Int J Comput Vis (2009) 84: 1–20
DOI 10.1007/s11263-008-0195-8

Nonlinear Mean Shift over Riemannian Manifolds

Raghav Subbarao · Peter Meer

Received: 28 January 2008 / Accepted: 3 November 2008 / Published online: 28 March 2009

Abstract The original mean shift algorithm is widely applied for nonparametric clustering in vector spaces. In this paper we generalize it to data points lying on Riemannian manifolds. This allows us to extend mean shift based clustering and filtering techniques to a large class of frequently occurring non-vector spaces in vision. We present an exact algorithm and prove its convergence properties as opposed to previous work which approximates the mean shift vector. The computational details of our algorithm are presented for frequently occurring classes of manifolds such as matrix Lie groups, Grassmann manifolds, essential matrices and symmetric positive definite matrices. Applications of the mean shift over these manifolds are shown.

Keywords Mean shift · Clustering · Riemannian manifolds

1 Introduction

The mean shift algorithm was first proposed in Fukunaga and Hostetler (1975) and was further discussed in Cheng (1995). However, it became truly popular after (Comaniciu and Meer 2002; Comaniciu et al. 2003) used mean shift for color image segmentation and motion tracking. Since then, mean shift has been applied to many different problems including image segmentation (Wang et al. 2004; Yang et al. 2003), tracking (Birchfield and Rangarajan 2005; Collins 2003; Elgammal et al. 2003; Hager et al. 2004) and robust fusion (Chen and Meer 2005; Comaniciu 2003).

R. Subbarao (✉) · P. Meer
Electrical and Computer Engineering Department, Rutgers University, 94 Brett Road, Piscataway, NJ, 08854-8058, USA
e-mail: rsubbara@eden.rutgers.edu

P. Meer
e-mail: meer@caip.rutgers.edu

Mean shift is essentially a clustering technique for finding meaningful centers of arbitrarily distributed points in a vector space. As it makes no assumptions about the nature of the distribution which generated the points, mean shift belongs to the class of nonparametric clustering methods. Mean shift has alternatively been shown to be equivalent to gradient ascent optimization of a kernel density (Comaniciu and Meer 2002), bounds optimization (Fashing and Tomasi 2005) and expectation-maximization (Carreira-Perpinan 2007). In practice, the popularity of mean shift is due to the fact that the algorithm is easy to implement while exhibiting very good convergence properties.

In this paper we present a generalized form of mean shift which can be used to cluster points which do not lie on a vector space. For example, consider a set of points lying on the surface of a sphere. As we explain later, each iteration of mean shift requires the weighted sum of the data points around our current position. However, the weighted sum of points on the surface of the sphere does not lie on the sphere. In this case, it is possible to use our geometric intuition to estimate a mean, but geometric constraints often lead to much more complex curved surfaces where it is not possible to use our intuition.

A more complex example would be a motion segmentation problem. It has been shown that robust estimation of motion parameters with voting based techniques, where the votes from different parameter hypothesis are aggregated, give good results (Mordohai and Medioni 2007). This would require the clustering of motion hypotheses. To use mean shift for this purpose, we would need a formal definition of notions such as the distance between motions and the average of a set of motions. These ideas have been well studied in fields such as differential geometry, physics and robotics and are formalized by the motion of a Riemannian manifold.

In this paper we present a mean shift algorithm for clustering points on Riemannian manifolds.

Riemannian manifolds appear frequently in computer vision due to the geometric nature of the problems. In Ma et al. (2001), geometric cost functions for reconstruction were treated as real valued functions on appropriate manifolds and minimized over these manifolds. The idea of treating motions as points on manifolds was used in Govindu (2004) to smooth motions. In Pennec and Ayache (1998), it was shown that distances and distributions over manifolds can be defined based on the geometry they represent. The recent interest in the theory of manifolds is also due to the novel imaging data of new medical systems. Diffusion tensor images measure the diffusivity of water molecules and the smoothing of these images requires an understanding of the manifold nature of diffusion matrices (Arsigny et al. 2006; Lenglet et al. 2004; Pennec et al. 2006; Vemuri et al. 2001). New image representations have also been developed for the statistical analysis of medical data using appropriate manifolds (Fletcher et al. 2003, 2004; Davis et al. 2007).

In Tuzel et al. (2005) mean shift was extended to a particular class of Riemannian manifolds, matrix Lie groups. A similar, but more general algorithm was proposed in Subbarao and Meer (2006), which could handle points lying on any Riemannian manifold, not necessarily Lie groups. Simultaneously, Begelfor and Werman (2006) proposed a slightly different mean shift algorithm for a specific class of manifolds, Grassmann manifolds. Although these algorithms give good results, here has been no analysis of their theoretical properties. Here, we present a proper derivation of nonlinear mean shift and discuss its theoretical properties. In deriving the nonlinear mean shift algorithm we have concentrated on issues such as ease of implementation and computational efficiency. For example, a proper definition of kernel densities over manifolds would be too complicated. Consequently, we choose to define a reasonable approximation whose gradient can be easily computed. This work has various novelties as compared to previous work on nonlinear mean shift. These are listed below.

- A general mean shift algorithm valid for all Riemannian manifolds is derived.
- For symmetric matrices and Grassmann manifolds, the algorithm proposed here is different from previous work.
- The theoretical properties of the algorithm are discussed. We present a proof of convergence of the nonlinear mean shift for the first time.

The rest of the paper if organized as follows. Section 2 provides a brief introduction to the theory of manifolds. This introduction is extremely basic and where possible we try to supply some geometric intuition to the meanings of the various terms. A thorough introduction to Riemannian geometry can be found in O'Neill (1983). Examples of commonly occurring Riemannian manifolds in computer vision are introduced in Sect. 3. The mean shift algorithm of Comaniciu and Meer (2002) is derived in Sect. 4 and our nonlinear mean shift algorithm is presented in Sect. 5. The computational details of the nonlinear mean shift algorithm over different manifolds are discussed in Sect. 6 and the theoretical properties of our algorithm are discussed in Sect. 7. The results of applying nonlinear mean shift to different problems are shown in Sect. 8.

Note, we represent the points on manifolds by small bold letters, e.g., $\mathbf{x}, \mathbf{y}$. In some of our examples, the manifold is a set of matrices. Although matrices are conventionally represented by capital bold letters, when we consider them to be points on a manifold, we denote them by small letters. However, in certain cases such as the essential manifold, we continue to denote points on the manifold by capital bold letters since essential matrices are conventionally represented by capital bold letters. The meaning should be clear from the context, but we reiterate that small bold letters do not necessarily represent just vectors.

2 Analytic Manifolds

A *manifold*, $\mathcal{M}$ is a Hausdorff topological space (O'Neill 1983, Chap. 1), such that for every point $\mathbf{x} \in \mathcal{M}$ there exists a neighborhood $\mathcal{U} \subset \mathcal{M}$ containing $\mathbf{x}$ and an associated homeomorphism ϕ from $\mathcal{U}$ to some Euclidean space $\mathbb{R}^m$, such that $\phi(\mathcal{U})$ is an open set in $\mathbb{R}^m$. The neighborhood $\mathcal{U}$ and its associated mapping ϕ together form a *coordinate chart*. A set of coordinate charts which cover the whole manifold is an *atlas*. Intuitively, a manifold is a space that is locally similar to an Euclidean space and this similarity is defined by the coordinate charts at each point. It is generally not possible to define global coordinates which make the whole manifold look like Euclidean space.

Consider two different coordinate charts $(\mathcal{U}, \phi)$ and $(\mathcal{V}, \psi)$ such that $\mathcal{U} \cap \mathcal{V}$ is not empty. Then, the *transition map* $\phi \circ \psi^{-1}$ is a mapping from the open set $\psi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^m$ to the open set $\phi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^m$. Further structure can be introduced on the manifold through the behaviour of the transition maps.

An *analytic manifold* is a manifold such that for all coordinate charts $(\mathcal{U}, \phi)$ and $(\mathcal{V}, \psi)$, either $\mathcal{U} \cap \mathcal{V}$ is empty or $\mathcal{U} \cap \mathcal{V}$ is nonempty and the transition map $\phi \circ \psi^{-1}$ is analytic, *i.e.*, has a convergent Taylor series expansion (Fig. 1). From now on we restrict ourselves to analytic manifolds.

Consider a real valued function $f : \mathcal{M} \to \mathbb{R}$ on the manifold. Given a coordinate chart $(\mathcal{U}, \phi)$, the function $f \circ \phi^{-1}$ maps the open set $\phi(\mathcal{U}) \in \mathbb{R}^m$ to $\mathbb{R}$. The function f is said to be continuous, if for all coordinate charts $\tilde{f} = f \circ \phi^{-1}$ is continuous when viewed as a function from $\mathbb{R}^m$ to $\mathbb{R}$. Similarly, f is said to be *analytic* if $\tilde{f}$ is analytic for all coordinate charts.

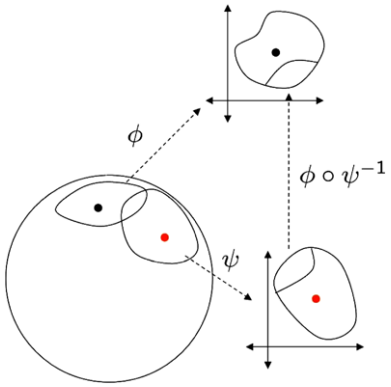


Fig. 1 Example of a two-dimensional manifold. Two overlapping coordinate charts are shown. If the manifold is analytic, the transition map $\phi \circ \psi^{-1}$ (and $\psi \circ \phi^{-1}$) from $\mathbb{R}^2$ to $\mathbb{R}^2$ should be analytic

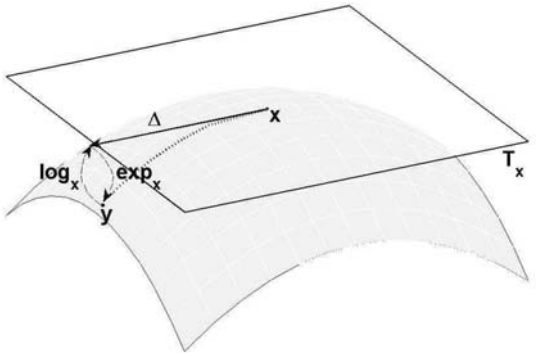


Fig. 2 Example of a two-dimensional manifold and the tangent space at the point **x**

The tangent space can be thought of as the set of allowed velocities for a point constrained to move on the manifold. Formally, the tangent space is a generalization of directional derivatives. A tangent of $\mathcal{M}$ at $\mathbf{x}$ is a real-valued operator on continuous functions satisfying

$$\Delta(af + bh) = a\Delta(f) + b\Delta(h), \tag{1}$$

$$\Delta(fh) = f\Delta(h) + h\Delta(f) \tag{2}$$

for all continuous functions f, h and $a, b \in \mathbb{R}$. The real number assigned to f can be thought of as the derivative of f in the direction represented by Δ. Tangent vectors are real-valued operators acting on continuous functions, although for intuitive purposes we can think of them as velocities of points constrained to move on the manifold. Properties (1) and (2) ensure that the mapping is linear and satisfies the Leibniz product rule of derivatives. The set of all tangents at $\mathbf{x}$ is denoted by $T_{\mathbf{x}}(\mathcal{M})$. It can be shown that for m-dimensional manifolds, the tangent space is a m-dimensional vector space (O'Neill 1983, Chap. 1). An example of a two-dimensional manifold embedded in $\mathbb{R}^3$ with the tangent space $T_{\mathbf{x}}(\mathcal{M})$, is shown in Fig. 2. The solid arrow Δ, is a tangent at $\mathbf{x}$.

When using the coordinate chart $(\mathcal{U}, \phi)$, a convenient basis for $T_{\mathbf{x}}(\mathcal{M})$ is the set of tangents $\partial_i, i = 1, \ldots, m$, where

$$\partial_i(f) = \left.\frac{\partial(f \circ \phi^{-1})}{\partial u^i}\right|_{\phi(\mathbf{x})}, \tag{3}$$

where, u^i is the i-th coordinate of the point $\phi(\mathbf{x})$ in $\mathbb{R}^m$. That is, ∂_i maps a function f to the directional derivative of $\tilde{f} = f \circ \phi^{-1}$ along the i-th coordinate, computed at $\phi(\mathbf{x}) \in \mathbb{R}^m$. The use of superscripts and subscripts for indexing components follows the standard rules based on whether they are covariant or contravariant tensors. However, we do not discuss this matter any further here.

A *Riemannian metric* on the manifold is defined by a set of inner products on the tangent spaces $T_{\mathbf{x}}(\mathcal{M})$ for each $\mathbf{x} \in \mathcal{M}$ (O'Neill 1983, Chap. 2). We denote this inner product by g and for two tangents $\Delta, \Gamma \in T_{\mathbf{x}}(\mathcal{M})$, the inner product is written as $g_{\mathbf{x}}(\Delta, \Gamma)$. The inner product induces a metric with the length of Δ given by $\sqrt{g_{\mathbf{x}}(\Delta, \Delta)}$. For continuity, the products vary smoothly with $\mathbf{x}$. For any $\mathbf{x} \in \mathcal{M}$, given a coordinate map ϕ, we can choose a basis for $T_{\mathbf{x}}(\mathcal{M})$. All tangents can now be represented as vectors in this basis and, relative to this basis, the inner product $g_{\mathbf{x}}$ can be written as a symmetric positive definite matrix. Note, the Riemannian metric is an inherently geometric notion. It does *not* require the definition of a coordinate chart or a basis for $T_{\mathbf{x}}(\mathcal{M})$. Different coordinate charts lead to different coordinates for tangents and the Riemannian metric matrix, but for a given pair of tangents the inner product is independent of the basis.

A *Riemannian manifold* is an ordered pair $(\mathcal{M}, g)$ consisting of the manifold and an associated metric. It is possible to define different metrics on the same manifold to obtain different Riemannian manifolds. However, in practice there exists a standard metric and the Riemannian manifold is denoted by the underlying manifold $\mathcal{M}$. This metric is chosen to have geometrical significance such as being invariant to a set of geometric transformations (Pennec and Ayache 1998).

A *curve* is a continuous mapping α from an open interval $\mathbf{S}$ of $\mathbb{R}$ to $\mathcal{M}$. For a particular $t \in \mathbf{S}$, $\alpha(t)$ lies on the manifold and $\alpha'(t)$ is the tangent at $\alpha(t)$ which maps a function f to $\partial(f \circ \alpha)/\partial t$. Physically $\alpha'(t)$ is the tangent which points along the curve $\alpha(t)$. The norm of $\alpha'(t)$ gives the velocity of the curve. The length of the curve is given by

$$\int_{t \in \mathbf{S}} \sqrt{g_{\alpha(t)}(\alpha'(t), \alpha'(t))}dt \tag{4}$$

where, $\alpha(t)$ is a point on the curve. The shortest path connecting two points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ is the *geodesic* between $\mathbf{x}$ and $\mathbf{y}$. The length of the geodesic is defined to be the *Riemannian distance* between the two points. Geodesics have the property that $g_{\alpha(t)}(\alpha'(t), \alpha'(t))$ is constant for all $t \in \mathbf{S}$, *i.e.*, the velocity is constant along the geodesic (O'Neill

1983, Chap. 3). This property of having zero acceleration is sometimes used to define a geodesic.

2.1 Exponential and Logarithm Operators

For Riemannian manifolds, tangents (on the tangent space) and geodesics (on the manifold) are closely related. For each tangent $\Delta \in T_{\mathbf{X}}(\mathcal{M})$, there is a unique geodesic $\alpha : [0, 1] \to \mathcal{M}$ starting at $\mathbf{x}$ with initial velocity $\alpha'(0) = \Delta$. The *exponential map*, $\exp_{\mathbf{X}}$, maps Δ to the point on the manifold reached by this geodesic

$$\exp_{\mathbf{X}}(\Delta) = \alpha(1). \tag{5}$$

The origin of the tangent space is mapped to the point itself, $\exp_{\mathbf{X}}(0) = \mathbf{x}$. For each point $\mathbf{x} \in \mathcal{M}$, there exists a neighborhood $\tilde{\mathcal{U}}$ of the origin in $T_{\mathbf{X}}(\mathcal{M})$, such that $\exp_{\mathbf{X}}$ is a diffeomorphism from $\tilde{\mathcal{U}}$ onto a neighborhood $\mathcal{U}$ of $\mathbf{x}$ (O'Neill 1983, Chap. 3). Over this neighborhood $\mathcal{U}$, we can define the inverse of the exponential and this mapping from $\mathcal{U}$ to $\tilde{\mathcal{U}}$ is known as the *logarithm map*, $\log_{\mathbf{X}} = \exp_{\mathbf{X}}^{-1}$. Note that the exponential and logarithm operators vary as the point $\mathbf{x}$ moves. This is made explicit by the subscript in the exponential and logarithm operators. The above concepts are illustrated in Fig. 2, where $\mathbf{x}$, $\mathbf{y}$ are points on the manifold and $\Delta \in T_{\mathbf{X}}(\mathcal{M})$. The dotted line shows the geodesic starting at $\mathbf{x}$ and ending at $\mathbf{y}$. This geodesic has an initial velocity Δ and we have $\exp_{\mathbf{X}}(\Delta) = \mathbf{y}$ and $\log_{\mathbf{X}}(\mathbf{y}) = \Delta$. The specific forms of these operators depend on the manifold. We present explicit formulae for certain manifolds in later sections.

The neighborhood $\tilde{\mathcal{U}}$ defined above is not necessarily convex. However, $\tilde{\mathcal{U}}$ is *star-shaped*, *i.e.*, that for any point lying in $\tilde{\mathcal{U}}$, the line joining the point to the origin is contained in $\tilde{\mathcal{U}}$ (O'Neill 1983, Chap. 3). The image of a star-shaped neighborhood under the exponential mapping is a neighborhood of $\mathbf{x}$ on the manifold. This neighborhood is known as a *normal neighborhood*.

The radius of the largest open ball in $T_{\mathbf{X}}(\mathcal{M})$, centered at the origin over which $\exp_{\mathbf{X}}$ is invertible, is known as the *injectivity radius* at $\mathbf{x}$ and denoted by $i(\mathbf{x}, \mathcal{M})$. The injectivity radius of the manifold, $i(\mathcal{M})$, is the minimum of the injectivity radii at all points on the manifold $\mathcal{M}$

$$i(\mathcal{M}) = \min_{\mathbf{x} \in \mathcal{M}} i(\mathbf{x}, \mathcal{M}). \tag{6}$$

For any open ball centered at the origin in $T_{\mathbf{X}}(\mathcal{M})$ with a radius less than $i(\mathcal{M})$, the exponential map is one-to-one and its inverse is given by the logarithm.

2.2 Normal Coordinates

The exponential map can be used to define convenient coordinates for normal neighborhoods, which simplify computation. Let $\tilde{\mathcal{U}}$ be a star shaped neighborhood at the origin in $T_{\mathbf{X}}(\mathcal{M})$ and let $\mathcal{U}$ be its image under the exponential map, *i.e.*, $\mathcal{U}$ is a normal neighborhood of $\mathbf{x}$. Let, $\mathbf{e}_i, i = 1, \ldots, m$ be any orthonormal coordinate system for $T_{\mathbf{X}}(\mathcal{M})$. Therefore,

$$g(\mathbf{e}_i, \mathbf{e}_j) = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases} \tag{7}$$

The *normal coordinate system* of $\mathbf{x}$ is the coordinate chart $(\mathcal{U}, \phi)$ which maps $\mathbf{y} \in \mathcal{U}$ to the coordinates of $\log_{\mathbf{X}}(\mathbf{y})$ in the orthonormal coordinate system,

$$\log_{\mathbf{X}}(\mathbf{y}) = \sum_{i=1}^{m} \phi^i(\mathbf{y}) \mathbf{e}_i \tag{8}$$

where, $\phi^i(\mathbf{y})$ is the i-th coordinate of $\phi(\mathbf{y}) \in \mathbb{R}^m$ (O'Neill 1983, Chap. 3). If we use (3) to define a define a basis of $T_{\mathbf{X}}(\mathcal{M})$ based on the normal coordinate mapping, we get $\partial_i = \mathbf{e}_i$.

2.3 Differential Operators on Manifolds

For a smooth, real valued function $f : \mathcal{M} \to \mathbb{R}$, the *gradient* of f at $\mathbf{x}$, $\nabla f \in T_{\mathbf{X}}(\mathcal{M})$, is the *unique* tangent vector satisfying

$$g_{\mathbf{X}}(\nabla f, \Delta) = \partial_\Delta f \tag{9}$$

for any $\Delta \in T_{\mathbf{X}}(\mathcal{M})$, where ∂_Δ is the directional derivative along Δ. The gradient is the unique tangent such that the directional derivative along any other tangent Δ is equal to the inner product of Δ with the gradient. It is possible to generalize higher order operators such as the Hessian and Laplacian for functions on manifolds (O'Neill 1983, Chap. 3). Using these operators the usual function optimization techniques such as gradient ascent, Newton iterations and conjugate gradient can be generalized to manifolds (Edelman et al. 1998; Smith 1994). However, the computation of these operators is more involved than the computation of the gradient (Ferreira and Xavier 2006).

As we show later, the function whose gradient we require is the Riemann squared distance. For two points, $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ let $d(\mathbf{x}, \mathbf{y})$ be the Riemannian distance between them. Consider the function $f(\mathbf{x}) = d^2(\mathbf{x}, \mathbf{y})$ as a function of $\mathbf{x}$ measuring the squared distance from $\mathbf{y}$. This is a real function on the manifold and we have the following theorem.

Theorem 1 *The gradient of the Riemann squared distance is given by*

$$\nabla f(\mathbf{x}) = \nabla_{\mathbf{X}} d^2(\mathbf{x}, \mathbf{y}) = -2 \log_{\mathbf{X}}(\mathbf{y}). \tag{10}$$

This property is well known, for example (Begelfor and Werman 2006; Ferreira and Xavier 2006). We present the proof in Appendix.

3 Types of Riemannian Manifolds

We briefly discuss the geometry of a few classes of Riemannian manifolds. Most frequently occurring manifolds in computer vision lie in one of these classes.

3.1 Lie Groups

A *Lie group* is a manifold which is also a group such that the group operation is an analytic mapping. The group operation gives Lie groups more algebraic structure than a manifold.

In practice, most frequently occurring Lie groups are sets of matrices, *i.e.*, each element in the group is a matrix and the group operation is matrix multiplication. Such groups are called *matrix Lie groups* (Rossmann 2003). An alternative definition of matrix Lie groups is that they are closed subgroups of the general linear group $\mathbf{GL}(n,\mathbb{R})$, the group of $n \times n$ nonsingular matrices. Common examples of matrix Lie groups include

- *Special Orthogonal Group*. The special orthogonal groups, $\mathbf{SO}(n)$, is the set of rotations in $\mathbb{R}^n$. Elements of $\mathbf{SO}(n)$ are $n \times n$ orthogonal matrices.
- *Special Euclidean Group*. The special Euclidean group, $\mathbf{SE}(n)$, is the set of rigid transformations in $\mathbb{R}^n$. Matrices in $\mathbf{SE}(n)$ are $(n+1) \times (n+1)$ matrices of the form

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{11}$$

  where $\mathbf{R} \in \mathbb{R}^{n \times n}$ is orthogonal and $\mathbf{t} \in \mathbb{R}^n$.
- *Affine Group*. The affine groups $\mathbf{A}(n)$ consists of $(n+1) \times (n+1)$ matrices of the form

$$\begin{bmatrix} \mathbf{H} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{12}$$

  where $\mathbf{H} \in \mathbb{R}^{n \times n}$ is invertible and $\mathbf{t} \in \mathbb{R}^n$.

The tangent space at the identity of the group is known as the *Lie algebra* of the Lie group. Lie algebras are important since the tangent space at any point on the manifold can be expressed in terms of the Lie algebra (O'Neill 1983; Rossmann 2003).

3.2 Homogeneous Spaces

The *group action* of a group G on a manifold $\mathcal{M}$ is a smooth mapping from $G \times \mathcal{M}$ to $\mathcal{M}$. For $g \in G$ and $\mathbf{x} \in \mathcal{M}$, the mapping is written as $(g, \mathbf{x}) \to g \cdot \mathbf{x}$ and satisfies $g \cdot (h \cdot \mathbf{x}) = (gh) \cdot \mathbf{x}$ and $\mathbf{e} \cdot \mathbf{x} = \mathbf{x}$, where $\mathbf{e}$ is the identity of the group G. Just as matrices are transformations of vector spaces, group actions are transformations of manifolds.

The *orbit* of a point $\mathbf{x} \in \mathcal{M}$ is

$$O(\mathbf{x}) = \{g \cdot \mathbf{x} | g \in G\}. \tag{13}$$

The group action divides the manifold into a set of disjoint orbits. For example, consider the Euclidean plane $\mathbb{R}^2$ to be the manifold under the action of the rotation group $\mathbf{SO}(2)$. The orbits consist of circles centered at the origin.

If the whole manifold forms a single orbit, then $\mathcal{M}$ is said to be a *homogeneous space* and the action is said to be *transitive*. The action is transitive, if for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$, there exists $g \in G$ such that $g \cdot \mathbf{x} = \mathbf{y}$. The Euclidean plane $\mathbb{R}^2$ under the action of the Euclidean motion group $\mathbf{SE}(2)$ would be a homogeneous space since for any two points on the plane, there exists some transformation mapping one to the other.

The *isotropy subgroup* of $\mathbf{x}$ is defined as

$$G_{\mathbf{x}} = \{g \in G | g \cdot \mathbf{x} = \mathbf{x}\}. \tag{14}$$

The isotropy subgroup is the set of elements which leave $\mathbf{x}$ fixed. In our previous example of $\mathbf{SE}(2)$ acting on $\mathbb{R}^2$, the isotropy subgroup of the origin would be the set of rotations. The isotropy subgroup around any other point on the Euclidean plane would be the set of rotations about that point.

Let H be a subgroup of G. The *left coset* of $g \in G$ is

$$gH = \{gh | h \in H\}. \tag{15}$$

The set of all left cosets of H in G is denoted by G/H. If G is a Lie group, then G/H forms a manifold known as a *coset manifold*. The coset manifold is a homogeneous space of G with the natural group action

$$g \cdot kH = (gk)H \tag{16}$$

i.e., g acting on the coset of k gives the coset of gk.

The above result lets us represent a coset manifold as a homogeneous space. For a manifold with a transitive Lie group action, it is possible to reverse the process and think of it as a coset manifold. Let $\mathcal{M}$ be a homogeneous space under the Lie group action of G. For any arbitrary point $\mathbf{x} \in \mathcal{M}$, the manifold $\mathcal{M}$ can be identified with the homogenous space $G/G_{\mathbf{x}}$ (O'Neill 1983, Chap. 9).

The reason for using this characterization of coset manifolds is that they inherit a lot of their operators from the Lie group. Once the geometry of the Lie group is understood, the geometry of its coset manifolds can be expressed in terms of the geometry of the Lie group.

3.3 Grassmann Manifolds

A point on the *Grassmann manifold*, $\mathbf{G}_{n,k}$, represents a k-dimensional subspace of $\mathbb{R}^n$. In practice an element of $\mathbf{G}_{n,k}$ is represented by an orthonormal basis as a $n \times k$ matrix, i.e., $\mathbf{x}^T\mathbf{x} = \mathbf{e}_{k \times k}$. Since many basis span the same subspace, this representation of points on $\mathbf{G}_{n,k}$ is *not* unique (Edelman et al. 1998).

Consider any element $\mathbf{U}$ of the group $\mathbf{SO}(n)$. Its columns form an orthonormal basis of the space $\mathbb{R}^n$. A k-dimensional subspace can be obtained by taking the span of the first k-columns of $\mathbf{U}$ and this is an element of $\mathbf{G}_{n,k}$. However, rotations of the form

$$\begin{bmatrix} \mathbf{U}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_{n-k} \end{bmatrix} \tag{17}$$

where, $\mathbf{U}_k \in \mathbf{SO}(k)$ and $\mathbf{U}_{n-k} \in \mathbf{SO}(n-k)$, leave the subspace spanned by the first k-columns unchanged. Multiplication by elements of this form do not change the point in $\mathbf{G}_{n,k}$ and the set of all such rotations is equivalent to $\mathbf{SO}(k) \times \mathbf{SO}(n-k)$. Therefore the Grassmann manifold $\mathbf{G}_{n,k}$ can be identified with the coset manifold $\mathbf{SO}(n)/(\mathbf{SO}(k) \times \mathbf{SO}(n-k))$.

3.4 Essential Manifold

An essential matrix encodes the epipolar geometry for a set of calibrated cameras. Let p and q be the normalized coordinates of corresponding points and $\mathbf{Q}$ be the essential matrix. The essential constraint is

$$p^T \mathbf{Q} q = 0 \tag{18}$$

and the 3×3 matrix $\mathbf{Q}$ is of rank-2 with two equal, positive singular values and a zero singular value. Let $\mathcal{E}$ denote the *essential space*, the set of all essential matrices. The essential space is an algebraic variety (Maybank 1992; Soatto et al. 1994) and a manifold of dimension six. If $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition of a 3×3 matrix $\mathbf{Q}$, then (Hartley and Zisserman 2000, Sect. 8.6.1)

$$\mathbf{Q} \in \mathcal{E} \quad \Longleftrightarrow \quad \mathbf{\Sigma} = \operatorname{diag}\{\lambda, \lambda, 0\}, \quad \lambda \in \mathbb{R}^+. \tag{19}$$

The essential matrix is a homogeneous quantity and scaling does not change the geometry. We use the scaling $\lambda = 1$ and define the *normalized essential space*, $\mathcal{E}_1$ as the set of 3×3 matrices with two unit singular values and one zero singular value

$$\mathbf{Q} \in \mathcal{E}_1 \quad \Longleftrightarrow \quad \mathbf{\Sigma} = \mathbf{\Sigma}_1 \tag{20}$$

where, $\mathbf{\Sigma}_1 = \operatorname{diag}\{1, 1, 0\}$.

Since the epipolar geometry depends on the relative pose of the two cameras, this can be recovered from the essential matrix except for two ambiguities. Firstly, since the epipolar geometry is a purely image based concept, there is no scale information and the baseline between the cameras can only be recovered upto a scale factor. Secondly, four different relative camera geometries give rise to the same essential matrix (Hartley and Zisserman 2000, p. 241). This is shown in Fig. 3. Given an essential matrix the camera geometry can only be recovered upto a four-fold ambiguity. Usually, further image information is required to disambiguate the four geometries and choose the true geometry based on the *positive depth constraint*.

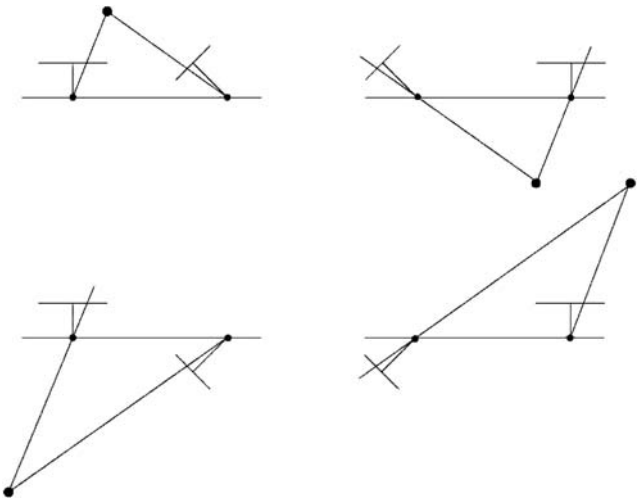


Fig. 3 Four different camera geometries which give the same essential matrix. In each row the geometries differ by changing the sign of the direction of translation. Each column is a twisted pair. The image was taken from Hartley and Zisserman (2000, p. 241)

A common parametrization of the essential manifold is based on the fact that each relative camera geometry corresponds to a tangent of $\mathbf{SO}(3)$ with unit norm. The set of all tangents of a manifold forms a manifold known as the *tangent bundle*. Therefore, the essential manifold can be identified with the unit tangent bundle of $\mathbf{SO}(3)$ (Ma et al. 2001; Soatto et al. 1994). Since each essential matrix corresponds to four different camera geometries, and each camera geometry corresponds to a different tangent of $\mathbf{SO}(3)$, this parametrization gives a four-fold covering of the essential manifold.

In Helmke et al. (2007) a geometric optimization method over essential matrices was proposed using an alternate parametrization. This parametrization was also used in Geyer et al. (2004) for harmonic analysis over the essential manifold. The parametrization is based on a singular value decomposition of the essential matrix. We further develop this idea here and show that this gives a one-to-one correspondence between the points on the manifold and essential matrices. Furthermore, this parametrization makes the essential manifold a homogeneous space under the action of the group $\mathbf{SO}(3) \times \mathbf{SO}(3)$. We will use this to obtain geometrically meaningful Riemannian metrics for the essential manifold.

Consider $\mathbf{Q} \in \mathcal{E}_1$ with the singular value decomposition $\mathbf{U}\mathbf{\Sigma}_1\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and $\det(\mathbf{U})$, $\det(\mathbf{V}) = \pm 1$. As the third singular value is zero, we can change the sign of the third columns of $\mathbf{U}$ and $\mathbf{V}$ to ensure $\det(\mathbf{U}), \det(\mathbf{V}) = 1$ without changing the SVD.

Since $\mathbf{SO}(3)$ is a Lie group, the manifold $\mathbf{SO}(3) \times \mathbf{SO}(3)$ is also a Lie group with the topology and group operation in-

herited from $\mathbf{SO}(3)$ (Rossmann 2003, Sect. 4.3). We define the mapping

$$\Phi : \mathbf{SO}(3) \times \mathbf{SO}(3) \to \mathcal{E}_1 \tag{21}$$

which maps $(\mathbf{U}, \mathbf{V}) \in \mathbf{SO}(3) \times \mathbf{SO}(3)$ to $\mathbf{U}\mathbf{\Sigma}_1\mathbf{V}^T \in \mathcal{E}_1$. The inverse mapping from $\mathcal{E}_1$ to $\mathbf{SO}(3) \times \mathbf{SO}(3)$ is not well defined as there is one degree of freedom in choosing the basis of the space spanned by the first two columns of $\mathbf{U}$ and $\mathbf{V}$. A rotation of the first two columns of $\mathbf{U}$ can be offset by a rotation of the first two columns of $\mathbf{V}$, such that $\mathbf{U}\mathbf{\Sigma}_1\mathbf{V}^T$ does not change. Consider the rotations $\mathbf{R}_1, \mathbf{R}_2$ such that

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \det(\mathbf{A}) \end{bmatrix}, \qquad \mathbf{R}_2 = \begin{bmatrix} \pm\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \det(\mathbf{A}) \end{bmatrix},$$

and $\mathbf{A}\mathbf{A}^T = \pm\mathbf{e}_2$. Then,

$$\mathbf{U}\mathbf{R}_1\mathbf{\Sigma}_1\mathbf{R}_2^T\mathbf{V}^T = \mathbf{U}\begin{bmatrix} \pm\mathbf{A}\mathbf{A}^T & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}\mathbf{V}^T = \pm\mathbf{U}\mathbf{\Sigma}_1\mathbf{V}^T \tag{22}$$

which leaves the essential matrix unchanged and Φ maps $(\mathbf{U}\mathbf{R}_1, \mathbf{V}\mathbf{R}_2) \in \mathbf{SO}(3) \times \mathbf{SO}(3)$ to the same point in $\mathcal{E}_1$.

Let $\mathbf{H}_\Phi$ be the group of transformations which leaves Φ invariant. It consists of elements which leave the third columns of $\mathbf{U}$ and $\mathbf{V}$ unchanged, and rotate the first two columns by angles which differ by $k\pi, k \in \mathbb{Z}$

$$\mathbf{H}_\Phi = \{(\mathbf{R}_1, \mathbf{R}_2) | \mathbf{R}_1, \mathbf{R}_2 \in \mathbf{S}_z, \mathbf{R}_1^T\mathbf{R}_2 = \mathbf{R}_z(k\pi)\} \tag{23}$$

where, $\mathbf{S}_z$ is the set of rotations around the z-axis and $\mathbf{R}_z(k\pi)$ denotes a rotation by $k\pi$ around the z-axis.

The manifold $\mathcal{E}_1$ is identified with the coset manifold $\mathbf{SO}(3) \times \mathbf{SO}(3)/\mathbf{H}_\Phi$, *i.e.*, elements of $\mathbf{SO}(3) \times \mathbf{SO}(3)$ which differ by group multiplication by an element in $\mathbf{H}_\Phi$ are considered to be the same on $\mathbf{SO}(3) \times \mathbf{SO}(3)/\mathbf{H}_\Phi$. Multiplication of $(\mathbf{U}, \mathbf{V})$ by elements of $\mathbf{H}_\Phi$ generates the equivalence class of $(\mathbf{U}, \mathbf{V})$, and all the elements in an equivalence class represent the same essential matrix.

3.5 Symmetric Positive Definite (SPD) Matrices

The set of $n \times n$ symmetric positive definite matrices forms a manifold known as the *symmetric manifold*, Sym_n^+. Recently, there has been a considerable amount of research aimed at understanding the geometry of this manifold due to the development of *diffusion tensor MRI* (DT-MRI), a widely used medical imaging method which measures the diffusivity of the water molecules in three dimensional space (Basser et al. 1994; Lenglet et al. 2004; Tschumperle and Deriche 2003; Vemuri et al. 2001). The diffusivity is encoded as a 3×3 SPD matrix and the image is a 3D grid of 3×3 SPD matrices. The filtering of these images is an important step in their processing and requires an understanding of the noise model and the geometry of the manifold, Sym_3^+.

A metric for this manifold was first proposed in Forstner and Moonen (1999). Later, in Pennec et al. (2006) it was shown that this metric is invariant to affine transformations and that Sym_n^+ is a Riemannian manifold. A different Riemannian metric for this manifold was proposed in Arsigny et al. (2006), and this is the metric we will use later.

4 The Original Mean Shift

The mean shift algorithm is based on the theory of kernel density estimation. Here we briefly describe the derivation of mean shift as in Comaniciu and Meer (2002).

Let $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, n$ be n independent, identically distributed points generated by an unknown probability distribution f. The *kernel density estimate*

$$\hat{f}_k(\mathbf{y}) = \frac{c_{k,h}}{n}\sum_{i=1}^{n} k\left(\frac{\|\mathbf{y} - \mathbf{x}_i\|^2}{h^2}\right) \tag{24}$$

based on a *profile function* k satisfying $k(z) \geq 0$ for $z \geq 0$, is a nonparametric estimator of the density $f(\mathbf{y})$ at $\mathbf{y}$. The constant $c_{k,h}$ is chosen to ensure that $\hat{f}_k$ integrates to one.

Define $g(\cdot) = -k'(\cdot)$. Taking the gradient of (24) we get

$$\mathbf{m}_h(\mathbf{y}) = C\frac{\nabla\hat{f}_k(\mathbf{y})}{\hat{f}_g(\mathbf{y})} = \frac{\sum_{i=1}^n \mathbf{x}_i\, g\left(\|\mathbf{y} - \mathbf{x}_i\|^2/h^2\right)}{\sum_{i=1}^n g\left(\|\mathbf{y} - \mathbf{x}_i\|^2/h^2\right)} - \mathbf{y} \tag{25}$$

where, C is a positive constant and $\mathbf{m}_h(\mathbf{x})$ is the *mean shift* vector. The expression (25) shows that the mean shift vector is proportional to a normalized density gradient estimate. The iteration

$$\mathbf{y}_{j+1} = \mathbf{m}_h(\mathbf{y}_j) + \mathbf{y}_j \tag{26}$$

is a gradient ascent technique converging to a stationary point of the density. Saddle points can be detected and removed, to obtain only the modes.

4.1 Mean Shift as Bounds Optimization

In the previous section mean shift was derived as a gradient ascent technique. The advantage of mean shift is that it avoids the computationally intensive line search step which other gradient ascent techniques require. The magnitude of the mean shift step adapts to the surrounding data and it can be shown that the mean shift iterations are guaranteed to converge to a local maxima of the kernel density (Comaniciu and Meer 2002).

An alternative view of mean shift was proposed in Fashing and Tomasi (2005). It was shown that for Epanechnikov kernels, the mean shift vector not only lies along the gradient but is in fact a Newton step. Furthermore, when using a general kernel, the mean shift step optimizes a lower bound

on the kernel density function. This idea of mean shift as a variational bounds optimization was further developed in Singh et al. (2004).

Finally, in Carreira-Perpinan (2007), it was shown that for Gaussian kernels, the mean shift step is the same as Expectation-Maximization. In the M-step of the EM-algorithm a tight lower bound on the function is computed and in the E-step this bound is maximized. For non-Gaussian kernels, mean shift is equivalent to generalized EM.

All these approaches yield the same update rule when the data lies in Euclidean space, but for Riemannian manifolds this is not true. Generalizing each of these different algorithms to manifolds lead to different update rules. However, the reason for the widespread use of mean shift is due to its ease of implementation. It offers a simple iterative update rule with provable convergence behavior. We should take this into consideration when developing a mean shift update rule for manifolds. The computation of Hessians over manifolds can be hard to implement, and should be avoided. We propose a simple gradient based mean shift rule while making sure that the convergence properties of mean shift continue to hold. Specifically we show the following.

- Kernel density estimation over manifolds is more complex than in the Euclidean case. It requires the computation of a point dependent volume density function leading to a complex update rule.
- The weighted average (not the weighted sum) of points on the manifold is well defined. Replacing a point by the weighted mean of the data points in a neighborhood still leads to convergence to the maxima of a cost function.
- For homogeneous spaces the update rule is equivalent to expectation-maximization.

The nonlinear mean shift algorithm for Riemannian manifolds is discussed next.

5 Nonlinear Mean Shift

The reason the previous mean shift algorithm is not directly applicable to manifolds is that manifolds are not vector spaces and the sum of points on the manifold, in general, does not lie on the manifold. Consequently, the mean shift vector of (25) is not valid. However, it is possible to define the weighted mean of points as the minimum of an appropriate cost function (Karcher 1977). In this section, we use this concept to derive the mean shift vector as the weighted sum of *tangent vectors*. Since tangent spaces are vector spaces, a weighted sum of tangents is possible and can be used to update the mode estimate. This method is valid over *any* Riemannian manifold.

5.1 Kernel Density Estimation over Riemannian Manifolds

Consider a Riemannian manifold with a metric d. Given n points on the manifold, $\mathbf{x}_i, i = 1, \ldots, n$, the kernel density estimate with profile k and bandwidth h is

$$\hat{f}_k(\mathbf{y}) = \frac{c_{k,h}}{n} \sum_{i=1}^{n} k\left(\frac{d^2(\mathbf{y}, \mathbf{x}_i)}{h^2}\right). \tag{27}$$

The bandwidth h can be included in the distance function as a scaling. However, we write it in this form since it gives us a parameter to tune performance in applications. If the manifold is an Euclidean space with the Euclidean distance metric, (27) is the same as (24). The constant $c_{k,h}$ is chosen to ensure that $\hat{f}_k$ is a density, *i.e.*, the integral of $\hat{f}_k$ over the manifold is one. Note, our definition of a kernel density is restricted to points lying on the manifold. Since the Riemannian distance on the right hand side is not defined for points not on the manifold, the domain of definition for the equation above is the manifold.

Strictly speaking, $\hat{f}_k$ is not a true kernel density. In Euclidean space the integral of the kernel is independent of the point at which it is centered. For a general Riemannian manifold, the integral of the kernel depends on the point at which it is centered. It is possible to ensure the integral of the kernel is the same irrespective of where it is centered by using the *volume density function* (Pelletier 2005). We do not do this, since the computation of the volume density function would limit the applicability of the algorithm to manifolds where explicit expressions for the volume densities are available. Also, the computation of the gradient would become very complicated. We prefer to use a modified kernel density function of (27) which is similar to the one proposed in Chikuse (2003, Chap. 10) for Grassmann manifolds. We use the same expression over all Riemannian manifolds.

5.2 Mean Shift over Riemannian Manifolds

Calculating the gradient of $\hat{f}_k$ at $\mathbf{y}$, we get

$$\begin{aligned}\nabla \hat{f}_k(\mathbf{y}) &= \frac{1}{n} \sum_{i=1}^{n} \nabla k\left(\frac{d^2(\mathbf{y}, \mathbf{x}_i)}{h^2}\right) \\ &= -\frac{1}{n} \sum_{i=1}^{n} g\left(\frac{d^2(\mathbf{y}, \mathbf{x}_i)}{h^2}\right) \frac{\nabla d^2(\mathbf{y}, \mathbf{x}_i)}{h^2} \\ &= \frac{2}{n} \sum_{i=1}^{n} g\left(\frac{d^2(\mathbf{y}, \mathbf{x}_i)}{h^2}\right) \frac{\log_{\mathbf{y}}(\mathbf{x}_i)}{h^2}\end{aligned} \tag{28}$$

where, $g(\cdot) = -k'(\cdot)$, and in the final step we use (10). The gradient of the distance is taken with respect to $\mathbf{y}$. Analogous

MEAN SHIFT OVER RIEMANNIAN MANIFOLDS

Given: Points on a manifold $\mathbf{x}_i, i = 1, \ldots, n$

for $i \leftarrow 1 \ldots n$

$\mathbf{y} \leftarrow \mathbf{x}_i$

repeat

$$\mathbf{m}_h(\mathbf{y}) \leftarrow \frac{\sum_{i=1}^{n} g\left(d^2(\mathbf{y}, \mathbf{x}_i)/h^2\right) log_{\mathbf{y}}(\mathbf{x}_i)}{\sum_{i=1}^{n} g\left(d^2(\mathbf{y}, \mathbf{x}_i)/h^2\right)}$$

$\mathbf{y} \leftarrow exp_{\mathbf{y}}\left(\mathbf{m}_h(\mathbf{y})\right)$

until $\|\mathbf{m}_h(\mathbf{y})\| < \epsilon$

Retain $\mathbf{y}$ as a local mode

Report distinct local modes.

Fig. 4 Mean shift algorithm for Riemannian manifolds

to (25), define the nonlinear mean shift vector as

$$\mathbf{m}_h(\mathbf{y}) = \frac{\sum_{i=1}^{n} g(\frac{d^2(\mathbf{y},\mathbf{x}_i)}{h^2}) \log_{\mathbf{y}}(\mathbf{x}_i)}{\sum_{i=1}^{n} g(\frac{d^2(\mathbf{y},\mathbf{x}_i)}{h^2})}. \tag{29}$$

All the operations in the above equation are well defined. The $\log_{\mathbf{y}}(\mathbf{x}_i)$ terms lie in the tangent space $T_{\mathbf{y}}(\mathcal{M})$ and the kernel terms $g(d^2(\mathbf{y}, \mathbf{x}_i)/h^2)$ are scalars. The mean shift vector is a weighted sum of tangent vectors, and is itself a tangent vector in $T_{\mathbf{y}}(\mathcal{M})$. The noneuclidean mean shift iteration is

$$\mathbf{y}_{j+1} = \exp_{\mathbf{y}_j}\left(\mathbf{m}_h(\mathbf{y}_j)\right). \tag{30}$$

The iteration (30) updates $\mathbf{y}_j$ by moving along the geodesic defined by the mean shift vector to get the next estimate, $\mathbf{y}_{j+1}$. A mean shift iteration is started at each data point by initializing $\mathbf{x} = \mathbf{x}_i$. The inner loop then iteratively updates $\mathbf{x}$ till convergence. The complete algorithm is shown in Fig. 4.

Strong modes have high kernel density scores and a large number of iterations converge to these locations. Spurious modes having low densities and few iterations converging to them can be pruned to obtain only the strong modes. So, mean shift returns the number of modes and their locations.

6 Computational Details of Nonlinear Mean Shift

The nonlinear mean shift algorithm is valid for any Riemannian manifold. A practical implementation requires the computation of the exponential operator $\exp_{\mathbf{x}}$ and the logarithm operator $\log_{\mathbf{x}}$. The computational details of this are discussed next.

6.1 Lie Groups

Lie groups are the most well known examples of Riemannian manifolds and the first nonlinear mean shift algorithm was proposed for Lie groups in Tuzel et al. (2005). Let exp and log be the matrix operators

$$\exp(\mathbf{\Delta}) = \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{\Delta}^i, \tag{31}$$

$$\log(\mathbf{y}) = \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i} (\mathbf{y} - \mathbf{e})^i \tag{32}$$

where, $\mathbf{e}$ is the identity matrix. These are standard matrix operators which can be applied to any square matrix and no subscript is necessary to define them. They should not be confused with the manifold operators, $\exp_{\mathbf{x}}$ and $\log_{\mathbf{x}}$ for Lie groups, which are given by

$$\exp_{\mathbf{x}}(\Delta) = \mathbf{x} \exp(\mathbf{x}^{-1}\Delta), \tag{33}$$

$$\log_{\mathbf{x}}(\mathbf{y}) = \mathbf{x} \log(\mathbf{x}^{-1}\mathbf{y}) \tag{34}$$

where, $\mathbf{y}$ is any point on the manifold and $\Delta \in T_{\mathbf{X}}(\mathcal{M})$ (O'Neill 1983, Chap. 11). The distance function is given by

$$d(\mathbf{x}, \mathbf{y}) = \| \log(\mathbf{x}^{-1}\mathbf{y}) \|_F \tag{35}$$

where, $\|.\|_F$ denotes the Frobenius norm of a matrix. This definition of d can be shown to be the distance corresponding to an inner product on $T_{\mathbf{X}}(\mathcal{M})$, as necessary for Riemannian manifolds (O'Neill 1983, Chap. 11). Substituting the expressions of (33) and (34) into (29), the mean shift vector for matrix Lie groups can be obtained.

The major bottleneck for the nonlinear mean shift algorithm is the computation of exp and log. There exist iterative techniques for a computing the exp or log of general square matrices (Golub and Van Loan 1989, Sect. 11.3). In practice, considerable improvements can be obtained by taking advantage of the structure of the matrix Lie group under consideration. We show this with the special orthogonal group, $\mathbf{SO}(3)$. Let $\mathfrak{so}(3)$ be the Lie algebra, *i.e.*, the tangent space at the identity. Elements of $\mathfrak{so}(3)$ are 3×3 skew-symmetric matrices of the form

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{36}$$

where the vector $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]$ is the axis of rotation and $\|\boldsymbol{\omega}\|$ is the magnitude of the rotation. The structure of $\mathbf{SO}(3)$ allows us to compute exp using the *Rodriguez* formula (Kanatani 1990, p. 204)

$$\exp([\boldsymbol{\omega}]_\times) = \mathbf{e} + \frac{\sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|} [\boldsymbol{\omega}]_\times + \frac{1 - \cos\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2} [\boldsymbol{\omega}]_\times^2, \tag{37}$$

where, $\mathbf{e}$ is the identity matrix. The matrix logarithm can be computed by inverting the above equation. Using (37) rather than iterative techniques to compute the matrix exponential can speed up an implementation by an order of 20.

The mean shift update proposed here is the same as the one proposed in Subbarao and Meer (2006), Tuzel et al. (2005). However, in Subbarao and Meer (2006), Tuzel et al. (2005) it was assumed that the relation between the Riemann distance and the log operator was an approximation. As we show later, this relation is exact, and the algorithm is provably convergent to a local maxima of the kernel density (27).

6.2 Grassmann Manifold

The *Grassmann manifold*, $\mathbf{G}_{n,k}$, consists of $n \times k$ orthonormal matrices. The set of tangents at $\mathbf{x}$ are $n \times k$ matrices Δ satisfying $\mathbf{x}^T \Delta = 0$. For a tangent $\Delta \in T_{\mathbf{X}}(\mathcal{M})$ the exponential at $\mathbf{x}$ is

$$\exp_{\mathbf{x}}(\boldsymbol{\Delta}) = \mathbf{x}\mathbf{v}\cos(\mathbf{s})\mathbf{v}^T + \mathbf{u}\sin(\mathbf{s})\mathbf{v}^T \tag{38}$$

where, $\mathbf{u}\mathbf{s}\mathbf{v}^T$ is the compact SVD of Δ and the sin and cos act element-by-element along the diagonal of $\mathbf{s}$ (Edelman et al. 1998).

The log operator is the inverse of the exp operator. Let $\mathbf{x}$ and $\mathbf{y}$ be two points on the manifold $\mathbf{G}_{n,k}$. The logarithm of $\mathbf{y}$ at $\mathbf{x}$ is given by

$$\log_{\mathbf{X}}(\mathbf{y}) = \mathbf{u}\sin^{-1}(\mathbf{s})\mathbf{v}^T \tag{39}$$

where, $\mathbf{u}\mathbf{s}\mathbf{d}^T = \mathbf{y} - \mathbf{x}\mathbf{x}^T\mathbf{y}$ and $\mathbf{v}\mathbf{c}\mathbf{d}^T = \mathbf{x}^T\mathbf{y}$ is the generalized SVD with $\mathbf{c}^T\mathbf{c} + \mathbf{s}^T\mathbf{s} = \mathbf{e}$ and the $\sin^{-1}$ acts element-by-element along the diagonal of $\mathbf{s}$. It can be verified that this tangent does satisfy the two properties, $\mathbf{x}^T \log_{\mathbf{X}}(\mathbf{y}) = 0$ and $\exp_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{y})) = \mathbf{y}$ as required in Edelman et al. (1998).

The distance between two points on the manifold is (Absil et al. 2003; Edelman et al. 1998)

$$d(\mathbf{x}, \mathbf{y}) = \left\| \log_{\mathbf{x}}(\mathbf{y}) \right\|_F. \tag{40}$$

These computational details are different from those proposed in Subbarao and Meer (2006). Over there, an arc-length approximation to the Riemannian distance was used for computational purposes. This approximation was treated as a function of one of the points and the gradient was computed as in Edelman et al. (1998). This holds for points close to each other and in all experiments the algorithm converges quickly. However, theoretically the convergence of the approximation of Subbarao and Meer (2006) is not assured, while the method proposed here is provably convergent.

6.3 Essential Matrices

The manifold $\mathbf{SO}(3) \times \mathbf{SO}(3)$ consists of two copies of $\mathbf{SO}(3)$ and the tangent space of $\mathbf{SO}(3) \times \mathbf{SO}(3)$ will consist of two copies of the tangent space of $\mathbf{SO}(3)$. Since $\mathbf{SO}(3)$ has three-dimensional tangent spaces, $\mathbf{SO}(3) \times \mathbf{SO}(3)$ will have six-dimensional tangent spaces. Consider $(\mathbf{U}, \mathbf{V}) \in \mathbf{SO}(3) \times \mathbf{SO}(3)$ and a tangent represented as a six vector

$$\Delta = \begin{bmatrix} \mathbf{u}^T & \mathbf{v}^T \end{bmatrix}^T \tag{41}$$

where, $\mathbf{u} = [u_x \ u_y \ u_z]^T$ and $\mathbf{v} = [v_x \ v_y \ v_z]^T$. The exponential for $\mathbf{SO}(3) \times \mathbf{SO}(3)$ is computed by performing the exponential of $\mathbf{SO}(3)$ twice, once each for $\mathbf{U}$ and $\mathbf{V}$

$$\exp_{(\mathbf{U},\mathbf{V})}(\Delta) = (\mathbf{U}\exp([\mathbf{u}]_\times), \mathbf{V}\exp([\mathbf{v}]_\times)) \tag{42}$$

where, the exp on the right represents the matrix exponential computed by the Rodriguez formula (37) and $[\cdot]_\times$ is defined by (36). The first three elements of the tangent vector correspond to $\mathbf{U}$ and the last three to $\mathbf{V}$. This ordering is equivalent to choosing a basis for the tangent space.

The tangent space of $\mathbf{SO}(3) \times \mathbf{SO}(3)$ can be divided into two complementary subspaces, the *vertical* and the *horizontal* space. The horizontal space contains tangents of the form

$$[u_x \quad u_y \quad u_z \quad v_x \quad v_y \quad -u_z], \qquad \|u_z\| < \pi/2. \tag{43}$$

The vertical space consists of tangents of the form

$$[0 \quad 0 \quad u_z \quad 0 \quad 0 \quad k\pi + u_z] \tag{44}$$

which lie in the Lie algebra of $\mathbf{H}_\Phi$ (Edelman et al. 1998). When $k = 0$, the vertical and horizontal spaces form complementary subspaces around the origin of the tangent space. Moving along geodesics defined by tangents in the vertical space is equivalent to multiplying by elements of $\mathbf{H}_\Phi$ and leaves the equivalence class unchanged. Vectors in the horizontal space are tangent to the equivalence class and all tangents of $\mathbf{SO}(3) \times \mathbf{SO}(3)/\mathbf{H}_\Phi$ must lie in the horizontal space of $\mathbf{SO}(3) \times \mathbf{SO}(3)$. Given a tangent in the horizontal space, its exponential can be computed like in (42) to get an element in a different equivalence class (which will be a different essential matrix).

Let $(\mathbf{U}, \mathbf{V})$ and $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ represent two elements of $\mathbf{SO}(3) \times \mathbf{SO}(3)/\mathbf{H}_\Phi$. These can be any points in their respective equivalence classes. The logarithm operator for $\mathbf{SO}(3) \times \mathbf{SO}(3)/\mathbf{H}_\Phi$ should give a tangent in the horizontal space. To do this we first compute the logarithm on the manifold $\mathbf{SO}(3) \times \mathbf{SO}(3)$. Define

$$\delta\mathbf{U} = \mathbf{U}^T\hat{\mathbf{U}}, \qquad \delta\mathbf{V} = \mathbf{V}^T\hat{\mathbf{V}}. \tag{45}$$

Taking the matrix logarithms of $\delta\mathbf{U}$ and $\delta\mathbf{V}$, and rearranging the elements into a six-vector, we get

$$[\delta u_x \quad \delta u_y \quad \delta u_z \quad \delta v_x \quad \delta v_y \quad \delta v_z]^T \tag{46}$$

which lies in the tangent space of $\mathbf{SO}(3) \times \mathbf{SO}(3)$. Since $(\mathbf{U}, \mathbf{V})$ and $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ are arbitrary elements of their equivalence classes, it is not necessary that this vector lie in the horizontal space. We need to remove the component lying in the vertical space. Using Givens rotations (Hartley and Zisserman 2000, App. 3) $\delta\mathbf{U}$ and $\delta\mathbf{V}$ are decomposed into rotations around the z-axis and rotations around axes in the xy-plane. Now, $(\mathbf{U}, \mathbf{V})$ is moved using z-rotations differing by $k\pi$, according to (42). On recomputing $\delta\mathbf{U}$ and $\delta\mathbf{V}$, they have opposite z-rotations less than $\pi/2$. This can be done in a single step and ensures that for the new $\delta\mathbf{U}$ and $\delta\mathbf{V}$, $\delta u_z \approx -\delta v_z$ upto a few degrees. Due to the nonlinearity of the manifold $\delta u_z = -\delta v_z$ will not hold exactly. This can be improved by moving $(\mathbf{U}, \mathbf{V})$ along tangents of the form

$$[0 \quad 0 \quad (\delta u_z + \delta v_z)/2 \quad 0 \quad 0 \quad (\delta u_z + \delta v_z)/2]^T \tag{47}$$

and recomputing $\delta\mathbf{U}$ and $\delta\mathbf{V}$. The tangents of (47) lie in the vertical space and do not change the equivalence class of $(\mathbf{U}, \mathbf{V})$. After the initial step with Givens rotations, $\delta u_z + \delta v_z$ is very small. Three or four iterations generally give an acceptable accuracy of the order of 10^{-4}. At convergence we obtain the log, which is a six-vector of the form

$$[\delta u_x \quad \delta u_y \quad \delta u_z \quad \delta v_x \quad \delta v_y \quad -\delta u_z] \tag{48}$$

pointing from one equivalence class to the other. The intrinsic distance between $(\mathbf{U}, \mathbf{V})$ and $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ is given by the norm of the five dimensional vector

$$d((\mathbf{U}, \mathbf{V}), (\hat{\mathbf{U}}, \hat{\mathbf{V}})) = \|\delta u_x \quad \delta u_y \quad \delta u_z \quad \delta v_x \quad \delta v_y\|_2. \tag{49}$$

6.4 Symmetric Positive Definite (SPD) Matrices

We use the framework proposed in Arsigny et al. (2006) to work with the manifold of $n \times n$ symmetric positive definite matrices, Sym_n^+. The idea is that the matrix log operator of (32) is a diffeomorphism (a one-to-one, continuous, differentiable mapping with a continuous, differentiable inverse) over the space Sym_n^+. The range of the log is the space of $n \times n$ symmetric matrices, not necessarily positive definite. All operations are carried out by mapping SPD matrices to symmetric matrices using the log operator. Means and interpolation can be done in the space of symmetric matrices, which is a vector space, and mapped back to Sym_n^+ using the exp operator. It is shown in Arsigny et al. (2006) that this corresponds to giving Sym_n^+ a Lie group structure. However, the group operation is *not* matrix multiplication and Sym_n^+ is not a matrix Lie group. The reason for using this Riemannian framework as opposed to the one proposed in Pennec et al. (2006) is that the numerical results are almost identical but the Lie group structure is more computationally efficient. Mean shift over Sym_n^+ can be carried out using the Riemannian structure of Pennec et al. (2006), and the results of this can be seen in Subbarao and Meer (2007).

The computation of the matrix exponential and logarithm can be simplified in the case of the SPD matrices. Let $\mathbf{usu}^T$ be the singular value decomposition of a symmetric positive definite matrix $\mathbf{x} \in \mathrm{Sym}_n^+$. Then the matrix logarithm is

$$\log(\mathbf{x}) = \mathbf{u} \log(\mathbf{s}) \mathbf{u}^T \tag{50}$$

where, the log acts on element-by-element along the diagonal of $\mathbf{s}$. Since the singular values of a SPD matrix are always positive, the log can act along the diagonal. The matrix exponential is computed similarly. Let Δ be a symmetric matrix with SVD given by $\Delta = \mathbf{udu}^T$. The matrix exponential becomes

$$\exp(\mathbf{\Delta}) = \mathbf{u} \exp(\mathbf{d}) \mathbf{u}^T, \tag{51}$$

where, the exp acts element-by-element along the diagonal of $\mathbf{d}$. It is easy to see that this form of the exponential is defined for any symmetric matrix and the result is a SPD matrix.

Let $\mathbf{x}$ and $\mathbf{y}$ be two $n \times n$ SPD matrices. The manifold logarithm according to Arsigny et al. (2006) is

$$\log_{\mathbf{x}}(\mathbf{y}) = \log(\mathbf{y}) - \log(\mathbf{x}). \tag{52}$$

Given a tangent $\mathbf{\Delta}$, the manifold exponential operator is

$$\exp_{\mathbf{x}}(\mathbf{\Delta}) = \exp(\log(\mathbf{x}) + \Delta). \tag{53}$$

The distance between $\mathbf{x}$ and $\mathbf{y}$ is given by

$$d(\mathbf{x}, \mathbf{y}) = \|\log(\mathbf{y}) - \log(\mathbf{x})\|_2. \tag{54}$$

In our implementation, the log operator is used in a preprocessing step to map all matrices to their matrix logarithms. Mean shift is then carried out in the vector space of symmetric matrices and the modes are mapped back to Sym_n^+ by the exp operator. This is equivalent to performing nonlinear mean shift with the Lie group structure of Sym_n^+ (Arsigny et al. 2006). However the matrix logarithms and exponentials are used only once during the preprocessing and post-processing of data and the time taken for mean shift is much lower.

7 Theoretical Properties of Nonlinear Mean Shift

One of the reasons of for the popularity of the original mean shift is the provable convergence of the mean shift iterations. In Comaniciu and Meer (2002) it was shown that the iterations will converge to a local maxima of a kernel density. This is rather surprising since other gradient ascent methods such as Newton steps or conjugate gradient require a line search along the gradient to make sure that the iteration does not take a big step and miss the maximum. In the mean

shift algorithms there is no line search. The mean shift iteration of (25) adapts itself to the data points surrounding it. In regions with a high density of points, the mean shift steps are small to make sure that it does not move too far and miss the maximum, while in regions with a low density of points the steps are much larger.

The nonlinear mean shift step of (29) ensures similar convergence properties. Let $\mathbf{y}_j$, $j = 1, \ldots$ be the successive estimates of the mode obtained through mean shift iterations given by (29) and (30). The following theorem is proved in Appendix.

Theorem 2 *If the kernel K has a convex and monotonically decreasing profile and the bandwidth h is less than the injectivity radius $i(\mathcal{M})$ of the manifold, the sequence $\{f(\mathbf{y}_j)\}_{j=1,2,\ldots}$ is convergent and monotonically non-decreasing.*

The proof proceeds is a manner similar to the proof for the Euclidean case (Comaniciu and Meer 2002). Given the convexity of the profile we only need to show that each mean shift step minimizes a weighted error measure, which is the same as that minimized by the weighted mean finding algorithm over manifolds. The necessary conditions for this have already been shown in Karcher (1977). In Comaniciu and Meer (2002) it was also shown that the mean shift iterations will converge to a spatially to a single point. This property may not extend to the manifold mean shift algorithm although we have not experienced such a situation in practice.

7.1 Nonlinear Mean Shift and Expectation Maximization

In Carreira-Perpinan (2007) it was shown that the original mean shift over vector spaces can be viewed as a special case of expectation maximization (EM). Due to the complexity of the idea we cannot completely explain it here but briefly outline the conditions under which manifold mean shift is equivalent to EM. The equivalence is shown by considering a slightly different problem. Assume that the kernel density function is given to us with one degree of freedom which allows us to translate the whole distribution. Therefore, the true distribution is of the form

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = f(\mathbf{x} + \boldsymbol{\theta}) \tag{55}$$

where, f is the kernel density and $\boldsymbol{\theta}$ is the translational degree of freedom. We are given a point $\mathbf{z}$ sampled from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ with unknown $\boldsymbol{\theta}$. Assume that the sampled point lies at the origin $\mathbf{z} = \mathbf{0}$ and use EM to find the most likely value of the parameter $\boldsymbol{\theta}$. Intuitively it is easy to see that the maximum likelihood estimate of $\boldsymbol{\theta}$ will be so as to move the point of maximum density to the origin. Therefore, we pick $\boldsymbol{\theta}$ such that $p_{\boldsymbol{\theta}}(\mathbf{0})$ is the highest mode. Since $p_{\boldsymbol{\theta}}(\mathbf{0}) = f(\boldsymbol{\theta})$, our best estimate of $\boldsymbol{\theta}$ will be the mode of f.

To generalize this procedure to arbitrary Riemannian manifolds, we need to generalize the concept of 'translating' the distribution like in (55). This notion is captured by a group action. We now take f to represent kernel density (27) over some manifold $\mathcal{M}$ and assume that a group G acts transitively over $\mathcal{M}$. We define the 'translated' version of f as

$$p_g(\mathbf{x}) = f(g \cdot \mathbf{x}) \tag{56}$$

where $g \in G$. Given that the sampled point lies at a particular $\mathbf{x}_0 \in \mathcal{M}$, EM attempts to find the group element which moves the mode of f to the point $\mathbf{x}_0$. We use an arbitrary point $\mathbf{x}_0$ since there is no special point equivalent to the origin of Euclidean space. Like in Carreira-Perpinan (2007) it is now easy to show that the EM iterations are the same as the mean shift iterations.

8 Applications

We present two applications of the nonlinear mean shift algorithm. The first application is a motion segmentation algorithm which can be used with many different motion models. *It recovers both the number of motions and the motion parameters in a single step.* The second application is a discontinuity preserving filtering algorithm which works with non-vector valued images.

8.1 Motion Segmentation

This motion segmentation algorithm was first proposed in Tuzel et al. (2005) although the algorithm was only valid for parameter spaces which were matrix Lie groups. This algorithm was extended in Subbarao and Meer (2006) to all Riemannian manifolds. However, Subbarao and Meer (2006) makes a few approximations, while applying the nonlinear mean shift algorithm to different manifolds. The mean shift algorithm proposed here is not an approximation and is theoretically valid for any Riemannian manifold.

The input to the algorithm consists of point matches belonging to different motions. Some of these correspondences may also be outliers. The algorithm proceeds in two stages. In the *first stage*, the matches are randomly sampled to generate minimal sets of points which define motion hypotheses. Such minimal sets are known as *elemental subsets*. The sampling and hypothesis generation can be improved by a *validation* step which reduces computation in the second stage (Tuzel et al. 2005).

In the *second stage*, the parameter estimates are clustered using the algorithm proposed in Sect. 5.2. The number of dominant modes gives the number of motions in the data and the position of the modes corresponds to the motion parameters. Modes are considered not to be dominant if they

do not have high kernel densities or do not have a large number of mean shift iterations converging to them.

The inliers for each mode are found based on the residuals. For each motion parameter returned by clustering the motions, the residual errors of the inliers will be close to zero in the Euclidean space of the residuals. This will lead to a mode around the origin in the space of residuals. We use the original mean shift in the Euclidean space of the residual errors to find the basin of attraction of the mode at zero. Points with residuals in this basin of attraction are declared inliers. Since the inliers for each motion are decided independently, it is possible for a correspondence to be assigned to two motions. In such a case the tie is broken by assigning it to the motion which gives a lower error.

The performance of our algorithm is tested by verifying that the number of strong modes is equal to the number of motions present in the data. Since mean shift returns all local maxima of the kernel density estimate, for a data set with m motions the first m modes should clearly dominate the $(m+1)$th mode, so that these extraneous modes can be pruned.

We also quantitatively compare the result of the algorithm with a manual segmentation in two ways. Firstly, the classification of the correspondences is compared based on the number of misclassifications by the algorithm. Secondly, we compute the squared error using only the points declared inliers by the manual segmentation. Let $\mathbf{M}_j$ be the j-th mode returned by the clustering. If the correspondence $\mathbf{p}_i$ is an inlier for this motion, the residual $\mathbf{M}_j(\mathbf{p}_i)$ should be small. We measure the *residual squared error* of $\mathbf{M}_j$ as

$$\epsilon_{res}^{j} = \frac{1}{n_j} \sum_{i=1}^{n_j} \left| \mathbf{M}_j(\mathbf{p}_i) \right|^2 \tag{57}$$

where the sum is taken only over correspondences which are inliers according to the manual segmentation and n_j is the number of inliers for the j-th motion. The lower limit for this error is

$$\epsilon_{LS}^{j} = \frac{1}{n_j} \sum_{i=1}^{n_j} |\hat{\mathbf{M}}_j(\mathbf{p}_i)|^2 \tag{58}$$

where, $\hat{\mathbf{M}}_j$ is the least squares estimate based on the inliers given by the manual segmentation. By definition, the least square estimate minimizes the squared error.

This algorithm can be used with different motion models, with the difference being the manifold over which the hypotheses are clustered. This depends on the application and the assumptions made about the type of motion. We present a examples of using this algorithm, and other examples of motion models and their corresponding manifolds can be found in Subbarao and Meer (2006), Tuzel et al. (2005). In all our experiments, we generate 1000 hypotheses and cluster them using nonlinear mean shift.

8.1.1 Camera Pose Segmentation

The pose of a calibrated camera is the rigid body transformation between the world coordinate system to the camera coordinate system and the set of camera poses is the special Euclidean group, $\mathbf{SE}(3)$. We used the OpenCV camera calibration routines based on Zhang (2000) to triangulate three different point clouds. A 10×7 checkerboard pattern was placed next to each of three objects and 25 images from different angles were taken for each object. A few of the images are shown in Fig. 5. The OpenCV routine returns the internal camera parameters and the pose for each of the frames. We then used SIFT (Lowe 2004) to match features across the images and used the calibrated pose estimates to triangulate features in 3D. This gives us a data set of three different point clouds along with the SIFT descriptors for each feature. Each of the three point clouds is triangulated in a different 3D coordinate system.

Fig. 5 *Camera Pose Segmentation.* Images used to reconstruct 3D point clouds using the OpenCV implementation of the calibration technique of Zhang (2000)

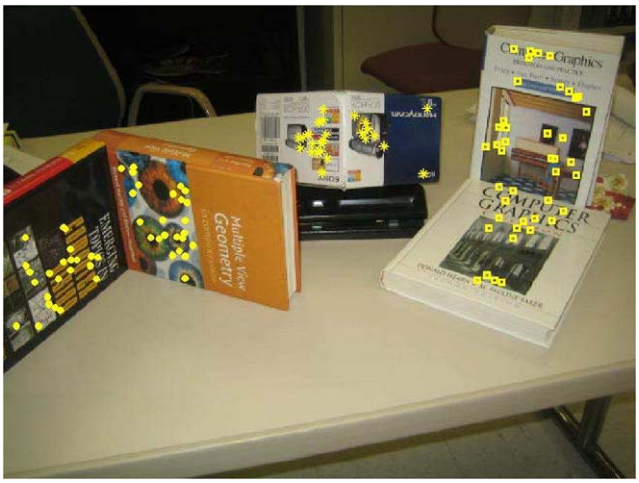


Fig. 6 *Camera Pose Segmentation. Mean shift over* **SE**(3). The figure shows inliers for the different motions found. The table on the *left* contains the properties of the first four modes. The first mode $\mathbf{M}_1$ corresponds to both books on the *right*, $\mathbf{M}_2$ corresponds to both books on the *left* and $\mathbf{M}_3$ is the box in the middle. Only the first three modes are valid motions. The table on the *right* compares the result with the manual segmentation

	Mot. hyp.	kde
$\mathbf{M}_1$	46	0.0388
$\mathbf{M}_2$	65	0.0387
$\mathbf{M}_3$	28	0.0223
M_4	16	0.0092

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	*Out*	ϵ_{res}	ϵ_{LS}
$\mathbf{M}_1$	42	0	0	1	2.46e−4	9.48e−5
$\mathbf{M}_2$	0	49	0	2	1.15e−4	3.90e−5
$\mathbf{M}_3$	0	0	25	0	4.63e−5	9.79e−6
Out	0	0	0	44		

For segmentation, we use an image in which all three objects are present as shown in Fig. 6. The relative pose of the camera with respect to the world coordinate systems of each of the objects is different. Therefore, the pose obtained using only the correspondences from each of the objects will be different. This property can be used to segment the three objects in the 2D image. Using SIFT we match the image features to the previously computed 3D features. This gives 3D to 2D correspondences lying across all three objects. Pose hypotheses are generated by the three-point method of Haralick et al. (1991). Each elemental subset of three points gives upto *two* solutions. The hypothesis are clustered using mean shift over the Lie group **SE**(3) with a bandwidth of $h = 0.1$. The results of the segmentation are shown in Fig. 6.

In the table on the left the number of hypotheses which converge to each mode and the kernel density at the mode are shown. Since the data set has three motions, there are three dominant modes with the fourth mode having a much lower score. The segmentation results are on the right. Each row represents a motion and the row labeled *Out* represents outliers. The first four columns show the classification results. For example, the first row indicates of the 43 inliers for the first motion, 42 are true inliers and one outlier is misclassified as an inlier. Values along the diagonal are correctly classified, while off-diagonal values are misclassifications. The last two columns show the residual errors for our estimates ϵ, and for the least squares estimate, ϵ_{LS}. These residuals are the average reprojection errors in mm^2 for the inliers based on the manual segmentation. Our algorithm's performance, with *no knowledge of the segmentation*, is comparable to manually segmented least squares estimates.

8.1.2 Multi-body Factorization

Here we use mean shift over Grassmann manifolds. The positions of points tracked over F frames of an uncalibrated affine camera define a feature vector in $\mathbb{R}^{2F}$. For points sharing the same motion, these vectors lie in a four dimensional subspace of $\mathbb{R}^{2F}$, and for planar scenes this subspace is only three dimensional (Sugaya and Kanatani 2004). In a scene with multiple planar moving bodies, each motion defines a different subspace, which can be represented by a point in the Grassmann manifold $\mathbf{G}_{2F,3}$, *i.e.*, $n = 2F$ and $k = 3$. An elemental subset consists of the feature vectors defined by three points tracked across F frames. The basis is obtained through singular value decomposition.

The result of multibody factorization with three motions is shown in Fig. 7. The system detects corners in the first frame. Points on the background were identified as having zero displacement and removed leaving 257 corners. These corners are tracked across five frames, *i.e.*, $F = 5$. The planar assumption holds due to low depth variation, and each motion defines a three-dimensional subspace of $\mathbb{R}^{10}$. The three motions contain 100, 46 and 54 points with 57 outliers. We clustered the motion hypotheses on $\mathbf{G}_{10,3}$ with a bandwidth of 0.07. The segmentation returned by the system perfectly matches the manual segmentation. The squared errors in pixel^2 for the system's estimates are comparable to the least squared errors.

	Mot. hyp.	kde
$\mathbf{M}_1$	581	0.469
$\mathbf{M}_2$	51	0.040
$\mathbf{M}_3$	54	0.036
M_4	8	0.016

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	*Out*	ϵ_{res}	ϵ_{LS}
$\mathbf{M}_1$	100	0	0	0	2.187	2.100
$\mathbf{M}_2$	0	46	0	0	1.025	0.901
$\mathbf{M}_3$	0	0	54	0	3.364	1.445
Out	0	0	0	57		

Fig. 7 *Multibody Factorization. Mean shift over* $\mathbf{G}_{10,3}$. The *left* figure shows the first frame with all the points which are tracked. The *right* image shows the fifth and last frame with only the inliers. The table on the *left* contains the properties of the first four modes. The first mode $\mathbf{M}_1$ corresponds to the napkin, $\mathbf{M}_2$ corresponds to the orange book and $\mathbf{M}_3$ to the white book. Only the first three modes are valid motions. The table on the *right* compares the results with the manual segmentation

8.1.3 Epipolar Segmentation

We do mean shift over the essential manifold. We calibrated a camera offline using the method of Zhang (2000). Two images of a scene with two moving bodies were taken. The points on each motion define an essential matrix due to their relative motion with respect to the camera. For hypotheses generation we used the method of Nister (2005). Each elemental subset consists of five point and returns upto ten essential matrices. The hypotheses are clustered over the essential manifold using the theory developed in Sect. 6.3.

The two images used for motion segmentation are shown in Fig. 8. The toy cars move together and the book has a separate motion. Using SIFT, and removing points in the background as having zero displacement, we get 100 point matches with 39 on the book and 42 on the cars and 19 outliers according to the manual segmentation. The mean shift was done with a bandwidth of $h = 0.001$. The clustering returns two dominant modes as expected. The first two modes are clearly more dominant than the third. Some of the outliers are misclassified as inliers since they satisfy the epipolar constraint.

8.2 Discontinuity Preserving Filtering

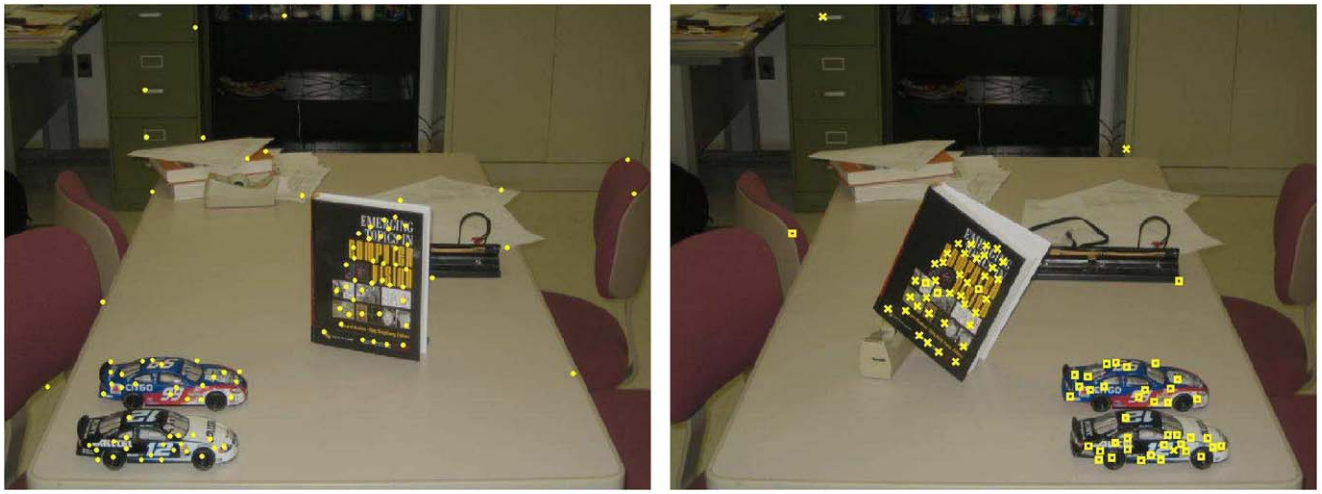
The original mean shift has been used for the discontinuity preserving filtering of color images (Christoudias et al. 2002; Comaniciu and Meer 2002). This algorithm was extended to manifold valued images in Subbarao et al. (2007).

The *image* $\mathbf{I}$ is considered to be a mapping on a n-dimensional lattice which assigns a value to each lattice point. Typically, $n = 2$ or 3. At each location $\mathbf{z}_i$, the data values $\mathbf{I}(\mathbf{z}_i)$ are assumed to lie on a Riemannian manifold, $\mathcal{M}$. A pixel $\mathbf{I}(\mathbf{z}_i)$ along with its location $\mathbf{z}_i$ is considered as a single data point $\mathbf{x}_i = (\mathbf{z}_i, \mathbf{I}(\mathbf{z}_i))$, in the *joint domain* $\mathbb{R}^n \times \mathcal{M}$.

We do mean shift in this joint space to cluster the pixels. Consider an iteration starting at the point $\mathbf{x}_i = (\mathbf{z}_i, \mathbf{c}_i)$, $\mathbf{c}_i = \mathbf{I}(\mathbf{z}_i)$. Let this iteration converge to $(\hat{\mathbf{z}}_i, \hat{\mathbf{c}}_i)$. In the filtered image $\hat{\mathbf{I}}$, we set $\hat{\mathbf{I}}(\mathbf{z}_i) = \hat{\mathbf{c}}_i$. The profile in the joint domain is the product of a *spatial profile* defined on the Euclidean part of the joint domain and a *parameter profile* defined on the manifold, as

$$k(\mathbf{x}, \mathbf{x}_i) = k_s\left(\frac{\|\mathbf{z} - \mathbf{z}_i\|^2}{h_s^2}\right) k_p\left(\frac{d^2(\mathbf{c}, \mathbf{c}_i)}{h_p^2}\right). \tag{59}$$

The bandwidth in the joint domain consists of a spatial bandwidth h_s and a parameter bandwidth h_p. In practice, we use



	Mot. hyp.	kde
$\mathbf{M}_1$	459	0.0215
$\mathbf{M}_2$	409	0.0051
M_3	92	0.0026

	$\mathbf{M}_1$	$\mathbf{M}_2$	*Out*	ϵ_{res}	ϵ_{LS}
$\mathbf{M}_1$	36	1	2	5.31e−5	3.82e−5
$\mathbf{M}_2$	3	38	2	9.86e−4	1.64e−4
Out	0	3	15		

Fig. 8 *Mean shift over the Essential Manifold.* The *left* figure shows the first frame with all the points which are matched. The *right* image shows the second frame with only the inliers returned by the segmentation. The table on the *left* contains the properties of the first three modes. The first mode $\mathbf{M}_1$ is due to the cars and the second mode $\mathbf{M}_2$ is due to the book. Only the first two modes correspond to motions. The table on the *right* compares the results with the manual segmentations

a truncated normal kernel and the performance of the algorithm can be controlled by varying h_p and h_s.

To optimize performance, we used the heuristic suggested in Christoudias et al. (2002) and used in the EDISON system. The filtering step was not applied to pixels which are on the mean shift trajectory of another (already processed) pixel. These pixels were directly associated with the mode to which the path converged. The approximation does not noticeably change the filtered image but reduces processing time.

8.2.1 Chromatic Noise Filtering

Chromatic noise filtering involves mean shift over $\mathbb{R}^2 \times \mathbf{G}_{3,1}$. Chromatic image noise affects the direction (chromaticity) of the color vector but not its intensity. The direction of a 3D vector can be represented by a unit vector in 3D and these form the *Grassmann manifold*, $\mathbf{G}_{3,1}$. By filtering chromaticity we obtain better results than original mean shift which smooths both chromaticity and brightness.

The results for the *jellybeans* image are shown in Fig. 9. The image is corrupted with chromatic noise of standard deviation 0.2. The original mean shift image filtering algorithm from EDISON (Christoudias et al. 2002), was run with a spatial bandwidth $h_s = 11.0$ and color bandwidth $h_p = 10.5$, to get the middle image. Using a larger h_p leads to oversmoothing and using smaller values does not change the image much. The nonlinear mean shift algorithm was run with the same $h_s = 11.0$ but with $h_p = 0.3$ to get the image on the right. Our filtering is clearly better than EDISON due to the averaging of the right noise model.

8.2.2 Diffusion Tensor-MRI Filtering

DT-MRI filtering involves mean shift over $\mathbb{R}^3 \times \mathrm{Sym}_3^+$. Our real data set is a DTI of the human heart obtained from Helm et al. (2006). The lattice size is $128 \times 128 \times 67$ and we ran the smoothing with bandwidth values $h_s = 9.0$ and $h_p = 1.0$. For visualization purposes, each 3×3 diffusion matrix is converted to some scalar value and planes of the 3D lattice are drawn. Here, we use the *fractional anisotropy* (Tschumperle and Deriche 2005)

$$\sqrt{\frac{3}{2}\frac{(\lambda_1-\bar{\lambda})^2+(\lambda_2-\bar{\lambda})^2+(\lambda_3-\bar{\lambda})^2}{\lambda_1^2+\lambda_2^2+\lambda_3^2}} \tag{60}$$

where, λ_1, λ_2 and λ_3 are the eigenvalues and $\bar{\lambda} = (\lambda_1+\lambda_2+\lambda_3)/3$. The fractional anisotropy for a particular plane $z = 47$ is shown in Fig. 10.

Fig. 9 *Chromatic Noise Filtering. Mean Shift over* $\mathbb{R}^2 \times \mathbf{G}_{3,1}$. The *jellybeans* image corrupted with chromatic noise is shown on the *left*. The results of using standard mean shift filtering with EDISON are in the *middle* and the result of nonlinear mean shift filtering is on the *right*

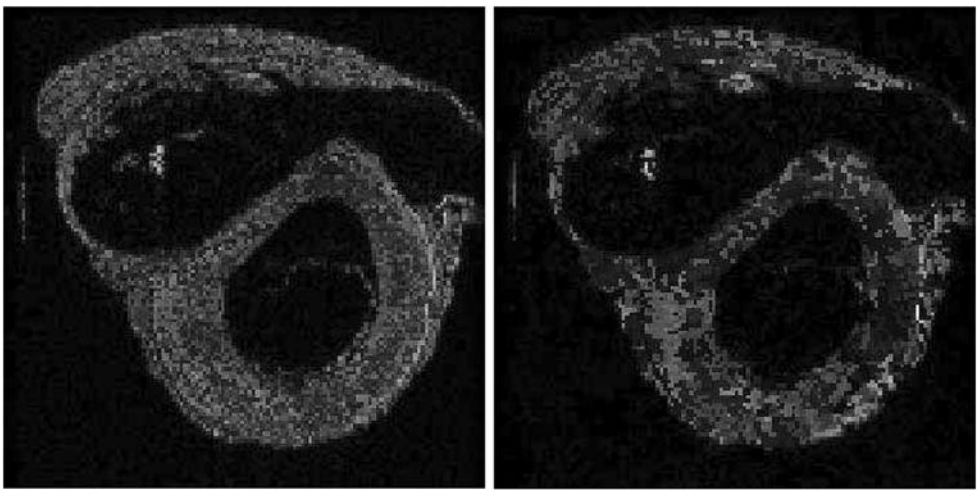


Fig. 10 *DT-MRI Filtering. Mean Shift over* $\mathbb{R}^3 \times \mathrm{Sym}_3^+$. Real DTI data of a human heart before and after smoothing. The jitter in the *left* image is due to noisy voxels having different anisotropies from their surroundings. These are removed by the smoothing and more continuous regions of uniform anisotropy are visible on the *right*

9 Conclusion

We present a new clustering algorithm which works with points lying on Riemannian manifolds. The algorithm is a generalization of mean shift to Riemannian manifolds and it inherits the properties of the mean shift algorithm such as being provably convergent to a maxima of the kernel density. We also discussed a novel parametrization of the essential manifold which is necessary for the application of the nonlinear mean shift to the space of essential matrices. The code IS available at www.caip.rutgers.edu/riul/research/code.html.

We only discussed a few of the many manifolds which occur in applications. It is possible to use the same mean shift algorithm for clustering shapes in shape space (Fletcher et al. 2004; Srivastava et al. 2005) or to cluster probability distributions over the manifolds of distributions (Srivastava et al. 2007). Consequently, we believe the nonlinear mean shift algorithm can be used to solve many different problems in the same way that the original mean shift has so widely been used.

Appendix: Proofs

Proof of Theorem 1 The gradient of the Riemann squared distance is given by

$$\nabla f(\mathbf{x}) = \nabla_{\mathbf{x}}\, d^2(\mathbf{x}, \mathbf{y}) = -2 \log_{\mathbf{x}}(\mathbf{y}). \tag{61}$$

Let $\alpha : [0, 1] \to \mathcal{M}$ be the geodesic from $\mathbf{x}$ to $\mathbf{y}$. Then, the geodesic from $\mathbf{y}$ to $\mathbf{x}$ is given by $\beta(t) = \alpha(1 - t)$. For any $t \in [0, 1]$, we have

$$\beta'(t) = -\alpha'(1 - t). \tag{62}$$

It states that, for any point on the geodesic between $\mathbf{x}$ and $\mathbf{y}$, the tangents along the curves $\alpha(t)$ and $\beta(t)$ differ by multiplication by -1. Specifically, at $t = 1$ we get

$\beta'(1) = -\alpha'(0)$. Recall that $\alpha'(0)$ and $\log_{\mathbf{x}}(\mathbf{y})$ are two different expressions for the same initial velocity of the geodesic from $\mathbf{x}$ to $\mathbf{y}$. Therefore, $\beta'(1) = -\log_{\mathbf{x}}(\mathbf{y})$ and it is sufficient to prove

$$\nabla f(\mathbf{x}) = 2\beta'(1). \tag{63}$$

Let $\tilde{\mathcal{V}}$ be the star-shaped neighborhood of the origin in $T_{\mathbf{y}}(\mathcal{M})$ and $(\mathcal{V}, \phi)$ be the corresponding normal neighborhood of $\mathbf{y} \in \mathcal{M}$. The function $f(\mathbf{x})$ measures the squared distance of $\mathbf{x}$ from $\mathbf{y}$. Since, the velocity of the geodesic is constant, the length of the geodesic is equal to the velocity along the geodesic

$$f(\mathbf{x}) = g_{\mathbf{y}}(\log_{\mathbf{y}}(\mathbf{x}), \log_{\mathbf{y}}(\mathbf{x})). \tag{64}$$

We now define, $\tilde{f} = f \circ \phi^{-1} : \mathbb{R}^m \to \mathbb{R}$. Given a point $\mathbf{u} \in \mathbb{R}^m$, $\phi^{-1}(\mathbf{u}) \in \mathcal{M}$ is a point on the manifold. By definition, we have $\log_{\mathbf{y}}(\phi^{-1}(\mathbf{u})) = \sum_i u^i \mathbf{e}_i$, where u^i is the i-th coordinate of $\mathbf{u}$ and $\mathbf{e}_i$ is the orthonormal basis of $T_{\mathbf{y}}(\mathcal{M})$. Therefore,

$$\begin{aligned}
\tilde{f}(\mathbf{u}) &= f(\phi^{-1}(\mathbf{u})) \\
&= g_{\mathbf{y}}(\log_{\mathbf{y}}(\phi^{-1}(\mathbf{u})), \log_{\mathbf{y}}(\phi^{-1}(\mathbf{u}))) \\
&= g_{\mathbf{y}}\left(\sum_{i=1}^{m} u^i \mathbf{e}_i, \sum_{i=1}^{m} u^i \mathbf{e}_i\right) \\
&= \sum_{i=1}^{m} (u^i)^2.
\end{aligned} \tag{65}$$

In the final step, we used the orthonormality of the basis $\mathbf{e}_i$. From O'Neill (1983, p. 85), the j-th component of the gradient is given by

$$(\nabla f)^j = \sum_{i=1}^{m} g^{ij} \frac{\partial \tilde{f}}{\partial u^i} = \frac{\partial \tilde{f}}{\partial u^j} = 2u^j \tag{66}$$

where, we use the fact that the Riemannian metric matrix is the identity when expressed in normal coordinates. By definition, u^j is the component of the tangent $\log_{\mathbf{y}}(\mathbf{x})$ along $\mathbf{e}_j$. Therefore, $\nabla f = 2\beta'(1)$ and (63) holds. □

Proof of Theorem 2 If the kernel K has a convex and monotonically decreasing profile and the bandwidth h is less than the injectivity radius $i(\mathcal{M})$ of the manifold, the sequence $\{f(\mathbf{y}_j)\}_{j=1,2,\ldots}$ is convergent and monotonically non-decreasing.

To prove this theorem we use a result shown in Karcher (1977). Consider a set of points $\mathbf{x}_i, i = 1, \ldots, n$ lying on a Riemannian manifold. The *weighted Karcher mean* of these points with weights $w_i \in \mathbb{R}, i = 1, \ldots, n$ is defined as the points which minimize the cost function

$$C(\mathbf{y}) = \sum_{i=1}^{n} w_i d^2(\mathbf{y}, \mathbf{x}_i). \tag{67}$$

It was shown in Karcher (1977) that if all the data points lie within an injectivity radius of each other, then the minimizer is unique and can be found using an iterative procedure. Let, $\mathbf{y}_j$ be the current estimate of the Karcher mean, then the updated estimate is given by

$$\mathbf{y}_{j+1} = \exp_{\mathbf{y}_j}\left(\sum_{i=1}^{n} w_i \log_{\mathbf{y}_j}(\mathbf{x}_i) \Big/ \sum_{i=1}^{n} w_i\right). \tag{68}$$

If $\mathbf{y}_{j+1}$ is obtained from $\mathbf{y}_j$ using the above rule, then $C(\mathbf{y}_{j+1}) \le C(\mathbf{y}_j)$ and

$$\sum_{i=1}^{n} w_i (d^2(\mathbf{y}_j, \mathbf{x}_i) - d^2(\mathbf{y}_{j+1}, \mathbf{x}_i)) \ge 0. \tag{69}$$

Using this result we prove the convergence of the mean shift iterations. Each kernel is a real-valued function with a maximum value of 1 and the kernel density (27) is a sum of n such kernels. Since n is finite, the value of $f(\mathbf{y}_j)$ is bounded above. To prove convergence of the sequence $\{f(\mathbf{y}_j)\}_{j=1,2,\ldots}$ it is sufficient to prove that

$$f(\mathbf{y}_{j+1}) \ge f(\mathbf{y}_j). \tag{70}$$

Using (27), we can write

$$\begin{aligned}
&f(\mathbf{y}_{j+1}) - f(\mathbf{y}_j) \\
&\quad = \frac{c_{k,h}}{n} \sum_{i=1}^{n} \left[k\left(\frac{d_{j+1,i}^2}{h^2}\right) - k\left(\frac{d_{j,i}^2}{h^2}\right)\right]
\end{aligned} \tag{71}$$

where, we use the notation $d_{j+1,i} = d(\mathbf{y}_{j+1}, \mathbf{x}_i)$. Due to the convexity of the kernel we have

$$k(z_2) \ge k(z_1) + k'(z_1)(z_2 - z_1) \tag{72}$$

for any $z_1, z_2 \in \mathbb{R}$. Since, $g(\cdot) = -k(\cdot)$, we rewrite (72) as

$$k(z_2) - k(z_1) \ge g(z_1)(z_1 - z_2). \tag{73}$$

Using this identity in (71) for each of the n terms we get

$$\begin{aligned}
&f(\mathbf{y}_{j+1}) - f(\mathbf{y}_j) \\
&\quad \ge \frac{c_{k,h}}{nh^2} \sum_{i=1}^{n} \left[g\left(\frac{d_{j,i}^2}{h^2}\right)(d_{j,i}^2 - d_{j+1,i}^2)\right].
\end{aligned} \tag{74}$$

We now use (69) with the data points $\mathbf{x}_i, i = 1, \ldots, n$ and the weights $w_i = g(d_{j,i}^2/h^2)$. These weights change at each iteration, but for a single iteration they are constant and the

inequality holds. In this case (68) is the same as the mean shift step (30) and therefore the right side of the above inequality if nonnegative. Therefore, (70) is true and the sequence of values $f(\mathbf{y}_j)$ is nondecreasing.

The only condition that needs to be verified is whether all the points lie within an injectivity radius of each other as required for the result of Karcher (1977) to hold. This can be ensured by using a value of h less than the injectivity radius. In this case, the weights $g(d_{j,i}^2/h^2)$ are zero for all points further than h away and all points with nonzero weights will lie within an injectivity radius of each other as required. □

References

Absil, P.-A., Mahony, R., & Sepulchre, R. (2003). Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, *80*(2), 199–220.

Arsigny, V., Fillard, P., Pennec, X., & Ayache, N. (2006). Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, *29*(1), 328–347.

Basser, P. J., Mattiello, J., & LeBihan, D. (1994). MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, *66*, 259–267.

Begelfor, E., & Werman, M. (2006). Affine invariance revisited. In *Proc. IEEE conf. on computer vision and pattern recognition*, New York, NY (Vol. II, pp. 2087–2094).

Birchfield, S., & Rangarajan, S. (2005). Spatiograms vs. histograms for region-based tracking. In *Proc. IEEE conf. on computer vision and pattern recognition*, San Diego, CA, June 2005 (Vol. II, pp. 1158–1163).

Carreira-Perpinan, M. A. (2007). Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(5), 767–776.

Chen, H., & Meer, P. (2005). Robust fusion of uncertain information. *IEEE Transactions on Systems, Man, Cybernetics Part B*, *35*, 578–586.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 790–799.

Chikuse, Y. (2003). *Statistics on special manifolds*. Berlin: Springer.

Christoudias, C. M., Georgescu, B., & Meer, P. (2002). Synergism in low level vision. In *Proc. 16th intl. conf. on pattern recognition*, Quebec, Canada (Vol. IV, pp. 150–155).

Collins, R. (2003). Mean shift blob tracking through scale space. In *Proc. IEEE conf. on computer vision and pattern recognition*, Madison, WI (Vol. II, pp. 234–240).

Comaniciu, D. (2003). Variable bandwidth density-based fusion. In *Proc. IEEE conf. on computer vision and pattern recognition*, Madison, WI, June 2003 (Vol. 1, pp. 59–66).

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 603–619.

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 564–577.

Davis, B., Fletcher, P. T., Bullitt, E., & Joshi, S. (2007). Population shape regression from random design data. In *Proc. 11th intl. conf. on computer vision*, Rio de Janeiro, Brazil, Oct. 2007.

Edelman, A., Arias, T. A., & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, *20*(2), 303–353.

Elgammal, A., Duraiswami, R., & Davis, L. S. (2003). Efficient kernel density estimation using the efficient kernel density estimation using the color modeling and tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(11), 1499–1504.

Fashing, M., & Tomasi, C. (2005). Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(3), 471–474.

Ferreira, R., & Xavier, J. (2006). Hessian of the Riemannian squared-distance function on connected locally symmetric spaces with applications. In *Controlo 2006, 7th Portuguese conference on automatic control*.

Fletcher, P., Joshi, S., Lu, C., & Pizer, S. (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, *23*(8), 995–1005.

Fletcher, P. T., Lu, C., & Joshi, S. (2003). Statistics of shape via principal geodesic analysis on Lie groups. In *Proc. IEEE conf. on computer vision and pattern recognition*, Madison, WI (pp. 95–101).

Forstner, W., & Moonen, B. (1999). *A metric for covariance matrices* (Technical report). Dept. of Geodesy and Geoinformatices, Stuttgart University.

Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, *21*, 32–40.

Geyer, C., Bajcsy, R., & Sastry, S. (2004). Euclid meets Fourier: Applying harmonic analysis to essential matrix estimation in omnidirectional cameras. In *Proc. of workshop on omnidirectional vision, camera networks and non-classical cameras*.

Golub, G. H., & Van Loan, C. F. (1989). *Matrix computations* (2nd ed.). Baltimore: John Hopkins Press.

Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. In *Proc. IEEE conf. on computer vision and pattern recognition*, Washington, DC (Vol. I, pp. 684–691).

Hager, G., Dewan, M., & Stewart, C. (2004). Multiple kernel tracking with SSD. In *Proc. IEEE conf. on computer vision and pattern recognition*, Washington, DC (Vol. I, pp. 790–797).

Haralick, R., Lee, C., Ottenberg, K., & Nolle, M. (1991). Analysis and solutions of the three point perspective pose estimation problem. In *Proc. IEEE conf. on computer vision and pattern recognition*, Maui, HA (pp. 592–598).

Hartley, R. I., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Helm, P. A., Winslow, R. L., & McVeigh, E. (2006). *Center for cardiovascular bioinformatics and modeling* (Technical report). Johns Hopkins University.

Helmke, U., Huper, K., Lee, P., & Moore, J. (2007). Essential matrix estimation using Gauss-Newton iterations on a manifold. *International Journal of Computer Vision*, *74*(2), 117–136.

Kanatani, K. (1990). *Group theoretical methods in image understanding*. Berlin: Springer.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, *30*(5), 509–541.

Lenglet, C., Deriche, R., & Faugeras, O. (2004). Inferring white matter geometry from diffusion tensor MRI: Application to connectivity mapping. In *Proc. European conf. on computer vision*, Prague, Czech Republic (Vol. IV, pp. 127–140).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Ma, Y., Kosecka, J., & Sastry, S. (2001). Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, *44*(3), 219–249.

Maybank, S. J. (1992). *Theory of reconstruction from image motion*. Berlin: Springer.

Mordohai, P., & Medioni, G. (2007). *Tensor voting: a perceptual organization approach to computer vision and machine learning*. Morgan and Claypool Publishers.

Nister, D. (2005). Preemptive RANSAC for live structure from motion. *Machine Vision and Applications*, *16*(5), 321–329.

O'Neill, B. (1983). *Semi-Riemannian manifolds:with applications to relativity*. San Diego: Academic Press.

Pelletier, B. (2005). Kernel density estimation on Riemannian manifolds. *Statistics and Probability Letters*, *73*(3), 297–304.

Pennec, X., & Ayache, N. (1998). Uniform distribution, distance and expectation problems for geometric feature processing. *Journal of Mathematical Imaging and Vision*, *9*(1), 49–67.

Pennec, X., Fillard, P., & Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, *66*(1), 41–66.

Rossmann, W. (2003). *Lie groups: an introduction through linear groups*. London: Oxford University Press.

Singh, M., Arora, H., & Ahuja, N. (2004). A robust probabilistic estimation framework for parametric image models. In *Proc. European conf. on computer vision*, Prague, Czech Republic, May 2004 (Vol. I, pp. 508–522).

Smith, S. T. (1994). Optimization techniques on Riemannian manifolds. *Fields Institute Communications*, *3*, 113–146.

Soatto, S., Perona, P., Frezza, R., & Picci, G. (1994). Motion estimation via dynamic vision. In *Proc. European conf. on computer vision*, Stockholm, Sweden (Vol. II, pp. 61–72).

Srivastava, A., Jermyn, I., & Joshi, S. (2007). Riemannian analysis of probability density functions with applications in vision. In *Proc. IEEE conf. on computer vision and pattern recognition*, Minneapolis, MN.

Srivastava, A., Joshi, S., Mio, W., & Liu, X. (2005). Statistical shape analysis: Clustering, learning and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 590–602.

Subbarao, R., Genc, Y., & Meer, P. (2007). Nonlinear mean shift for robust pose estimation. In *8th IEEE workshop on applications of computer vision*, Austin, TX, February 2007.

Subbarao, R., & Meer, P. (2006). Nonlinear mean shift for clustering over analytic manifolds. In *Proc. IEEE conf. on computer vision and pattern recognition*, New York, NY (Vol. I, pp. 1168–1175).

Subbarao, R., & Meer, P. (2007). Discontinuity preserving filtering over analytic manifolds. In *Proc. IEEE conf. on computer vision and pattern recognition*, Minneapolis, MN.

Sugaya, Y., & Kanatani, K. (2004). Geometric structure of degeneracy for multi-body motion segmentation. In *LNCS: Vol. 3247. The 2nd workshop on statistical methods in video processing (SMVP 2004)* (pp. 13–25). Berlin: Springer.

Tschumperle, D., & Deriche, R. (2003). DT-MRI images: Estimation, regularization and application. In *Computer aided systems theory—EUROCAST 2003* (pp. 531–541).

Tschumperle, D., & Deriche, R. (2005). Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 506–517.

Tuzel, O., Subbarao, R., & Meer, P. (2005). Simultaneous multiple 3D motion estimation via mode finding on Lie groups. In *Proc. 10th intl. conf. on computer vision*, Beijing, China (Vol. 1, pp. 18–25).

Vemuri, B. C., Chen, Y., Rao, M., McGraw, T., Wang, Z., & Mareci, T. (2001). Fiber tract mapping from diffusion tensor MRI. In *Proceedings of the IEEE workshop on variational and level set methods* (pp. 81–88).

Wang, J., Thiesson, B., Xu, Y., & Cohen, M. (2004). Image and video segmentation by anisotropic kernel mean shift. In *Proc. European conf. on computer vision*, Prague, Czech Republic (Vol. II, pp. 238–249).

Yang, C., Duraiswami, R., DeMenthon, D., & Davis, L. (2003). Mean-shift analysis using quasi-Newton methods. In *Intl. conf. on image processing* (Vol. II, pp. 447–450).

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.