Building Roadmaps of Local Minima of Visual Models

Cristian Sminchisescu and Bill Triggs

INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot, France.
{Cristian.Sminchisescu,Bill.Triggs}@inrialpes.fr
http://www.inrialpes.fr/movi/people/{Sminchisescu,Triggs}

Abstract. Getting trapped in suboptimal local minima is a perennial problem in model based vision, especially in applications like monocular human body tracking where complex nonlinear parametric models are repeatedly fitted to ambiguous image data. We show that the trapping problem can be attacked by building 'roadmaps' of nearby minima linked by *transition pathways* — paths leading over low 'cols' or 'passes' in the cost surface, found by locating the *transition state* (codimension-1 saddle point) at the top of the pass and then sliding downhill to the next minimum. We know of no previous vision or optimization work on numerical methods for locating transition states, but such methods do exist in computational chemistry, where transitions are critical for predicting reaction parameters. We present two families of methods, originally derived in chemistry, but here generalized, clarified and adapted to the needs of model based vision: *eigenvector tracking* is a modified form of damped Newton minimization, while *hypersurface sweeping* sweeps a moving hypersurface through the space, tracking minima within it. Experiments on the challenging problem of estimating 3D human pose from monocular images show that our algorithms find nearby transition states and minima very efficiently, but also underline the disturbingly large number of minima that exist in this and similar model based vision problems.

Keywords: Model based vision, global optimization, saddle points, 3D human tracking.

1 Introduction

Many visual modelling problems can be reduced to cost minimization in a high dimensional parameter space. Local minimization is usually feasible, but practical cost functions often have large numbers of local minima and it can be very difficult to ensure that the desired one is found. Exhaustive search rapidly becomes impracticable in more than 2–3 dimensions, so most global optimization methods focus on heuristics for finding 'good places to look next'. This includes both deterministic techniques like branch-and-bound and pattern search, and stochastic importance samplers like simulated annealing, genetic algorithms and tabu search.

Unfortunately, global optimization remains expensive with any of these methods. In this paper we develop an alternative strategy based on building 1-D 'roadmaps' of the salient minima, linked by paths passing over **saddle points**: stationary (zero gradient) points that have one or more negative curvature directions, so that they represent 'cols' rather than 'hollows' in the cost surface. We will restrict attention to **transition states**

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 566–582, 2002.

(saddles with just one negative curvature direction), as these give the minimum-peak-cost pathways between local minima. Our focus is on methods for finding the salient transitions surrounding an initial minimum. Given these, adjacent minima can be found simply by sliding downhill using local minimization.

Despite the omnipresence of local minima, we know of no previous vision or optimization work on systematic numerical algorithms for locating transition states. As far as we can judge, this was generally considered to be intractable. However such methods *do* exist in the computational chemistry / solid state physics community, where transitions are central to the theory of chemical reactions[1]. We will describe two families of transition-finding algorithms that have roots in computational chemistry: **eigenvector tracking** is a modified form of damped Newton minimization, while **hypersurface sweeping** sweeps a moving hypersurface through the space, tracking minima within it. These methods are potentially useful in almost any visual modelling problem where local minima cause difficulties. Examples include model based tracking, reconstruction under correspondence ambiguities, and various classes of camera pose and calibration problems. We present experimental results on monocular model based human pose estimation.

1.1 Literature Review

We start with a brief overview of the computational chemistry / solid state physics literature on locating transition states. This literature should be accessible to vision workers with high-school chemistry and a working knowledge of optimization. However the underlying ideas can be difficult to disentangle from chemistry-specific heuristics, and some papers are rather naive about numerical optimization issues. We therefore give a self-contained treatment of two of the most promising approaches below, in numerical analysts language.

A transition state is a local minimum along its $n-1$ positive curvature directions in parameter space, but a local maximum along its remaining negative curvature one. So transition state search methods often reduce to a series of $(n-1)$-D minimizations, while moving or maximizing along the remaining direction. The main differences lie in the methods of choosing the directions to use.

Eigenvector tracking methods [10, 17, 6, 39, 40, 23, 16, 31, 24, 15, 11, 5] are modified Newton minimizers designed to increase the cost along one of the curvature eigendirections, rather than reducing it along all eigendirections. If the lowest curvature direction is chosen they attempt to find the lowest gradient path to a transition state by walking along the 'floor' of the local cost 'valley'. However this behaviour can not be guaranteed [19, 36, 18] and valleys need not even lead to saddle points: they might be 'blind', with the transition states located off to one side. An early method [17] used explicit Newton

[1] Atomic assemblies can be modelled in terms of the potential energy induced by interactions among their atoms, *i.e.* by an energy function defined over the high-dimensional configuration space of the atoms' relative positions. A typical assembly spends most of its time near an energy minimum (a stable or quasi-stable state), but thermal perturbations may sometimes cause it to cross a transition state to an adjacent minimum (a chemical reaction). The energy of the lowest transition joining two minima determines the likelihood of such a perturbation, and hence the reaction pathway and rate.

minimization in the $(n-1)$-D space obtained by eliminating the coordinate with the largest overlap with the desired up-hill direction. Later quasi-Newton methods use Lagrange multipliers [6, 39, 40] or shifted Hessian eigenvalues [31, 24, 15, 11, 5] to ensure that the cost function is increased along the chosen 'uphill eigenvector' direction while being minimized in all orthogonal ones. Maintaining a consistent direction to follow can be delicate and several competing methods exist, including using a fixed eigenvector index [17, 6, 39, 40, 23, 16] and attempting to track corresponding eigenvectors from step to step [31, 24, 15, 11, 5]. Eigenvector tracking can be motivated as a 'virtual cost minimization' obtained by inverting the sign of the negative Hessian eigenvalue and the corresponding gradient component [22, 15]. This gives an intuitive algebraic analogy with minimization, but none of its convergence guarantees as the virtual cost function changes at each step.

Constraint based methods [10, 1, 2, 3, 22, 16] aim to use some form of constrained optimization to guarantee more systematic global progress towards a transition state. Crippen & Sheraga's early method [10] builds an uphill path by minimizing in the orthogonal hyperplane of a ray emanating from the initial minimum and passing through the current configuration. Mousseau [22] uses a similar but less rigorous technique based on changing the gradient sign in one direction followed by conjugate root-finding in the other directions. Barkema [3] uses a biased repulsive spherical potential and optimizes subject to this soft constraint. New minima are found but the method does not attempt to pass exactly through a saddle point. Abashkin & Russo [1, 2] minimize on successively larger radius hyperspheres centred at a minimum, and also include a method for refining approximately located saddle points. The use of hyperspheres forces the search to move initially along the valley floor of the cost surface [18], so usually at most two distinct saddles can be found. Below we show how to steer the initial search along any desired direction using ellipsoidal surfaces.

There are also stochastic search methods designed to find transition states. See our companion paper [35] for references.

2 Algorithms for Finding Transition States

Many efficient methods exist for finding local minima of smooth high dimensional cost surfaces. Minimization allows strong theoretical guarantees as the reduction in the function value provides a clear criterion for monitoring progress. For example, for a bounded-below function in a bounded search region, any method that ensures 'sufficient decrease' in the function at each step is 'globally convergent' to some local minimum [13]. Finding saddle points is much harder as there is no universal progress criterion and no obvious analogue of a 'downhill' direction. Newton-style iterations provide rapid *local* convergence near the saddle, but it is not so obvious how to find sufficiently nearby starting points. We will consider several methods that extend the convergence zone. We are mainly interested in saddles as starting points for finding adjacent minima, so we will focus on methods that can be started from a minimum and tuned to find nearby transition states. (Efficient 'rubber band relaxation' methods also exist for finding the transition state(s) linking two given minima [29]).

2.1 Newton Methods

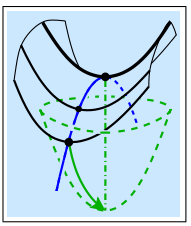
Let $f(\mathbf{x})$ be the cost function being optimized over its n-D parameter vector $\mathbf{x}$, $\mathbf{g} \equiv \frac{\partial f}{\partial \mathbf{x}}$ be the function's **gradient** and $\mathbf{H} \equiv \frac{\partial^2 f}{\partial \mathbf{x}^2}$ be its **Hessian**. We seek **transition states**, stationary points $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ at which the Hessian has one negative and $n-1$ positive eigenvalues. If there is a stationary point at $\mathbf{x}+\boldsymbol{\delta}\mathbf{x}$, a first order Taylor approximation at $\mathbf{x}$ gives $\mathbf{0} = \mathbf{g}(\mathbf{x}+\boldsymbol{\delta}\mathbf{x}) \approx \mathbf{g}(\mathbf{x}) + \mathbf{H}\,\boldsymbol{\delta}\mathbf{x}$. Solving this linear system for $\boldsymbol{\delta}\mathbf{x}$ and iterating to refine the approximation gives the **Newton iteration**: $\mathbf{x} \leftarrow \mathbf{x}+\boldsymbol{\delta}\mathbf{x}$ with update $\boldsymbol{\delta}\mathbf{x} = -\mathbf{H}^{-1}\mathbf{g}$. When started sufficiently close to any regular[2] stationary point, Newton's method converges to it, but how close you need to be is a delicate point in practice.

For Newton-based minimization, convergence can be globalized by adding suitable damping to shorten the step and stabilize the iteration. The standard methods use the **damped Newton** update $\boldsymbol{\delta}\mathbf{x} = -(\mathbf{H}+\lambda\mathbf{D})^{-1}\mathbf{g}$, where $\mathbf{D}$ is a positive diagonal matrix (often the identity). The damping factor $\lambda > 0$ is manipulated by the algorithm to ensure stable and reliable progress downhill towards the minimum. Damping can be viewed as Newton's method applied to a modified local model for f, whose gradient at $\mathbf{x}$ is unchanged but whose curvature is steepened to $\mathbf{H}+\lambda\mathbf{D}$.

Similarly, to reduce the step and stabilize the iteration near a saddle point, negative curvatures must be made more negative, and positive ones more positive. In a Hessian eigenbasis $\mathbf{H} = \mathbf{V}\,\mathbf{E}\,\mathbf{V}^\top$, where $\mathbf{E} = \mathrm{diag}(\lambda_1, ..., \lambda_n)$ are the eigenvalues of $\mathbf{H}$ and the columns of $\mathbf{V}$ are its eigenvectors, the undamped Newton update becomes $\boldsymbol{\delta}\mathbf{x} = -\mathbf{V}\,(\bar{g}_1/\lambda_1, \ldots, \bar{g}_n/\lambda_n)^\top$ where $\bar{g}_i \equiv (\mathbf{V}^\top\mathbf{g})_i$ are the eigen-components of the gradient. Damping can be introduced by replacing this with[3]:

$$\boldsymbol{\delta}\mathbf{x} = -\mathbf{V}\,\mathbf{u}(\lambda), \quad \mathbf{u}(\lambda) \equiv \left(\frac{\bar{g}_1}{\lambda_1+\sigma_1\lambda}, ..., \frac{\bar{g}_n}{\lambda_n+\sigma_n\lambda}\right)^\top = \left(\frac{\sigma_1\bar{g}_1}{\sigma_i\lambda_1+\lambda}, ..., \frac{\sigma_n\bar{g}_n}{\sigma_n\lambda_n+\lambda}\right)^\top \quad (1)$$

where $\sigma_i = \pm 1$ is a desired sign pattern for the λ_i. Damping $\lambda > \max_i(-\sigma_i\lambda_i, 0)$ ensures that the denominators are positive, so that the iteration moves uphill to a maximum along the eigendirections with $\sigma_i = -1$ and downhill to a minimum along the others. At each step this can be viewed as the minimization of a virtual local function with curvatures $\sigma_i\lambda_i$ and sign-flipped gradients $\sigma_i\bar{g}_i$. But the model changes at each step so none of the usual convergence guarantees of well-damped minimization apply: f itself is *not* minimized.



As in minimization, λ must be varied to ensure smooth progress. There are two main strategies for this: **Levenberg-Marquardt** methods manipulate λ directly, while the more sophisticated **trust region** ones maintain a local region of supposed-'trustworthy' points and choose λ to ensure that the step stays within it, *e.g.* $\|\boldsymbol{\delta}\mathbf{x}(\lambda)\| = \|\mathbf{u}(\lambda)\| \lesssim r$ where r is a desired 'trust radius'. (Such a λ can be found efficiently with a simple 1-D Newton iteration started at large λ [13]). In both cases, convergence criteria and model accuracy metrics such as the relative f-prediction error:

[2] 'Regular' means that $\mathbf{H}$ is nonsingular and 2^{nd} order Taylor expansion converges.

[3] There is nothing absolute about eigenvalues! Affine changes of coordinates leave the original Newton method unchanged but produce essentially inequivalent eigen-decompositions and dampings.

$$\beta = \left| \frac{f(\mathbf{x}+\boldsymbol{\delta}\mathbf{x}) - f(\mathbf{x})}{\mathbf{g}^\top \boldsymbol{\delta}\mathbf{x} + \boldsymbol{\delta}\mathbf{x}^\top \mathbf{H} \boldsymbol{\delta}\mathbf{x}/2} - 1 \right| \tag{2}$$

are monitored, and the damping is increased (larger λ or shorter r) if the accuracy is low, decreased if it is high, and left unchanged if it is intermediate (*e.g.*, by scaling λ or r up or down by fixed constants).

As in minimization, if the exact Hessian is unavailable, quasi-Newton approximations based on previously computed gradients can be used. Positive definiteness is not required so update rules such as Powell's are preferred [5]: $\mathbf{H} \leftarrow \mathbf{H} - \frac{\boldsymbol{\delta}\mathbf{x}^\top \boldsymbol{\xi}}{\|\boldsymbol{\delta}\mathbf{x}\|^4} \boldsymbol{\delta}\mathbf{x}\, \boldsymbol{\delta}\mathbf{x}^\top + \frac{\boldsymbol{\xi}\, \boldsymbol{\delta}\mathbf{x}^\top + \boldsymbol{\delta}\mathbf{x}\, \boldsymbol{\xi}^\top}{\|\boldsymbol{\delta}\mathbf{x}\|^2}$, where $\boldsymbol{\xi} = \mathbf{g}(\mathbf{x} + \boldsymbol{\delta}\mathbf{x}) - \mathbf{g}(\mathbf{x}) - \mathbf{H}(\mathbf{x})\boldsymbol{\delta}\mathbf{x}$.

2.2 Eigenvector Tracking

Now consider the choice of the signs σ_i. Roughly speaking, the damped iteration moves uphill to a maximum along directions with $\sigma_i = -1$ and downhill to a minimum along directions with $\sigma_i = +1$, *i.e.* it tries to find a stationary point whose principal curvatures λ_i have signs σ_i. To find minima we need $\sigma_i = +1$, and for transition states exactly one σ_i should be made negative[4]. The question is, which one.

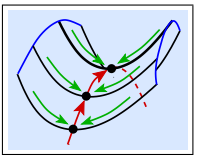
This question is thornier than it may seem. To ensure continued progress we need to track and modify "the same" eigenvector(s) at each step. Unfortunately, there is no globally well defined correspondence rule linking eigenvectors at different points, especially given that the trajectory followed depends strongly on the eigenvector(s) chosen. So in practice we must resort to one of several imperfect correspondence heuristics. For transition state searches we need only track one eigenvector (the one that is given $\sigma_i = -1$), so we will concentrate on this case. A simple approach would be to choose a fixed direction in space (perhaps the initial eigendirection) and take the eigenvector with maximal projection along this direction. But many such directions are possible and the most interesting saddles may happen not to have negative curvature along the particular direction chosen. Alternatively, we can try to track a given eigenvector as it changes. The problem is that globally, eigendirections are by no means stable. Eigenvalues change as we move about the space, but generically (in codimension 1) they never cross. When they approach one another, the eigenbasis of their 2D subspace becomes ill-conditioned and slews around through roughly 90° to avoid the collision. Seen from a large enough scale, the eigenvalues do seem to cross with more or less constant eigenvectors, but on a finer scale there is no crossing, only a smooth but rapid change of eigendirection that is difficult to track accurately. Whichever of the two behaviours is desired, it is difficult to choose a step length that reliably ensures it, so the numerical behaviour of eigenvector-tracking methods is often somewhat erratic. In fact, the imprecise coarse scale view is probably the desired one: if we are tracking a large eigenvalue and hoping to reduce it to something negative, it will have to "pass through" each of the smaller eigenvalues. Tracking at too fine a scale is fatal as it (correctly) prevents such crossings, instead making the method veer off at right angles to the desired trajectory.

Even without these problems there would be no guarantee that a saddle point of the desired signature was found (*e.g.* the trajectory could diverge to infinity). Also, as with

[4] This holds irrespective of the λ_i and $\bar{g}_i$ at the *current* state, which affect only the damping required for stability.

other damped Newton methods, the whole process is strongly dependent on the affine coordinate system used. Nevertheless, eigenvector tracking is relatively lightweight, simple to implement, and it often works well in practice.

2.3 Hypersurface Sweeping

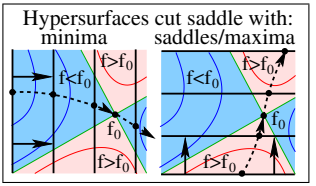


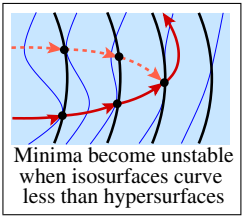
Eigenvector trackers do not enforce any notion of global progress, so they can sometimes behave erratically, *e.g.* cycling or stalling. To prevent this we can take a more global approach to the '$(n-1)$-D minimization and 1D maximization' required for transition state search. 'Hypersurface sweeping' approaches sweep an $(n-1)$-D hypersurface across the parameter space — typically a moving hyperplane or an expanding hyper-ellipsoid centred at the initial minimum — tracking local minima within the hypersurface and looking for temporal maxima in their function values. The intuition is that as the hypersurface expands towards a transition state, and assuming that it approaches along its negative curvature direction, the $(n-1)$-D minimization forces the hypersurface-minimum to move along the lowest path leading up to the saddle's 'col', and the 1-D maximization detects the moment at which the col is crossed. The method can not stall or cycle as the hypersurface sweeps through each point in the space exactly once.

The moving hypersurface can be defined either implicitly as the level sets $c(\mathbf{x}) = t$ of some function $c(\mathbf{x})$ on the parameter space (a linear form for hyperplanes, a quadratic one for hyper-ellipsoids...), or explicitly in terms of a local parametrization $\mathbf{x} = \mathbf{x}(\mathbf{y}, t)$ for some hypersurface-t-parametrizing $(n-1)$-D vector $\mathbf{y}$. The minimum-tracking problem becomes:

$$\text{local\_max}_t \; f_c(t) \qquad \text{where} \qquad f_c(t) \equiv \begin{cases} \text{local\_min}_{c(\mathbf{x})=t} \, f(\mathbf{x}) \\ \text{local\_min}_{\mathbf{y}} \, f(\mathbf{x}(\mathbf{y}, t)) \end{cases} \tag{3}$$

Different local minima on the hypersurface typically lead to different transition states. To find the lowest cost transition we would in principle have to track every minimum. More seriously, transitions that are cut by the hypersurfaces in negative curvature directions are missed: they appear as saddle points or local maxima within the hypersurface, and so can not be found by tracking only minima. Nor do local maxima of $f_c(t)$ always indicate true transition states. At any hypersurface-stationary point, f's isosurface is necessarily tangent to the local hypersurface: $\mathbf{g} \propto \frac{\partial c}{\partial \mathbf{x}}$ or $\mathbf{g}^\top \frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \mathbf{0}$. The point is a hypersurface-minimum, -saddle, or -maximum respectively as the cost isosurface has higher/mixed/lower signed curvature than the local hypersurface (*i.e.* as the isosurface is locally inside/mixed/outside the hypersurface). At points where the moving hypersurface transitions from being outside to being mixed w.r.t. the local isosurface, the minimum being tracked abruptly disappears and the solution drops away to some other local minimum on the hypersurface, causing an abrupt 'sawtooth' maximum in $f_c(t)$ (generically, the hypersurface-minimum collides with a hypersurface-saddle and is annihilated). The search





can continue from there, but it is important to verify that the maxima found really are saddle points.

As any given family of hypersurfaces is necessarily blind to some saddle orientations, it is wise to try a range of different families. Hyperplanes search preferentially along a fixed direction whereas hyper-ellipsoids can find saddles lying in any direction. The initial direction of the minimum trajectory is determined by the hyperplane normal or the ellipsoid shape. Near a minimum $\mathbf{x}_0$ with Hessian $\mathbf{H}_0$, consider the ellipsoids $c(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_0)^\top \mathbf{A}\,(\mathbf{x}-\mathbf{x}_0) = t$, where $\mathbf{A}$ is some positive definite matrix. To second order, $f(\mathbf{x})$ generically has exactly two local minima on an infinitesimal ellipsoid $c(\mathbf{x}) = t$: the $\pm$ directions of the smallest eigenvector of the matrix pencil[5] $\mathbf{A}+\lambda\,\mathbf{H}$. For most $\mathbf{A}$ there are thus only two possible initial trajectories for the moving minimum, and so at most two first saddles will be found. To find additional saddles we need to modify $\mathbf{A}$. We can enforce any desired initial direction $\mathbf{u}$ by taking the 'neutral' search ellipsoids $\mathbf{A} = \mathbf{H}$ (on which f is constant to second order, so that all initial directions are equally good) and flattening them slightly relative to the cost isosurfaces in the $\mathbf{u}$ direction. *E.g.*, to satisfy the Lagrange multiplier condition for a constrained minimum, $\frac{\partial c}{\partial \mathbf{x}} \propto \frac{\partial f}{\partial \mathbf{x}}$, we can take $\mathbf{A} = \mathbf{H} + \mu \frac{\mathbf{g}\,\mathbf{g}^\top}{\mathbf{u}^\top \mathbf{g}}$ where $\mathbf{g} = \mathbf{H}\,\mathbf{u}$ is the cost gradient (and hence isosurface normal) at displacements along $\mathbf{u}$ and μ is a positive constant, say $\mu \sim 0.1$ for mild flattening. Similarly, for hyperplanes $c(\mathbf{x}) = \mathbf{n}^\top(\mathbf{x}-\mathbf{x}_0) = t$ with normal $\mathbf{n}$, the initial minimum direction is $\mathbf{u} = \pm\mathbf{H}^{-1}\,\mathbf{n}$, so to search in direction $\mathbf{u}$ we need to take $\mathbf{n} = \mathbf{H}\,\mathbf{u}$.

The minimum tracking process is a fairly straightforward application of constrained optimization, but for completeness we summarize the equations needed in the appendix.

Summary: None of the current methods are foolproof. Damped Newton iteration is useful for refining estimated saddles but its convergence domain is too limited for general use. Eigenvector tracking extends the convergence domain but it is theoretically less sound (or at least, highly dependent on the step size and 'same eigenvector' heuristics). Hypersurface sweeping is better founded and provides at least weak guarantees of global progress, but it is more complex to implement and no single sweep finds all saddle points.

2.4 Implementation Details

We have tested several variants of each of the above methods. In the experiments below we focus on just two, which are summarized in fig. 1:

Hyper-ellipsoid sweeping: We start at a local minimum and use centred, curvature-eigenbasis-aligned ellipsoidal hypersurfaces flattened along one eigendirection, say the e^{th}. This restricts the initial search to an eigendirection (the e^{th}). This limitation could easily be removed, but gives a convenient, not-too-large set of directions to try. All calculations are performed in eigen-coordinates and the minimum is tracked using variable elimination (6) on x_e. In eigen-coordinates, the on-hypersurface constraint becomes $\sum_i (x_i/\sigma'_i)^2 = t^2$, where the σ'_i are the principal standard deviations, except that the e^{th} (eliminated) one is shrunk by say 20%. Solving for x_e gives $x_e(\mathbf{y}, t) = \pm\sigma'_e(t^2 - \sum_{i\neq e}(x_i/\sigma'_i)^2)^{1/2}$ where $\mathbf{y} = (x_1, ..., x_{e-1}, x_{e+1}, ..., x_n)$.

[5] Numerically, these can be found by generalized eigen-decomposition of $(\mathbf{A}, \mathbf{H})$, or standard eigen-decomposition of $\mathbf{L}^{-\top}\mathbf{H}\mathbf{L}^{-1}$ where $\mathbf{L}\mathbf{L}^\top$ is the Cholesky decomposition of $\mathbf{A}$.

Hyper-ellipsoid Sweeping Transition State Search

1. Initialization

Given initial minimum $\mathbf{x}_0$ with Hessian $\mathbf{H}$, eigen-decompose $\mathbf{H}$ to $(\lambda_i, \mathbf{v}_i)$ with principal radii $\sigma_i = 1/\sqrt{\lambda_i}$. Choose an initial search eigen-direction e. Shrink σ_e by say 20% and prepare to eliminate x_e. Set initial step $\mathbf{x}_1 = \mathbf{x}_0 + t_1\sigma_e\mathbf{v}_e$ where t_1 is say 3. Go to step 2.B.

2. Loop, Updating Hypersurface and Minimizing

A. k=k+1. Estimate an initial $\mathbf{x}_k$ by linear extrapolation to the trust radius. Compute the resulting t_k.

B. Minimize f on the t_k ellipsoid to get $f_c(t_k)$: $\mathbf{y}_k = \arg\min_{\mathbf{y}} f(\mathbf{x}_k(\mathbf{y}, t_k))$.

C. Compute $f_c' = \frac{\partial}{\partial t} f_c(t_k)$. If $f_c' < \epsilon$ we are near or past saddle: go to step 3.A. Otherwise go to step 2.A.

3. Line Search for Transition State Refinement

A. If $|f_c'| < \epsilon$, exit.

B. k=k+1. Estimate t_{saddle} by linear interpolation of last two f_c' values.

B. Optimize $\mathbf{y}_k$ as in step 2.B and go to step 3.A.

Eigenvector Tracking Transition State Search

Initialization

Set starting point $\mathbf{x}_0$, initial tracking direction $\mathbf{t}$ and initial trust radius r.

Eigenvector Tracking Loop

A. At $\mathbf{x}_k$, find $f_k, \mathbf{g}_k, \mathbf{H}_k$, the Hessian eigen-decomposition $(\lambda_i, \mathbf{v}_i)$ and eigen-basis gradient $\bar{g}_k$. Set n_- to the number of negative eigenvalues. If the problem has active internal constraints, project $\mathbf{t}$ onto the constraint surface.

B. If $k > 0$ choose $e = \max_i |\mathbf{v}_i^\top \mathbf{t}|$. Set $\alpha = |\mathbf{v}_e^\top \mathbf{t}|$ and $\mathbf{t} = \mathbf{v}_e$.

C. If $\lambda_e > 0$ set $\bar{g}_e = -\bar{g}_e$. Take an undamped Newton step if n_-=1 and $\|\boldsymbol{\delta}\mathbf{x}\| \leq r$. Otherwise take a damped one with λ chosen so that $\|\boldsymbol{\delta}\mathbf{x}\| \leq r$.

D. Find the new f and the modelling error β (2). If $\beta > 0.3$ (say) or $\alpha < 0.7$ (say), shrink the trust radius r by say 50%. Otherwise, if $\beta < 0.2$ (say) and we took a damped step, grow r by say 40%.

E. If $\beta \geq 1$ go to step C. If $\|\mathbf{g}_k\| < \epsilon$ return success if n_-=1, failure otherwise. Otherwise, go to step A.

Fig. 1. Our ellipsoid sweeping and eigenvector tracking algorithms for transition state search.

Derivatives are easily found. At each time step we predict the new minimum by linear extrapolation from the previous two, $\mathbf{x} = \mathbf{x}_k + r\,\frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|}$ where r is a trust region radius for $\boldsymbol{\delta}\mathbf{x}$, then solve for the corresponding t_{k+1} using the ellipsoid constraint.

Eigenvector tracker: We use the damped Newton saddle step (1), moving away from the minimum by reversing the sign of the gradient in the tracked eigendirection if this has positive curvature. The damping $\lambda > 0$ is controlled to keep the step within a trust radius r and to dominate any undesired negative eigenvalues. The trust radius is set by monitoring the accuracy (2) of the local model for f.

In some of our target applications, the underlying problem has bound constraints that must be maintained. For hypersurface sweeping this just adds additional constraints to the within-hypersurface minimizations. For eigenvector following, our trust region step routine uses a projection strategy to handle constraints on $\mathbf{x}$, and also projects the eigenvector-tracking direction $\mathbf{t}$ along the constraints to ensure stability.

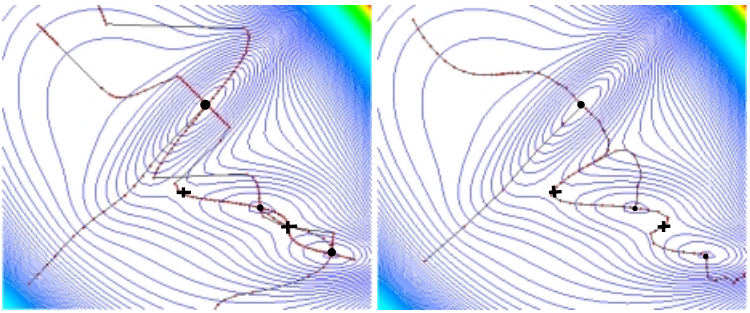


Fig. 2. Trajectories for the hyper-ellipsoid sweeping (left) and eigenvector following (right) algorithms on the Müller cost surface, initialized along the $\pm$ eigendirections of the 3 minima.

3 Human Domain Modelling

Representation: The 3D body model used in our human tracking experiments consists of a kinematic 'skeleton' of articulated joints controlled by angular joint parameters, covered by a 'flesh' built from superquadric ellipsoids with additional global deformations [4]. A typical model has 30–35 joint parameters; 8 internal proportions encoding the positions of the hip, clavicle and skull tip joints; and 9 deformable shape parameters for each body part. The complete model is encoded in a single large parameter vector $\mathbf{x}$. During tracking and static pose estimation we usually estimate only joint parameters, but during initialization some length ratios are also estimated. In use, the superquadric surfaces are discretized into 2D meshes and the mesh nodes are mapped to 3D points using the kinematic body chain then projected to predicted image points $\mathbf{r}_i(\mathbf{x})$ using perspective image projection.

Observation Likelihood: Robust model-to-image matching cost metrics are evaluated for each predicted image feature $\mathbf{r}_i$, and the results are summed over all observations to produce the image contribution to the parameter space cost function. Cost gradient and Hessian contributions $\mathbf{g}_i, \mathbf{H}_i$ are also computed and assembled. We use a robust combination of extracted-feature-based metrics and intensity-based ones such as optical flow, robustified normalized edge energy and potentials derived from silhouette distance transforms [32]. The feature-based terms associate the predictions $\mathbf{r}_i$ with nearby image features $\bar{\mathbf{r}}_i$, the cost being a robust function of the prediction errors $\boldsymbol{\Delta}\mathbf{r}_i(\mathbf{x}) = \bar{\mathbf{r}}_i - \mathbf{r}_i(\mathbf{x})$. We also give results for a simpler likelihood designed for model initialization, based on squared distances between reprojected model joints and their specified image positions.

Priors and Constraints: Our model [33, 34] incorporates both hard constraints (for joint angle limits) and soft priors (penalties for anthropometric model proportions, collision avoidance between body parts, and stabilization of useful but hard-to-estimate model parameters such as internal d.o.f. of the clavicle complex). In the experiments below we use mainly joint angle limits and body part non-interpenetration constraints. The priors provide additional cost, gradient and Hessian contributions for the optimization.

Estimation: We apply Bayes rule and maximize the total posterior probability to give locally MAP parameter estimates:

$$\log p(\mathbf{x}|\bar{\mathbf{r}}) \;\propto\; \log p(\mathbf{x}) + \log p(\bar{\mathbf{r}}|\mathbf{x}) \;=\; \log p(\mathbf{x}) - \int e(\bar{\mathbf{r}}_i|\mathbf{x})\, di \tag{4}$$

Here, $p(\mathbf{x})$ is the prior on the model parameters, $e(\bar{\mathbf{r}}_i|\mathbf{x})$ is the cost density associated with observation i, and the integral is over all observations (assumed independent). Equation (4) gives the model likelihood in a single image, under the model priors but without initial state or temporal priors. During tracking, the temporal prior at time t is determined by the previous posterior $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$ and the system dynamics $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, where we have collected the observations at time t into vector $\mathbf{r}_t$ and defined $\mathbf{R}_t = \{\mathbf{r}_1, \ldots, \mathbf{r}_t\}$. The posterior at t becomes:

$$p(\mathbf{x}_t|\mathbf{R}_t) \;\propto\; p(\bar{\mathbf{r}}_t|\mathbf{x}_t)\, p(\mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$$

Together $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$ form the time t prior $p(\mathbf{x}_t|\mathbf{R}_{t-1})$ for the image correspondence search (4).

4 Experiments

We illustrate our transition state search algorithms on a 2 d.o.f. toy problem, and on 3D human pose and motion estimation from monocular images.

The Müller Potential: This simple analytical 2D cost function[6] is often used to illustrate transition state methods in chemistry. Fig. 2 shows its 3 minima and 2 saddles (the black dots and crosses) and plots the trajectories of the two methods starting from each minimum. The hypersurface sweeping algorithm is run for extended trajectories through several saddles and minima (left plot). In this simple example, a single sweep started at the top left minimum successfully finds all of the other minima and transition states.

Articulated 3D Human Motion Estimation: There is a large literature on human motion tracking but relatively little work on the thorny issue of local minima in the difficult 3D-from-monocular case. Cham & Rehg [7] combine local optimization and condensation sampling for 2D tracking. Deutscher *et al* [12] use an annealed sampling method and multiple cameras. Sidenbladh *et al* [30] use particle filtering with importance sampling based on a learned walking model. Sminchisescu & Triggs [34] combine robust constraint-consistent local continuous optimization with a covariance-scaled sampling

[6] It has the form: $V(x,y) = \sum_{i=1}^{4} A_i\, e^{a_i(x-x_i)^2 + b_i(x-x_i)(y-y_i) + c_i(y-y_i)^2}$, where: $\mathbf{A} = (-200, -100, -170, 15)$, $\mathbf{a} = (-1, -1, -6.5, 0.7)$, $\mathbf{b} = (0, 0, 11, 0.6)$, $\mathbf{c} = (-10, -10, -6.5, 0.7)$, $\mathbf{x} = (1, 0, -0.5, -1)$, $\mathbf{y} = (0, 0.5, 1.5, 1)$.

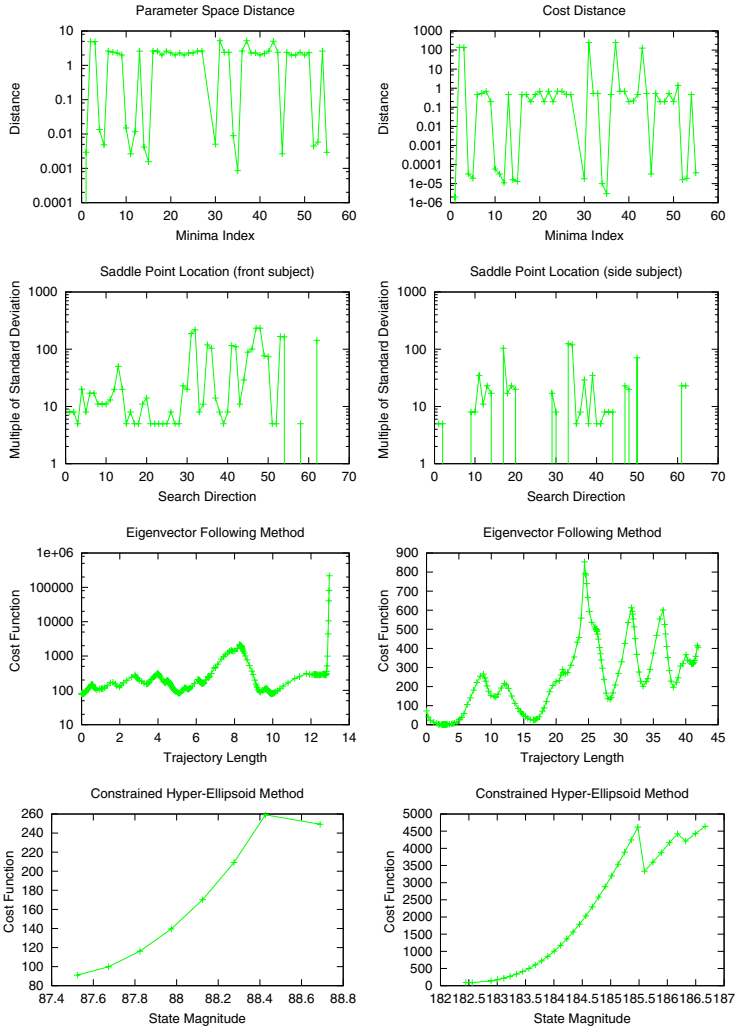


Fig. 3. Transition state and minimum location algorithms for ± 32-eigendirection search trials. Top row: parameter space distance and cost difference between initial and current minimum. Second row: saddle point distances in standard deviations for a frontal and a partly side-on 3D pose. Third and fourth rows: cost profiles for different trajectories and constraints (see text).

method that focuses samples in regions likely to have low cost. All of these works note the difficulty of the multiple-minimum problem and attempt to develop techniques or constraints (on the scene, motion, number of cameras or background) to tackle it.

Here we show a few examples from a larger set of experiments with a 32 d.o.f. articulated full-body model, including pose estimation and tracking in monocular images using cost surfaces based on different combinations of image cues. The figures show

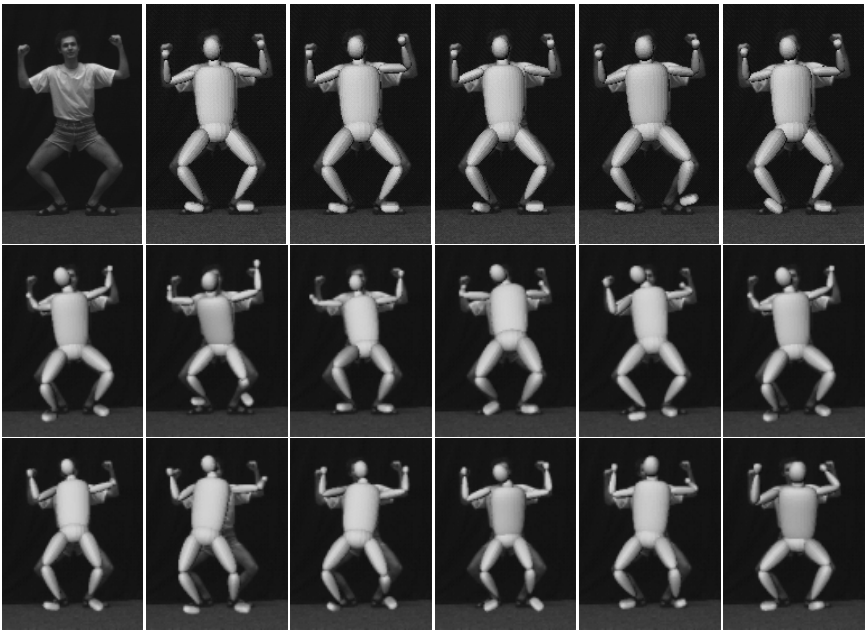


Fig. 4. Minima of image-based cost functions. Top row: contour and optical flow likelihood. Middle and bottom rows: silhouette and edge likelihood.

examples of minima found in likelihood models based on image contours and optical flow (fig. 4, top row), contours and silhouette-image data (fig. 4, middle and bottom rows), and model-to-image joint correspondences (fig. 5).

Fig. 3 attempts to capture some more quantitative information about the methods, here for the joint correspondence cost function. The first row displays the parameter space and cost distances of the 56 minima found during a set of 64 constrained searches (the $\pm$ directions of the 32 eigenvectors of an initial minimum, distances being measured w.r.t. this minimum, in radians and metres for the parameter space). The second row again shows parameter space distances, but now measured in standard deviations and for saddles rather than minima, for the same frontal view and for a slightly more side-on one (fig. 5, respectively top and bottom). The plots reveal the structure of the cost surface, with nearby saddles at 4–8 standard deviations and progressively more remote ones at 20–50, 80–100 and 150–200 standard deviations. It follows that no multiple-minimum exploration algorithm can afford to search only within the 'natural' covariance scale of its current minima: significantly deeper sampling is needed to capture even nearby minima (as previously noted, *e.g.* by [33, 34]).

The last two rows of fig. 3 show some sample cost profiles for typical runs of the eigenvector following (row 3) and constrained hyper-surface (row 4) saddle search methods. In the eigenvector method, it is preferable to represent the joint limits using a 'hard' active set strategy (row 3 right) rather than soft constraints (row 3 left): the stiff 'cost walls' induced by the soft constraints tend to force the eigenvector follower into head-on

collision with the wall, with the cost climbing rapidly to infinity. The active set strategy avoids this problem at the price of more frequent direction changes as the joint limits switch on and off. The hyper-ellipsoid method (row 4) produces more stable trajectories that do not require special joint limit processing, but its cost profiles have characteristic sawtooth edges (row 4 right) associated with sudden state readjustments on the hypersphere at points where the tracked minimum becomes locally unstable.

Fig. 4 shows minima for costs based on various combinations of image cues. In the first row the minima correspond to a small interframe motion, using contour and robust optical flow information. This case has relatively few, but closely spaced local minima owing to the smoothing/quadratic effect of the flow. (Remoter minima do still exist at points where the robust contributions of sets of flow measurements turn off, particularly when these coincide with incorrect edge assignments). The second and third rows show minima arising from a silhouette and edge based cost function. The minima shown include 'reflective' (depth-related) ambiguities, incorrect edge assignments and singular 'inside-silhouette' configurations (which can be alleviated to some extent by augmenting the likelihood term as in [32]).

Finally, fig. 5 shows depth ambiguities for articulated 3D frontal and side poses, under model to image joint correspondences. The arm-shoulder complex is very flexible and therefore tends to induce more minima than the legs. We also find that side views tend to generate fewer minima than frontal ones, perhaps due to presence of body-part non-self-intersection and joint constraints that render many 'purely reflective' minima infeasible.

5 Conclusions and Future Work

We have described two families of deterministic-optimization-based algorithms for finding 'transition states' (saddle points with 1 negative eigenvalue) in high-dimensional multi-modal cost surfaces. These allow us to build topological 'roadmaps' of the nearby local minima and the transition states that lead to them. The methods are based on ones developed in computational chemistry, but here generalized, clarified and adapted for use in computational vision. Experiments on the difficult problem of articulated 3D human pose from monocular images show that our algorithms can stably and efficiently recover large numbers of transition states and minima, but also serve to underline the very large numbers of minima that exist in this problem.

We are currently applying our methods to other multimodal problems in vision, including structure from motion. We are also trying to use them to quantify the degrees of ambiguity of different cost functions, with the longer term goal of designing better cost functions based on higher-level features and groupings.

Acknowledgements. This work was supported by an EIFFEL doctoral grant and European Union FET-Open project VIBES. We would like to thank Alexandru Telea for discussions on implementation.

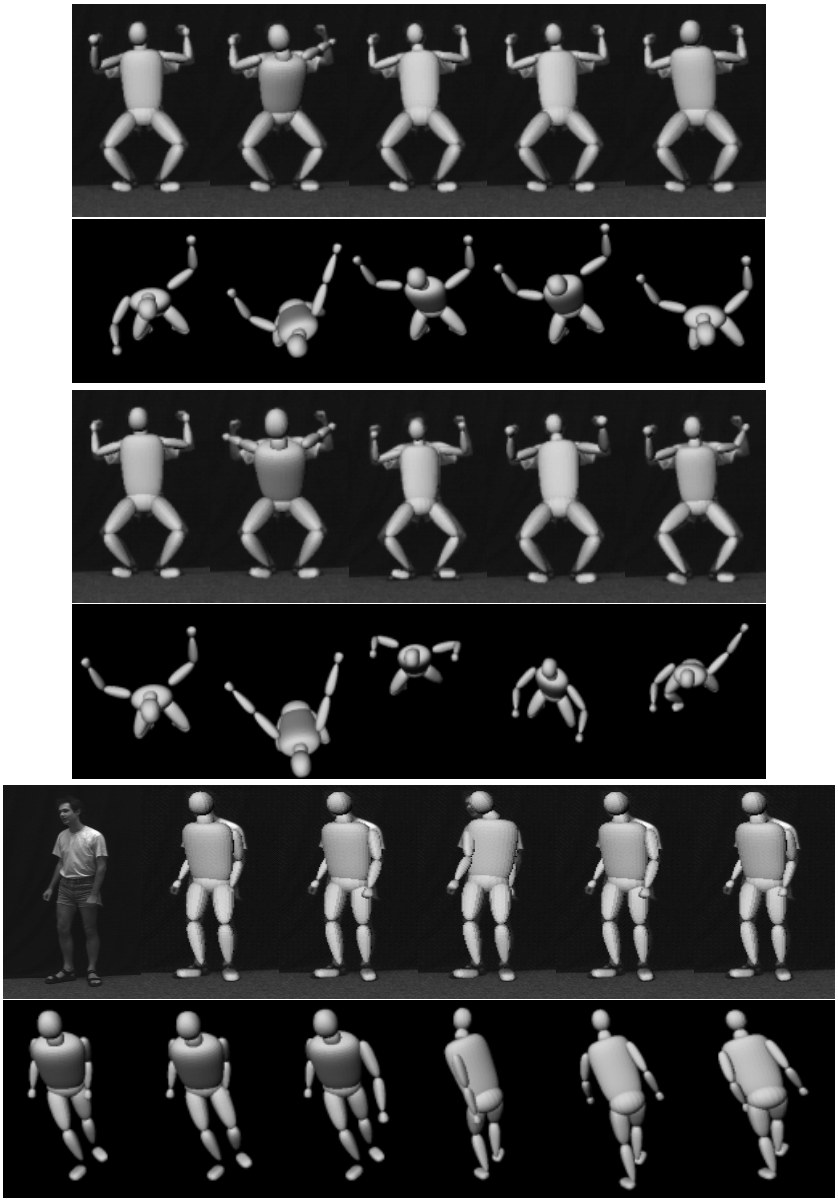


Fig. 5. 'Reflective' kinematic ambiguities under the model/image joint correspondence cost function. Each pair of rows displays the original image overlayed with the projected model, and the 3D model position seen from a fixed synthetic overhead camera. Note the pronounced forwards-backwards character of these reflective minima, and the large parameter space distances that often separate of them.

Appendix: Implementation of Hypersurface Sweeping

Here we summarize the equations needed to implement hypersurface sweeping for both implicit and parametric hypersurfaces. For the implicit approach, let $\mathbf{g}_c \equiv \frac{\partial c}{\partial \mathbf{x}}$ and also $\mathbf{H}_c \equiv \frac{\partial^2 c}{\partial \mathbf{x}^2}$. The hypersurface constraint is enforced with a Lagrange multiplier λ, solving $\frac{\partial}{\partial \mathbf{x}}(f+\lambda c) = \mathbf{g} + \lambda \mathbf{g}_c = \mathbf{0}$ subject to $c = t$. If we are currently at $(\mathbf{x}, \lambda)$, second order Taylor expansion of these equations for a constrained minimum at $(\mathbf{x}+\boldsymbol{\delta}\mathbf{x}, \lambda+\delta\lambda)$ gives the standard **sequential quadratic programming** update rule for $(\boldsymbol{\delta}\mathbf{x}, \delta\lambda)$:

$$\begin{pmatrix} \mathbf{H}_\lambda & \mathbf{g}_c \\ \mathbf{g}_c^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}\mathbf{x} \\ \delta\lambda \end{pmatrix} = -\begin{pmatrix} \mathbf{g} + \lambda \mathbf{g}_c \\ c - t \end{pmatrix} \qquad \text{where} \qquad \mathbf{H}_\lambda \equiv \mathbf{H} + \lambda \mathbf{H}_c \tag{5}$$

(The $\lambda \mathbf{H}_c$ term in the Hessian is often dropped for simplicity. This slows the convergence but still gives correct results). Similarly, in the parametric approach let $\mathbf{J} \equiv \frac{\partial}{\partial \mathbf{y}} \mathbf{x}(\mathbf{y}, t)$. The chain rule gives the reduced gradient $\mathbf{g}_\mathbf{y} = \mathbf{J}\,\mathbf{g}$ and Hessian $\mathbf{H}_\mathbf{y} = \mathbf{J}\,\mathbf{H}\,\mathbf{J}^\top + (\frac{\partial}{\partial \mathbf{y}}\mathbf{J})\,\mathbf{g}$. These can be used directly in the Newton update rule $\boldsymbol{\delta}\mathbf{y} = -\mathbf{H}_\mathbf{y}^{-1}\mathbf{g}_\mathbf{y}$. In particular, if we eliminate one $\mathbf{x}$-variable — say x_n so that $\mathbf{y} = (x_1, ..., x_{n-1})$ and $x_n = x_n(\mathbf{y}, t)$ — we have:

$$\mathbf{J} = \left(\mathbf{I} \,|\, \frac{\partial x_n}{\partial \mathbf{y}}\right), \qquad \mathbf{g}_\mathbf{y} = \frac{\partial f}{\partial \mathbf{y}} + g_n \frac{\partial x_n}{\partial \mathbf{y}}, \qquad \mathbf{H}_\mathbf{y} = \mathbf{J}\,\mathbf{H}\,\mathbf{J}^\top + g_n \frac{\partial x_n}{\partial \mathbf{y}} \tag{6}$$

To save optimization work and for convergence testing and step length control, it is useful to be able to extrapolate the position and value of the next minimum from existing values. This can be done, *e.g.*, by linear extrapolation from two previous positions, or analytically by solving the constrained minimum state update equations $(\mathbf{g}+(\lambda+\delta\lambda)\mathbf{g}_c)(\mathbf{x}+\boldsymbol{\delta}\mathbf{x}) = \mathbf{0}$ or $\mathbf{g}_\mathbf{y}(\mathbf{y}+\boldsymbol{\delta}\mathbf{y}, t+\delta t) = \mathbf{0}$ to first order, assuming that $\mathbf{x}, t$ is already a minimum and $t \to t+\delta t$:

$$(\boldsymbol{\delta}\mathbf{x},\ \delta\lambda) = \frac{\delta t}{\mathbf{g}_c^\top \mathbf{H}_\lambda^{-1} \mathbf{g}_c} (\mathbf{H}_\lambda^{-1}\mathbf{g}_c,\ -1) \tag{7}$$

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{J}\,\boldsymbol{\delta}\mathbf{y} + \frac{\partial \mathbf{x}}{\partial t}\,\delta t, \qquad \boldsymbol{\delta}\mathbf{y} = -\mathbf{H}_\mathbf{y}^{-1}\left(\frac{\partial \mathbf{J}}{\partial t}\,\mathbf{g} + \mathbf{J}\,\mathbf{H}\,\frac{\partial \mathbf{x}}{\partial t}\right)\delta t \tag{8}$$

Taylor expansion of $f(\mathbf{x}+\boldsymbol{\delta}\mathbf{x})$ then gives $f_c(t+\delta t) \approx f_c(t) + f_c'\,\delta t + \frac{1}{2} f_c''\,\delta t^2$ with $f_c' = \mathbf{g}\,\frac{\delta \mathbf{x}}{\delta t}$ and $f_c'' = \frac{\delta \mathbf{x}}{\delta t}^\top \mathbf{H}\,\frac{\delta \mathbf{x}}{\delta t}$. For step length control, we can either fix δt and solve for $\boldsymbol{\delta}\mathbf{x}$ or $\boldsymbol{\delta}\mathbf{y}$ (and hence $\mathbf{x}+\boldsymbol{\delta}\mathbf{x} \equiv \mathbf{x}(\mathbf{y}+\boldsymbol{\delta}\mathbf{y}, t+\delta t)$), or fix a desired trust region for $\boldsymbol{\delta}\mathbf{x}$ or $\boldsymbol{\delta}\mathbf{y}$ and work backwards to find a δt giving a step within it.

References

[1] Y. Abashkin and N. Russo. Transition State Structures and Reaction Profiles from Constrained Optimization Procedure. Implementation in the Framework of Density Functional Theory. *J. Chem. Phys.*, 1994.

[2] Y. Abashkin, N. Russo, and M. Toscano. Transition States and Energy Barriers from Density Functional Studies: Representative Isomerization Reactions. *International Journal of Quantum Chemistry*, 1994.

[3] G. T. Barkema. Event-Based Relaxation of Continuous Disordered Systems. *Physical Review Letters*, 77(21), 1996.
[4] A. Barr. Global and Local Deformations of Solid Primitives. *Computer Graphics*, 18:21–30, 1984.
[5] J. M. Bofill. Updated Hessian Matrix and the Restricted Step Method for Locating Transition Structures. *Journal of Computational Chemistry*, 15(1):1–11, 1994.
[6] C. J. Cerjan and W. H. Miller. On Finding Transition States. *J. Chem. Phys.*, 75(6), 1981.
[7] T. Cham and J. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In *CVPR*, volume 2, pages 239–245, 1999.
[8] A. Chiusso, R. Brockett, and S. Soatto. Optimal structure from motion: Local ambiguities and global estimates. *IJCV*, 39(3):195–228, 2000.
[9] K. Choo and D. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *ICCV*, 2001.
[10] G. M. Crippen and H. A. Scheraga. Minimization of Polypeptide Energy. XI. The Method of Gentlest Ascent. *Archives of Biochemistry and Biophysics*, 144:462–466, 1971.
[11] P. Culot, G. Dive, V. H. Nguyen, and J. M. Ghuysen. A Quasi-Newton Algorithm for First-Order Saddle Point Location. *Theoretica Chimica Acta*, 82:189–205, 1992.
[12] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *CVPR*, 2000.
[13] R. Fletcher. Practical Methods of Optimization. In *John Wiley*, 1987.
[14] D. Gavrila and L. Davis. 3-D Model Based Tracking of Humans in Action:A Multiview Approach. In *CVPR*, pages 73–80, 1996.
[15] T. Helgaker. Transition-State Optimizations by Trust-Region Image Minimization. *Chemical Physics Letters*, 182(5), 1991.
[16] G. Henkelman and H. Jonsson. A Dimer Method for Finding Saddle Points on High Dimensional Potential Surfaces Using Only First Derivatives. *J. Chem. Phys.*, 111(15):7011–7022, 1999.
[17] R. L. Hilderbrandt. Application of Newton-Raphson Optimization Techniques in Molecular Mechanics Calculations. *Computers & Chemistry*, 1:179–186, 1977.
[18] F. Jensen. Locating Transition Structures by Mode Following: A Comparison of Six Methods on the Ar_8 Lennard-Jones potential. *J. Chem. Phys.*, 102(17):6706–6718, 1995.
[19] P. Jorgensen, H. J. A. Jensen, and T. Helgaker. A Gradient Extremal Walking Algorithm. *Theoretica Chimica Acta*, 73:55–65, 1988.
[20] J. MacCormick and M. Isard. Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracker. In *ECCV*, volume 2, pages 3–19, 2000.
[21] D. Morris and J. Rehg. Singularity Analysis for Articulated Object Tracking. In *CVPR*, pages 289–296, 1998.
[22] N. Mousseau and G. T. Berkema. Traveling Through Potential Energy Lanscapes of Desordered Materials: The Activation-Relaxation Technique. *Physical Review E*, 57(2), 1998.
[23] L. J. Munro and D. J. Wales. Defect Migration in Crystalline Silicon. *Physical Review B*, 59(6):3969–3980, 1999.
[24] J. Nichols, H. Taylor, P. Schmidt, and J. Simons. Walking on Potential Energy Surfaces. *J. Chem. Phys.*, 92(1), 1990.
[25] J. Oliensis. The Error Surface for Structure from Motion. Technical report, NECI, 2001.
[26] R. Plankers and P. Fua. Articulated Soft Objects for Video-Based Body Modeling. In *ICCV*, pages 394–401, 2001.
[27] J. Rehg and T. Kanade. Model-Based Tracking of Self Occluding Articulated Objects. In *ICCV*, pages 612–617, 1995.
[28] R. Rosales and S. Sclaroff. Inferring Body Pose without Tracking Body Parts. In *CVPR*, pages 721–727, 2000.

[29] E. M. Sevick, A. T. Bell, and D. N. Theodorou. A Chain of States Method for Investigating Infrequent Event Processes Occuring in Multistate, Multidimensional Systems. *J. Chem. Phys.*, 98(4), 1993.
[30] H. Sidenbladh, M. Black, and D. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *ECCV*, 2000.
[31] J. Simons, P. Jorgensen, H. Taylor, and J. Ozmen. Walking on Potential Energy Surfaces. *J. Phys. Chem.*, 87:2745–2753, 1983.
[32] C. Sminchisescu. Consistency and Coupling in Human Model Likelihoods. In *CFGR*, 2002.
[33] C. Sminchisescu and B. Triggs. A Robust Multiple Hypothesis Approach to Monocular Human Motion Tracking. Technical Report RR-4208, INRIA, 2001.
[34] C. Sminchisescu and B. Triggs. Covariance-Scaled Sampling for Monocular 3D Body Tracking. In *CVPR*, 2001.
[35] C. Sminchisescu and B. Triggs. Hyperdynamics Importance Sampling. In *ECCV*, 2002.
[36] J. Q. Sun and K. Ruedenberg. Gradient Extremals and Stepest Descend Lines on Potential Energy Surfaces. *J. Chem. Phys.*, 98(12), 1993.
[37] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In Springer-Verlag, editor, *Vision Algorithms: Theory and Practice*, 2000.
[38] S. Wachter and H. Nagel. Tracking Persons in Monocular Image Sequences. *CVIU*, 74(3):174–192, 1999.
[39] D. J. Wales. Finding Sadle Points for Clusters. *J. Chem. Phys.*, 91(11), 1989.
[40] D. J. Wales and T. R. Walsh. Theoretical Study of the Water Pentamer. *J. Chem. Phys.*, 105(16), 1996.