paper the Jacobian based models are compared to the well known affine models. Hence we start by defining our terminology for the affine model in Section 2.1. In Section 2.2 we introduce the visual-motor Jacobian model, and in Section 2.3 we show how it can be estimated incrementally on-line purely from observations. Section 2.4 discusses the motor kinematics effects on the linearity of the VM model. Section 2.5 addresses how to transform between affine and Jacobian based representations, and last in Section 3 we show experiments with both composite affine and Jacobian based representations used for tracking motion.

2 Theory

Consider a setup as in Fig. 1 where a video camera is supplying a sequence of intensity images $\mathbf{I}_t$ of a moving object. Whether the object or the camera moves is irrelevant here, but we assume there exists some motor space in which the motion can be described. Let $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots)$ be the corresponding sequence of states in motor space.[1] When the motion is caused by a machine (e.g. robot arm), the motor state can often be read out from motor encoders. In other cases an observation process can be used.

Let $\mathbf{y} = (y_1, \ldots, y_m)$ be a vector of image row (x) and column (y) coordinates of $\frac{m}{2}$ tracked points in the image, and $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t, \ldots)$ be the time sequence of

[1] Capitals denote matrices, bold face vectors. Images when treated as vectors are flattened along the column direction.

tracking vectors. For the affine model we consider the first 4 points (8 elements in $\mathbf{y}$) to be fiducial points as shown in Fig. 1.

For the visual-motor case, hereforth called VM, we assume that there exists some (to us usually unknown) function f such that $\mathbf{y} = f(\mathbf{x})$, that is, all visual states can be explained from motor space. Hence f represents the composition of object and world geometry, camera calibration, and motor/manipulator kinematics. Conceptually one can think of this as having divided up the forward image formation function $\mathbf{I} = \Phi(\mathbf{x})$ into $\mathbf{I} = g(f(\mathbf{x}))$, where g is the inverse tracking function, and the image projections of tracked points $\mathbf{y}$ is an intermediate representation of system state.

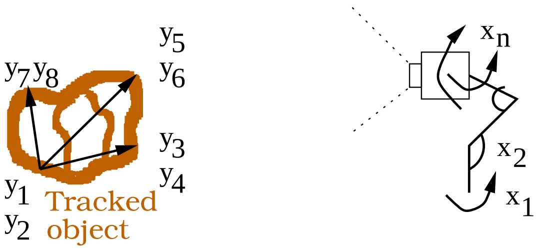

Figure 1: In the affine case image appearance changes are modeled as transforms to an affine basis. In the VM case the transforms are to an external (motor) coordinate frame.

2.1 Geometric Affine Modeling

It is well-known that, given two or more views of a set of points on a rigid object, four non-coplanar points can be chosen as an affine basis, and all other points can be described in that basis [1]. For subsequent views the projection of every point can be determined knowing only the projection of the four basis points.

Given four affine basis points $(\mathbf{e}_0, \ldots, \mathbf{e}_3) = ((0,0,0),(0,0,1),\ldots,(1,0,0))$ and their image projections $\mathbf{y} = (y_1, \ldots, y_8)$, where y_1 and y_2 is the image column and row projection respectively of $\mathbf{e}_0$, y_3 to y_8 similarly for the remaining three basis points. Let the origin be:

$$P_0(\mathbf{y}) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{1}$$

and the three basis vectors be:

$$P_1(\mathbf{y}) = \left[\begin{pmatrix} y_3 & y_5 & y_7 \\ y_4 & y_6 & y_8 \end{pmatrix} - P_0(\mathbf{y}) \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \right]. \tag{2}$$

Any point $\mathbf{q} = (q_1, q_2, q_3)$ defined in the affine basis can be projected into an image point by:

$$\mathbf{p} = P_1(\mathbf{y})\mathbf{q} + P_0(\mathbf{y}) = P(\mathbf{y}, \mathbf{q}) \tag{3}$$

Lines are represented by their two endpoints in both affine and image coordinates. They are transformed from affine to image space by the above equation as row vectors of points: $(\mathbf{p}_1, \mathbf{p}_2) = P(\mathbf{y}, (\mathbf{q}_1, \mathbf{q}_2))$. The object wireframe drawings used for video overlay are represented as a collection of lines.

Several possibilities exist for finding the affine model of an object. We consider two. In the first case the affine structure is known in some basis (e.g. from a drawing or model of the object). Then we need the basis change transform to the currently tracked affine basis.

In the second case affine points, or a simple wire mesh can be defined by the operator by drawing the desired lines and points on top of two or more intensity images from different poses using a graphics editor, as follows. Let $\mathbf{p}^1 \ldots \mathbf{p}^w$, $w \geq 2$ be the image projections of a point $\mathbf{q}$. Similarly $\mathbf{y}^1 \ldots \mathbf{y}^w$ are the projections of the basis. The coordinates of $\mathbf{q}$ are given by solving the overdetermined equation system:

$$\begin{pmatrix} \mathbf{p}^1 - P_0(\mathbf{y}^1) \\ \vdots \\ \mathbf{p}^w - P_0(\mathbf{y}^w) \end{pmatrix} = \begin{pmatrix} P_1(\mathbf{y}^1) \\ \vdots \\ P_1(\mathbf{y}^w) \end{pmatrix} \mathbf{q} \tag{4}$$

In practice a densely sampled set of points $\mathbf{p}$ and $\mathbf{y}$ can be obtained by having a human assign the points once and then tracking them through a smooth object motion. If just two sample views are used, they need to be from two significantly different poses. In this case the two views are shown to the operator and the operator enters the points in the same order for each view. More alternatives for finding models can be found in [2].

2.2 Direct Visual-Motor Image Modeling

An alternative to defining image structure in terms of fiducial image points as above is to parameterize in some other coordinate space. For active motion, the motor space is particularly relevant. Let $\mathbf{x}_t$ describe the motor configuration or pose at time t. A motor command $\Delta\mathbf{x}$ causes a change in motor pose, $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta\mathbf{x}$. Let $\mathbf{y}_t$ be image projections of points as in the previous section. The image change caused by $\Delta\mathbf{x}$ can locally be approximated linearly:[2]

$$\mathbf{y}_{t+1} \approx J\Delta\mathbf{x} + \mathbf{y}_t \tag{5}$$

The components of the (Jacobian) matrix J can in principle be found by solving an equation system similar to Eq. 4, but now requiring the point projection from at least n images, where n is the number of motor freedoms of the system. However, since this image model is only locally valid it has to be incrementally updated as

[2]In general of course the relationship $\mathbf{y} = f(\mathbf{x})$ is non-linear (e.g. because the joints in hands and arms are revolute), for some function f. J then is the Jacobian of f.

the system state changes anyway. Our recursive Jacobian estimation procedure is described in Section 2.3.

Similarly to the affine model, the possible visual configurations are restricted to a subspace. Given only J but no predicted $\mathbf{\Delta x}$, the set of $\mathbf{y}$ consistent with the VM is the range of J, eg the set of possible solutions $\mathbf{y}$ to Eq. 5 for some $\mathbf{\Delta x}$. Hence our visual search space is lowered from m independent visual features values to the set of n coupled changes described by J. Furthermore, if more information is known such as the direction of movement $\frac{\mathbf{x}}{\|\mathbf{x}\|}$ the search space is reduced to one dimensional. For example, in Fig. 4 in the experiments section the 22-dimensional visual space from tracking 11 point features is reduced to 1-D.

The structure information about a feature is coded in the rows of J, which serves function similar to the affine coordinates $\mathbf{q}$, but unlike the affine case there is no direct image geometric interpretation (except in special cases where J represents affine structure as in e.g. Eq. 6). Instead the interpretation depends on the task specific choice of the motor space.

The affine model and the Motor-Visual model are both linear models of image structure, but with different properties. To compare them, note that by taking the affine model (Eq. 3), transposing and rearranging the m visual feature projections $\mathbf{y} = (p_{1x}, p_{1y} \ldots)^T$ (of $\frac{m}{2}$ affine points $\mathbf{q}$), it can be written in the same form as Eq. 5.

$$\begin{pmatrix} p_{1x} \\ \vdots \\ p_{\frac{m}{2}x} \\ p_{1y} \\ \vdots \\ p_{\frac{m}{2}y} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_1^T & \\ \vdots & 0 \\ \mathbf{q}_{\frac{m}{2}}^T & \\ & \mathbf{q}_1^T \\ 0 & \vdots \\ & \mathbf{q}_{\frac{m}{2}}^T \end{pmatrix} \begin{pmatrix} y_3 - y_1 \\ y_5 - y_1 \\ y_7 - y_1 \\ y_4 - y_2 \\ y_6 - y_2 \\ y_8 - y_2 \end{pmatrix} + \begin{pmatrix} y_1 \\ \vdots \\ y_1 \\ y_2 \\ \vdots \\ y_2 \end{pmatrix} \tag{6}$$

As can be seen, while J in Eq. 5 is potentially an arbitrary matrix, in the equivalent conventional affine case, the mapping is significantly constrained (e.g., preserves parallelism) and has only one quarter as many distinct values. However, while the above model can be represented compactly, it is not general enough for visual-motor transforms. To allow these, while still preserving the space efficiency of the affine encoding, we cascade the two types of linear transformations, and write the result compactly as (J_i is row i of J):

$$\begin{pmatrix} p_{1x} & p_{1y} \\ \vdots & \\ p_{\frac{m}{2}x} & p_{\frac{m}{2}y} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (\, y_1 + J_0\mathbf{x}, \quad y_2 + J_1\mathbf{x}\,) +$$

$$+ \left(\begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_{\frac{m}{2}}^T \end{pmatrix} (\, J_{2..4}\mathbf{x}, \quad J_{5..7}\mathbf{x}\,) + \begin{pmatrix} y_3, & y_4 \\ y_5, & y_6 \\ y_7, & y_8 \end{pmatrix} \right) \tag{7}$$

where we can identify $J = (J_0^T..J_7^T)^T$ as the motor-visual Jacobian of the form of Eq. 5 for the affine basis points $(\mathbf{e_0}, \ldots, \mathbf{e_3})$ of Eq. 3.

Jointed objects or objects consisting of several independent rigid bodies can be modeled by extending Eq. 7 with one set of affine coordinates for each rigid body, and a VM Jacobian describing how their motions are coupled. Essentially this is just the column concatenation of several Eq. 7. The extended Jacobian can also naively be formed by column concatenating the independent Jacobians for each rigid body, but a cleverer approach here is to specify the affine frames so that coordinate axes of the different rigid bodies are aligned when possible, e.g. for a 1-D revolute joint one of the affine axes of each link can be defined along the axis of revolution, hence reducing the total basis size from 6 to 5 vectors and m from 16 to 12.

2.3 On-line Motor-Visual Calibration

Assume that during operation of the visual-motor system we have access to the control commands $\mathbf{\Delta x}_t$, and after some delay the images of the resulting movements $\mathbf{I}$. For each of the images we can compute (track) $\mathbf{y}$. Hence we can compute the corresponding $\mathbf{\Delta y}_t$. Each such pair constrains the possible Jacobians J by: (Secant equation)

$$\mathbf{\Delta y}_t = J\mathbf{\Delta x} \tag{8}$$

Using this one can derive a Broyden[9] type update formula:

$$\hat{J}_{t+1} = \hat{J}_t + \frac{(\mathbf{\Delta y}_{measured} - \hat{J}_t\mathbf{\Delta x})\mathbf{\Delta x}^T}{\mathbf{\Delta x}^T\mathbf{\Delta x}} \tag{9}$$

For a set of orthogonal movements about a point $\{\mathbf{\Delta x}_t, t = 1, \ldots, n\}$, Eq. (9) is identical to a finite difference approximation $\hat{J}(\mathbf{x}, \mathbf{d}) = (f(\mathbf{x} + d_1\mathbf{e}_1) - f(\mathbf{x}), \ldots, f(\mathbf{x} + d_n\mathbf{e}_n) - f(\mathbf{x}))D^{-1}$, where $D = diag(\mathbf{d})$ in coordinate frame rotated so that $d_t\mathbf{e}_t = \mathbf{x}_t$. Note however that our estimation accepts movements along arbitrary directions $\mathbf{\Delta x}$, and thus is applicable also during the execution of any task.

Over a course of time our estimation method will generate a piecewise linear estimate of the motor-visual model. This is illustrated in fig. 2. The size of the mesh element on which a particular Jacobian is used is determined using a trust region method [10]. Let α_t bound the trajectory length ($\|\mathbf{\Delta x}_t\| \leq \alpha_t$) between Jacobian updates, and define a model agreement $d_t = \frac{\|\hat{J}\mathbf{\Delta x}\|}{\|\mathbf{\Delta y}_{measured}\|}$. The trust region method adjusts α_t

according to (d_{lower} and d_{upper} are predefined bounds):

$$\alpha_{t+1} = \begin{cases} \frac{1}{2}\alpha_t & \text{if } d_t \leq d_{lower} \\ \alpha_t & \text{if } d_{lower} < d_t \leq d_{upper} \\ \max(2\|\Delta \mathbf{x}_t\|, \alpha_t) & \text{if } d_t > d_{upper} \end{cases} \tag{10}$$

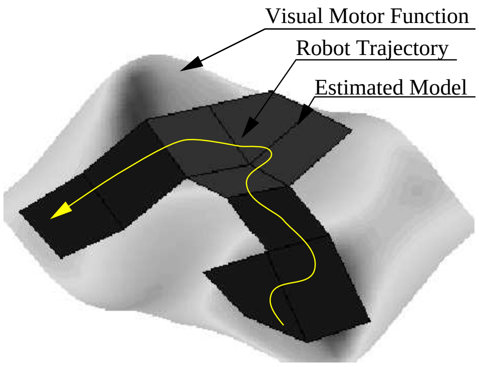

Figure 2: During movement the motor-visual model is estimated along the trajectory using piece-wise linear segments

2.4 Effects of kinematics on VM

The function $\mathbf{y} = f(\mathbf{x})$ models how the image plane projection of world or object points varies with robot configuration $\mathbf{x}$. It depends on object and world geometry, camera calibration, and manipulator kinematics. The first few of these are dealt with in other viewing geometry models. The last one, kinematics, is unique to our approach and we discuss its effects here. Since we estimate f there is no need to exactly analyze kinematics (difficult in the real world). We only need to argue that f is smooth enough. Actuated joints are typically one of two kinds: Translatory linear or revolute. Linear joints result in a Cartesian motor frame and sometimes machines or robots with revolute joints are also calibrated in a Cartesian frame .

If $\mathbf{x}$ is represented in a Cartesian frame, f is an almost linear function, i.e. equality would hold in Eq. 5. If that were the case the Jacobian would only have to be estimated once. However, in practice f is slightly nonlinear due to real cameras not being orthographic, and more important, if (a grasped) object slips, and hence changes its relative pose to the actuators (fingers in case of the robot hand), or any other un-modeled change or system disturbance, then f needs to be re-estimated.

If control commands are in motor joint angle space, f is a composition of link transforms, i.e. the non-linear trigonometric functions. Except for near singular configurations the resulting f is easily approximated by a piecewise linear function as in the previous section. For instance, consider a revolute joint movement in the image plane. Points on the joint draw a circle, while the Jacobian VM will approximate this with linear segments. The image movement can be estimated within 5% relative accuracy for up to 0.5 radian movements between model updates. In a setup for manipulation typically a detailed view of the object is used. This means that the manipulator kinematics will be relatively linear within the field of view and overall it is quite easy to reliably maintain an estimate of f.

2.5 Transforming between VM and affine models

An affine model can be directly expanded to a visual-motor one by using Eq. 6. For reducing a visual-motor model to an affine some kind of best fit approach have to be chosen. A least square fit of the Jacobian basis to an affine can be obtained as follows. Assume that the Jacobian measurement space is from tracking features of a rigid object and arranged so that the first 8 rows come from the 4 feature points selected as affine basis points.[3] Form an explicit feature measurement matrix, and partition it:

$$M = \begin{pmatrix} M_b \\ M_d \end{pmatrix} = J_t X + f(x_t)(1, \dots, 1) \tag{11}$$

where $X = (\mathbf{x}_1, \dots, \mathbf{x}_k)$ is a basis spanning the prior movements $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t$, from which J was estimated or a subset thereof.

Identify the first 8 rows of M as the basis projections $M_b = (\mathbf{y}^1 \dots \mathbf{y}^w)$ with $w = n$ of Eq. 4. It can now be rewritten as:

$$M_d^T - \begin{pmatrix} P_0(\mathbf{y}^1) \\ \vdots \\ P_0(\mathbf{y}^w) \end{pmatrix} (1, \dots, 1) = \begin{pmatrix} P_1(\mathbf{y}^1) \\ \vdots \\ P_1(\mathbf{y}^w) \end{pmatrix} Q \tag{12}$$

Where $Q = (\mathbf{q}_5, \dots, \mathbf{q}_{\frac{m}{2}})$ are the affine coordinates of the tracked points $\begin{pmatrix} y_9 \\ y_{10} \end{pmatrix}, \dots, \begin{pmatrix} y_{m-1} \\ y_m \end{pmatrix}$. For $w = n = 2$ and J at least rank 2, Eq. 12 is well determined and can be solved for the affine coordinates Q. More typically $n > 2$ and the equation system is over determined, and can be solved using one's favorite method for such (e.g. QR decomposition). Particularly if Eq. 12 is solved with a least square error metric it makes sense to choose X so that the condition number $\kappa(JX)$ is low. X then scales J[10]. If scaling is not critical and the Jacobian is full rank the identity matrix can be used, $X = I$.

The combination of the Jacobian estimation in Eq. 9 and Eq. 12 can be shown to be equivalent to the standard affine coordinate estimation, Eq. 4, for a set $t = k \leq n$ of orthogonal movements.[4] In the more

[3] The basis can also be selected as linear combinations of several features, or the whole feature set.

[4] Let $X = (\mathbf{x}_1, \dots, \mathbf{x}_k)$, and note that the original tracking data is preserved in the Jacobian along the directions in X.

typical situation, after many images $t >> n$ and the recursive Jacobian estimate will weigh the most recent image data higher, and hence the method will estimate an affine model which fits best to the current image and motor state, not a compromise to all states which is the case in Eq. 4.

3 Experiments

We show experiments with two temporal tracking applications. Linear systems methods have been used in tracking before (e.g. see [5]), but typically using visual state. Using the VM Jacobian one can track and predict in motor space, which is often lower dimensional than visual space. In addition motions which appear complex when projected into a fixed camera are often much easier described in a natural motor frame.[5]

In the first experiment a piece of 2x4 wood is manipulated by a Utah/MIT robot hand. Although the hand has a 16 DOF workspace the rigid wood block can move in only 6. The composite affine and Jacobian based model (Eq. 7) is used to predict and draw a wire frame onto a wooden block. Four non-coplanar corners shown in [**www-video-1**] are tracked. An initial linear model $\hat{J}_1$ is estimated by performing some linearly independent twiddles of the grasped object. During subsequent hand motion the Jacobian is estimated based on these points using Eq. 9. [**www-video-2**] shows the composite result during a few movements of the robot hand. The error between the model predicted and actual tracked frame was measured for 729 movements and found to be within 2 to 4 pixels.

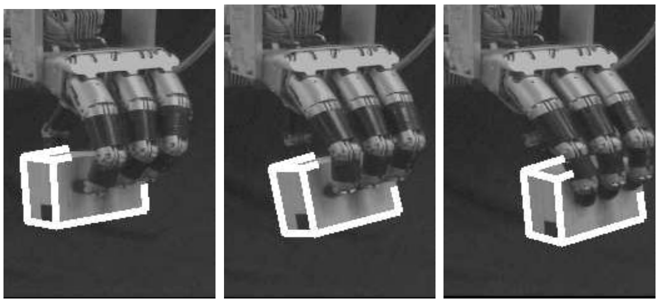

Figure 3: Composite Affine and Jacobian based model used to predict and render a wireframe drawing [**www-video-2**]

The second experiment illustrates how the piece wise linear Jacobian model can be applied in a non-linear problem. The Jacobian based model is used to track non-rigid deformations of a face drawn on a bath sponge. The sponge was squeezed by the author. Instead of using a direct motor state, $\mathbf{x}$ in f was parameterized by the first feature, and all other feature movements are estimated therefrom. Fig. 4 shows three frames from the video [**www-video-3**]. Eleven points were tracked from the face features for a total of 22 feature values. Measured tracker positions are used to update the Jacobian directional information for every 5 to 10 pixels of motion. The directions are overlaid as green arrows in the video. The next tracker positions are predicted using the linear system model Eq.5, with $\Delta\mathbf{x}_{t+1} = \Delta\mathbf{x}_{t,mesured}$. The predicted positions are illustrated with blue crosses. The Jacobian is updated again when new video tracking information (drawn as yellow stars) is available and the cycle repeated. Prediction accuracy of individual feature positions had a 0.5 pixel average and 4 pixel max error.

4 Discussion

We have introduced and argued for the usefulness of visual-motor (VM) transformations as a way of exploring and organizing visual structure. The transform matrix, or VM Jacobian, is a linear image model much like other linear imaging models, such as the affine. VM Jacobians have previously been used in robotics for visual servo control[6]. In those cases the Jacobian is typically modeled analytically, between the motor frame and as few visual features as possible, hence no learning of visual structure has been exploited.

In vision the purpose is to learn something from the images. Hence in this paper we proposed to estimate the VM Jacobians directly from images and control signals, and do so for a large set of features. Thus no cumbersome calibration, or a-priori modeling, of the manipulation system, objects, tools and scene is needed.We described how the Jacobian, when overdetermined, then represents image structure, and how to transform between the VM representation and traditional affine structure.

Some key points are illustrated in Table 1. For many viewing conditions and cameras an affine model is valid over most of the viewing space. The VM Jacobian represents structure locally linearly. It only extrapolates globally if the underlying f is globally linear. Motor parameterizations with non-linear f are however usually much easier to find. Many vision problems are or can be recast as local problems comparing spatially nearby structure, or temporally predicting changes.

One standard affine model can represent the image transformations of one rigid moving object based on how the four fiducial points move. These are represented as eight scalar image features. In the VM case it is advantageous to use as few DOF as possible. A rigid object can move in 6 DOF. If the actual motion is e.g. only translatory or rotational then 3 DOF suffices. Two rigid objects linked by one revolute joint is

[5] Consider for example the 1D motion the points on a wheel or cylinder rolling on a surface. The point motion looks complex to the camera, but can be described by just one motor parameter around the axle of rotation.

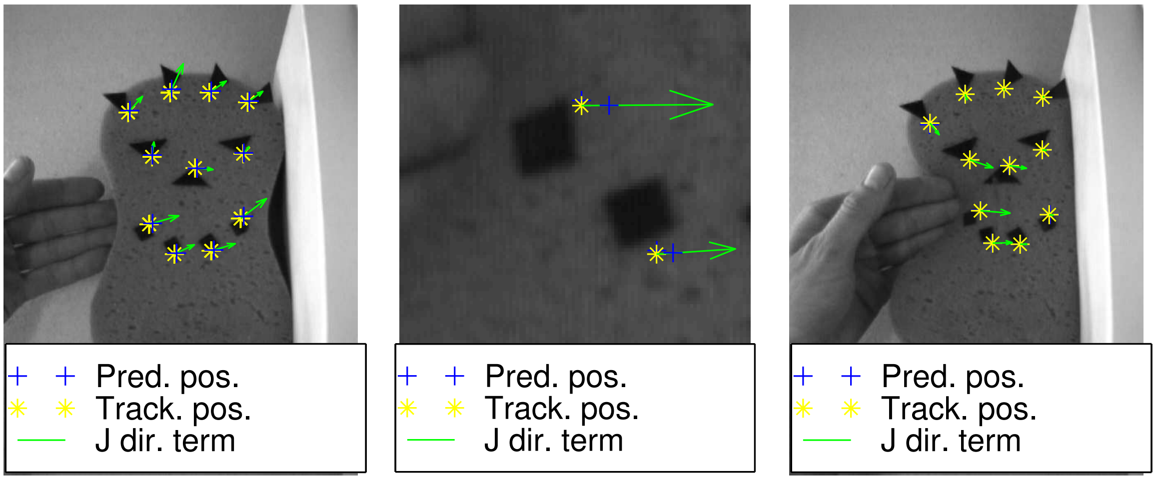

Figure 4: VM model estimation and prediction of non-rigid deformations. Left: Small deformation, Middle: Detail of two features close to the human hand. Left: With larger deformations the relative feature movement is considerable non-linear globally, but the VM adapts to this changing the local linear Jacobian. [**www-video-3**]

	Affine	Jacobian VM
Spatial validity	Global for many cameras	Local for most VM functions
System validity	One rigid object	Rigid, linked and some non-rigid
DOF's	8 for rigid object	Any. Up to 6 per rigid object
Temporal processing	Typ. Batch off-line	On-line recursive local estimation
Size	Compact	Typ. bigger

Table 1: Comparison of affine and VM models

at most 7 DOF etc. For elastic non-rigid deformation the VM is parameterized in the actuated forces.

For many practical problems we propose that a composite affine and VM model (Eq. 7) is the best compromise. Many complex objects can be divided into simpler parts. In the experiments we showed how a grasped object manipulated by a Utah/MIT hand could be represented by an affine wire mesh coupled to a VM for the motions. Here the recursive estimation technique used updates the model to compensate for slips in the grasp points and other disturbances. High DOF robot hand motion has otherwise proven to be difficult to accurately model analytically a-priori[7].

In the future we plan to apply these techniques to more complex systems. A human can for instance be modeled as mostly a jointed structure. Limbs are, although somewhat non-rigid, representable as linked rigid objects. Each link can be represented by an affine frame. The affine frames are related by the VM. The VM could perhaps also be directly applied to very non-rigid parts like the face.

References

[www-video-] On-line m-peg movies of the experiments are available on www.cs.ualberta.ca/~jag/submission/

[1] Koendrink J. J. van Doorn A. J. "Affine Structure from Motion" *L. Opt. Soc. Am.* v8 n2 p377-385, 1991.

[2] Tomasi C. and Kanade T. "Shape and motion from image streams under orthography: A factorization method" *IJCV* 9, 137-154, 1992.

[3] Kutulakos K. N. Vallino J. "Affine Object Representations for Calibration-Free Augmented Reality" *Proc. of IEEE Virtual Reality Symp*, pp. 25-36. the 1996

[4] *CVonline*
On-Line Compendium of Computer Vision R. Fisher (ed). Available: http://www.dai.ed.ac.uk/CVonline/

[5] G. Hager, K. Toyama, "The XVision System: A General-Purpose Substrate for Portable Real-Time Vision Applications" *Computer Vision and Image Understanding*, 69(1), pp. 23-37, 1998.

[6] Weiss L. E. Sanderson A. C. Neumann C. P. "Dynamic Sensor-Based Control of Robots with Visual Feedback" *J. of Robotics and Aut.* v. RA-3 1987

[7] Fuentes O. Nelson R. " Experiments on Dextrous Manipulation without Prior Object Models" *Proc. ISIC* 1996.

[8] Jain A.K. *Fundamentals of Digital Image Processing* Prentice Hall 1989.

[9] Fletcher R. *Practical Methods of Optimization* Chichester, second ed. 1987

[10] Dahlquist G. Björck Å. *Numerical Methods* Second Ed, Prentice Hall, 199x, preprint.