

Flexible flow for 3D nonrigid tracking and shape recovery

Matthew Brand and Rahul Bhotika

TR-2001-38 December 2001

Abstract

We introduce linear methods for model-based tracking of nonrigid 3D objects and for acquiring such models from video. 3D motions and flexions are calculated directly from image intensities without information-lossy intermediate results. Measurement uncertainty is quantified and fully propagated through the inverse model to yield posterior mean (PM) and mode (MAP) pose estimates. A Bayesian framework manages uncertainty, accommodates priors, and gives confidence measures. We obtain highly accurate and robust closed-form estimators by minimizing information loss from non-reversible (inner-product and least-squares) operations, and, when unavoidable, performing such operations with the appropriate error norm. For model acquisition, we show how to refine a crude or generic model to fit the video subject. We demonstrate with tracking, model refinement, and super-resolution texture lifting from low-quality low-resolution video.

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Information Technology Center America; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Information Technology Center America. All rights reserved.

1st draft circulated August 2000; submitted to CVPR May 2001; accepted September 2001; published December 2001.



Flexible flow for 3D nonrigid tracking and shape recovery

Matthew Brand

Mitsubishi Electric Research Labs
Cambridge, MA 02139 USA

Rahul Bhotika

U. Rochester Computer Science
Rochester, NY 14627 USA

Abstract

We introduce linear methods for model-based tracking of nonrigid 3D objects and for acquiring such models from video. 3D motions and flexions are calculated directly from image intensities without information-lossy intermediate results. Measurement uncertainty is quantified and fully propagated through the inverse model to yield posterior mean (PM) and/or mode (MAP) pose estimates. A Bayesian framework manages uncertainty, accommodates priors, and gives confidence measures. We obtain highly accurate and robust closed-form estimators by minimizing information loss from non-reversible (inner-product and least-squares) operations, and, when unavoidable, performing such operations with the appropriate error norm. For model acquisition, we show how to refine a crude or generic model to fit the video subject. We demonstrate with tracking, model refinement, and super-resolution texture lifting from low-quality low-resolution video.

1. Overview

Knowledge of 3D shape and modes of deformation should be a valuable constraint in visual tracking of nonrigid objects. Here we reformulate the problem in a weak perspective context and show that by careful propagation of evidence—including uncertainty—through a properly inverted forward model, one can achieve robust 3D nonrigid tracking directly from intensity changes in relatively unconstrained low-quality consumer-grade video—with lightweight computations. Our treatment offers a complete and correct characterization of the inverse problem (from intensity values to 3D motion parameters) as an inference task.

1.0.1. Model-based flow: We develop a linear approach to 3D flex-constrained optical flow that enables online monocular 3D tracking, model refinement, and super-resolution texture lifting. Given a 3D flexible cloud-of-points model and video, we solve directly for translation, rotation, scale, flexions (deformation coefficients), and confidence measures in each frame. Posterior mean (PM) and Bayesian maximum a posteriori (MAP) motion/flexion estimates are computed directly from intensity gradients without information-lossy intermediate results, e.g., without computing the flow. The formulation also supports motion priors and multi-frame/multiview constraints.

1.0.2. Uncertainty-informed estimation: To make robust inference from a limited set of image measurements we must

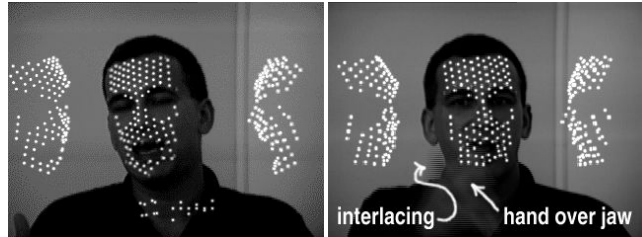


Figure 1: Model-based tracking is robust to degraded images and transient occlusions. Dots show flexed model in 3/4, frontal, and profile view. Dots on face show where the image is sampled. Dots on neck encode 3D motion parameters.

make full use of information about the uncertainty in those measurements. Rather than treat the measurements as points, we treat them as samples from a probability density function (PDF) whose covariance is determined from the image. We carry the entire PDF all the way through the inverse model so that evidence and constraints can interact without premature collapse of the PDF to point estimates.

1.0.3. Maximizing information state: The key to propagating PDFs is the use of matrix transforms to maximize information state: Nonreversible matrix operations (i.e., multiplication, division, thin SVD) drop information about the PDF and compound numerical error. Overconstrained division and thin SVD are particularly nettlesome because their results are correct only in a least-squares sense, whose spherical error norm is often known to be the *wrong* error norm for vision problems (e.g., see §2). We show how to structure chains of matrix operations so that information state *grows* rather than collapses, principally by judicious substitution of reversible analogues, for example, eliminating inner products by substituting Kronecker products for matrix multiplications. This allows us to eliminate or delay least-squares operations until the information state is finally collapsed to give a motion/flexion/shape estimate. To do so we introduce several useful identities that enable one to factor information out of the expanded arrays *under arbitrary elliptical error norms*.

1.0.4. Robust tracking without features: In tracking, we use these methods to propagate image uncertainties back through the projection model until they can be narrowed or resolved via interaction with global geometric invariants, then integrated out to yield robust posterior mean (PM) parameter estimates. The resulting tracker uses whatever information is available in an arbitrary sampling of image regions and gives

accurate motion estimates even when most of these regions are degenerate (textureless) and many are self-occluded.

1.0.5. Acquiring geometry and texture: These methods also lead to a procedure for extracting 3D linearly deformable shape models from video: *Model refinement* combines a generic or poorly fit model with its tracking residuals to yield a much more accurate model with increased detail (more points). The refined model supports subpixel-accurate tracking, which in turn allows us to integrate a super-resolution texture map from nonrigid motion in many frames.

1.1. Prior flexible model-based trackers

1.1.1. 2D: Several notable papers have used flexible 2D models for object matching or tracking. These are either based on eigenspace representations of variability of shape and texture [7, 3, 6] or parametric representations of variability [4, 16]. Most require a large database of hand-marked images for training and/or robustizing statistics to discard outliers; there is a price to pay for using 2D models of 3D variability.

1.1.2. 3D: Nonrigid 3D tracking has been treated mostly as a optimization problem. Pighin, Szeliski, & Salesin [15] used Levenberg-Marquardt optimization to align a texture-mapped morphable 3D model to video frames. DeCarlo and Metaxas [8] set the dynamics of a deformable 3D model against the intensity gradients used to calculate optical flow and solved a linearization of this equation for approximate velocities between two frames. Edge information is incorporated into the system to keep the model from drifting off the subject, and a large SVD is needed to invert the equations in each frame. “Due to serious non-linearities” and non-robustness to flow noise [9, §4.2], this was put into an extended iterative Kalman filter that used flow uncertainty to balance edge forces against flow forces, but apparently the flow forces themselves were not uncertainty-informed. We will show that nonrigid motion can be estimated directly and robustly from intensity gradients using purely linear methods. Eisert, Wiegand, and Girod [10] use the same dynamical as [8], but calculate motion from a synthesized image that combines the model with a texture map. This follows the analysis-by-synthesis approach of Li, Riovainen, & Forchheimer [13]. Bascle & Blake [1] effect an interesting compromise between 2D and 3D by factoring the motion of tracked contours into flexions and 2D affine-with-parallax warps via SVD, and suggest the problem we solve (without SVD) in §3—factoring 3D motion and flexions from image data. Our results advance the literature with: closed-form estimators for pose and flexion, integration over uncertainty in the measurables, and matrix transforms that minimize information loss in calculations.

1.2. Notation

We use standard notation taken from [11, 14]: a is a scalar, \mathbf{a} is a vector, \mathbf{A} is a matrix; $[\Rightarrow_i \mathbf{A}_i]$, $[\Downarrow_i \mathbf{A}_i]$, $[\bowtie_i \mathbf{A}_i]$ are horizontal, vertical, and diagonal concatenations, respectively. \mathbf{I}

is the identity matrix; $\mathbf{0}$ and $\mathbf{1}$ are the zero and one matrices. When not specified, matrix dimensions are determined by conformance. \mathbf{A}^\top denotes transpose; vector-transpose $\mathbf{A}^{(i)}$ transposes matrix \mathbf{A} with each vertical group of i elements treated as a unit. \otimes denotes Kronecker product; \odot denotes Hadamard product; \oplus denotes tiled addition, e.g., $\mathbf{A}_{6 \times 2} \oplus \mathbf{B}_{2 \times 2} = \mathbf{A}_{6 \times 2} + (\mathbf{1}_{3 \times 1} \otimes \mathbf{B}_{2 \times 2})$. $\text{vec } \mathbf{A}$ vectorizes \mathbf{A} by stacking its columns and $\text{vec}_i \mathbf{A}_{r \times c} = (\text{vec } \mathbf{A})^{(i)}$ folds $(\text{vec } \mathbf{A})_{rc \times 1}$ into a matrix having rc/i columns of i elements each; we have generalized vec to be consistent with common practice for matrix reshaping.

1.3. Uncertainty propagation

In state estimation it is common to treat measurements as points whose coordinates are put through an inverted forward model to obtain an estimate of the state variables. It is well understood that this is a mathematically convenient fiction—measurements are samples from a distribution. For correct inference one should push the entire distribution through the inverse model to form a posterior PDF over the state variables. One can then minimize or integrate out the uncertainty to obtain meaningful estimates such as the posterior mode (MAP) or mean (PM) respectively, plus measures of their information content such as the posterior variance and entropy. Only under rare and special conditions will point-estimated values coincide with the PM or MAP values. In fact, a point-estimated value may have very little information content, because the amount of nearby posterior probability mass is unknown.

In space-variant experimental settings such as photogrammetry, the measurement uncertainty can be quantified directly from the signal, and it is a very rich channel of information. We will show how to design inverse models that correctly propagate quantified uncertainty along with measurements, leading estimates of physical variables that are considerably more accurate—and often quite different—than those obtained by point estimation.

1.4. Information state

One difference between propagating point estimates and PDFs through inverse models is that PDFs require special measures to prevent information loss. The information state, or number of bits used to describe the PDF, should increase with each operation that introduces new information, e.g., measurements or physical constraints. But in inverse modeling a large set of measurements is usually reduced to a small number of state parameters, implying a shrinking information state. Information state shrinks in two ways: Bits are lost in *non-reversible* operations such as sums and matrix products (e.g., adding two numbers each having b bits of precision produces a result having at most $b + 1$ reliable bits), while entire dimensions are lost in *non-invertible* operations such as projections and least-squares fits, simply because it takes fewer numbers to specify the result than the operands.

To ensure that information state grows with each operation, we replace non-reversible operators with reversible analogues. For example, we substitute Kronecker for matrix products, e.g., for $\mathbf{C} = \mathbf{A}_{r \times c} \mathbf{B}_{m \times n}$ we write $\mathbf{C}' = \mathbf{A} \otimes \mathbf{B}$. This avoids bit loss because there are no sums and dimension drop because all pairwise products from the two operands are explicitly represented. The matrix product \mathbf{C} can be extracted from \mathbf{C}' by adding elements; more importantly, the operands can also be recovered by averaging elements or, if there is noise, by factoring $\text{vec}_{mn}(\mathbf{C}'^{(m)}) = (\text{vec } \mathbf{B})(\text{vec } \mathbf{A}^\top)^\top$. A forward model rewritten in terms of reversible operators can be inverted to yield estimates that are considerably more robust to numerical and measurement noise, due to overcomplete representation of the PDF. Furthermore:

Lemma: One can solve for any matrix of interest within an arbitrary chain of matrix sums, products, reshapings, and rearrangements with a single least-squares operation.

Proof sketch: The result of the operator chain is a linear function of the matrix of interest. All linear operators commute under suitable transforms (e.g., $(\mathbf{A}\mathbf{B})^\top = \mathbf{B}^\top \mathbf{A}^\top$), so the chain can be re-ordered until the matrix of interest is exposed to division.

Remarks: (A) Many of the algebraic and conformance constraints that make it difficult to reorder a matrix formula are handled in the course of converting the formula to reversible operators. (B) Though matrices grow, they are often sparse and their size is bounded by the number of measurements.

2. 3D nonrigid motion

2.0.1. A forward model: We begin with a single 3D point on a nonrigid surface. Let $\mathbf{D}_{K \times 3}$ be a K -morph basis for point i ; by convention the first row specifies the point's mean 3D location and subsequent rows are displacement vectors establishing its modes of deformation. The projection \mathbf{p} of that point onto the image is

$$\mathbf{p} = \mathbf{O}(\mathbf{R}\mathbf{c}^\top \mathbf{D} + \mathbf{t}), \quad (1)$$

where $\mathbf{c}_{K \times 1}$ is a vector of morph coefficients, $\mathbf{R}_{3 \times 3}$ is a rotation; $\mathbf{t}_{3 \times 1}$ is a translation; and \mathbf{O} is a projection operator. We will consider the case of weak perspective, where $\mathbf{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and scale changes due to perspective looming are folded into \mathbf{c} . This provides a good approximation to full perspective when depth variation within the surface is small relative to surface/camera distance.

2.0.2. Optical flow: A point \mathbf{p} has no direct signature in the image, but the motion of the 3D surface immediately around it can be related to local intensity changes from image I_0 to image I_1 via the chain rule:

$$\left(\underbrace{\mathbf{p}_1 - \mathbf{p}_0}_{\text{projected motion}} \approx \underbrace{\mathbf{f} \doteq \frac{d\mathbf{p}}{dt}}_{\text{optical flow}} \right)^\top \cdot \underbrace{\frac{dI}{d\mathbf{p}}}_{\text{spatial gradient}} = \underbrace{\frac{dI}{dt}}_{\text{temporal gradient}} \quad (2)$$

We write \approx because optical flow is differential but the light field is sampled at discrete intervals in space and time. Substituting eqn. (1) for \mathbf{p}_1 in eqn. (2) connects pose parameters directly to the optic flow. By calculating the optical flow $\mathbf{F} \doteq [\downarrow_i \mathbf{f}_i]$ at many sites, one can set up an overconstrained system of equations and solve for pose parameters. This works poorly, because: (A) The spatial gradient may be degenerate, with zero components that corrupt the flow estimate. (B) The system of equations for pose parameters treats all flow estimates equally, while some are more reliable than others. (C) It takes two divisions to solve for \mathbf{R} and three for \mathbf{c} ; each successive least-squares operation introduces error by amplifying the influence of outliers and previous errors.

3. Inference from measurements

3.0.3. Normal equations for symmetric flow: We begin by reformulating optical flow to get a measure of uncertainty. Assuming normally distributed noise in the image intensities the distribution over possible texture flows at a local image patch is

$$\text{flow distribution} \quad \mathbf{f} \sim \mathcal{N}(\mu_f, \mathbf{X}^{-1}), \quad \text{where} \quad (3)$$

$$\text{spatial gradient} \quad \nabla_{\mathbf{p}} \doteq \frac{1}{2} \frac{d}{d\mathbf{p}} (I_0(\mathbf{p}) + I_1(\mathbf{p})), \quad (4)$$

$$\text{spatial variation} \quad \mathbf{X}_{2 \times 2} \doteq \int \nabla_{\mathbf{p}} \nabla_{\mathbf{p}}^\top d\mathbf{p}, \quad (5)$$

$$\text{temporal variation} \quad \mathbf{y}_{2 \times 1} \doteq \int (I_1(\mathbf{p}) - I_0(\mathbf{p})) \cdot \nabla_{\mathbf{p}} d\mathbf{p}, \quad (6)$$

μ_f is linear in \mathbf{y} , and the extent of the patch is defined by the integration measure $d\mathbf{p}$, typically a positive function that monotonically declines with distance from the patch center. This slightly generalizes a formulation due to Tomasi [2] and inherits its virtues: (A) It is symmetric, using texture information in both frames. (B) Solutions for μ_f take the form of *normal equations* in which possible degeneracies can be made well-behaved via a choice of error norm. (C) \mathbf{X} is the *precision* (inverse covariance) of the flow estimate; its eigenvectors and eigenvalues give the directions and magnitudes of the flow's greatest and least certainty.

The mean of the distribution over \mathbf{f} , if determined purely from local image cues, would be estimated as $\hat{\mu}_f \leftarrow \mathbf{X} \backslash \mathbf{y}$. However, by using a global motion model, we assert that

$$\text{vec}(\mathbf{P}_1 - \mathbf{P}_0)^\top \mathbf{X} = \mathbf{F}^\top \mathbf{X} = \mathbf{Y}^\top, \quad (7)$$

which is the uncertain, multiframe version of eqn. (2), describing flows at N different sites simultaneously using the vertically stacked $\mathbf{F}_{2N \times 1} \doteq [\downarrow_i \mathbf{f}_i]$, $\mathbf{Y}_{2N \times 1} \doteq [\downarrow_i \mathbf{y}_i]$ and diagonally stacked $\mathbf{X}_{2N \times 2N} \doteq [\downarrow_i \mathbf{X}_i]$. Note that this is an *uncertain equation* because \mathbf{F} is a multivariate PDF, *not* a numerically instantiable vector.

3.0.4. Reversible formulation: To maximize information state, we rewrite the forward model as

$$\mathbf{P} = (\mathbf{c}^\top \otimes \mathbf{R}) \mathbf{S} \oplus \mathbf{t}, \quad (8)$$

where $\mathbf{S}_{3K \times N} \doteq [\downarrow_i \text{vec } \mathbf{D}_i^\top]$ is a morph basis for N points, and $\mathbf{R}_{D \times 3}$, $\mathbf{t}_{D \times 1}$ are truncated for projection to $D = 2$ dimensional images. This is the multipoint weak perspective

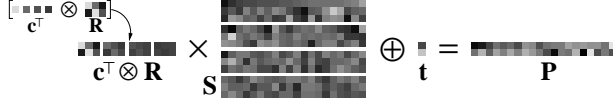


Figure 2: Depiction of eqn. (8) with $D = 2, K = 4, N = 20$.

equivalent of eqn. (1). Note that all variables of interest now participate in reversible operators.

We substitute eqn. (8) for \mathbf{P}_1 in eqn. (7) to obtain

$$\text{vec}((\mathbf{c}^\top \otimes \mathbf{R})\mathbf{S} \oplus \mathbf{t} - \mathbf{P}_0) = \mathbf{F} = \mathbf{X}^{-1}\mathbf{Y}. \quad (9)$$

Eqn. (9) summarizes all the evidence and invariants available. All inference tasks—motion estimation, shape estimation, flow, etc., reduce to solving eqn. (9) for the appropriate set of variables. Instead of direct algebraic solution, we will use l.h.s. eqn. (9) to reduce the multivariate PDF $\mathbf{F} \sim \mathcal{N}(\mu_{\mathbf{F}}, \Sigma_{\mathbf{F}} = \mathbf{X}^{-1})$ to gaussian posteriors over the unknowns. We then solve for the posterior mode. Since the gaussian mean and mode coincide, solving for the posterior mode is equivalent to integrating out all uncertainty.

3.0.5. Translation: Without loss of generality, we assume \mathbf{S} is a zero-mean shape/morph basis (e.g., the row sums $\mathbf{S}\mathbf{1} = \mathbf{0}$), so that rotation and deformation are decoupled from translation, which can be estimated as the mean motion of all points. The posterior PDF over translation is

$$\mathbf{t} \sim \mathcal{N}(\hat{\mathbf{t}}, \mathbf{G}^\top \mathbf{X}^{-1} \mathbf{G} / N^2), \quad \text{where} \quad (10)$$

$$\hat{\mathbf{t}} \leftarrow \arg \min_{\mathbf{t}'} \prod_i \mathcal{N}(\mathbf{X}_i \setminus \mathbf{y}_i; \mathbf{t}', \mathbf{X}_i^{-1}) = (\mathbf{X}\mathbf{G}) \setminus \mathbf{Y}, \quad (11)$$

and the sum operator $\mathbf{G} \doteq (\mathbf{1}_{N \times 1} \otimes \mathbf{I}_D)$. Eqn. (11) is derived by taking the log of the product of gaussians and solving for the minimum of the resulting quadratic form; the uncertainty is calculated as the covariance of the average of a set of random multivariates. §B shows how to combine this uncertainty with that of the flow and propagate it into the rotation and deformation estimates. We relegate it to an appendix because the translation uncertainty is very small compared to the flow uncertainty; we found that such calculations have negligible benefit at noticeable computational expense.

3.0.6. Certainty warping: §A shows that for least-squares operations such as division, there exists a ‘‘certainty warp’’ of a normally distributed uncertain dividend and its divisor such that the quotient having minimum squared-error in the warped space is equal to the quotient having minimum mahalanobis residual in the original space, i.e., the result is the posterior mode with respect to the uncertainty in the images¹. The certainty warp for eqn. (9) is as follows: Let $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top \stackrel{\text{eig}}{\leftarrow} \Sigma_{\mathbf{F}}^{-1} = \mathbf{X}$ decompose the precision matrix into eigenvectors \mathbf{V} and eigenvalues $\mathbf{\Lambda}$. In 2D, the decomposition requires three scalar square roots per point in the model. The left-handed certainty warp $\mathbf{Q} \doteq \sqrt{\mathbf{\Lambda}}\mathbf{V}^\top$ scales eqn. (9) in the directions where the flow is the most certain:

¹If there is uncertainty in the divisor, division is no longer a linear operator and finding the mode of the non-gaussian posterior becomes a cross-entropy minimization problem [5].

$$\mathbf{Q} \text{vec}((\mathbf{c}^\top \otimes \mathbf{R})\mathbf{S} \oplus \mathbf{t} - \mathbf{P}_0) = \mathbf{Q}\mathbf{X}^{-1}\mathbf{Y} = \mathbf{\Lambda}^{-1/2}\mathbf{V}^\top\mathbf{Y}, \quad (12)$$

which we rearrange and rewrite as

$$\mathbf{Q}((\mathbf{c}^\top \otimes \mathbf{R})\mathbf{S})^{(D)} = \mathbf{Y}' \doteq \mathbf{\Lambda}^{-1/2}\mathbf{V}^\top\mathbf{Y} + \mathbf{Q}(\mathbf{P}_0 \oplus \mathbf{t}), \quad (13)$$

where D is the dimensionality of the projection. To solve for pose parameters we use the identity

$$(\mathbf{A}\mathbf{B})^{(D)} = (\mathbf{B}^\top \otimes \mathbf{I}_D)\mathbf{A}^{(D)} \quad (14)$$

to rearrange l.h.s. eqn. (13) to expose pose variables to division. Repeated applications give

$$((\mathbf{c}^\top \otimes \mathbf{R})\mathbf{S})^{(D)} = (\mathbf{S}^\top \otimes \mathbf{I}_D)(\mathbf{c}^\top \otimes \mathbf{R})^{(D)}, \quad (15)$$

$$= ((\mathbf{I}_K \otimes \mathbf{R})\mathbf{S})^{(D)}\mathbf{c}, \quad \text{and} \quad (16)$$

$$= (((\mathbf{c}^\top \otimes \mathbf{I}_3)\mathbf{S})^\top \otimes \mathbf{I}_D)\mathbf{R}^{(D)}. \quad (17)$$

3.0.7. Rotation and deformation: Since $\hat{\mathbf{c}}$ and $\hat{\mathbf{R}}$ are strongly coupled in planar projection, the physically meaningful estimate is the mean of their joint posterior PDF (the means of their marginal posterior PDFs may not be physically consistent). So we first use eqn. (15) to calculate the posterior over $\hat{\mathbf{M}} \doteq \text{vec}_{3D}(\mathbf{c}^\top \otimes \mathbf{R}) = (\text{vec } \mathbf{R})\mathbf{c}^\top$, the outer product of the rotation and flexion parameters:

$$\hat{\mathbf{M}} \sim \mathcal{N}(\hat{\mathbf{M}}, ((\mathbf{S} \otimes \mathbf{I}_D)\mathbf{X}(\mathbf{S}^\top \otimes \mathbf{I}_D))^{-1}) \quad \text{where} \quad (18)$$

$$\hat{\mathbf{M}} \leftarrow [(\mathbf{Q}(\mathbf{S}^\top \otimes \mathbf{I}_D)) \setminus \mathbf{Y}']^{(3D)}. \quad (19)$$

The posterior mean $\hat{\mathbf{M}}$ is easily factored to yield $\hat{\mathbf{R}}, \hat{\mathbf{c}}$. The virtue of factoring is that it avoids concentration of noise-based error in any one pose parameter.

3.0.8. Orthonormal decomposition: The factorization $(\text{vec } \hat{\mathbf{R}})\hat{\mathbf{c}} \leftarrow \hat{\mathbf{M}}$ (vectorized orthonormal matrix times coefficient vector) is usually performed by rank-1 thin SVD $\mathbf{U}\mathbf{S}\mathbf{V}^\top \stackrel{\text{svd}_1}{\leftarrow} \hat{\mathbf{M}}$, followed by orthonormalization of $\text{vec}_2 \mathbf{U}$ to yield $\hat{\mathbf{R}}$, then corrective redivision $\hat{\mathbf{c}} \leftarrow (\text{vec } \hat{\mathbf{R}}) \setminus \hat{\mathbf{M}}$. This finds the rotation closest to the vector that best factors $\hat{\mathbf{M}}$, which is not necessarily the rotation that best factors $\hat{\mathbf{M}}$. Instead we propose solving directly for a rotational factor. We find $\hat{\mathbf{R}}$ by projecting $\hat{\mathbf{M}}$ onto the rotation manifold \mathcal{S}^3 :

Theorem: Let $\mathbf{u}_{k \times 1}$ satisfy $\mathbf{c}^\top \mathbf{u} \neq 0$ and define

$$\mathbf{A} \doteq (\mathbf{I}_D \otimes \hat{\mathbf{u}}^\top)[\mathbf{M}^{\leftrightarrow}, \mathbf{M}^1]^\top = \text{vec}_D(\mathbf{M}\mathbf{u}). \quad (20)$$

Using the eigen-decomposition $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top \stackrel{\text{eig}}{\leftarrow} (\mathbf{A}\mathbf{A}^\top)_{D \times D}$,

$$\hat{\mathbf{R}} \leftarrow \sqrt{\mathbf{A}\mathbf{A}^\top} \setminus \mathbf{A} = \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^\top \mathbf{A}, \quad (21)$$

$$\hat{\mathbf{c}} \leftarrow ((\text{vec } \hat{\mathbf{R}}) \setminus \mathbf{M})^\top = ((\text{vec } \hat{\mathbf{R}})^\top \mathbf{M})^\top / D. \quad (22)$$

Proof sketch: Eqn. (20) projects onto \mathcal{R}^{3D} and eqn. (21) projects onto \mathcal{S}^3 (see [12]). Set $\mathbf{u} = 1/(Kc)$ (element-wise inverses) so that $\mathbf{c}^\top \mathbf{u} = 1$. Substituting $\mathbf{M} = (\text{vec } \mathbf{R})\mathbf{c}^\top$ into eqn. (20) yields $\mathbf{A} = \text{vec}_D((\text{vec } \mathbf{R})\mathbf{c}^\top \mathbf{u}) = \mathbf{R}$, and therefore $\hat{\mathbf{R}} = \mathbf{A} = \mathbf{R}$ in eqn. (21). Due to cancellation of \mathbf{u} in $\sqrt{\mathbf{A}\mathbf{A}^\top} \setminus \mathbf{A}$, the equality $\hat{\mathbf{R}} = \mathbf{R}$ immediately generalizes to any \mathbf{u} yielding a full rank- D division of any $\mathbf{A} \in$ or $\notin \mathcal{S}^3$ in eqn. (21) (hence the restriction $\mathbf{c}^\top \mathbf{u} \neq 0$).

Remarks: (A) $\hat{\mathbf{R}}$ is determined up to one sign ambiguity (reflecting the bas-relief ambiguity of weak perspective). (B)

Choices of \mathbf{u} *do* matter in the presence of noise because the cancellation becomes inexact. Because eqn. (19) estimates an $\hat{\mathbf{M}}$ having whitened uncertainty, $\mathbf{u} = \mathbf{1}$ works quite well and the optimal value $\mathbf{u} = \pm \sqrt{\text{trace}(\hat{\mathbf{M}}^\top \hat{\mathbf{M}})}$ is quickly approximated as $\hat{\mathbf{u}} \leftarrow \hat{\mathbf{M}}^\top \hat{\mathbf{M}}_{\text{col}(1)}$. (C) For images, $\hat{\mathbf{R}}$ is computed in an $O(1)$ 2D eigen-decomposition requiring just three scalar square roots. Factorization via SVD and orthonormalization takes $O(K^2)$. (D) We find that in Monte Carlo numerical trials with noise added to \mathbf{M} , orthonormal decomposition comes closest to a known true rotation with $p \ll 0.01$ levels of statistical significance.

3.0.9. A basic tracker: At this point, eqns. (4-6, 11, 19, 20-22) constitute a simple but quite functional nonrigid 3D motion tracker which can be implemented in one page of Matlab code. The rest of the paper explores other inferences and optimizations this framework supports.

3.0.10. Motion refinement: Because the image evidence is taken from small windows around each point, for large motions there is the risk that relevant regions in image I_1 are not being sampled. Therefore it is desirable to resample the gradients at a physical offset given by eqns. (11, 19). While doing so, one can quickly reestimate $\hat{\mathbf{R}}, \hat{\mathbf{c}}$ by computing their conditional posterior means from eqns. (17, 16):

$$\hat{\mathbf{R}}|\mathbf{c} \leftarrow [(\mathbf{Q}((\mathbf{c}^\top \otimes \mathbf{I}_3)\mathbf{S})^\top \otimes \mathbf{I}_D) \setminus \mathbf{Y}']^{(D)} \quad (23)$$

$$\hat{\mathbf{c}}|\mathbf{R} \leftarrow (\mathbf{Q}((\mathbf{I}_K \otimes \mathbf{R})\mathbf{S})^{(D)}) \setminus \mathbf{Y}' \quad (24)$$

If iterated, this cycle can be viewed as a Gauss-Newton method with quadratic convergence rate; in practise we find we only need to do it once or twice per frame².

3.0.11. Rapid calculation: Eqns. (19, 23) & (24) can be recast in normal form (e.g., $\mathbf{A} \setminus \mathbf{Y} = (\mathbf{A}^\top \mathbf{A}) \setminus (\mathbf{A}^\top \mathbf{Y})$) to yield small divisions by symmetric matrices that are the size of the unknowns. This also eliminates the need to eigen-decompose \mathbf{X} (because in normal form, we get the cancellations $\mathbf{Q}^\top \mathbf{Q} \rightarrow \mathbf{X}$ and $\mathbf{Q}^\top \Lambda^{-1/2} \mathbf{V}^\top \rightarrow \mathbf{I}$). For example, let \mathbf{B} be the cheaply computed inverse Cholesky factor satisfying $\mathbf{B}^{-\top} \mathbf{B}^{-1} = [(\mathbf{S} \otimes \mathbf{I}_D)\mathbf{X}(\mathbf{S}^\top \otimes \mathbf{I}_D)]_{3DK \times 3DK}$. Then substituting eqn. (19) in normal form into eqn. (20) gives

$$\mathbf{A} \leftarrow \text{vec}_D[(\mathbf{u}^\top \otimes \mathbf{I}_{3D})\mathbf{B}\mathbf{B}^\top(\mathbf{S} \otimes \mathbf{I}_D)(\mathbf{Y} + \mathbf{X}(\mathbf{P}_0 \ominus \mathbf{t}))], \quad (25)$$

evaluable as a sequence of shrinking matrix-vector multiplies. In the naive case of isotropic uncertainty, matrix divisions can be eliminated entirely if \mathbf{S} is made row-orthogonal.

3.0.12. Priors and Bayesian inference: Thus far, we have worked without a prior, so posterior mean and maximum marginal likelihood estimates are identical. One might want a prior on pose, motion, or expression; such priors are easily folded into the estimators given above. For example, consider a gaussian prior probability on flexions

²These are MAP/PM estimates w.r.t. the simplifying assumption that the other pose parameters are certain. It is also possible to solve for MAP estimates w.r.t. uncertainty in the other pose parameters [5] (but not PM—the posterior loses normality), but the increased computational cost appears to outweigh the increased convergence rate.

$p_c(\mathbf{c}) \doteq \mathcal{N}(\mu'_c, \Sigma'_c)$. Since the log-posterior is a sum balancing two quadratic forms (log-likelihood against log-prior), and the least-squares division in the estimator is similarly a balance between constraints expressed in the rows of the divisor, the maximum *a posteriori* estimator $\hat{\mathbf{c}}'_{\text{MAP}} \doteq \arg \max_c p(\mathbf{c}|\mathbf{X}, \mathbf{Y}, \mathbf{R}, \mathbf{t}, \mu'_c, \Sigma'_c)$ is constructed by concatenating additional constraints (rows) to the system of equations. Of course, these constraints must also be certainty-warped into the spherically normed space of the least-squares solver. For example, eqn. (24) is rewritten

$$\hat{\mathbf{c}}_{\text{MAP}} \leftarrow \left[\begin{array}{c} \mathbf{Q}((\mathbf{I}_K \otimes \mathbf{R})\mathbf{S})^{(D)} \\ \underline{\mathbf{Q}}_c \end{array} \right] \setminus \left[\begin{array}{c} \mathbf{Y}' \\ \underline{\mathbf{Q}}_c \mu'_c \end{array} \right]. \quad (26)$$

where \mathbf{Q}_c is the left-handed certainty warp derived from $\Sigma'_c{}^{-1}$. MAP estimators can be similarly constructed for translations and rotations³.

3.0.13. Residuals and likelihoods: Given a flow estimate $\hat{\mathbf{F}} \leftarrow ((\hat{\mathbf{c}}^\top \otimes \hat{\mathbf{R}})\mathbf{S} \oplus \hat{\mathbf{t}}) - \mathbf{P}_0$, the unaccounted temporal intensity information is $\mathbf{H} \doteq \mathbf{Y} - \mathbf{X}(\text{vec } \hat{\mathbf{F}})$ intensity-levels \times pixel-lengths. Working forward from the uncertainty model of the low-level flow (eqn. (3)), the tracking residue, or mahalonobis distance, is $\sqrt{\mathbf{H}^\top \mathbf{X}^{-1} \mathbf{H}}$ intensity-levels (per frame). This implies that the likelihood of the image evidence given a motion estimate is $p(\mathbf{X}, \mathbf{Y}|\mathbf{R}, \mathbf{C}, \mathbf{T}) = e^{-(\mathbf{H}^\top \mathbf{X}^{-1} \mathbf{H} + 2n \log 2\pi - \log |\mathbf{X}|)/2}$.

3.1. Other sources of information

3.1.1. Oblique/occluded surface patches: Backfacing and silhouette-edge patches can be discounted in calculations on a frame-by-frame basis by adding information about surface normals to the geometric model. One can then weight each flow window's contribution to \mathbf{X}, \mathbf{Y} by $\max(0, z)$, z being the depth component of its associated unit normal. With occluded points, $\hat{\mathbf{t}}$ must be refined as well.

3.1.2. Multiframe/multiview constraints: One may choose any set of previously processed frames and produce virtual frames by warping them into the pose of frame $t-1$ using their motion/flexion estimates. Then the multiframe motion/flexion estimators for frame t are built by stacking the dividends (and similarly, the divisors) that relate each virtual frame with frame t . The matrices are already weighted by their certainties, so the result is a proper expectation instead of a mere average. Evidence from multiple cameras can be combined in the flexion estimator by similar stacking (assuming identical optics and distances to the surface).

4. Model refinement

With rotations and flexions computed for many frames, the model \mathbf{S} can be refined to better fit the video subject. Let $\mathbf{R}_{0 \rightarrow t}$ and $\mathbf{c}_{0 \rightarrow t}$ be the estimated rotation and flexion taking frame 0 into frame t , and

³Caveat: A rotational prior will be gaussian in \mathcal{R}^6 rather than fisherian on the \mathcal{S}^3 manifold of rotations (a negligible difference for small Σ'_R).



$\mathbf{N}_{0 \rightarrow t} \doteq \mathbf{c}_{0 \rightarrow t}^\top \otimes \mathbf{R}_{0 \rightarrow t} - \mathbf{c}_{0 \rightarrow t-1}^\top \otimes \mathbf{R}_{0 \rightarrow t-1}$. Let $\mathbf{F}_{t-1 \rightarrow t}$ be the flow taking frame $t-1$ into frame t , and $\mathbf{t}_{t-1 \rightarrow t}$ be its translational component. Then it is a tautology from eqn. (9) that $\mathbf{S} = [\downarrow_{t=1}^T \mathbf{N}_{0 \rightarrow t}] \setminus [\downarrow_{t=1}^T \mathbf{F}_{t-1 \rightarrow t} \ominus \mathbf{t}_{t-1 \rightarrow t}]$, where \downarrow signifies vertical stacking. However, if we rewrite this using the uncertainty information, then we can solve for the model that minimizes tracking residuals for the estimated motion:

$$\hat{\mathbf{S}} \leftarrow \underset{3K \times N}{\text{vec}} \left[\left[\downarrow_{t=1}^T \mathbf{Q}_{t-1 \rightarrow t} (\mathbf{I}_N \otimes \mathbf{N}_{0 \rightarrow t}) \right] \right. \\ \left. \setminus \left[\downarrow_{t=1}^T \mathbf{Q}_{t-1 \rightarrow t} (\mathbf{F}_{t-1 \rightarrow t} \ominus \mathbf{t}_{t-1 \rightarrow t}) \right] \right] \quad (27)$$

(\mathbf{Q} and $\mathbf{\Lambda}$ are those of the flow-based certainty warp.) One can calculate shape directly from image gradients by replacing the numerator with $[\downarrow_{t=1}^T \mathbf{\Lambda}_{t-1 \rightarrow t}^{-1} \mathbf{Q}_{t-1 \rightarrow t} (\mathbf{Y}_{t-1 \rightarrow t} - (\mathbf{1}_{1 \times n} \otimes \mathbf{t}_{t-1 \rightarrow t}) \mathbf{X}_{t-1 \rightarrow t})]$. However, we found this can be sensitive to brightness constancy violations (e.g., it causes the model to bulge at specularities). We also found it useful to constrain eqn. (27) to retain the x, y coordinates of the original model and to solve only for depth and deformations by stacking heavily weighted rows with frontal-plane-only rotations.

4.0.3. Adding detail: Model refinement makes it possible to increase the level of detail of a model: Interpolate or extrapolate new points, track, then refine to get corrected depths and deformations for those points.

5. Experiments

5.0.4. Model acquisition: We were given some 320×240 29.97Hz video of a subject who had been motion-captured several years ago. The video and some 23-marker motion capture data was donated by the studio because calibration errors had made the data too noisy and warped to be usable for the client’s application. The images are also low quality, having low contrast and resolution, autofocus gaffes and interlacing artifacts from an early consumer video camera. The image region containing the face is roughly 60×100 pixels and is quite dim, with a dynamic range of about 40 gray levels. We obtained a 23-point model  from a principal components analysis of a random subset of motion capture frames whose rigid-body motion had been approximately removed. (We used motion capture data and video taken at different times.) We also guessed 3 nose points. To get better coverage of the face, we linearly interpolated between these points to obtain a new hexagonally-gridded 128-point model —at cost of having depths and deformations that are somewhat incorrect. None of the points in the original model were retained. This model predominantly samples the forehead, nose, cheeks, and lower jaw—less than 15% of the sampling windows overlap high-texture features.

5.0.5. Initialization and tracking: The model was superimposed on the face in one frame with an incorrect frontal/neutral pose and a 20-frame subsequence was tracked backwards and forwards several times until the model

“seated” properly on the face. The remaining 1700 frames were then tracked without difficulty, despite interlacing artifacts, large rotations, motion blur, and partial self-occlusion by hands and head turns (see figure 1). This is more than twice the longest sequence reported in [9], at roughly 1/10 the resolution (facial area), demonstrating very good resistance to drift. Image gradients were sampled twice per frame; motion estimators (eqns. (24–23)) were applied once per sampling. Figure 4 shows the motion parameters recovered from a subsequence with and without uncertainty propagation. Using a refined version of the model (see below), the certainty-weighted tracker proved to be subpixel accurate with average residual of 0.0062 intensity values per sampling window. There was one occasional systematic source of error: When the subject turned his shiny forehead toward a spotlight, violations of the brightness constancy assumption kept the model about 1 pixel short of the full rotation. The tracker was implemented in interpreted Matlab code on a vintage 1998 Alpha 400MHz CPU; tracking rates ranged from 5-12Hz, depending on the number of pixels sampled per frame.

5.0.6. Model refinement: We combined the model with the tracking residuals as per §4 to refine the geometry, thereby recovering the shape and depth of the nose as well as the curvature of the forehead and cheeks. The morph basis was similarly improved. Figure 5 and the accompanying videos contrast the original model with the results of refinement.

5.0.7. Super-resolution texture lifting: To confirm that the tracking was subpixel-accurate, we took 24 tracked frames (frames 8-31 in figure 4), warped the images to a common 3D pose and expression, then combined the results to construct a super-resolution image of the nose and upper lip (the highest-texture part of the face covered by the model) shown in figure 6 (and more clearly in the electronic stills).

5.0.8. Model refinement II: A 68-frame 320×240 video of a 3-year-old child was tracked using the same 26-point model as above and then resampled and refined to 100 points. The sequence was far too short to sufficiently constrain the smallest deformation modes—but it did give a good shape and first deformation estimate, as depicted in figure 3 and shown rotating in the accompanying video. Note that this is quite difficult because modeling a child requires substantial changes to the shape/deformation basis and because youthful faces have very little texture.

6. Summary

We have examined flexible 3D model-based flow and model acquisition from video in the context of linear deformable models viewed in weak perspective. The main results are: (A) 3D motion/flexion estimators that operate directly on image gradients and make full use of image uncertainty to yield posterior mean estimates; (B) methods for minimizing information loss about the measurement PDF as it is propagated through chains of matrix operations in the inverse model;

(C) model refinement for boosting the detail and accuracy of models. The uncertainty-informed calculations in this paper are fast, accurate, and robust in the face of noise and degeneracies. The implementation tracks accurately for thousands of frames in low-res low-quality video, giving results that appear to compare favorably with the state-of-the-art. We are now studying more interesting camera models and the problem of integrating over uncertainty through time.

Acknowledgments

Thanks to Ilya Baran and Jeroen Baar for programming support, and to John Hershey for a careful reading of this paper. Rahul Bhotika was supported by a MERL internship and by NSF grant IIS-9875628.

References

- [1] B. Bascle and A. Blake. Separability of pose and expression in facial tracking and animation. In *Proc. ICCV*, pages 323–328, Mumbai, India, 1998.
- [2] S. Birchfeld. Derivation of Kanade-Lucas-Tomasi tracking equation. Web-published manuscript at <http://robotics.stanford.edu/~birch/kl/>, 1996.
- [3] M. J. Black and A. D. Jepson. Eigenttracking: Robust matching and tracking of articulated objects using a view-based representation. In *IJCV*, pages 63–84, 1998.
- [4] M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proc. ICCV*, pages 374–381, Cambridge, MA, 1995.
- [5] M. Brand. An uncertainty algebra. In *In preparation*, 2001.
- [6] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. In *Proc. ECCV*, volume 2, pages 484–498, 1998.
- [7] M. Covell. Eigen-points: Control-point location using principal component analysis. In *Proc. 2nd IWAFGR*, pages 122–127, Vermont, USA, 1996.
- [8] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proceedings, CVPR96*, pages 231–238, 1996.
- [9] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127, 2000.
- [10] P. Eisert, T. Wiegand, and B. Girod. Model-aided coding: A new approach to incorporate facial animation into motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3):344–358, April 2000.
- [11] G. Golub and A. van Loan. *Matrix Computations*. Johns Hopkins U. Press, 1996.
- [12] B. Horn, H. Hilden, and S. Negahdaripour. Closed form solution of absolute orientation using orthonormal matrices. *J. of the Optical Society A*, 5(7):1127–1135, 1988.
- [13] H. Li, P. Roivainen, and R. Forchheimer. 3-D motion estimation in model-based facial image coding. *IEEE Trans. PAMI*, 15(6):545–555, June 1993.
- [14] J. R. Magnus and H. Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. Wiley, 1999.

- [15] F. Pighin, R. Szeliski, and D. H. Salesin. Resynthesizing facial animation through 3D model-based tracking. In *Proc., ICCV*, pages 143–150, Corfu, Greece, 1999. IEEE Computer Society.
- [16] S. Sclaroff and J. Isidoro. Active blobs. In *Proc., ICCV*, pages 1146–1153, Mumbai, India, 1998.

A. Elliptical error norms

Least-squares problems such as division and SVD are stated: Solve $\mathbf{E} \doteq \mathbf{JK} - \mathbf{L}$ for \mathbf{J} or \mathbf{K} (or both factors) such that error norm $\|\mathbf{E}\| = \text{tr}(\mathbf{E}^\top \mathbf{E})$ is minimized. This spherical norm tolerates error in all directions equally, whereas spatial inference problems typically present elliptical error norms, with error tolerated less in directions along which measurements are more certain. We replace the least-squares objective with an elliptical error norm $(\text{vec } \mathbf{E})^\top \boldsymbol{\Sigma}^{-1} (\text{vec } \mathbf{E})$ specified by symmetric positive definite matrix $\boldsymbol{\Sigma}$. The vec permits arbitrary constraints between *all* variables, even if they are in different columns of \mathbf{E} (the unvectorized case is known as weighted least-squares [11]). Setting the matrix derivative to zero, we find that the solution must satisfy $\mathbf{0} = \mathbf{Q}(\text{vec}(\mathbf{JK} - \mathbf{L}))$, where \mathbf{Q} , a factor of $\boldsymbol{\Sigma}^{-1}$, determines the error norm that the solution will minimize (e.g., $\mathbf{Q} = \mathbf{I}$ implies the spherical norm). Decompose $\boldsymbol{\Sigma}^{-1}$ into unitary eigenvectors \mathbf{V} and eigenvalues $\boldsymbol{\Lambda}$ such that $\boldsymbol{\Sigma}^{-1} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$ and define $\mathbf{Q} \doteq \sqrt{\boldsymbol{\Lambda}}\mathbf{V}^\top$. Then the least-squares problem $(\mathbf{Q} \text{vec } \mathbf{E})^\top (\mathbf{Q} \text{vec } \mathbf{E})$ is algebraically equivalent to the elliptical objective. Geometrically, \mathbf{Q} rotates the problem’s directions of greatest and least uncertainty into axis-alignment, then scales each axis proportional to its certainty. For this reason it is often called a “certainty warp,” “covariance-weighted solution,” or more traditionally, “directionally weighted least-squares,” a solution method for vector/matrix problems. We generalize the warp to matrix/matrix problems such as $\mathbf{JK} = \mathbf{L}$ by vectorizing with the identity $\text{vec } \mathbf{L} = \text{vec}(\mathbf{JK}) = (\mathbf{I} \otimes \mathbf{J}) \text{vec } \mathbf{K} = (\mathbf{K}^\top \otimes \mathbf{I}) \text{vec } \mathbf{J}$; premultiplying everything by \mathbf{Q} and dividing for the variable of interest gives solutions $\hat{\mathbf{K}} \leftarrow \text{vec}_{\text{rows}(\mathbf{K})}((\mathbf{Q}(\mathbf{I}_{\text{cols}(\mathbf{K})} \otimes \mathbf{J})) \setminus (\mathbf{Q} \text{vec } \mathbf{L}))$ and $\hat{\mathbf{J}} \leftarrow \text{vec}_{\text{rows}(\mathbf{J})}((\mathbf{Q}(\mathbf{K}^\top \otimes \mathbf{I}_{\text{rows}(\mathbf{J})})) \setminus (\mathbf{Q} \text{vec } \mathbf{L}))$.

B. Propagating translation uncertainty

Removing translation from the flow changes the uncertainty of the measurement PDF:

$$\mathbf{F} \ominus \mathbf{t} \sim \mathcal{N}(\mu_{\mathbf{F} \ominus \mathbf{t}}, \boldsymbol{\Sigma}_{\mathbf{F} \ominus \mathbf{t}}), \text{ where} \quad (28)$$

$$\boldsymbol{\Sigma}_{\mathbf{F} \ominus \mathbf{t}} = \boldsymbol{\Sigma}_{\mathbf{F}} \oplus \boldsymbol{\Sigma}_{\mathbf{t}} \oplus \mathbf{G}^\top \boldsymbol{\Sigma}_{\mathbf{F}} / N \oplus \boldsymbol{\Sigma}_{\mathbf{F}} \mathbf{G} / N; \quad (29)$$

$\boldsymbol{\Sigma}_{\mathbf{F}} = \mathbf{X}^{-1}$; $\mu_{\mathbf{t}} = (\mathbf{X}\mathbf{G}) \setminus \mathbf{Y}$; $\boldsymbol{\Sigma}_{\mathbf{t}} = \mathbf{G}^\top \mathbf{X}^{-1} \mathbf{G} / N^2$; and $\mathbf{G} \doteq (\mathbf{1}_{N \times 1} \otimes \mathbf{I}_D)$. The last two summands in the covariance account for correlation between $\boldsymbol{\Sigma}_{\mathbf{F}}$ and $\boldsymbol{\Sigma}_{\mathbf{t}}$. Using this, eqn. (9) is rewritten

$$\text{vec}((\mathbf{c}^\top \otimes \mathbf{R})\mathbf{S} - \mathbf{P}_0) = \mathbf{F} \ominus \mathbf{t} = \boldsymbol{\Sigma}_{\mathbf{F} \ominus \mathbf{t}}(\mathbf{X} \setminus \mathbf{Y} \ominus \hat{\mathbf{t}}) \quad (30)$$

and all subsequent calculations use $\boldsymbol{\Sigma}_{\mathbf{F} \ominus \mathbf{t}}$ instead of $\boldsymbol{\Sigma}_{\mathbf{F}}$ as the uncertainty.

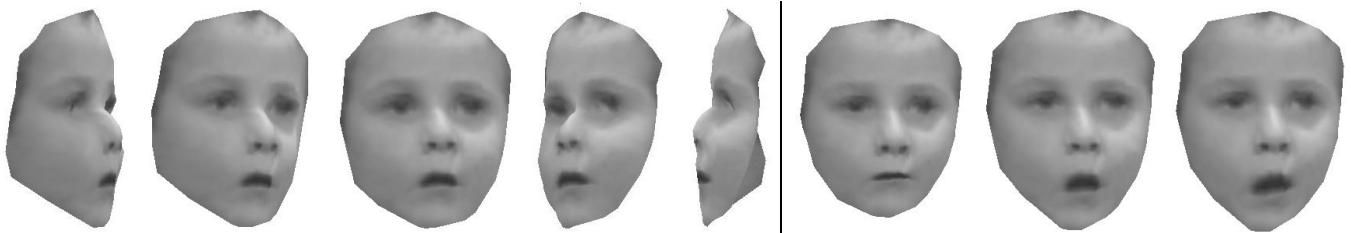


Figure 3: Five views of 3D shape and three principal deformations (closing mouth, opening mouth & raising eyebrows, pulling down mouth corner) recovered from model refinement. All have synthetic geometry and texture.

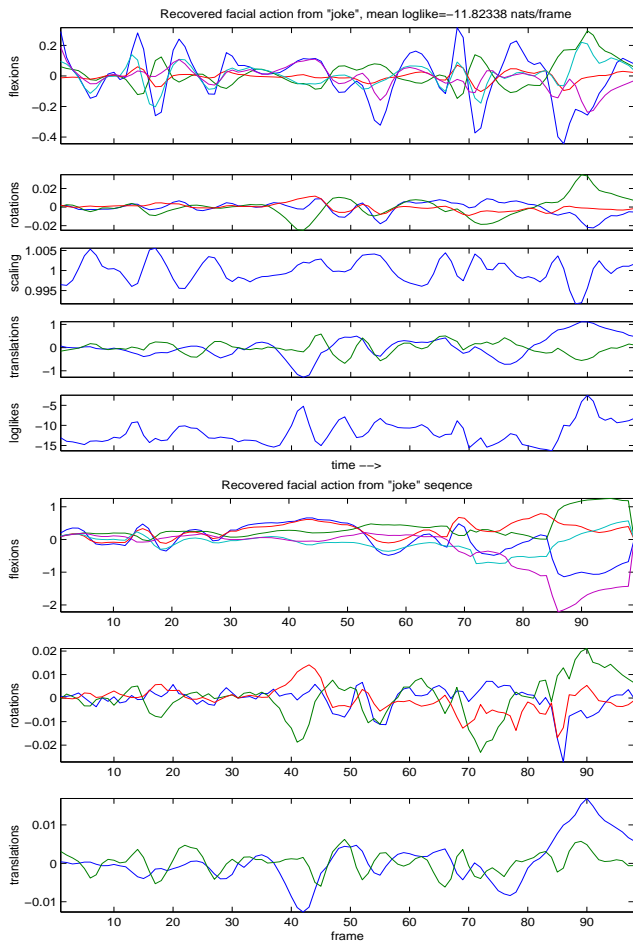


Figure 4: Model-based tracking results from the middle 100 frames of a joke-telling sequence using certainty-weighted (top) and naive (bottom) estimators. Top five graphs show (top to bottom) certainty-weighted flexions, rotation, scale, translation, and log-likelihood (note likelihood peaks for rigid changes such as head rotations). Bottom three graphs show naively calculated flexions, rotation, and translation. The certainty-weighted tracker produces a very clean rotation estimate and registers the flexions in precise detail; the naive tracker has trouble separating rotations from flexions, particularly near the end of the segment.

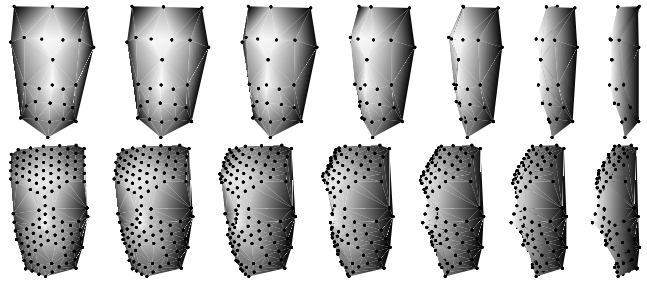


Figure 5: The model before and after one iteration of tracking and refinement, which recovers nose shape and curvature of the cheeks and forehead. Deformations improve as well; here the refined model is posed with the jaw raised partway. We have also added a shaded surface to indicate depth.

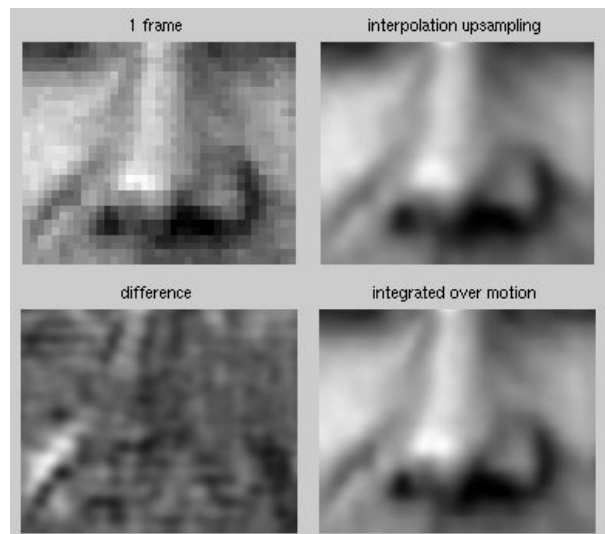


Figure 6: Super-resolution from tracking. Clockwise from top-left: (1) Close-up of a single frame. (2) The best single-frame upsampled bicubic interpolation we could achieve by hand. (3) A $64\times$ super-resolution image constructed by combining 24 tracked frames with motions and flexions warped out. Note the added detail at the cleft of the nose, the contour between the nose and left eye, and the curve of the skin fold that runs from nose to mouth corner. (4) Differencing (2) & (3) reveals some pixelation artifacts and a small flexion widening the upper lip.