Human Pose Extraction from Monocular Videos using Constrained Non-Rigid Factorization

Appu Shaji
Dept. of Computer Science and Engineering
IIT Bombay
Powai, Mumbai 400 076, India
appu@cse.iitb.ac.in

Behjat Siddiquie
Dept. of Computer Science
University of Maryland
College Park, MD 20742, USA
behjat@cs.umd.edu

Sharat Chandran
Dept. of Computer Science and Engineering
IIT Bombay
Powai, Mumbai 400 076, India
sharat@cse.iitb.ac.in

David Suter
Dept. of Electrical and Computer Systems Engineering
Monash University
Clayton 3800, Victoria, Australia
d.suter@eng.monash.edu.au

Abstract

We focus on the problem of automatically extracting the 3D configuration of human poses from 2D image features tracked over a finite interval of time . This problem is highly non-linear in nature and confounds standard regression techniques. Our approach effectively marries a non-rigid factorization algorithm with prior learned statistical models from archival motion capture database. We show that a stand alone non-rigid factorization algorithm is highly unsuitable for this problem. However, when coupled with the learned statistical model in the form of a constrained non-linear programming method, it yields a substantially better solution.

1 Introduction

Given a monocular video which features a single human in motion, our goal in this work is to reconstruct the 3D configuration (seen from an arbitrary choice of a world coordinate system). We assume that we have as input anatomically well-defined landmark points (such as major joints) recorded from an orthographic or weak-perspective camera. Our emphasis is not in feature tracking, but rather on recovering the lost depth during image formation from noisy and possibly incomplete data.

Human motion comprises of an enormous amount of inherent subtlety and variability. Consequently the problem of inferring 3D pose from 2D image sequences is highly non-linear in nature and confounds standard regression techniques. Besides, even if we have a good knowledge about the projection matrix of the camera, for any single input observation of a human pose in 2D, there are possibly multiple valid body configurations. Correlate this with our lack of judgment when we see the Necker cube. From a numerical point of view, estimating 3D structure and motion from image sequences is a higher order (quartic) non-linear optimization problem (§Eq. 5), prone to local minima. These local minima are intrinsic to the problem (termed as true illusions [1]).

Previous Work: A variety of statistical as well as deterministic methods have been developed for extracting pose from single view image sequences. We can define a gross dichotomy on the class of approaches: Ones that concentrate on learning a mapping from silhouette feature space to 3D pose [2], and others that try to map feature points, usually localized to anatomically meaningful landmark points such as elbows position, limb end-point position etc. to 3D poses [3, 4]. Our approach falls in the second category. For a curious reader, we suggest [5] which catalogs most of the important works on 3D human tracking.

BMVC 2007 doi:10.5244/C.21.92

The solution approach in all of the above cases sans [4] is formulated as an (approximate) probabilistic inference problem. Given an observation, they try to pick a pose from a prior distribution which best fits the current likelihood. Though this is an extremely powerful tool, we note that the methods do not explicitly address geometric properties or algebraic details of the data. Rather, the methods rely on these details being captured during the training stage and appear as latent parameters. In essence, this transfers too much importance to the training stage.

An alternative less explored, is to borrow techniques from structure from motion (SfM) and couple them with prior statistical knowledge. SfM [6] techniques are able to produce highly accurate solution when the object is rigid, and is widely regarded as one of biggest success story of computer vision. But, extending SfM to non-rigid scenario has turned out to be quite tricky. One popular flavor of SfM algorithm is the Factorization algorithm [7–10].

In this work, we use a variant of recently proposed [10] non-rigid factorization method (NRF, hereafter) for performing SfM.

Methodology: Factorization methods attempt to capture the implicit geometric invariants present in a wide temporal window of input data. (An example invariant might be that two feature points from a single rigid body should have similar motion trajectories. These invariants uncover themselves as reduced rank constraints [7, 8, 10] on the data observation matrix consisting of stacked (x,y) points. This matrix can be *factorized* into two matrices, one representing the rotation, and the other representing the shape of the object. A straightforward Singular Value Decomposition (SVD) of this matrix results in the recovery of this factorization only up-to a *generalized linear corrective transform* (§Eq. 3). Solving this linear transform is a non-trivial task for several reasons as has been recently observed in the literature.

Further, the current factorization based solutions are not directly adaptable to the human movement problem (our interest) since the quality of the solution degenerates very rapidly when the "deformations" are large[1] .

Contribution: In this paper we propose a novel *constrained factorization* algorithm, which effectively couples prior learned statistical knowledge about human shape variability (and the subspace it spans) from the ground truth motion capture data, with non-rigid factorization algorithm. Specifically, we make use of motion capture data to build a prior *reference pre-shape* (§Sec. 3.1) . We assume that the recovered shape from the NRF algorithm should be structurally similar to the reference pre-shape. This is formulated as a constrained non-linear programming problem. These constraints on the structure of shape subspaces reduces the search domain and renders the problem well-posed (Eq. 6). We provide qualitative and quantitative results to demonstrate the validity of our scheme.

Notation: We follow the notation used in [10]. a is a scalar, $\mathbf{a}$ is a vector and $\mathbf{A}$ is a matrix. $\otimes$ denotes Kronecker product. $\odot$ denotes Hadamard product. $\text{vec}(\mathbf{A})$ vectorizes $\mathbf{A}$ by stacking its columns and $\text{vech}(\mathbf{A})$ vectorizes only its lower triangular portion. $\mathbf{A}^{\dagger}$ denotes the generalized inverse. $\text{vc}(\mathbf{x},\mathbf{y}) = \text{vech}(\mathbf{xy^T} + \mathbf{yx^T} - \text{diag}(\mathbf{x} \odot \mathbf{y}))$. Note that $\text{vc}(\mathbf{x},\mathbf{y})$ operator helps to represent equations of the form $\text{vec}(\mathbf{x^T A y})$ when $\mathbf{A}$ is symmetric, more concisely as $\text{vc}(\mathbf{x},\mathbf{y})^{\text{T}}.\text{vech}(\mathbf{A})$

Road Map In Section 2 we outline two different applications of existing NRF methods, which are relevant in our context. We first describe how NRF can be used to de-noise and fill in missing entries of a noisy and possibly incomplete data sequence. This is followed up with a brief overview of a straightforward way of using prior NRF methods, with our experiments that exposes some problems. Section 3 formalizes our notion of *shape* and describes how shape variability of an ensemble of data can be captured. Section 4 gives the details of a Sequential Quadratic Programming based constrained optimization scheme which couples NRF algorithm with the learned statistical data. We discuss our experiments and results in Section 5 and conclude in Section 6.

2 Non Rigid Factorization

Apart from structure from motion, factorization techniques can be applied to a wide range of application like data segmentation, data de-noising and data imputation. Data de-noising and imputation are of significant interest to us since the feature tracks from the off-the shelf trackers are

[1] There has been some recent work on extending factorization methods for articulated structures [11, 12]. But these methods require a very large number of features, whereas we work with a very sparse number of features and assume the human body to be a deforming object

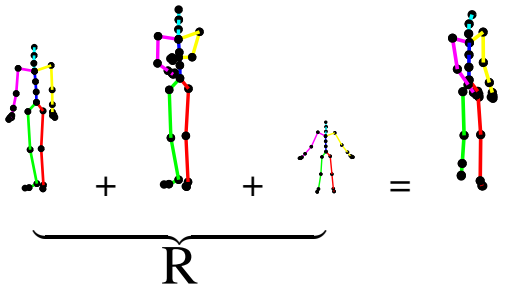


Figure 1: A pictorial representation of a *morphable model*. The right hand side is the actual data seen but can be obtained by modifying "basis" shapes.

typically noisy and contain missing information due to occlusion. The de-noising and structure recovering capability of the factorization algorithm is reviewed in this section.

The Basics: A popular representation for image formation (for either non-rigid or articulated objects) under orthographic or weak projective camera models is to write

$$\mathbf{W}_f = \mathbf{R}_f(\sum_{i=1}^{K} c_{fi}\mathbf{S}_i)$$

where $\mathbf{W}_f$ is the observed 2D feature in frame f (out of F given frames), $\mathbf{R}_f \in \mathbb{R}^{2\times 3}$ is the truncated row-orthonormal rotation matrix. K is the number of morph shapes needed to fully represent the object, $\mathbf{S}_i \in \mathbb{R}^{3\times P}$ the ith morph shapes (where P refers to the number of feature points tracked), and c_{fi}, the morph weights corresponding to $\mathbf{S}$ in the fth frame. This is pictorially represented in Fig. 1.

We build an *observation matrix* $\mathbf{W} \in \mathbb{R}^{2F\times P}$ by stacking the position of P landmark points observed in F frames. The structure of the observation matrix $\mathbf{W}$ appears in the left hand side of Eq. 1. Here (x_{ij}, y_{ij}) refers to the 2D co-ordinates of the point j in frame i.

$$\mathbf{P} = \begin{pmatrix} x_{11} & \cdots & x_{1P} \\ y_{11} & \cdots & y_{1P} \\ \vdots & \ldots & \vdots \\ x_{F1} & \cdots & x_{FP} \\ y_{F1} & \cdots & y_{FP} \end{pmatrix} = \mathbf{MS} = \underbrace{\begin{pmatrix} \mathbf{c}_1^T \otimes \mathbf{R}_1 \\ \vdots \\ \mathbf{c}_F^T \otimes \mathbf{R}_F \end{pmatrix}}_{2F\times 3K} \underbrace{\begin{pmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_F \end{pmatrix}}_{3K\times P} \tag{1}$$

This factorization can be performed modulo a *gauge factor* of $\mathbf{G} \in GL(3K, 3K)$ [8](§Sec.2.2) using SVD, if we assume an isotropic and Gaussian noise model[2]. But when there are outliers and missing data, which indeed is the case with most real-life measurements due to tracking failure and outliers, a straightforward SVD is no longer applicable.

2.1 Data denoising and missing information recovery

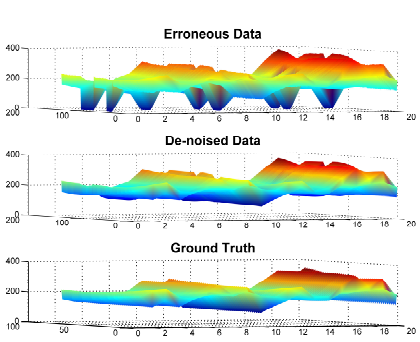
The most commonly used approach is to re-write the above problem with some robust ρ-function where the contribution of each item is weighted according to its fitness to the subspace [13, 14]. The modified factorization problem is now to compute the maximum likely estimator of a weighted L_2 norm cost function.

$$\varepsilon_{\text{mle}}(\tilde{\mathbf{M}}, \tilde{\mathbf{S}}) = ||\mathbf{W} \odot (\mathbf{P} - \tilde{\mathbf{M}}\tilde{\mathbf{S}})||_F^2 \tag{2}$$

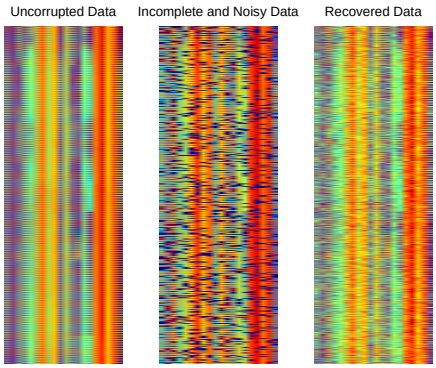
where $w_{ij} \geq 0$ is a weighing factor which specifies the uncertainty in $\mathbf{p}_{ij}$ and $w_{ij} = 0$ if $\mathbf{p}_{ij}$ is missing

The literature on factorization with missing data falls into several categories: close-form solutions, imputation methods, EM-akin alteration methods and direct non-linear minimization methods. An excellent comparative study between these various method can be found in [14].

[2]Note that though the factorization assumes that temporal dependices in the data are caught by the tracker, the rank constraint enforces another layer of weak and subtle constraint on the contunity of motion.



(a)



(b)

Figure 2: Surface and matrix plots (left and right hand side respectively) of noisy+incomplete data, de-noised data and Ground Truth. Notice that the recovered data has a high similarity to the ground truth

Our Denoising Method: We make use of the second order damped Newton algorithm introduced in [14] to de-noise the noisy point tracks. But we additionally perform modified Gram-Schmidt orthogonalization on the current estimate of both $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ at each iteration. Note that Eq. 2 does not impose any structure on $\tilde{\mathbf{M}}$ or $\tilde{\mathbf{S}}$, whereas SVD based solutions ensured that $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ are orthonormal and form good bases. We find that enforcing the orthonormality at each step makes the algorithm more numerically robust, rather than performing one single SVD toward the end. We initialize the optimization with left and right subspace estimate from a sparse SVD [15] of the incomplete data matrix. We weigh the visible features isotropically. These weights are estimated by contrasting the singular value spectrum of the sparse SVD with the mean value of a prior computed ensemble of spectrum of non-noisy, complete and typical data sets. The features which are not visible are assigned zero weights. A typical example is shown in Fig. 2(a). The deep trenches in the top figure corresponds to the missing data feature points. Observe that these valleys disappear after the de-noising step (middle figure). Moreover, the recovered data has a high similarity to the ground truth (bottom figure).

2.2 Recovering Motion and Shape

As mentioned earlier, unfortunately this factorization is not unique, but determinable only up to a non-singular linear corrective transformation $\mathbf{G}$ as

$$\mathbf{M} = \tilde{\mathbf{M}}.\mathbf{G} \qquad \mathbf{S} = \mathbf{G}^{-1}.\tilde{\mathbf{S}} \tag{3}$$

where we have the true scaled rotation matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$. The heart of the non-linear factorization algorithm lies in solving for this corrective transform $\mathbf{G} \in \mathbb{GL}^{(3K\times 3K)}$ as described briefly below.

Let $\mathbf{x}_f^T$ and $\mathbf{y}_f^T$ be the pair of rows in $\mathbf{M}$ which gives the projection for frame f. Notice that $\mathbf{M}$ is made up of blocks of 2×3 scaled rotation matrices. Hence rows of each of these 2×3 blocks must be orthogonal and of equal norm.

$$\begin{aligned}
\mathbf{x}_f^T\mathbf{y}_f = 0 \quad &\text{(orthogonality constraint)}\\
\tilde{\mathbf{x}}_f^T\mathbf{G}\mathbf{G}^T\tilde{\mathbf{y}}_\mathbf{f} = 0 \Rightarrow \mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f},\tilde{\mathbf{y}}_\mathrm{f})\mathrm{vech}(\mathbf{G}\mathbf{G}^\mathbf{T}) = 0 &\\
\mathbf{x}_f^T\mathbf{x}_f = \mathbf{y}_f^T\mathbf{y}_f \Rightarrow (\mathbf{x}_f-\mathbf{y}_f)^T(\mathbf{x}_f+\mathbf{y}_f) = 0 \quad &\text{(equal norm constraint)}\\
(\tilde{\mathbf{x}}_f-\tilde{\mathbf{y}}_f)^T\mathbf{G}\mathbf{G}^T(\tilde{\mathbf{x}}_f+\tilde{\mathbf{y}}_f) \Rightarrow \mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f}-\tilde{\mathbf{y}}_\mathrm{f},\tilde{\mathbf{x}}_\mathrm{f}+\tilde{\mathbf{y}}_\mathrm{f})\mathrm{vech}(\mathbf{G}\mathbf{G}^\mathbf{T}) = 0 &\\
\textit{Let } \mathbf{L} = [\mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f},\tilde{\mathbf{y}}_\mathrm{f}),\mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f}-\tilde{\mathbf{y}}_\mathrm{f},\tilde{\mathbf{x}}_\mathrm{f}+\tilde{\mathbf{y}}_\mathrm{f})]^T\forall f \text{ and } \mathbf{Q_A} = \mathbf{L}\mathbf{L}^T &
\end{aligned} \tag{4}$$

Note that $\mathbf{M}_{1:3} = \tilde{\mathbf{M}}\mathbf{G_{1:3}} \in \mathbb{R}^{2\mathrm{F}\times 3}$. It turns out that solving for $\mathbf{G}_{1:3}$ is sufficient to solve for the rest of $\mathbf{G}$ [10]. The vanilla NRF computes $\mathbf{G_{1:3}}\mathbf{G_{1:3}^T}$ that minimizes the sum squared deviation from orthogonality in the final motion matrix by least squares solving the system of equations given by

$$\mathrm{OrthErr}_{\mathbf{Q_A}}(\mathbf{G}_{1:3}) = \mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^\mathrm{T})^\mathrm{T}\mathbf{Q}_\mathrm{A}\mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^\mathrm{T}) \tag{5}$$

The symmetric matrix $\mathbf{G}_{1:3}\mathbf{G}_{1:3}$ is later decomposed to $\mathbf{G}_{1:3}$ by performing a rank-3 EVD ($\mathbf{G}_{1:3} = \mathbf{V}\Lambda^{0.5}$)

Significantly, it was recently shown [9] that these rotation constraints are not sufficient to *uniquely* solve for the corrective transform $\mathbf{G}$ for articulate and non-rigid motions. More specifically the general solution of the rotation constraints is $\mathbf{GHG}^T$, where $\mathbf{H}$ is the summation of an arbitrary block skew symmetric matrix and an arbitrary block scale identity matrix. The culprit being the redundancy in the constraint matrix which leaves the solution to Eq. 5 under-constrained. One way to overcome this ill-posedness is a heuristic scheme proposed by the authors of [9] where shapes in K frames are assumed to be independent and will act as a set of bases. Unfortunately, in general, this is not a good practice, since it tries to represent the shape space non-parsimoniously with a finite set of local diffeomorphisms, and hence has questionable subspace representation ability [16].

An alternative appears in [10] where Brand makes another relevant observations that approximation of Eq. 5 as a nested *linear* least square solution doesn't do justice to the physical reality. It overlooks a lot of co-variance information encoded in $\mathbf{Q}_\mathbf{A}$. Instead, the author solves $\mathbf{G}_{1:3}$ directly from Eq. 5 using a variant of first order line search global optimization framework (the step sizes are calculated by direct root finding). But, our experiments showed that the error surfaces generally have a rough terrain and many a times converge to the dreaded local minima. An example is show in Fig. 3.

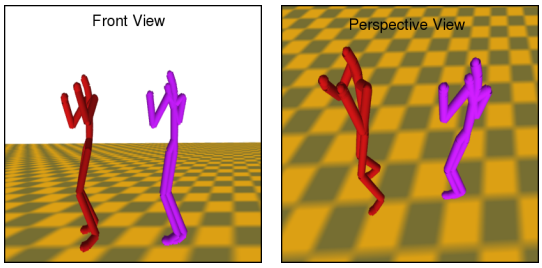


Figure 3: Non-rigid factorization algorithms have the tendency to flatten the body structure (notice the legs). The red colored human model is the representation of the actual data and the pink colored model is a reconstruction from 2D data.

The vanilla NRF, does not make any assumption about the shape of the object in scene. But a huge chunk of vision related engineering problems (in our case human pose extraction) do allow us to make *valid* assumption regarding object shape subspaces and possibly get an estimate of the subspace apriori. In the next section we describe how a good prior estimate of shape subspace can be obtained.

3 Shape Analysis

The word "shape" is very commonly used in everyday language, usually referring to the appearance of an object. Following Kendall [17] the definition of shape that we consider is:

> *Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object*

Important aspects of shape analysis are to obtain a measure of distance between shapes, to estimate average shapes from a random sample and to estimate shape variability from a random sample.

Procrustes analysis involves matching configurations with similarity transformations to be as close as possible according to Euclidean distance, using least squares techniques. More formally, given two mean centered configuration matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, the *full Procrustes distance* between $\mathbf{X}_1$ and $\mathbf{X}_2$ is

$$\mathrm{D}_{\mathrm{pro}} = \inf_{\Gamma \in SO(3), \beta \in \mathbb{R}} ||\mathbf{Z}_2 - \beta \mathbf{Z}_1 \Gamma||$$

where $\mathbf{X}_r = \mathbf{Z}_r / ||\mathbf{X}_r||, r = 1,2$

Similarly, the full Procrustes estimate of mean shape $[\hat{\mu}]$ is obtained by minimizing (over μ) the sum of square full Procrustes distance from each $\mathbf{X}_i$ to an unknown unit mean configuration μ, i.e

$$[\hat{\mu}] = \arg\inf_{\mu} \sum_{i=1}^{n} d_F^2(\mathbf{X}_i, \mu)$$

For a more detailed exposition, we refer the readers to [18] and the original work of Kendall [17]

3.1 Creating The Reference Pre-Shape

In the last decade or so, principal component analysis (PCA) has become a favorite tool for computer vision and graphics researchers [19, 20]. PCA is a simple, yet powerful technique to collect and investigate the statistically variability of data which resides in linear spaces ($\mathbb{R}^3$ in our case). To learn a good set of bases we need a corpus of accurate data with wide variability, which now a days is publicly available in the form of archival motion capture data.

Each pose is parametrized as a single observation 60 dimensional column vector ($\text{vec}(\mathbf{Q}_{\text{train}})$) containing the Euclidean positional information of all the land mark points[3]. We borrow techniques from Procrustes Analysis introduced in the previous section to strip these vectors of positional, scale, and orientation details.

If $\hat{\mu}$ be a pre-shape corresponding to the full Procrustes mean shape, the aligned vectors can be computed as

$$v_F = (\mathbf{1} - \text{vec}(\hat{\mu}\hat{\mu}^T))\text{vec}(\hat{\beta}_i \mathbf{Q}_{\text{train}} \hat{\Gamma}_i)$$

These aligned vectors are stacked into a data matrix $\mathbf{X}_{\text{mocap}}$ and we compute the principal components of this data matrix. PCA performs a basis transformation to an orthogonal co-ordinate system formed by the eigen vectors $\mathbf{V}_i$ of the covariance matrix of $\mathbf{X}_{\text{mocap}}$. These orthogonal components are ordered with respect to the descending values of their eigenvalues and are arranged into $\mathbf{S}_{\text{ref}}$. We call $\mathbf{S}_{\text{ref}}$ as *Reference Pre-shape*. For a full body motion with just 5 bases we are able to represent more than 94% variation in the data.

4 Constrained Factorization

The primal idea behind our method is that shapes recovered by the NRF should having significant similarity to the pre-learned *Reference Pre-Shape*. We express this as a constrained non-linear programming problem.

More formally, we rewrite Eq. 5 as

$$\begin{array}{rcl} E(G_{1:3}) & = & \text{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^T)^T \mathbf{Q}_A \text{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^T) \\ S.T & & \text{trace}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^T) = 1 \\ & & D_{\text{pro}}^2(\mathbf{G}_{1:3}, \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}) \leq -d \end{array} \tag{6}$$

where $D_{\text{pro}}(\mathbf{X}, \mathbf{Y})$ gives the orthogonal Procrustes distance between $\mathbf{X}$ and $\mathbf{Y}$ and d is an user-set parameter, which specifies the tolerance level for the structural variation and defines the feasible area or the domain of the cost function (smaller the tolerance, narrower the feasible area) . In our experiments we used 0.2 as the threshold. Though it is tempting to decrease the tolerance, lesser tolerance makes the algorithm more prone to over-fitting (especially if the training set is not exhaustive enough).

Notice that both our cost function and constraints are non-linear. While the cost function is quartic, the constraints are of quadratic nature. Though constrained non-linear optimization (in general) is still an open problem, many efficient, but approximate numerical schemes exist [21] especially for relatively lower order cost function (quartic, in our case) and near linear constraint functions (quadratic). We make use of *Sequential Quadratic Programming*, a well known and used numerical solution for optimizing smooth non-linear cost functions under smooth non-linear constraints [21,22]. It is Newton like in that it requires second derivatives of the cost function and potentially provides quadratic convergence.

The goal is to extremize a scalar cost function $E(\mathbf{x})$ subject to a vector of constraints $\mathbf{c}(\mathbf{x}) \leq 0$. (Note that inequality constraints can be treated at par with the equality constraint by assuming its

[3]Note that the ordering (or the meta-knowledge about it) of this vector has to be consistent with the 2D observation vector.

respective Lagrange multiplier vanishes whenever the inequality is not strict [21] and is strictly positive whenever the inequality is strict). Lagrange multipliers λ give an implicit solution.

$$\bigtriangledown E + \lambda \bigtriangledown \mathbf{c} = \mathbf{0} \text{ with } \mathbf{c}(\mathbf{x}) = \mathbf{0}$$

We resolve this iteratively starting from some initial guess b$\mathbf{x_0}$. We approximate the cost to second order and the constraints to first order at $\mathbf{x_0}$, giving a quadratic optimization sub-problem with linear constraints.

$$\min_{\delta \mathbf{x}} \left(\bigtriangledown E.\delta x + \frac{1}{2} \delta \mathbf{x}^T \bigtriangledown^2 f.\delta \mathbf{x} \right) |_{\mathbf{c}+\bigtriangledown \mathbf{c}.\delta \mathbf{x}}$$

This sub-problem has an exact linear solution

$$\begin{pmatrix} \bigtriangledown^2 E & \bigtriangledown \mathbf{c}^T \\ \bigtriangledown \mathbf{c} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \mathbf{x} \\ \lambda \end{pmatrix} = - \begin{pmatrix} \bigtriangledown \mathbf{E} \\ \mathbf{c} \end{pmatrix} \tag{7}$$

We solve for $\delta \mathbf{x}$, update $\mathbf{x}_0$ to $\mathbf{x}_1 = \mathbf{x}_0 + \delta \mathbf{x}$, re-estimate derivatives and iterate to converge.

The first order and second order derivatives of the Lagrange function in Eq. 6 are given in Appendix A.

5 Experiments

Training Data: We use 700 frames from motion capture data included in the HumanEva dataset [5] for learning the pre-shapes. These frames are selected by randomly sampling from the training set provided in the dataset. Selected frames span poses from various set of human action like walking, boxing, making hand gestures etc.

Testing: We test the performance of our algorithm on synthetic data with ground truth included in the testing set of the HumanEva dataset, as well as videos which give us only 2D information.

Motion-Capture Based Synthetic Data: In any choice of motion clip (from the motion capture data base) we know the 3D positions. We synthetically created a two dimensional projection by randomly choosing a center of projection. To simulate tracking errors and the like, the resulting "features" are further corrupted by adding Gaussian noise and frames dropped randomly to simulate quantifiable error and occlusion errors in the tracking process. This constituted the process of creating the observation matrix. The incomplete and noisy observation matrix is denoised using the method described in Section 2. Recall that the output of factorization is only accurate up-to an arbitrary rotation and scale. So the error at each frame is defined to be the Procrustes distance from the recovered orientation to the ground truth. We compare the performance of our algorithm to that of the unconstrained case [10].

Fig. 4 shows ground truth (left) contrasted with the output of our algorithm (middle) and the unconstrained case (right). The recovered pose by the unconstrained algorithm is nearly planar (notice the stick figure's left arm piercing its torso). The newly introduced boundary conditions ensured that the recovered solution did not collapse into a degenerate solution unlike the unconstrained state, and is found to be quite similar to the ground truth.

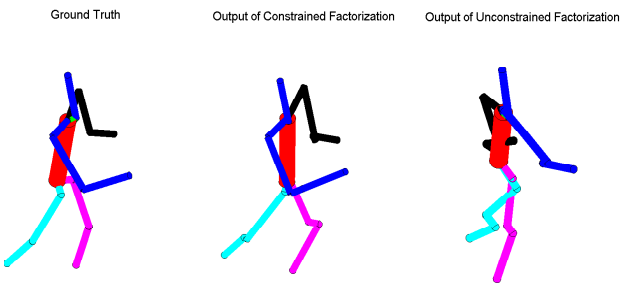
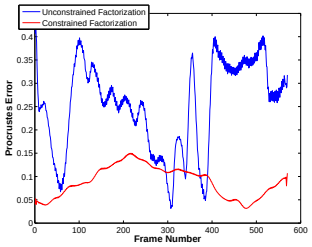


Figure 4: Ground truth data contrasted with the output from the constrained (our method) and non-constrained factorization (prior method) respectively.

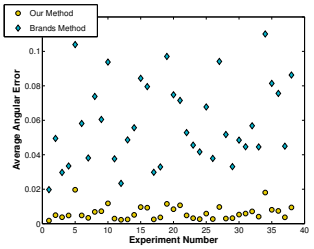
Next, we compared the performance of both algorithms over a novel long sequence (show in Fig. 5(a)). This sequence is novel in that it was not used for the computation of the reference

pre-shape. We selected a complicated clip of a boxing motion consisting of 577 frames sampled at 30Hz. The data is corrupted with 10% additive Gaussian noise and around 15% of its observations are masked out. Note that average performance of the constrained factorization algorithm hovers around the 5–15% reconstruction error mark. One interesting variation in the plot is that occasionally (frame numbers 290–320, 348–355 and 380–395) the error of the unconstrained algorithm dips somewhat below that of its constrained counter part (our method). The reason for this unexpected better performance is that during these frames, the actor is assuming a near planar pose and the degenerate shape base extracted by the unconstrained factorization algorithm is better able to explain these frames. Nevertheless, the unconstrained algorithm rapidly loses accuracy in the more common situation, when the actor resumes his or her flexible movements.

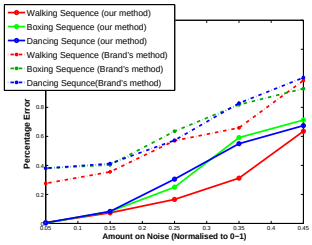
The scatter diagram in Fig. 5(b) plots the average error recorded by the constrained factorization algorithm (shown in yellow) and its unconstrained counterpart (shown in cyan) for various data input (a total of 39 different inputs). Each of the data input was seeded with 2% additive Gaussian error, and no occlusion condition was assumed. While carrying out these experiments we further assumed that the inequality constraints are strict. Fig. 5(c) shows the performance of both the version of the algorithm with three different sequence (walking , boxing and running) when subjected to different amount of synthetic noise. While the dotted line records the performance of the unconstrained version of the algorithm, the regular line record that of the constrained one. Walking, Boxing and Dancing motion sequence are represented by the red, green and blue lines respectively. Superior performance by the constrained version of the algorithm is amply recorded in every experiment.



(a) Error comparison of constrained and unconstrained NFM over a 577 frames boxing sequence



(b) Average error of constrained and unconstrained NFM from different experiments



(c) The performance of the algorithm with various amount of noise levels

Figure 5: Comparative Performance Evaluation

5.1 Data With No Ground Truth

In this experiment an 80 frame video sequence was semi-automatically tracked using the KLT based tracker. We hand picked the features which conformed to the anatomically relevant landmark points. We re-picked the lost features after every 10 frames. Note that far superior tracking schemes exist [23] for tracking humans from video. The purpose of this experiment was to test the performance under non-linear error models which often appear in real data sequences. Two different 'pigeon' views of the recovered orientation of the actor is shown along with actual data is show in Fig. 6. As a post-processing step, the recovered data is smoothed out using a Kalman smoother. More output including the video of the just explained experiment can be found at `http://www.cse.iitb.ac.in/appu/bmvc07/`

6 Conclusion and Future Work

We have given a novel constrained non-rigid factorization algorithm that extracts 3D human poses from 2D video sequences. Both qualitative and quantitative results were provided. Note that our method can be applied to any deforming data sequences (apart from human motion), provided accurate motion capture or similar high precision quantized data exists.

Future Work: The strength and weakness of factorization based techniques lies in its block based nature. This potentially rules out any online scheme. We are currently exploring the possibility of having a windowed scheme, thereby making the algorithm semi-online. We are also considering having an iterative refinement of reference pre-shape, hence equipping the algorithm

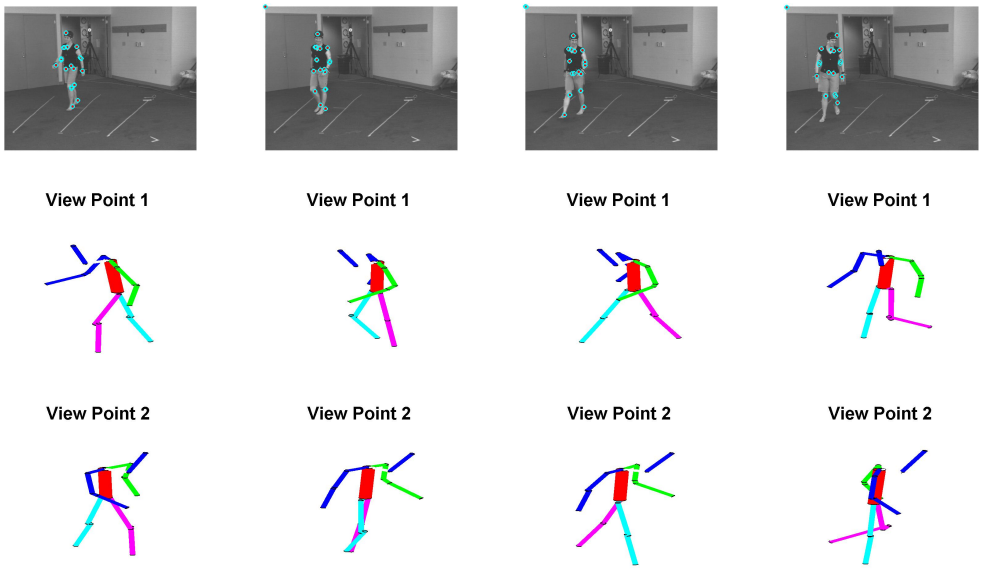


Figure 6: The top row shows the raw frames with features overlayed. The middle and bottom shows the recovered 3d pose rendered from two novel view points. The front view is identical and not shown.

to handle non-stationary data, and previously unseen data. Another possibility we wish to explore is to merge the optimization given in Eq. 2 and Eq. 6 as a single optimization problem.

A Derivatives

The corresponding Lagrange function of Eq. 6 can be written as

$$\begin{aligned}\mathscr{L} = \operatorname{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\mathrm{T}})^{\mathrm{T}}\mathbf{Q}_{\mathrm{A}}\operatorname{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\mathrm{T}}) + \lambda(\operatorname{vec}(\mathbf{G}_{1:3})^{\mathrm{T}}\operatorname{vec}(\mathbf{G}_{1:3}) - 1) \\ + \mu(\operatorname{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma)^{\mathrm{T}}\operatorname{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma) - \mathrm{d}) \quad (8)\end{aligned}$$

where $\Gamma_i \in \mathbb{SO}(3)$. Let $\mathbf{Z} = \mathbf{G}_{1:3}$ and $\mathbf{J}_{ij} \in \{0,1\}^{3K\times 3}$ is all zeros except for element $J_{ij} = 1$

$$\begin{aligned}\frac{\partial\mathscr{L}(\mathbf{Z},\lambda,\mu)}{\partial Z_{ij}} =& 2\operatorname{vech}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{\mathrm{T}}\mathbf{Q}_{\mathrm{A}}\operatorname{vech}(\mathbf{Z}\mathbf{J}_{\mathrm{ij}}^{\mathrm{T}} + \mathbf{J}_{\mathrm{ij}}\mathbf{Z}^{\mathrm{T}}) + \lambda\operatorname{vec}(\mathbf{Z}\mathbf{J}_{\mathrm{ij}}^{\mathrm{T}} + \mathbf{J}_{\mathrm{ij}}\mathbf{Z}^{\mathrm{T}}) \\ &+ \mu(\operatorname{vec}((\mathbf{Z} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma)\mathbf{J}_{\mathrm{ij}}^{\mathrm{T}} + \mathbf{J}_{\mathrm{ij}}\operatorname{vec}((\mathbf{Z} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma)^{\mathrm{T}}) \\ \frac{\partial\mathscr{L}(\mathbf{Z},,\lambda,\mu)}{\partial\lambda} =& \operatorname{vec}(\mathbf{G}_{1:3})^{\mathrm{T}}\operatorname{vec}(\mathbf{G}_{1:3}) \\ \frac{\partial\mathscr{L}(\mathbf{Z},,\lambda,\mu)}{\partial\mu} =& \operatorname{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma)^{\mathrm{T}}\operatorname{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}\Gamma) \\ \frac{\partial\mathscr{L}(\mathbf{Z},\lambda,\mu)}{\partial Z_{\mathrm{ij}}Z_{\mathrm{kl}}} =& 2.\operatorname{vech}(\mathbf{Z}\mathbf{J}_{\mathrm{kl}}^{\mathrm{T}}) + \mathbf{J}_{\mathrm{kl}}\mathbf{Z}^{\mathrm{T}})\mathbf{Q}_{\mathbf{A}}.\operatorname{vech}(\mathbf{Z}\mathbf{J}_{\mathrm{ij}}^{\mathrm{T}} + \mathbf{J}_{\mathrm{ij}}\mathbf{Z}^{\mathrm{T}}) \\ &+ (\operatorname{vech}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{\mathrm{T}}\mathbf{Q}_{\mathrm{A}} + \lambda + \mu)\operatorname{vech}(\mathbf{J}_{\mathrm{kl}}\mathbf{J}_{\mathrm{ij}}^{\mathrm{T}} + \mathbf{J}_{\mathrm{ij}}\mathbf{J}_{\mathrm{kl}}^{\mathrm{T}})\end{aligned} \quad (9)$$

References

[1] Soatto, S., Brockett, R.: Optimal structure from motion: Local ambiguites and global estimates. In: CVPR. (1998) 282–288

[2] Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Transactions on Pattern Analysis & Machine Intelligence **28**(1) (2006)

[3] Sigal, L., Black, M.J.: Predicting 3d people from 2d pictures. In: Articulated Motion and Deformable Objects, 4th International Conference. (2006) 185–195

[4] Taylor, C.J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. Computer Vision and Image Understanding **80**(3) (2000) 349–363

[5] Sigal, L., Black, M.J.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Dept. of Computer Science, Brown University, Providence, Rhode Island 02912 (2006)

[6] Ma, Y., Soatto, S., Košecká, J., Sastry, S.: An Invitation to 3-D Vision. From Images to Geometric Models. Springer (2004)

[7] Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision (1992) 137–154

[8] Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE CVPR. (2000) 690–696

[9] Xiao, J., Chai, J.X., Kanade, T.: A closed form solution to non-rigid shape and motion recovery. In: ECCV. (2004) 573–587

[10] Brand, M.: A direct method of 3D factorization of nonrigid motion observed in 2D. In: Computer Vision and Pattern Recognition. (2005) 122–128

[11] Yan, J., Pollefeys, M.: A factorization-based approach to articulated motion recovery. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, Washington, DC, USA, IEEE Computer Society (2005) 815–821

[12] Tresadern, P., Reid, I.: Articulated structure from motion by factorization. In: Proc. 23nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego. (2005)

[13] Brandt, S.: Closed-form solutions for affine reconstruction under missing data. In: European Conference on Computer Vision, Springer-Verlag, 2002 (2002) 109–114

[14] Buchanan, A., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. In: CVPR. Volume 2. (2005) 316–322

[15] Larsen, R.: Lanczos bidiagonalization with partial reorthogonalization. PhD thesis, Dept. Computer Science, University of Aarhus, DK-8000 Aarhus C, Denmark, (1998)

[16] Soatto, S., Yezzi, A.J.: DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In: ECCV (3). (2002) 32–57

[17] Kendall, D.: Shape manifolds, procrustean metrics and complex projective spaces. Statistical Science **16** (1984) 81 – 121

[18] Dryden, I., Mardia, K.: Statistical Shape Analysis. Number ISBN 0-471-95816-6 in Wiley series in proability and Statistics. John Wiley and Sons (1998)

[19] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models and their training and application. Computer Vision and Image Understanding **61**(1) (1995) 38–59

[20] Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH, ACM Transaction on Graphics, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1999) 187–194

[21] Fletcher, R.: Practical Methods of Optimization. 2nd edition edn. John Wiley & Sons (1987)

[22] Triggs, B., McLauchlan, P., Hartley, R.I., A.W., F.: Bundle adjustment - a modern synthesis. In Triggs, B., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, Springer (1999) 298–373

[23] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision **61**(1) (2005) 55–79