Iterative Extensions of the Sturm/Triggs Algorithm: Convergence and Nonconvergence

John Oliensis[1] and Richard Hartley[2,⋆]

[1] Department of Computer Science, Stevens Institute of Technology,
Castle Point on Hudson, Hoboken, NJ 07030
[2] Australian National University and National ICT Australia

Abstract. We show that SIESTA, the simplest iterative extension of the Sturm/Triggs algorithm, descends an error function. However, we prove that SIESTA does not converge to usable results. The iterative extension of Mahamud et al. has similar problems, and experiments with "balanced" iterations show that they can fail to converge. We present CIESTA, an algorithm which avoids these problems. It is identical to SIESTA except for one extra, simple stage of computation. We prove that CIESTA descends an error and approaches fixed points. Under weak assumptions, it converges. The CIESTA error can be minimized using a standard descent method such as Gauss–Newton, combining quadratic convergence with the advantage of minimizing in the projective depths.

1 Introduction

The Sturm/Triggs (**ST**) algorithm [9] is a popular example of the factorization strategy [10] for estimating 3D structure and camera matrices from a collection of matched images. The factorization part of the algorithm needs starting estimates of the *projective depths* λ_n^i, which [9] obtained originally from image pairs. After [9], researchers noted that the λ_n^i can be taken equal or close to 1 for important classes of camera motions [11][1][5]. For these motions, the algorithm becomes almost a direct method, since it computes the structure/cameras directly from the λ_n^i whose values are approximately known.

To improve the results of **ST**, several researchers proposed iterative extensions of the method which: initialize the λ_n^i (typically) at 1, estimate the structure/cameras, use these estimates to recompute the λ_n^i, use the new λ_n^i to recompute the structure/cameras, etc. [11][1][5][8][4]. One common use is for initializing bundle adjustment [4]; for example, a few iterations can extend an affine estimate computed via Tomasi/Kanade [10] to a projective initialization. The iteration often gives much faster initial convergence than bundle adjustment does [2]. Variant iterative extensions include [1][5][8][4]. Notably, [4] recommends adding a "balancing" step [9] following the computation of the λ_n^i to readjust

[⋆] National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts And the Australian Research Council through Backing Australia's Ability And the ICT Research Centre of Excellence programs.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part IV, LNCS 3954, pp. 214–227, 2006.

their values toward 1. This keeps the λ_n^i near the correct values (for many classes of motions) and also reduces the bias of the estimates [4].

This paper discusses the convergence of these iterations. Our theorems and experiments show that the versions without balancing do not converge sensibly and that the balanced iteration [4] can fail to converge. We propose CIESTA, a simple algorithm which avoids these problems. We prove that CIESTA descends an error function, that it iterates toward a "best achievable" estimate, and that these "best" estimates are stationary points of the error. CIESTA extends **ST** to a sound iteration, replacing balancing with regularization. Since CIESTA descends a known error function, it can be replaced by a standard descent method such as Gauss–Newton, combining quadratic convergence with the advantage of minimizing in the projective depths.

Notation. Given N quantities ζ_a indexed by a, we use $\{\zeta\} \in \Re^N$ to denote the column vector whose ath element is ζ_a. if $A \in \Re^{M\times N}$ is a matrix, we define $\{A\} \in \Re^{MN}$ as the column vector obtained by concatenating the columns of A.

For multiview geometry, we use the notation of [4]. Let $\mathbf{X}_n \equiv \left(X_n; Y_n; Z_n; 1\right) \in \Re^4$ represent the homogenous coordinates of the nth 3D point (we use ';' to indicate a column vector), with $n = 1, 2, \ldots, N_p$, and let $\mathbf{x}_n^i \equiv \left(x_n^i; y_n^i; 1\right) \in \Re^3$ be its homogenous image in the ith image, for $i = 1, \ldots N_I$. Let $M^i \in \Re^{3\times 4}$ be the ith *camera matrix*, and let $\mathcal{M} \in \Re^{3N_I\times 4}$ consist of the M^i concatenated one on top of the other. Define the *structure matrix* $\mathcal{X} \in \Re^{4\times N_p}$ so that its nth column $\mathcal{X}_n$ is proportional to $\mathbf{X}_n$. Neglecting noise, we have $\lambda_n^i \mathbf{x}_n^i = M^i \mathcal{X}_n$, where the constants λ_n^i are the *projective depths*. We use $\lambda \in \Re^{N_I N_p}$ to denote the vector of all the projective depths ordered in the natural way. Let $\mathcal{W} = \mathcal{W}(\lambda) \in \Re^{3N_I\times N_p}$ be the scaled data matrix consisting of the $\mathbf{x}_n^i$ multiplied by the projective depths, with $\mathcal{W}_n^{(3i-2):3i} = \lambda_n^i \mathbf{x}_n^i$. **ST** exploits the fact that, for known λ_j^i and zero noise, the matrix $\mathcal{W}$ has rank ≤ 4 and factors into a camera matrix times a structure matrix.

2 Simplest Iterative Extension of the ST Algorithm

Let $\hat{\mathcal{W}}(\lambda)$ be a matrix with rank ≤ 4 that gives the best approximation to $\mathcal{W}(\lambda)$ under the Frobenius norm: $\hat{\mathcal{W}}(\lambda) \equiv \arg\min_{\mathrm{rank}(Y)\leq 4} \|\mathcal{W}(\lambda) - Y\|$. Given the SVD $W(\lambda) = UDV^T$, we have the standard result $\hat{\mathcal{W}}(\lambda) = U\hat{D}V^T$, where $\hat{D}$ is obtained from D by zeroing all but the first four diagonal entries.

SIESTA repeatedly adjusts the λ_n^i to make the scaled data matrix $\mathcal{W}$ closer to rank 4. Let $\lambda^{(k)}$ and $\mathcal{W}^{(k)} \equiv \mathcal{W}\left(\lambda^{(k)}\right)$ give the estimates of the λ_n^i and $\mathcal{W}$ in the the kth iteration. The algorithm is:

- **Initialize** the λ_n^i**.** By default we set all the $\lambda_n^{i(0)}$ to 1.
- **Iteration** k**, stage 1:** Given the scaled data matrix $\mathcal{W}^{(k-1)} \equiv \mathcal{W}\left(\lambda^{(k-1)}\right)$, compute its best rank ≤ 4 approximation $\hat{\mathcal{W}}$. Set $\hat{\mathcal{W}}^{(k-1)} = \hat{\mathcal{W}}$.
- **Iteration** k**, stage 2:** Given $\hat{\mathcal{W}}^{(k-1)}$, choose $\lambda^{(k)}$ to give the closest matrix of the form $\mathcal{W}(\lambda^{(k)})$, that is, $\lambda^{(k)} = \arg\min_\lambda \|\mathcal{W}(\lambda) - \hat{\mathcal{W}}^{(k-1)}\|$.

Remark 1. The SIESTA algorithm has a simple interpretation if one thinks of the $\mathcal{W}^{(k)}$ and $\hat{\mathcal{W}^{(k)}}$ as points in $\Re^{3N_I N_p}$. For fixed image points $\mathbf{x}_n^i$, the set of all $\{\mathcal{W}(\lambda)\}$ is a linear subspace of $\Re^{3N_I N_p}$ which has dimension $N_I N_p$ since its points are indexed by the $N_I N_p$ projective depths. We denote it by $\mathcal{L}^{N_I N_p}$. Let $\hat{\Omega}$ denote the set of all points $\{\hat{\mathcal{W}}\}$ in $\Re^{3N_I N_p}$ coming from matrices $\hat{\mathcal{W}} \in \Re^{3N_I \times N_p}$ of rank ≤ 4. The SIESTA iteration can be rewritten as:

- **Stage 1:** Given $\left\{\mathcal{W}^{(k-1)}\right\}$, find the closest point $\left\{\hat{\mathcal{W}}^{(k-1)}\right\}$ from the set $\hat{\Omega}$.
- **Stage 2:** Given $\left\{\hat{\mathcal{W}}^{(k-1)}\right\}$, find the closest point $\left\{\mathcal{W}^{(k)}\right\}$ from $\mathcal{L}^{N_I N_p}$.

Next we show that each SIESTA iteration "improves" the reconstruction.

Definition 1. *Define* $E(\mathcal{W}, Y) \equiv \|\mathcal{W} - Y\|/\|\mathcal{W}\|$ *and*

$$\hat{E}(\lambda) \equiv \min_{\operatorname{rank}(Y) \leq 4} E(\mathcal{W}(\lambda), Y) = E(\mathcal{W}, \hat{\mathcal{W}}) \qquad \textit{(SIESTA error).}$$

The SIESTA error $\hat{E}$ measures the fractional size of the non-rank 4 part of $\mathcal{W}$.

Proposition 1. *The SIESTA error* $\hat{E}\left(\lambda^{(k)}\right)$ *is nonincreasing with* k.

Proof (sketch). Let $\theta^{(k)} \equiv \theta(\mathcal{W}^{(k)}, \hat{\mathcal{W}}^{(k)})$ give the angle between the matrices $\mathcal{W}^{(k)}$ and $\hat{\mathcal{W}}^{(k)}$ considered as vectors in $R^{3N_I N_p}$. Its sine relates to the error $\hat{E}$:

$$\sin^2(\theta^{(k)}) = \left|\{\mathcal{W}^{(k)}\} - \{\hat{\mathcal{W}}^{(k)}\}\right|^2 / \left|\{\mathcal{W}^{(k)}\}\right|^2 = \hat{E}\left(\lambda^{(k)}\right). \tag{1}$$

SIESTA starts with a point in $\mathcal{L}^{N_I N_p}$, finds the closest point from $\hat{\Omega}$, finds the closest point to this from $\mathcal{L}^{N_I N_p}$, etc. Since it computes the best approximation each time, the angle between the two latest estimates from $\hat{\Omega}$ and $\mathcal{L}^{N_I N_p}$ is nonincreasing, so $\theta^{(k)}$ and $\hat{E}$ are nonincreasing. ∎

Discussion. Our result justifies the practice of applying a few iterations of SIESTA to extend an affine estimate based on $\lambda_n^i = 1$ to a projective one, which can be used to start bundle adjustment. Although we show below that SIESTA does not converge correctly, this is not be a fatal flaw, since the drift away from good estimates is extraordinarily slow and hence correctable.

3 Convergence Problems for Iterative Factorization

3.1 SIESTA Fails to Converge

Trivial minima. We begin by describing trivial minima. If we choose the λ_n^i zero except in four columns, the matrix $\mathcal{W}(\lambda)$ will have all columns but four composed of zeros. Then $\mathcal{W}(\lambda)$ will have rank ≤ 4, and the error $\hat{E}(\lambda) = E(\mathcal{W}(\lambda), \hat{\mathcal{W}}(\lambda)) = 0$ because $\mathcal{W}(\lambda)$ and its closest rank ≤ 4 matrix $\hat{\mathcal{W}}(\lambda)$ are equal.

This set of λ_n^i gives a *trivial minimum* of the SIESTA error. Choosing all the λ_n^i zero except in one row also gives a trivial minimum. Trivial minima are of no interest, since they don't give reasonable interpretations of the data. Unfortunately, the proposition below shows that unless a non-trivial solution exists with exactly zero error (meaning that the data admits a noise-free solution), then the SIESTA algorithm must approach a trivial minimum, or possibly, in rare circumstances, a saddle point of the error. Experiments on small problems show that the algorithm approaches trivial minima, though extremely slowly.

Proposition 2. *Every local minimum of the SIESTA error $\hat{E}$ is a global minimum with zero error.*

Proof. We can assume without loss of generality that every 3D point has nonzero λ_n^i in some images, since otherwise we can eliminate these points and apply the argument below to the remaining set of points.

We suppose that the SIESTA error $\hat{E}$ has a local minimum at λ. Let $\mathcal{W} = \mathcal{W}(\lambda)$ be the corresponding scaled data matrix. By the assumption just above, $\mathcal{W}$ has no columns consisting entirely of zeros. Under these two conditions, we will show that the error equals zero or, equivalently, that $\mathcal{W}$ has rank ≤ 4.

Consider a transformation that perturbs a matrix by multiplying its n-th column by a value s. We denote this transformation by $\tau_{n\kappa}$ where $\kappa = s^2 - 1$. The reason for introducing the variable κ is that the subsequent computations simplify when expressed in terms of κ. For $\kappa = 0$, the transformation $\tau_{n\kappa}$ is the identity transformation and leaves the original matrix unchanged. It is evident that applying $\tau_{n\kappa}$ to the matrix $\mathcal{W} = \mathcal{W}(\lambda)$ is equivalent to multiplying the nth column of the projective depths λ_n^i by s, so we can write $\tau_{nk}(\mathcal{W}) = \mathcal{W}(\lambda^{\tau_{nk}})$, where $\lambda^{\tau_{nk}}$ equals λ except for the appropriate scaling of the nth column.

For the remainder of the proof, we write simply $\mathcal{W}$, omitting the dependence on λ. We denote $\tau_{n\kappa}(\mathcal{W})$ by $\mathcal{W}^\tau$, and the nearest[1] rank ≤ 4 matrix to $\mathcal{W}^\tau$ by $\widehat{\mathcal{W}^\tau}$. Recall that, similarly, $\hat{\mathcal{W}}$ is the closest matrix to $\mathcal{W}$ having rank ≤ 4. We may also apply the transformation $\tau_{n\kappa}$ to $\hat{\mathcal{W}}$, resulting in a matrix $(\hat{\mathcal{W}})^\tau = \tau_{n\kappa}(\hat{\mathcal{W}})$. This matrix has the same rank as $\hat{\mathcal{W}}$ for $s \neq 0$ and hence has rank ≤ 4, but, as we shall see, it is in general distinct from $\widehat{\mathcal{W}^\tau}$. It is important to understand the difference between $(\hat{\mathcal{W}})^\tau$ and $\widehat{\mathcal{W}^\tau}$.

As a first step, we show (under our assumptions above) that *any* $\kappa \neq 0$ gives

$$\hat{E}(\mathcal{W}^\tau) \equiv E(\mathcal{W}^\tau, \widehat{\mathcal{W}^\tau}) \leq E(\mathcal{W}^\tau, (\hat{\mathcal{W}})^\tau) = E(\mathcal{W}, \hat{\mathcal{W}}) \equiv \hat{E}(\mathcal{W}) \ . \tag{2}$$

The inequality in (2) follows simply from the definition of the error E and the fact that $\widehat{\mathcal{W}^\tau}$ is the closest matrix to $\mathcal{W}^\tau$ having rank ≤ 4. Consider the equality $E(\mathcal{W}^\tau, (\hat{\mathcal{W}})^\tau) = E(\mathcal{W}, \hat{\mathcal{W}})$. Noting that $\mathcal{W}$ and $\mathcal{W}^\tau$ differ only in the overall scale of their nth columns, we may compute

$$E(\mathcal{W}^\tau, (\hat{\mathcal{W}})^\tau) = (\kappa|\mathcal{R}_n|^2 + \|\mathcal{R}\|^2)/(\kappa|\mathcal{W}_n|^2 + \|\mathcal{W}\|^2), \tag{3}$$

where $\mathcal{R} = \mathcal{W} - \hat{\mathcal{W}}$, and $\mathcal{R}_n$ and $\mathcal{W}_n$ are the nth columns of $\mathcal{R}$ and $\mathcal{W}$. Under our assumption that $\mathcal{W}$ gives a local minimum, the derivative of this expression

[1] The nearest matrix need not be unique.

with respect to κ must be zero. Computing the derivative at $\kappa = 0$, and setting the numerator to zero leads to $\|\mathcal{W}\|^2 \; |\mathcal{R}_n|^2 - \|\mathcal{R}\|^2 \; |\mathcal{W}_n|^2 = 0$, which gives

$$|\mathcal{R}_n|^2/|\mathcal{W}_n|^2 = \|\mathcal{R}\|^2/\|\mathcal{W}\|^2, \tag{4}$$

i.e., the left–hand ratio has the same value for any n. After substituting in (3),

$$E(\mathcal{W}^\tau, (\hat{\mathcal{W}})^\tau) = \|\mathcal{R}\|^2/\|\mathcal{W}\|^2 = E(\mathcal{W}, \hat{\mathcal{W}}) \tag{5}$$

for all values of κ, as required. This proves (2).

Suppose we could make the inequality in (2) strict for arbitrarily small values of κ. In fact, we cannot do this, since if we could the error $\hat{E}$ would be strictly decreasing at λ and $\mathcal{W}(\lambda)$ rather than having a local minimum as assumed. Therefore, for all κ less than some small value, we have the equality $E(\mathcal{W}^\tau, \widehat{\mathcal{W}^\tau}) = E(\mathcal{W}^\tau, (\hat{\mathcal{W}})^\tau)$. This means that $(\hat{\mathcal{W}})^\tau$ is a closest rank ≤ 4 matrix to $\mathcal{W}^\tau$ for all sufficiently small κ, regardless of which column n is scaled by the transform. We will prove the proposition by showing that this can hold only if $\mathcal{W}$ already has rank ≤ 4. First, we need a lemma.

Lemma 1. *If a matrix $\hat{\mathcal{W}}$ is a closest matrix having rank $\leq r$ to a matrix $\mathcal{W}$, then $\mathcal{R}^\top \hat{\mathcal{W}} = \mathcal{R}\hat{\mathcal{W}}^\top = 0$, where $\mathcal{R} = \mathcal{W} - \hat{\mathcal{W}}$.*

Proof (sketch). Write $\hat{\mathcal{W}} = AB$, where A has r columns, take derivatives of $\|\mathcal{W} - AB^\top\|^2$ with respect to the entries of A or B, and set them to zero.

We return to the proof of the proposition. Since $\hat{\mathcal{W}}$ is a closest rank ≤ 4 matrix to $\mathcal{W}$, the lemma gives $\mathcal{R}\hat{\mathcal{W}}^\top = 0$. As argued above, we can choose $\kappa \neq 0$ small enough so that $(\hat{\mathcal{W}})^\tau$ is a closest rank ≤ 4 matrix to $\mathcal{W}^\tau$, regardless of what n we choose for τ_{nk}. For such κ, the lemma gives $\mathcal{R}^\tau (\hat{\mathcal{W}})^{\tau\top} = 0$, where $\mathcal{R}^\tau = \mathcal{W}^\tau - (\hat{\mathcal{W}})^\tau$, and it follows that $\mathcal{R}\hat{\mathcal{W}}^\top - \mathcal{R}^\tau (\hat{\mathcal{W}})^{\tau\top} = 0$. Since $\mathcal{W}$ and $\mathcal{W}^\tau$, and similarly $\mathcal{R}$ and $\mathcal{R}^\tau$, differ only in the scaling of their n-th columns, we may easily compute the matrix $\mathcal{R}\hat{\mathcal{W}}^\top - \mathcal{R}^\tau (\hat{\mathcal{W}})^{\tau\top}$: Its (p, q)-th entry equals $\kappa \mathcal{R}_n^p \mathcal{W}_n^q$. Since $\kappa \neq 0$ and our arguments hold regardless of the n we choose for τ_{nk}, we have $\mathcal{R}_n^p \mathcal{W}_n^q = 0$ for all values of n, p, and q.

We assumed that $\mathcal{W}$ has no columns consisting entirely of zeros. Thus, each column n of $\mathcal{W}$ contains a non-zero entry $\mathcal{W}_n^q$, so for each n we must have $\mathcal{R}_n^p = 0$ for all p, which means that column n of $\mathcal{R}$ is zero. Hence $\mathcal{R} = 0$ and $\mathcal{W}$ gives zero error, which is what we set out to prove. ■

Remark 2. Intuitively, Proposition 2 holds because the trivial minima are so destabilizing that one can always reduce the error by moving toward one.

SIESTA can be useful despite our result. (5) suggests that the error can be very flat and SIESTA's descent to a trivial minimum extremely slow. In trials on realistic data, the SIESTA error drops quickly from its start at $\lambda_n^i = 1$ but never approaches a trivial minimum; in fact, it descends so slowly after a few hundred iterations (with $\Delta\hat{E} \leq O(10^{-11})$) that one can easily conclude wrongly that it has converged. What seems to happen is that SIESTA approaches an almost minimum—a saddle point that would be a minimum if it weren't destabilized by the trivial minima—and then slows, usually still with $\lambda_n^i \approx 1$.

All this suggests that the destabilization from the trivial minima is weak, only becoming important at small error values. If we can compensate for it, e.g., by 'balancing', this might turn the saddles into minima giving correct estimates. In trials, SIESTA does give good estimates once it slows. Although its error has no usable minima, the saddle points may serve as useful 'effective minima.'

3.2 Other Iterative Extensions of ST

Mahamud et al. [7][8] proposed an iteration similar to SIESTA that differs by maintaining a normalization constraint on the columns of $\mathcal{W}$.[2] The first stage of the iteration is the same as in SIESTA, and the second stage is:

- **Iteration k, stage 2:** Given $\hat{\mathcal{W}}^{(k-1)}$, choose new projective depths $\lambda^{(k)}$ so that $\mathcal{W}(\lambda^{(k)})$ optimally approximates $\hat{\mathcal{W}}^{(k-1)}$ subject to the N_p columns constraints $|\mathcal{W}_n| = 1$, $n \in \{1 \dots N_p\}$.

With the constraints, the SIESTA error $\hat{E}$ reduces in effect to $\|\mathcal{W} - \hat{\mathcal{W}}\|^2$. It is easy to show that the iteration descends this error [8]. The constrained error possibly does have nontrivial minima, but we argue below that it does not have usable minima corresponding to good structure/camera estimates.

[1][5] proposed a SIESTA variant roughly dual to [8] but did not give an error for it. A similar iteration that descends an error is SIESTA with a new stage 2:

- **Iteration k, stage 2:** Given $\hat{\mathcal{W}}^{(k-1)}$, choose new projective depths $\lambda^{(k)}$ so that $\mathcal{W}(\lambda^{(k)})$ optimally approximates $\hat{\mathcal{W}}^{(k-1)}$ subject to the N_I image constraints $\|\mathcal{W}^{(3i-2):3i}\| = 1$, where each matrix $\mathcal{W}^{(3i-2):3i} \in \Re^{3 \times N_p}$ gives the three rows of $\mathcal{W}$ for image i.

This iteration also descends the error $\|\mathcal{W} - \hat{\mathcal{W}}\|^2$. We have not analyzed its convergence, but we expect that it has the same problems as the previous one.

Convergence analysis for the iteration of [7][8].[2]

Our results are weaker than for SIESTA, so we just summarize them.

As for SIESTA, we start by considering a transformation that scales the λ_n^i toward a trivial minimum (see Remark 2). We define the transform so that it first scales all the projective depths for the kth image by s, and then scales the column of projective depths for each 3D point to maintain the norm constraints on the columns of $\mathcal{W}$. As before, we apply the same transform to $\hat{\mathcal{W}}$ as for $\mathcal{W}$. Assuming that λ gives a stationary point of the error, our transform must also give a stationary point at λ, and this leads to constraints on $\mathcal{W}$ and $\hat{\mathcal{W}}$ analogous to (4). Exploiting these constraints, we try to modify the transform so that it strictly decreases the error at λ.

This is much harder than for SIESTA. The initial transform τ_{nk} for SIESTA gave an error that was *constant* at a stationary point, so we could make the error

[2] Mahamud et al. [7] also proposed a different iteration that minimizes alternately with respect to the camera and structure matrices. This approach loses the advantage of minimizing in the λ_n^i—it cannot exploit prior knowledge that the λ_n^i are near one.

decrease, establishing the stationary point as a saddle, by an arbitrarily small change in τ_{nk}. For the algorithm of [8], the error at a stationary point usually has a minimum under our initial transform. We need a *large* change in the transform to make the error decrease, so this may not always be possible. However, we argue that we can make the error decrease at "desirable" stationary points, where the estimates of the structure/cameras are roughly correct and $\lambda_n^i \approx 1$.

We now describe how to modify the inital transform described above. Let $\hat{\mathcal{W}} = \hat{M}\hat{X}^T$ be the rank 4 factoring that comes from the SVD of $\mathcal{W}$. We modify the initial transform of $\hat{\mathcal{W}}$ by transforming $\hat{X}$ linearly before scaling it, where we choose this linear transform to minimize the error's second derivative with respect to the transform at the stationary point. We have derived upper bounds on the resulting second derivatives. We will argue that these are negative at a "desirable" stationary point, so such stationary points are saddles.

Define $\mathbf{w}_n^i \equiv [\mathcal{W}]_n^{(3i-2):3i}$ and $\hat{\mathbf{w}}_n^i \equiv \left[\hat{\mathcal{W}}\right]_n^{(3i-2):3i}$ and the residual $\mathbf{r}_n^i \equiv \mathbf{w}_n^i - \hat{\mathbf{w}}_n^i$; all are vectors in $\Re^3$. Without loss of generality, take the columns of $\hat{M}$ orthogonal and define $\hat{m}^i \equiv \hat{M}^{(3i-2):3i} \in \Re^{3\times 4}$. Let $\hat{\mu}_a^i$ be the ath singular value of $\hat{m}^i$ and let $\hat{\mathbf{m}}_a^i \in \Re^3$ be the ath column of $\hat{m}^i$. Choose image k so

$$\left\langle |\hat{\mathbf{m}}^k|^2 \right\rangle \geq \left\langle |\mathbf{w}^k|^2 \right\rangle, \tag{6}$$

where we use $\langle\cdot\rangle$ to denote the average, taken over the omitted index. Such an image always exists, since our normalizations give

$$1 = \sum_{i=1}^{N_I}\sum_{a=1}^{4} |\hat{\mathbf{m}}_a^i|^2/4 = \sum_{i=1}^{N_I} \left\langle |\hat{\mathbf{m}}^i|^2 \right\rangle = \sum_{i=1}^{N_I} \left\langle |\mathbf{w}^i|^2 \right\rangle = \sum_{i=1}^{N_I}\sum_{n=1}^{N_p} |\mathbf{w}_n^i|^2/N_P.$$

Our upper bound on the second derivative for the modified transform is

$$2\left(\sum_{n=1}^{N_p} |\mathbf{r}_n^k|^2\right)\left(2 \max_{n=1\ldots N_p} \left|\, |\mathbf{w}_n^k|^2 - \left\langle |\mathbf{w}^k|^2 \right\rangle \,\right| - \frac{4}{3}\frac{|\hat{\mu}_3^k|^2}{\left\langle |\hat{\mu}^k|^2 \right\rangle} \left\langle |\mathbf{w}^k|^2 \right\rangle \right) \tag{7}$$

for the chosen image k. In practice, [4][9] recommend normalizing the homogenous image points to a unit box before applying **ST**. Then, assuming a "desirable" stationary point with all λ_n^i near 1, the $|\mathbf{w}_n^k|^2$ will be approximately constant in k and n. If the singular values $\hat{\mu}_a^k$ all have roughly the same size, then $|\hat{\mu}_3^k|^2 / \left\langle |\hat{\mu}^k|^2 \right\rangle \approx 1$, and our bound is likely to be negative.

In our experiments on real sequences, the apparent convergence points of the Mahamud et al. iteration [7][8] always have $\left(\hat{\mu}_3^k\right)^2 / \left\langle |\hat{\mu}^k|^2 \right\rangle \approx 1$, and they almost always give a negative value of the bound (7), which rules out these "convergence points" as local minima. Note that the bound is conservative; in practice, we expect cancellations to reduce the second derivative below (7).

Why is the ratio $\left(\hat{\mu}_3^k\right)^2 / \left\langle |\hat{\mu}^k|^2 \right\rangle$ typically near 1? One contributing factor is that, after the standard scaling to a unit box, the image submatrix $w^i \equiv$

$\left[\mathbf{w}_1^i, \mathbf{w}_2^i, \ldots, \mathbf{w}_{N_P}^i\right] \in \Re^{3\times N_p}$ typically has three singular values of the same order. Another cause is the following. Write the SVD of the scaled data matrix as $\mathcal{W} = UDV^T$. Writing the singular values $\hat{\mu}_a^k$ in terms of the image data gives

$$\left(\hat{\mu}_a^k\right)^2 = N_I^{-1} \sum_{n=1}^{N_p} \left|\mathbf{s}_a^{kT} w^k V_n\right|^2 \Big/ \left\langle \left|wV_n\right|^2 \right\rangle,$$

where $\mathbf{s}_a^k \in \Re^3$ represent the ath left singular vector of $\hat{m}^k$ and V_n denotes the nth column of V. The average in the denominator is over all images i. Thus, $\left(\hat{\mu}_a^k\right)^2$ is proportional to a sum of projections of the (homogeneous) image data normalized by their average values. If the camera positions are spaced roughly uniformly, as they are in many sequences, the kth image is often close to "average," so the singular values $\hat{\mu}_a^k$ all have similar sizes. One *can* get $\hat{\mu}_3^k \ll \hat{\mu}_1^k$ if, for example, most of the camera positions cluster together but one is very far from the others.

We have also derived a second bound whose size is easier to estimate. Choose image k such that $\left\|w^k - \hat{w}^k\right\|^2 \leq \hat{E}\left\|w^k\right\|^2$, which is always possible since one can show that $\sum_i \left\|w^i - \hat{w}^i\right\|^2 = \hat{E}\sum_i \left\|w^i\right\|^2$. Denote the ath singular value of w^k by d_a^k and the ath singular value of $\mathcal{W}$ by D_a. Our new bound is

$$2\left(\sum_{n=1}^{N_p} |\mathbf{r}_n^k|^2\right)\left(2 \max_{n=1\ldots N_p} \left|\ |\mathbf{w}_n^k|^2 - \left\langle |\mathbf{w}^k|^2 \right\rangle\ \right| - \left(d_3^k/\left\|w^k\right\| - \hat{E}^{1/2}\right)^2 \frac{N_p}{D_1^2}\left\langle |\mathbf{w}^k|^2 \right\rangle\ \right). \tag{8}$$

After the standard scaling of the image data to a unit box, we expect $d_3^k/\left\|w^k\right\| \approx 3^{-1/2} \approx 0.58$. Even if the scene is planar, the first three singular values of $\mathcal{W}$ are usually substantial, causing $N_p/D_1^2 > 1$. Experimentally, we find $d_3^k/\left\|w^k\right\| \approx 0.3$ and $N_p/D_1^2 \approx 1.3$. Substituting the experimental values, and assuming $\mathcal{W}$ is close to a rank ≤ 4 matrix so $\hat{E} \ll 1/3$, we can approximate the bound as

$$2\left(\sum_{n=1}^{N_p} |\mathbf{r}_n^k|^2\right)\left(2 \max_{n=1\ldots N_p} \left|\ |\mathbf{w}_n^k|^2 - \left\langle |\mathbf{w}^k|^2 \right\rangle\ \right| - 0.13\left\langle |\mathbf{w}^k|^2 \right\rangle\ \right). \tag{9}$$

Table 1 shows results for the Mahamud et al. algorithm on real image sequences, see Figure 1. We obtained these by running the algorithm for 1000 iterations, after which the error was changing so slowly that the algorithm seemed to have converged. In all but one case, we found negative values for the bounds (8) and (9), proving the algorithm had not converged to a minimum. In the exceptional case, the trivial minima had produced small λ_n^i for a few points. Repeating the experiment without these points gave a negative bound. We have verified our upper bounds experimentally on several thousand synthetic sequences. We also used a standard nonlinear minimization routine (LSQNONLIN from MATLAB) to minimize the error for the Mahamud et al. algorithm. The routine converged to a trivial minimum in all cases. These results indicate that the convergence of the Mahamud et al. algorithm is problematic at best. [2]

Table 1. The bounds (7), (8) for five real sequences (Fig. 1). 'Ox0–10' is for 11 images; other 'Ox' rows are for image pairs. 'Ox0&8*' is for images 0 and 8, with 3 points subtracted. The 'μ ratio' column gives the least $|\hat{\mu}_3^k|^2 / \langle |\hat{\mu}^k|^2 \rangle$ over k satisfying (6).

	Est. Range	First	Least μ	$\hat{E}^{1/2}$	Range for $\max_n$		$\max_k$	Second
	λ	Bound	ratio	$(\times 10^{-3})$	$\left\lvert \frac{\lvert\mathbf{w}_n\rvert^2}{\langle\lvert\mathbf{w}\rvert^2\rangle} - 1 \right\rvert$	$\frac{N_p}{D_1^2}$	$\frac{d_3^2}{\lVert w\rVert^2}$	Bound
Ox0-10	0.79–1.15	−0.06	0.40	1.6	0.01–0.37	1.3	0.08	−0.09
Ox0&10	0.83–1.14	−0.24	0.78	1.8	0.29–0.31	1.3	0.10	0.44
Ox0&8	0.14–1.40	0.52	0.73	8.2	0.98–1.08	1.3	0.09	1.9
Ox0&8*	0.85–1.13	−0.28	0.78	1.6	0.24–0.25	1.3	0.09	0.37
Ox0&1	0.97–1.03	−0.46	0.75	0.5	0.04–0.04	1.3	0.12	−0.07
Rock	0.90–1.09	−0.07	0.46	6.0	0.03–0.20	1.4	0.08	0.01
Puma	0.99–1.01	−0.05	0.50	1.7	0.01–0.06	1.5	0.14	−0.21
MSTea	0.97–1.03	−0.43	0.75	0.7	0.05–0.05	1.5	0.16	−0.14
MSPlane	0.91–1.10	−0.13	0.44	0.6	0.05–0.13	1.5	0.18	−0.17

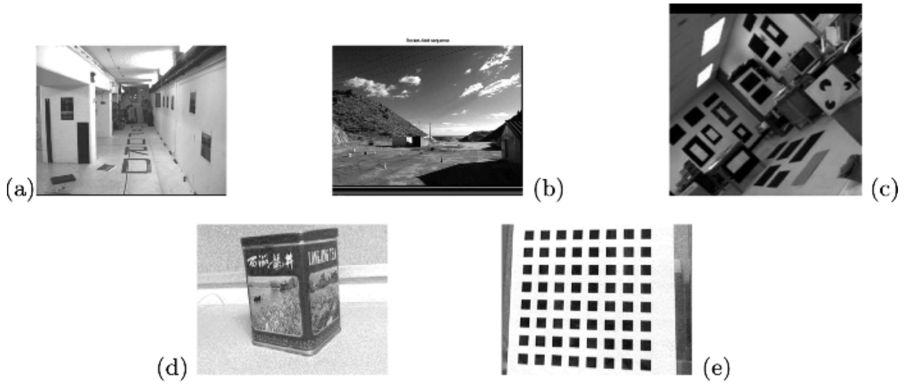

Fig. 1. Images from the five real sequences. (a) Oxford corridor; (b) Rocket–Field [3]; (c) PUMA [6]; (d) Microsoft tea [12]; (e) Microsoft Plane calibration [12].

3.3 Balancing

[4] modifies SIESTA by adding a third "balancing" stage [11] following stage 2 that rescales the λ_n^i to make them close to 1. This lessens the algorithm's bias and helps to steer it away from trivial minima. The balancing can be done in two passes, by first scaling $\lambda_n^i \longrightarrow \alpha^i \lambda_n^i$ for each image i so that $\sum_{n=1}^{N_p} |\lambda_n^i|^2 = N_I$, and then scaling $\lambda_n^i \longrightarrow \beta_n \lambda_n^i$ for each point so $\sum_{i=1}^{N_I} |\lambda_n^i|^2 = N_p$. Optionally, this can be repeated several times or iterated to convergence. Unfortunately, it seems likely that the rescaling conflicts with the error minimization in stages 1 and 2, and that the balanced iteration need not converge. To exaggerate this potential conflict and make it more observable, we implemented SIESTA with a balancing stage that iterates to near convergence, with up to 10 rounds of first balancing the rows and then the columns of λ_n^i. In one of our experiments, this algorithm apparently converged to a limit cycle which repeatedly passed through

three different values for λ with three different values of the error $\hat{E}(\lambda)$. For this strong version of balancing, it seems that an iteration of SIESTA–plus–balancing does not guarantee improvement in the projective–depth estimates.

Nevertheless, occasional balancing may serve as a useful "mid–course correction" that compensates for SIESTA's drift toward trivial minima.

4 CIESTA

An alternative to balancing is regularization. We define a new iteration CIESTA that descends the error $\hat{E}_{\mathrm{reg}}(\lambda)$, where

$$\hat{E}_{\mathrm{reg}}(\lambda) \equiv \min_{\mathrm{rank}(Y)\leq 4} E_{\mathrm{reg}}, \qquad E_{\mathrm{reg}}(\lambda, Y) \equiv E(\mathcal{W}(\lambda), Y) + \mu \sum_{i=1}^{N_I} \sum_{n=1}^{N_p} |\mathbf{x}_n^i|^2 (1-\lambda_n^i)^2, \tag{10}$$

and $\mu > 0$ is the regularization constant. The algorithm is the same as SIESTA except for a new third stage in the iteration.

Let $\lambda^{(k)} \in \Re^{N_i N_p}$ and $\mathcal{W}^{(k)} \equiv \mathcal{W}(\lambda^{(k)}$ now denote the output of the kth CIESTA iteration, and let $\lambda^{(0)}$ and $\mathcal{W}^{(0)}$ give the initialization. As before, let $\hat{\mathcal{W}}^{(k)}$ be the best approximation of rank ≤ 4 to $\mathcal{W}^{(k)}$. Define the constants

$$C_0 = \mu \sum_{i=1}^{N_I} \sum_{n=1}^{N_p} |\mathbf{x}_n^i|^2, \qquad C_1^{(k)} \equiv \mu \sum_{i=1}^{N_I} \sum_{n=1}^{N_p} \mathbf{x}_n^i \cdot \hat{\mathbf{w}}_n^{i(k)}, \tag{11}$$

$$C_2^{(k)} \equiv \mu \sum_{i=1}^{N_I} \sum_{n=1}^{N_p} \frac{\left(\mathbf{x}_n^i \cdot \hat{\mathbf{w}}_n^{i(k)}\right)^2}{|\mathbf{x}_n^i|^2}, \qquad C_3^{(k)} = \mu \sum_{i=1}^{N_I} \sum_{n=1}^{N_p} |\hat{\mathbf{w}}_n^{i(k)}|^2,$$

and $z^{(k)} \equiv C_3^{(k)} C_0 / C_2^{(k)}$. Define the function

$$b_+^{(k)}(a) \equiv a^{1/2} / \left(a^2 C_0 + 2a C_1^{(k)} + C_2^{(k)}\right)^{1/2}, \tag{12}$$

which is obtained as an intermediate result while minimizing $E_{\mathrm{reg}}(\lambda, \hat{\mathcal{W}}^{(k)})$ in λ.

Remark 3. One can show that CIESTA gives the following constraints:

1. $C_0 > 0, \qquad C_3^{(k)} \geq C_2^{(k)}, \qquad C_2^{(k)} > 0.$
2. $\mathcal{Q}^{(k)} \equiv a^2 C_0 + 2a C_1^{(k)} + C_2^{(k)} > 0, \qquad z^{(k)} > 0$

From the second line we see that b_+ is finite. CIESTA's new third stage is:

- **CIESTA** (iteration k, stage 3): With $\kappa \equiv k-1$, compute the roots of $P^{(\kappa)}(a)$

$$\equiv C_0 a^6 - \left(C_0^2 - 2C_1^{(\kappa)}\right) a^5 - \left(2C_0 C_3^{(\kappa)} - C_2^{(\kappa)}\right) a^4 - \left(4C_1^{(\kappa)} C_3^{(\kappa)} - 2C_2^{(\kappa)} C_0\right) a^3$$

$$+ \left(C_0 C_3^{(\kappa)2} - 2C_2^{(\kappa)} C_3^{(\kappa)}\right) a^2 + \left(2C_1^{(\kappa)} C_3^{(\kappa)2} - C_2^{(\kappa)2}\right) a + C_2^{(\kappa)} C_3^{(\kappa)2}. \tag{13}$$

Choose a root $a > 0$ such that: For $z^{(\kappa)} \neq 1$, the quantity $\bar{a} \equiv a(C_0/C_2^{(\kappa)})^{1/2}$ and $z^{(\kappa)}$ lie on the same side of 1; For $z^{(\kappa)} = 1$, when there is a choice, take either of the choices with $\bar{a} \neq 1$. Redefine $\lambda^{(k)} \longrightarrow \lambda^{(k)} = \left(a + \lambda^{(k)}\right) b_+^{(\kappa)}(a)$.

The four propositions below address the convergence of CIESTA (proofs omitted). Let $\hat{E}_\infty$ be the greatest lower bound of the errors $\hat{E}_{\text{reg}}(\lambda^{(k)})$, and let $\mathcal{A}$ be the set of accumulation points of the sequence $\lambda^{(k)}$.

Assumption 1 **(μ condition).** *CIESTA starts with all $\lambda_n^i = 1$, and*

$$\mu \|\mathcal{W}^{(0)}\|^2 > \|\mathcal{W}^{(0)} - \hat{\mathcal{W}}^{(0)}\|^2 / \|\mathcal{W}^{(0)}\|^2, \tag{14}$$

which is equivalent to $C_0^2 > \left(C_0 + C_3^{(0)} - 2C_1^{(0)}\right)$.

Remark 4. Our results below don't require that CIESTA start at $\lambda_n^i = 1$; we assume this just to simplify the theorems and proofs. The Assumption specifies how much regularization is needed to guarantee CIESTA's performance. Taking μ large enough rules out the trivial minima.

Proposition 3. *Suppose Assumption 1 holds. The errors $\hat{E}_{\text{reg}}\left(\lambda^{(k)}\right)$ are non-increasing with k and converge monotonically in the limit $k \to \infty$.*

Proposition 4. *Suppose Assumption 1 holds. Then: 1) Every $\lambda_{\mathcal{A}} \in \mathcal{A}$ has $\hat{E}_{\text{reg}}(\lambda_{\mathcal{A}}) = \hat{E}_\infty$; 2) For any $\epsilon > 0$, there exists a K such that $k > K$ implies $\left|\lambda^{(k)} - \lambda_{\mathcal{A}}\right| \leq \epsilon$ for some $\lambda_{\mathcal{A}} \in \mathcal{A}$.*

Proposition 5. *Suppose Assumption 1 holds. Let $\lambda_{\mathcal{A}} \in \mathcal{A}$. Let the fourth singular value of $\mathcal{W}(\lambda_{\mathcal{A}})$ be strictly greater than the fifth, and $z^{\mathcal{A}} \neq 1$, where $z^{\mathcal{A}}$ is the constant from (11) evaluated at $\lambda_{\mathcal{A}}$. Then $\lambda_{\mathcal{A}}$ is a fixed point of CIESTA and a stationary point of the error $\hat{E}_{\text{reg}}$ (not necessarily a minimum).*

Proposition 6. *Suppose the assumptions of Proposition 5 hold for some $\lambda_{\mathcal{A}} \in \mathcal{A}$. Proposition 5 states that $\hat{E}_{\text{reg}}$ has a stationary point at $\lambda_{\mathcal{A}}$. If in fact $\hat{E}_{\text{reg}}$ has a strict local minimum at $\lambda_{\mathcal{A}}$, then CIESTA converges uniquely to $\lambda_{\mathcal{A}}$.*

Propositions 3 and 4 show that CIESTA "converges" in a certain sense (discussed below). Proposition 5 states that its end results are sensible, that is, they are stationary points of the error. Proposition 6 shows that under weak assumptions CIESTA converges in a strict sense to a unique result.

The proof of Prop. 4 is easy and the proof of Prop. 3 is a calculation. The proof of Proposition 5 is more technical: We need to show that E_{reg} has a unique global minimum and that the output of stage 3 depends continuously on its input.

Discussion. Like balancing, CIESTA favors $\lambda_n^i \approx 1$, but it guarantees improved estimates with lower error. The error $\hat{E}_{\text{reg}}$ shows explicitly how CIESTA weights its preference for $\lambda_n^i \approx 1$ versus the data error $\hat{E}$. The extra computation of stage 3 is small: it just requires finding the eigenvalues of a 6×6 matrix.

We have not shown that CIESTA converges to a single λ (except under the assumptions of Prop. 6), and it is not clear whether this always happens. But our results have the same practical implications as a convergence proof.

A convergence proof would amount to the following guarantee: by iterating enough times, one can bring the algorithm as close as desired to a "best achievable result," i.e., to a λ with the lowest error reachable from its starting point. This does not forbid other equally good estimates with the same error as the "best result," though the algorithm happens not to converge to them.

Proposition 4 provides essentially the same guarantee: by iterating enough times, we can bring CIESTA arbitrarily close to a "best achievable result." The difference is that the nearest "best result" may change from iteration to iteration. This doesn't matter since all are good and we may choose any one as CIESTA's final output. Under the conditions of Prop. 6, we do have strict convergence.

One can minimize $\hat{E}_{\text{reg}}$ using a traditional quadratically convergent technique such as Gauss–Newton instead of CIESTA.

4.1 CIESTA Experiments

Table 2 shows results obtained using a standard quadratically convergent nonlinear minimization routine (from MATLAB) to minimize $\hat{E}_{\text{reg}}$ for the real image sequences of Figure 1. The algorithm always converged to a nontrivial minimum with $\lambda_n^i \approx 1$, though we used a value for μ that permitted some of the λ_n^i to go to zero. The value of μ was twice that needed to avoid $\lambda = 0$.

The iterative extensions of ST, including SIESTA, CIESTA, and the balanced iterations, all give similar results in practice, and iterating them to convergence (or apparent convergence) gives better results than a single iteration does. To illustrate this, we compared their results against ground truth on one synthetic and two real sequences, see Table 3. (We generated the synthetic OxCorr sequence in Table 3 using the Oxford Corridor ground truth structure and random translations and rotations. For OxDino, we extracted 50 points tracked over 6 images from the Oxford Dinosaur sequence and computed the ground truth by bundle adjustment.)

In all three cases: SIESTA gave the best agreement with the ground truth; the result at "convergence" improved on that obtained after a single iteration; the results of the "balanced" iteration did not depend on the number of rounds

Table 2. Results of using MATLAB's LSQNONLIN to minimize the CIESTA error $\hat{E}_{\text{reg}}$. Results show the values at convergence. The f values do not equal 1 exactly.

Sequence	μ/Bound	λ range	f	C_0	C_2	C_3
Rock	2	0.90–1.08	1	1.9786	1.9785	1.9785
PUMA	2	0.98–1.01	1	1.9954	1.9954	1.9954
Ox0&1	2	0.98–1.02	1	1.9993	1.9993	1.9993
Ox0&8	2	0.71–1.09	1	1.9358	1.9355	1.9357
Ox0&10	2	0.82–1.13	1	1.9770	1.9770	1.9770
Ox0-10	2	0.79– 1.13	1	1.9844	1.9844	1.9844
MSTea	2	0.98–1.02	1	1.9993	1.9993	1.9993
MSPlane	2	0.90–1.09	1	1.9958	1.9958	1.9958

Table 3. Fractional errors $\sum_n |P_n^{\mathrm{calc}} - P_n^{\mathrm{GT}}|^2 / \sum_n |P_n^{\mathrm{GT}}|^2$ $(\times 10^4)$ for the structure after "convergence." We compute $P_n^{\mathrm{calc}} \in \Re^3$ from the calculated homogeneous structure by applying a projective transform to minimize the error. SIESTA1 gives results after one iteration; other results are after 1000 iterations or convergence (CIESTA). Bal1 and 10 results are obtained using a single round or 10 rounds of column/row balancing in each iteration.

Sequence	SIESTA1	SIESTA	Bal1	Bal10	CIESTA
OxCorr	4.8	2.8	3.0	3.0	2.9
PUMA	0.97	0.47	0.47	0.47	0.48
OxDino	1.48	0.39	0.49	0.49	0.54

of balancing; and CIESTA (using μ computed as in Table 2) performed as well as the balanced iterations.

5 Conclusion

We showed that SIESTA, the simplest iterative extension of ST, descends an error function: Each iteration "improves" the estimates. However, we proved that the SIESTA doesn't converges to useful results. We showed that another proposed extension of **ST** [7] shares this problem.[2] [4] advocate "balancing" to improve convergence. Our experiments show that balancing need not yield a convergent algorithm.

We proposed CIESTA, a new iterative extension of ST, which avoids these problems. CIESTA replaces balancing by regularization. The algorithm is identical to SIESTA except for one additional and still simple stage of computation. We proved that CIESTA descends an error function and approaches nontrivial fixed points, and that it converges under weak assumptions.

CIESTA, like other iterative extensions of **ST**, has the advantage of minimizing in the λ_n^i, whose values are often known to be near one a priori. Thus, it often shows fast initial convergence toward estimates that are approximately correct. Like other iterative extensions, CIESTA has the disadvantage that it converges linearly. A quadratically convergent method such as Gauss–Newton will be faster near a fixed point or in narrow valleys of the error function. Using such a method instead of CIESTA can combine quadratic convergence with the advantage of minimizing in the λ_n^i. A hybrid strategy that uses CIESTA initially and then switches to a second–order method, or full bundle adjustment, can combine the speed advantages of both [2].

Unlike bundle adjustment, CIESTA needs regularization. This allows the user to incorporate a realistic preference for projective depth values near 1 but can bias the final estimate. However, our experiments indicate that just a small amount of regularization suffices to stabilize the error minima. CIESTA's regularization and SIESTA's trivial convergence generally do not have a big effect on the estimates obtained once the algorithms slow their progress.

References

1. R. Berthilsson, A. Heyden, G. Sparr, "Recursive Structure and Motion from Image Sequences Using Shape and Depth Spaces," *CVPR* 444–449, 1997.
2. A. Buchanan and A. Fitzgibbon, "Damped Newton Algorithms for Matrix Factorization with Missing Data," *CVPR* 2005.
3. R. Dutta, R. Manmatha, L.R. Williams, and E.M. Riseman, "A data set for quantitative motion analysis," *CVPR*, 159-164, 1989.
4. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge, 2000.
5. A. Heyden, R. Berthilsson, G. Sparr, "An iterative factorization method for projective structure and motion from image sequences," **IVC** 17 981–991,1999.
6. R. Kumar and A.R. Hanson, "Sensitivity of the Pose Refinement Problem to Accurate Estimation of Camera Parameters," *ICCV*, 365-369, 1990.
7. S. Mahamud, M. Hebert, Y. Omori, J. Ponce, "Provably-Convergent Iterative Methods for Projective Structure from Motion," *CVPR* I:1018-1025, 2001.
8. S. Mahamud, M. Hebert, "Iterative Projective Reconstruction from Multiple Views," *CVPR* II 430-437, 2000.
9. P. Sturm and B. Triggs, "A factorization based algorithm for multi–image projective structure and motion," *ECCV* II 709–720, 1996.
10. C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," **IJCV** 9, 137-154, 1992.
11. B. Triggs, "Factorization methods for projective structure and motion," *CVPR* 845–851, 1996.
12. Zhengyou Zhang, "A Flexible New Technique for Camera Calibration," **PAMI** 22:11, 1330-1334, 2000 and Microsoft Technical Report MSR-TR-98-71, 1998.