CVPR 2004 TUTORIAL ON

NONLINEAR MANIFOLDS IN COMPUTER VISION

Anuj Srivastava[*], Washington Mio[**] and Xiuwen Liu[***]

[*]Department of Statistics

[**]Department of Mathematics

[***]Department of Computer Science

Florida State University

Sunday, June 27th, 1:00-5:00pm, Washington DC.

PROGRAM OVERVIEW

Part I: Motivation (1:00 – 2:00pm)

presented by Anuj Srivastava

— Why should one study nonlinear manifolds?

— How can differential geometry be useful in vision applications?

— Examples, illustrations, and references

Part II: Tools from Differential Geometry (2:00 – 3:30pm)

presented by Washington Mio

— Differentiable manifolds, definitions and examples

— Tangent vectors and tangent spaces

— Riemannian metrics, gradients, geodesics, exponential map

— Integral curves and flows

Part III: Statistics on Nonlinear Manifolds (3:30 – 4:00pm)

presented by Anuj Srivastava

— Intrinsic and Extrinsic means

— Covariances in tangent spaces

— Probability distributions on manifolds

— Estimation and testing on manifolds, with examples

Part IV: Algorithms and Applications in Computer Vision (4:00 – 5:00pm)

presented by Xiuwen Liu

— Statistics on a unit circle

— Geodesic flows on shape spaces

— Optimal linear bases for image-based recognition

— Clustering on shape manifolds

NONLINEAR MANIFOLDS

- What are nonlinear manifolds?:

  Nonlinear manifolds are manifolds that are not vector spaces, i.e. if $x, y \in M$, then $ax + by$ may not be in M, for a pair $a, b \in \mathbb{R}$ [1, 14]. The usual Euclidean calculus may not apply. Conventional statistics may not apply. (More later in Part II)

- When do we need them?:

  In applications where certain constraints make the underlying space nonlinear. For example, elements of $\mathbb{R}^n$ with unit norm constraint. Or, matrices with orthogonality constraint.

- How do we deal with them?:

  Using differential geometry of the underlying manifold, one can usually derive most of the corresponding results from Euclidean spaces.

MOTIVATION – OUTLINE

A large number of problems in computer vision can be studied a problems in optimization or inference on nonlinear manifolds.

We will describe some examples:

- Pose estimation and recognition of 3D objects from static 2D images.

- Tracking of 3D objects using video sequences.

- Component Analysis: Is PCA better than ICA? Or vice-versa? – Search for optimal linear projection for reducing image size, before a statistical analysis.

  Non-negative matrix factorizations, sparse representations, etc are other examples.

- Statistical study of shapes: shapes of 2D and 3D objects can be represented via landmarks, curves, or surfaces.

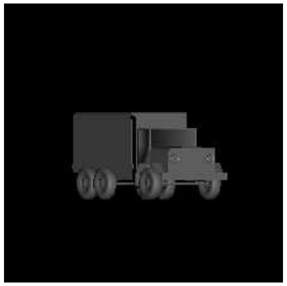
- Learning Image manifold: estimation and learning of manifolds formed by images embedded in a bigger Euclidean space.

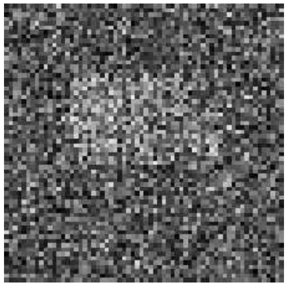
- Structure from motion, shape from shading, etc

PROBLEM 1: POSE ESTIMATION

We are interested in recognizing known objects in observed images. Consider a simple sub-problem from "transformable templates".

Goal: Given a 3D model of an object (truck in the figure), and a noisy image I, how to estimate its pose (orientation) in that image?





Mathematical Formulation:

Let I^{α} be a 3D mesh of the object (e.g. facial scan). Orientation of an object can be represented by a 3×3 orthogonal matrix [6]. The space of all such matrices is:

$$SO(3) = \{O \in \mathbb{R}^{3\times 3} | O^T O = I_3, \ \det(O) = 1\} .$$

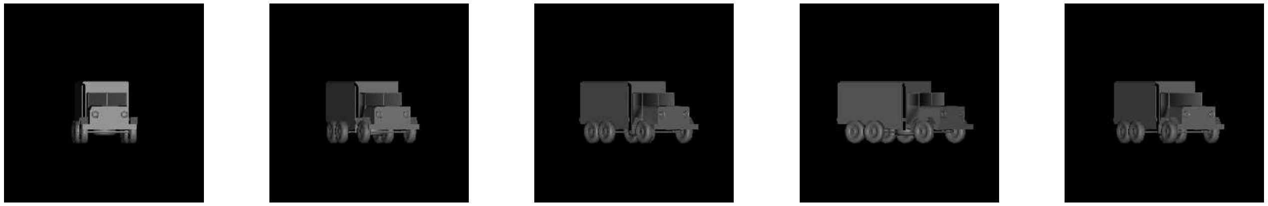


Figure 1: A truck rendered at different orientations in $SO(2)$.

Notation: Let $T(sI^\alpha)$ denote an image of an object α taken from a pose $s \in SO(3)$. T is a projection from $\mathbb{R}^3$ to $\mathbb{R}^2$.

Image Model: Let a simple image model be: $I = T(sI^\alpha) \oplus clutter$.

Problem: Given an image I, estimate $s \in SO(3)$ according to:

$$\hat{s}_\alpha = \operatorname*{argmax}_{s \in SO(3)} P(I|s, \alpha) \quad \text{(MLE)}\ .$$

$SO(3)$ is not a vector space. However, it is a Lie group! How to solve optimization problems on Lie groups?

Given a function $F : SO(n) \rightarrow \mathbb{R}$, we need to tools to solve optimization problems on $SO(n)$. We need definitions of derivatives (gradients), Hessians, increments, etc. [10, 5, 3, 15]

GRADIENT-BASED OPTIMIZATION

In $\mathbb{R}^n$, one can seek a local optimum of a function F using a gradient process:

$$\frac{dX(t)}{dt} = \nabla F(X(t)) \; ,$$

or its discrete implementation, for $\epsilon > 0$ small,

$$X(t+\epsilon) = X(t) + \epsilon \nabla F(X(t)) \; .$$

Addition is not a valid operation in $SO(3)$!

One has to use the differential geometry of $SO(3)$ to construct gradient flows.

OBJECT RECOGNITION

Goal: Given a noisy image I of an object, find the most probable object?

Several strategies. One is to define posterior probability:

$$P(\alpha|I) = P(\alpha) \int_S P(I|s,\alpha) P(s|\alpha) ds \; .$$

S can include pose, illumination, motion, etc. S is the set of nuisance variables in recognition.

Solve for:

$$\hat{\alpha} = \operatorname*{argmax}_{\alpha} P(\alpha|I) \; .$$

This requires tools for integration on a Lie group.

PROBLEM 2: OBJECT MOTION TRACKING

Goal: Given a time series of sensor measurements (images, range measurements), estimate a moving object's positions and orientations at each time.

Representation: Rotation and translation as treated as elements of special Euclidean group $SE(3) = SO(3) \ltimes \mathbb{R}^3$. Consider the stochastic process

$$\{s_t \in SE(3),\ t = 1, 2, \ldots, T\} ,$$

and the goal is to estimate the time series $\{s_t\}$ using observations.

Solution:

If s_t was in a Euclidean space, then the solution comes from filtering, smoothing, or prediction.

Classical Filtering: For Euclidean case,

Use a state equation and an observation equation:

$$\begin{aligned} \text{State equation :} \quad & s_{t+1} = A(s_t) + \mu_t \\ \text{Observation equation :} \quad & I_{t+1} = B(s_{t+1}) + \nu_t \end{aligned}$$

Then, find the mean and covariance associated with the posterior

$$P(s_t | I_1, I_2, \ldots, I_t) \; .$$

If A and B are linear functions, (and μ_t, ν_t are independent Gaussian), then Kalman filtering provides exact solution. Else, use Monte Carlo methods for approximate solutions.

Nonlinear Manifolds: Kalman filtering does not apply, but Monte Carlo methods do!

Use nonlinear filtering equations: Under usual Markov assumptions,

$$\begin{aligned} \text{Predict : } P(s_{t+1}|I_{[1:t]}) &= \int_{s_t} P(s_{t+1}|s_t) P(s_t|I_{[1:t]}) ds_t \\ \text{Update : } P(s_{t+1}|I_{[1:t+1]}) &= \frac{P(I_{t+1}|s_{t+1}) P(s_{t+1}|I_{[1:t]})}{P(I_{t+1}|I_{[1:t]})} \end{aligned}$$

Use Monte Carlo method (condensation, Jump-Diffusion, etc.) to solve for the posterior mean, covariance, etc, at each time.

We need tools to sample from probability distributions and to compute averages from these samples on nonlinear manifolds.

Statistics on circle: [9], statistics on $SO(n)$: [4], statistics on Grassmann manifolds: [16]

PROBLEM 3: OPTIMAL LINEAR PROJECTIONS

- Raw images are elements of a high-dimensional space.

- One need to reduce their dimension using a linear or a nonlinear method, before proceeding to a statistical analysis.

- In view of their simplicity, linear projections are popular, e.g. PCA, ICA, FDA, etc.

- Let $X \in \mathbb{R}^n$ be a random image (as a column vector), and $U \in \mathbb{R}^{n \times d}$ ($d << n$) be an orthogonal matrix. That is, $U^T U = I_d$. Let $u(X) = U^T X$ be a reduced representation of X.

- How to choose U in a particular application? For example, in recognition of people by their facial images, is PCA the best linear projection one can have?
- In general, the solution is difficult!

  However, assume that we have a well-defined performance function F that evaluates the choice of U.

  For example, F is the recognition performance on the training data. (More in Part IV). F can also be a measure of sparsity, or independence of components.
- How to solve for:

$$\hat{U} = \operatorname*{argmax}_{U} F(U) \ .$$

  On what space should this optimization problem be solved?

Representation: The set of all $n \times d$ orthogonal matrices forms a Stiefel manifold

$\mathcal{S}_{n,d}$. In case F depends on the subspace but not on a basis chosen to represent the subspace, then the search space is a Grassmann manifold $\mathcal{G}_{n,d}$.

There are quotient spaces:

Steifel: $\mathcal{S}_{n,d} = SO(n)/SO(n-d)$ and

Grassmann: $\mathcal{G}_{n,d} = SO(n)/(SO(n-d) \times SO(d))$.

Solution: Solve for

$$\hat{U} = \operatorname*{argmax}_{\mathcal{S}_{n,d}} F(U) \text{ or } \operatorname*{argmax}_{\mathcal{G}_{n,d}} F(U) \; .$$

Example: X denotes face images and $u(X)$ is used for recognition. Let F be a recognition performance and choose U to maximize F. U is an element of a Grassmann manifold $\mathcal{G}_{n,d}$. [8].

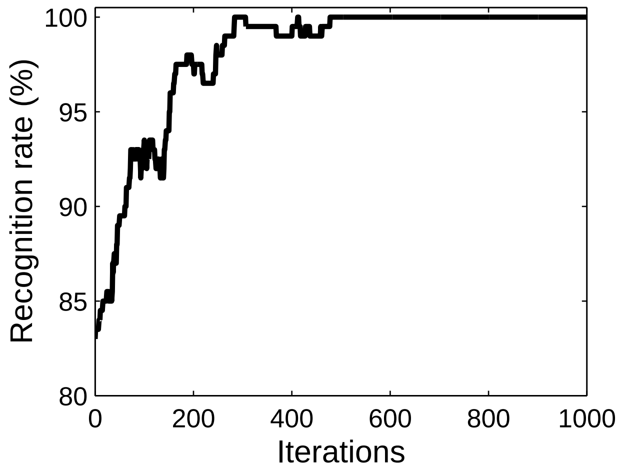
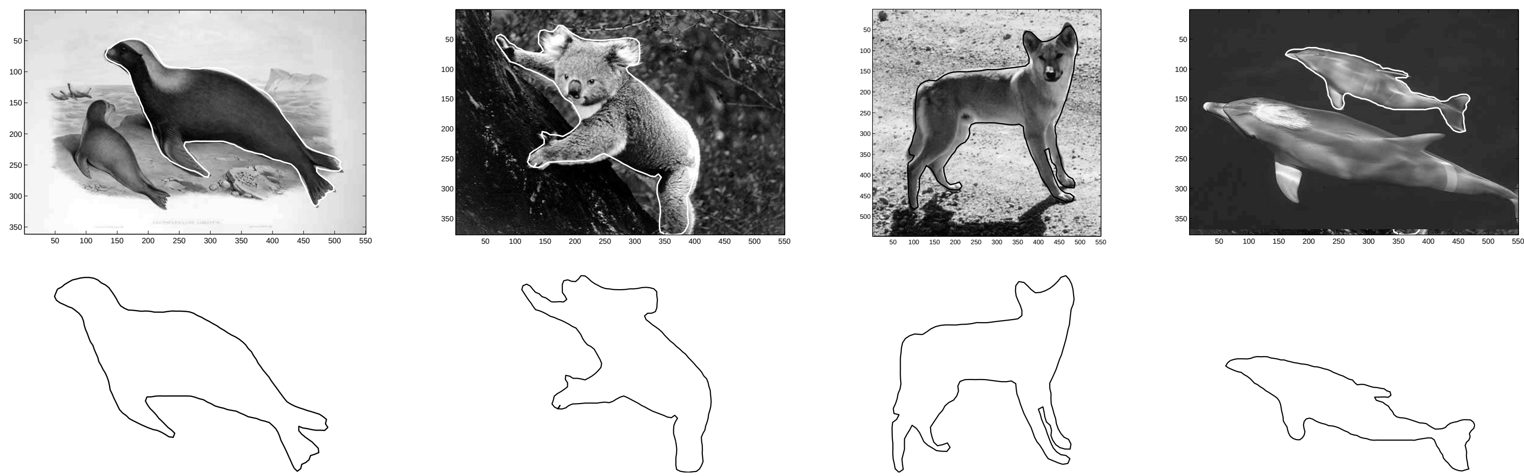


Figure shows evolution of $F(X_t)$, where X_t is a stochastic optimization process on $\mathcal{G}_{n,d}$.

(More in Part IV)

PROBLEM 4: STATISTICAL SHAPE ANALYSIS



We wish to use shapes of boundaries as features in object recognition.

Consider shapes in $\mathbb{R}^2$. The important issues are: how to represent shapes, how to compare them, how to analyze them statistically?

PROCRUSTES APPROACH

Representation: Let a shape be represented by k points on its boundary. A shape is an element of $x \in \mathbb{R}^{2k}$ or $z \in \mathbb{C}^k$. [2, 13].

Shape Preserving Transformations: Under rigid rotation and translation, and uniform scaling, the shape remains unchanged. We need a quantity that uniquely represents a shape. Assuming known registration:

- Remove translation by assuming $\frac{1}{k}\sum_{i=1}^{k} z_i = 0$. All shapes are centered.
- Remove scale by assuming $\|z\| = 1$, i.e. all shape vectors lie on a unit sphere.
- Remove rotation in a pairwise fashion. For example, to align $z^{(2)}$ to $z^{(1)}$ replace by $e^{j\hat{\theta}} z^{(2)}$, where

$$\hat{\theta} = \underset{\theta \in SO(2)}{\operatorname{argmin}} \|z^{(1)} - e^{j\theta} z^{(2)}\|^2 .$$

PROCRUSTES APPROACH Contd.

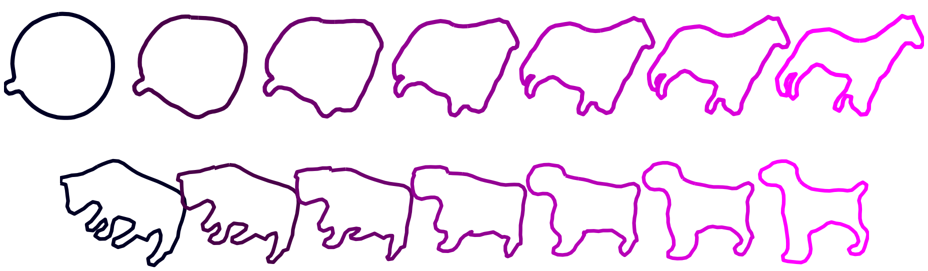
Pre-shape Space:

$$\mathcal{C} = \{z \in \mathbb{C}^k | \frac{1}{k} \sum_{i=1}^{k} z_i = 0, \|z\| = 1\} \,.$$

Shape Space: $\mathcal{S} = \mathcal{C}/SO(2)$. Shape space is a quotient space; all shapes that are within a planar rotation are considered equivalent.

How to quantify dissimilarities between two shapes?

Construct a geodesic path between them on $\mathcal{S}$, and compute the geodesic length. It serves as a shape metric.



ANOTHER APPROACH: GEOMETRY OF CLOSED CURVES

Goal: To analyze shapes of continuous boundaries of objects as they appear in images. [7]

Representation:

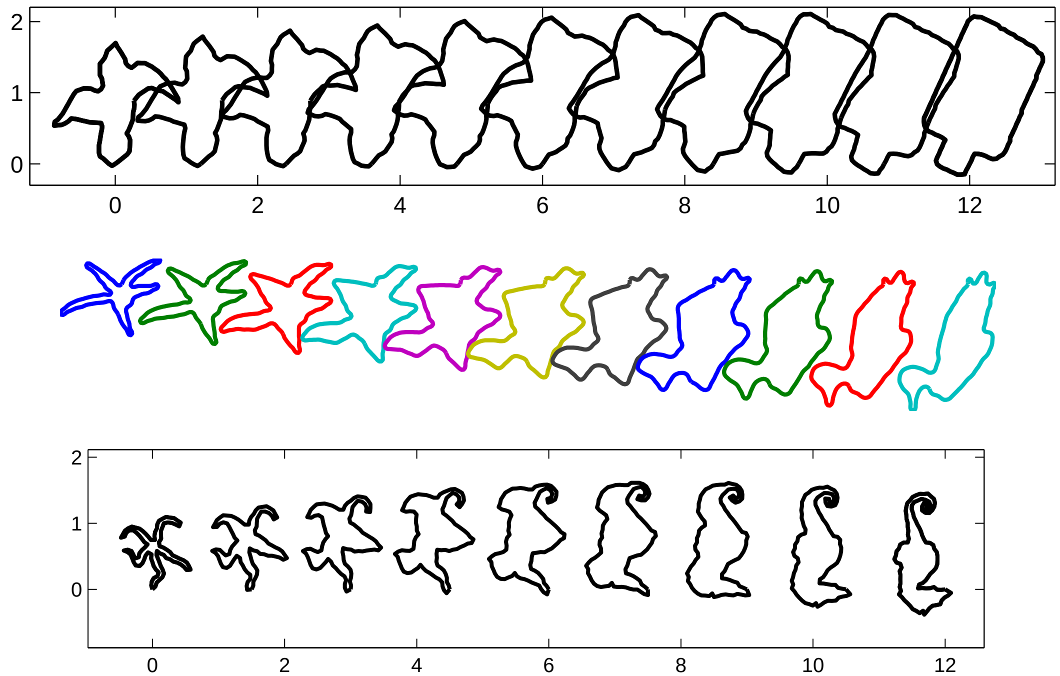
- Denote shapes by their angle functions θ (with arc-length parametrization). Consider

$$\mathcal{C} = \{\theta \in \mathbb{L}^2[0, 2\pi] | \int_0^{2\pi} \cos(\theta(s))ds = \int_0^{2\pi} \sin(\theta(s))ds = 0\}$$

- Shape space $\mathcal{S} = \mathcal{C}/S$, where S includes rotation (unit circle) and re-parametrization (unit circle) groups.

Solution: Compute exponential flows or geodesics on $\mathcal{S}$ to quantify shape differences.

Example



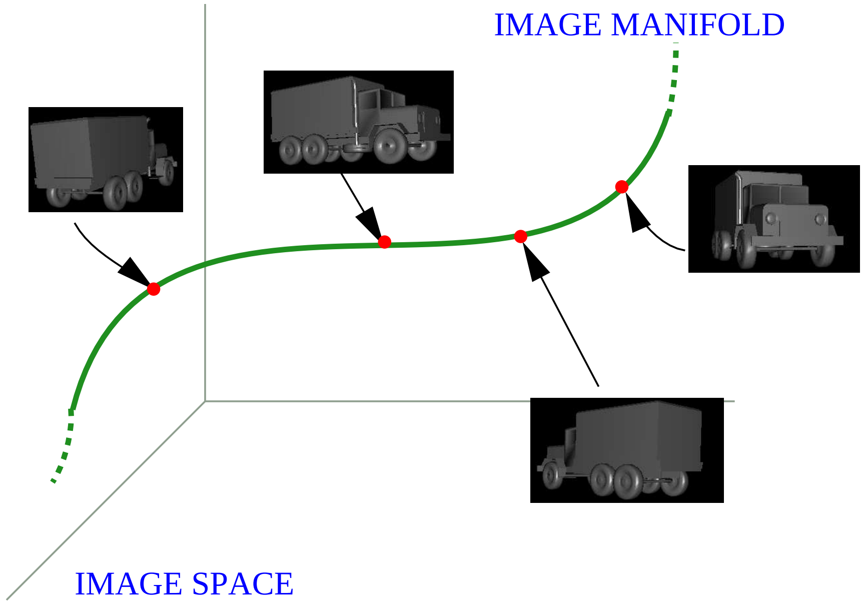
Geodesic paths between the end shapes. Geodesic lengths quantify shape differences, and help perform statistical analysis of shapes.

GEODESIC FLOWS ON SHAPE SPACES OF CUVES

PROBLEM 5: LEARNING IMAGE MANIFOLDS

Let $I(s, \alpha)$ denote an image of object α at variables (pose, location, illumination, etc) denoted by s. Let $\mathcal{I}^{\alpha} = \{I(s, \alpha) | s \in S\}$ be the collection of all such images of α.

$\mathcal{I}^{\alpha}$ is called image manifold of α.



Complete image manifold:

$$\mathcal{I} = \bigcup_{\alpha} \mathcal{I}^{\alpha} \, .$$

Goal: We are interested in estimating (or learning) $\mathcal{I}$ from the given observations.

Representation: $\mathcal{I}$ is a low-dimensional manifold embedded in a large Euclidean space. We wish to represent individual images by their coefficients in low-dimensional subspaces.

Solutions: [12]

http://www.cse.msu.edu/ lawhiu/manifold/

1. Isomap: Denote observed images as points on a discrete lattice. Approximate geodesic lengths on $\mathcal{I}$ by shortest graph lengths on this lattice. Perform dimension reduction, dimension estimation, clustering, etc. using this geodesic length. [11].

2. Locally-linear embedding: [17]

3. Pattern Theoretic Approach:

CVPR TUTORIAL ON

NONLINEAR MANIFOLDS IN COMPUTER VISION

II. DIFFERENTIABLE MANIFOLDS

Washington Mio

Anuj Srivastava and Xiuwen Liu

(Illustrations by D. Badlyans)

July 27th, 2004 – Washington, DC

WHY MANIFOLDS?

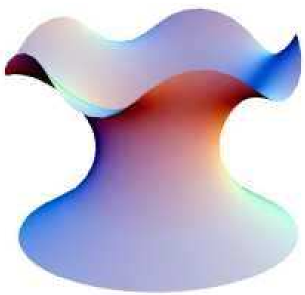
Non-linearity underlies many mathematical structures and representations used in computer vision. For example, many **image and shape descriptors** fall in this category, as discussed in the introduction to this tutorial.

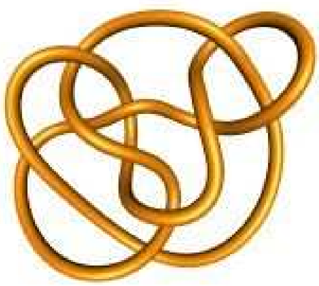
On the other hand, **differential calculus** is one of the most useful tools employed in the study of phenomena modeled on **Euclidean n-space** $\mathbb{R}^n$. The study of **optimization and inference problems, dynamics, and geometry** on linear spaces can all be approached with the tools of calculus.

A natural question arises:

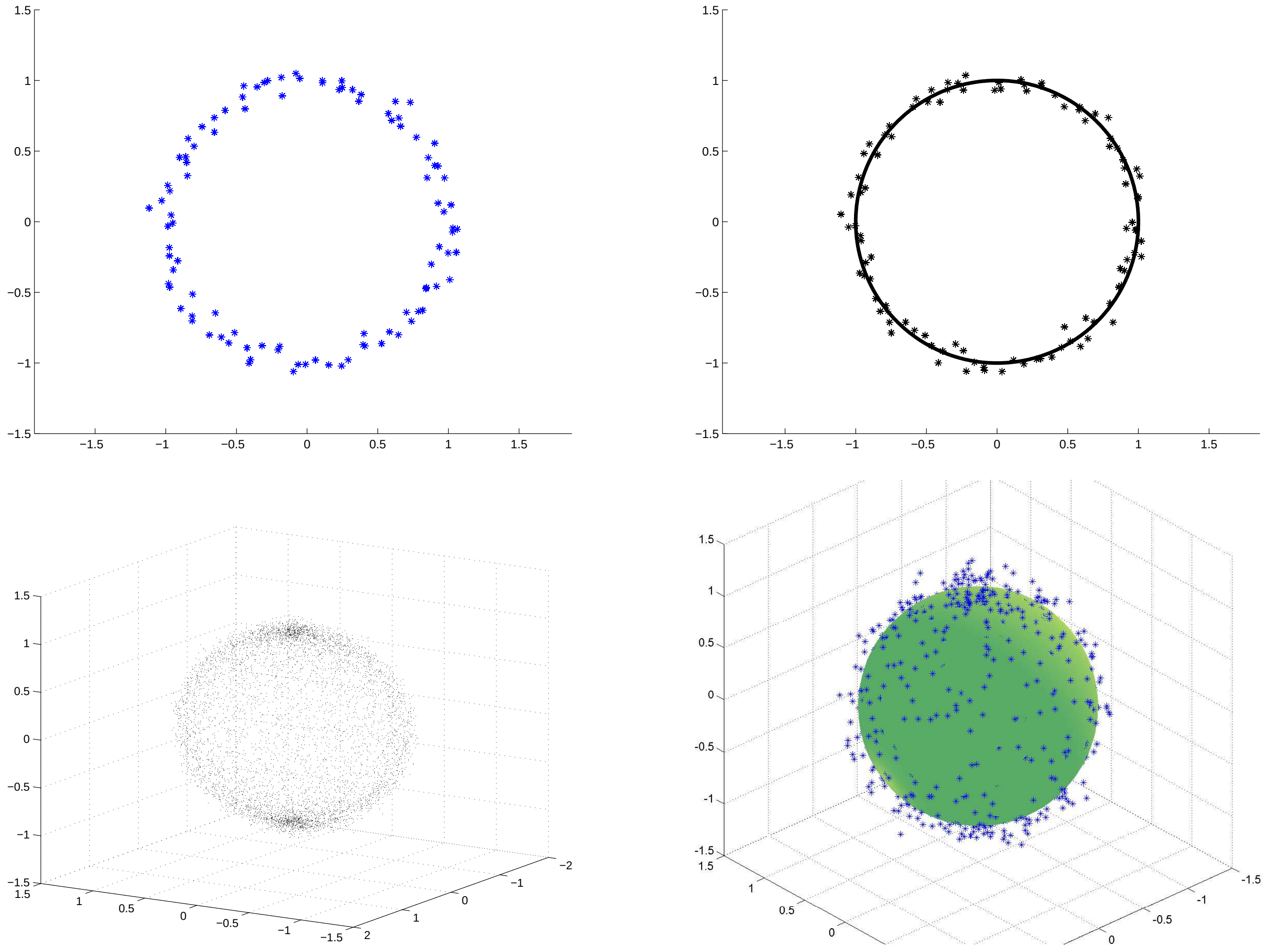
What spaces can the techniques of differential calculus be extended to?

Since ***derivative*** is a local notion, it is reasonable to expect that differential calculus can be developed in spaces that locally "look like" Euclidean spaces. These spaces are known as ***differentiable manifolds***.





Manifolds Underlying Data Points



For simplicity, we only consider ***manifolds in Euclidean spaces***, but a more general approach may be taken. Also, most concepts to be discussed extend to ***infinite-dimensional manifolds***, whose relevance to shape and vision problems already has been indicated in the introduction.

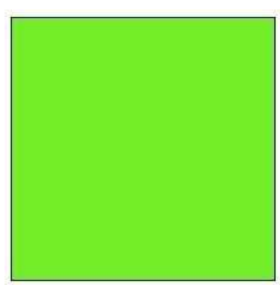
Our discussion will be somewhat informal. A few **references** to more complete and general treatments:

- **W. Boothby**, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, 2002.
- **F. Warner**, *Foundations of Differential Geometry and Lie Groups*, Graduate Texts in Mathematics, Springer-Verlag, 1983.
- **J. Lee**, *Introduction to Smooth Manifolds*, Springer-Verlag, 2002.

How to make sense of “locally similar” to an Euclidean space?

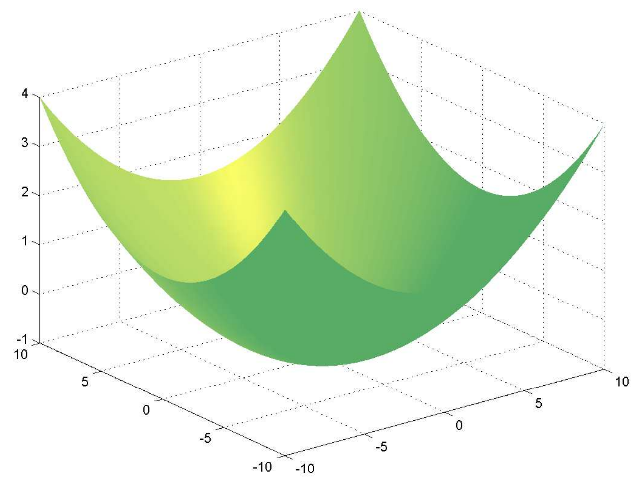
A map $\varphi\colon U \to \mathbb{R}^m$ defined on an open region $U \subseteq \mathbb{R}^n$, $n \leq m$, is said to be a ***parameterization*** if:

(i) φ is a ***smooth*** (i.e., infinitely differentiable), ***one-to-one*** mapping.



$U \subset \mathbb{R}^2$

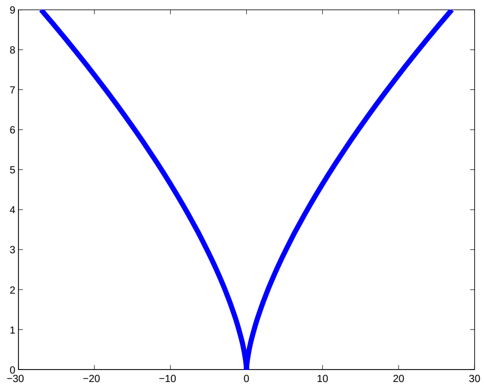
$\xrightarrow{\varphi}$



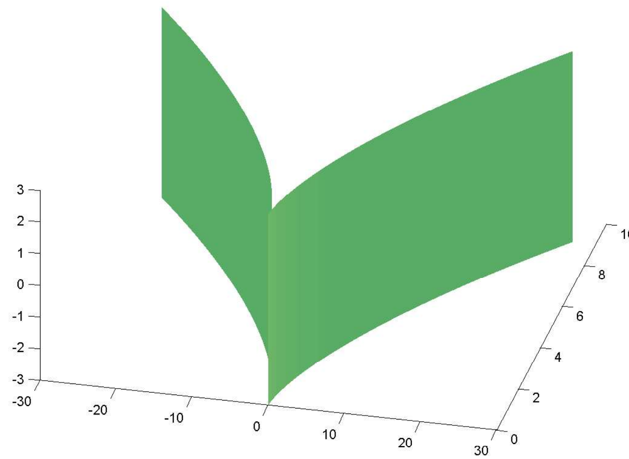
This simply says that $V = \varphi(U)$ is produced by ***bending and stretching*** the region U in a gentle, elastic manner, ***disallowing self-intersections***.

(ii) The $m \times n$ ***Jacobian*** matrix $J(x) = \left[\frac{\partial \varphi_i}{\partial x_j}(x)\right]$ has ***rank n***, for every $x \in U$.

Here, $x = (x_1, \ldots, x_n)$ and $\varphi = (\varphi_1, \ldots, \varphi_m)$.



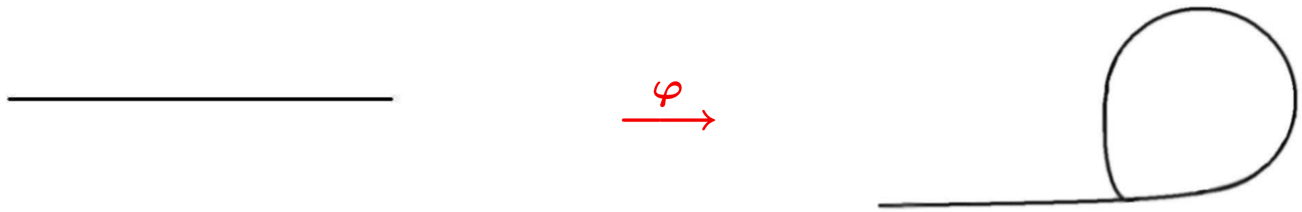
(a) $\boldsymbol{\varphi(t) = (t^3, t^2)}$



(b) $\boldsymbol{\varphi(t, s) = (t^3, t^2, s)}$

This condition further ensures that V has ***no sharp bends, corners, peaks***, or other singularities.

(ii) The ***inverse*** map $\boldsymbol{x} = \varphi^{-1} : V \to U$ is ***continuous***.

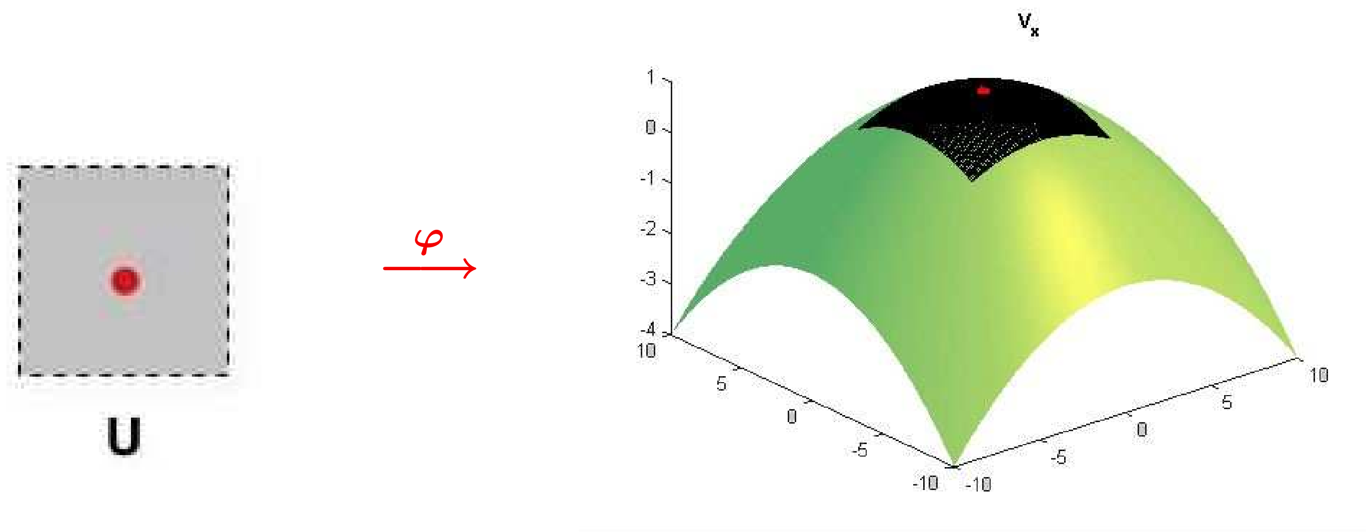


This condition has to do with the fact that we should be able to recover $\boldsymbol{U}$ from $\boldsymbol{V}$ with a ***continuous deformation***. In the illustration above, φ^{-1} is not continuous.

One often refers to φ as a ***parameterization*** of the set $\boldsymbol{V = \varphi(U)}$, and to the inverse mapping $\boldsymbol{x = \varphi^{-1}}$ as a ***local chart***, or a ***local coordinate system*** on V.

Definition of Differentiable Manifolds

A subspace $M \subseteq \mathbb{R}^m$ is an ***n-dimensional differentiable manifold*** if every point has a neighborhood that admits a parameterization defined on a region in $\mathbb{R}^n$. More precisely, for every point $x \in M$, there are a neighborhood $V_x \subseteq M$ of x, and a parameterization $\varphi \colon U \subseteq \mathbb{R}^n \to \mathbb{R}^m$, with $\varphi(U) = V_x$.



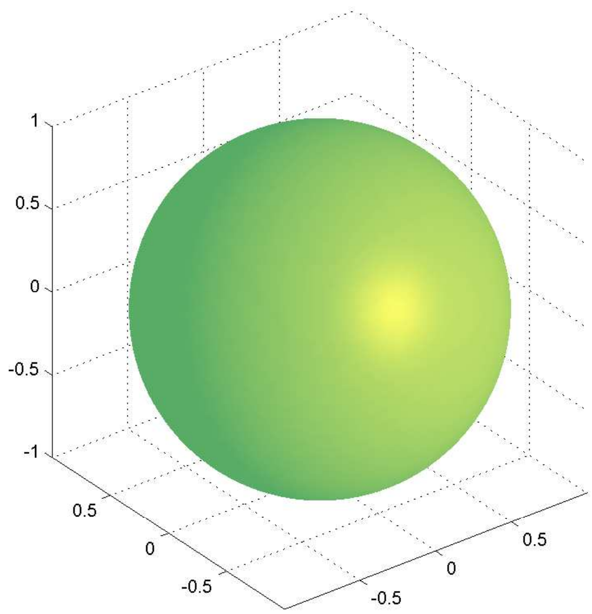
Examples of Manifolds

- **The Space of Directions in $\mathbb{R}^n$**

A direction is determined by a unit vector. Thus, this space can be identified with the unit sphere

$$\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \,:\, \|x\| = 1\}$$

Notice that $\mathbb{S}^{n-1}$ can be described as the ***level set*** $F^{-1}(1)$ of the function $F(x_1, \ldots, x_n) = x_1^2 + \ldots + x_n^2$.



This implicit description of the sphere is a special case of the following general construction.

Let $F\colon \mathbb{R}^m \to \mathbb{R}^k$, $k < m$, be a smooth mapping, and let $a \in \mathbb{R}^k$. If the $k \times m$ **Jacobian matrix** $J(x)$ of F has **rank** $\boldsymbol{k}$, for every $x \in F^{-1}(a)$, then

$$\boldsymbol{M = F^{-1}(a) \subset \mathbb{R}^m}$$

is an $(m-k)$-dimensional manifold. When this condition on J is satisfied, $\boldsymbol{a \in \mathbb{R}^k}$ is said to be a ***regular value*** of F.

• The Graph of a Smooth Function

Let $g\colon \mathbb{R}^n \to \mathbb{R}^\ell$ be smooth. Then,

$$\boldsymbol{M = \{(x, y) \in \mathbb{R}^{n+k} \;:\; y = g(x)\}}$$

is an n-dimensional manifold. Here, $F(x, y) = y - g(x)$ and $M = F^{-1}(0)$.

• The Group $O(n)$ of Orthogonal Matrices

Let $\mathcal{M}(n) \cong \mathbb{R}^{n^2}$ be the space of all $n \times n$ real matrices, and $\mathcal{S}(n) \cong \mathbb{R}^{n(n+1)/2}$ the subcollection of all symmetric matrices. Consider the map $\boldsymbol{F} \colon \boldsymbol{\mathcal{M}(n) \to \mathcal{S}(n)}$ given by $F(A) = AA^t$, where the superscript t indicates transposition.

Orthogonal matrices are those satisfying $AA^t = I$. Hence,

$$\boldsymbol{O(n) = F^{-1}(I)}.$$

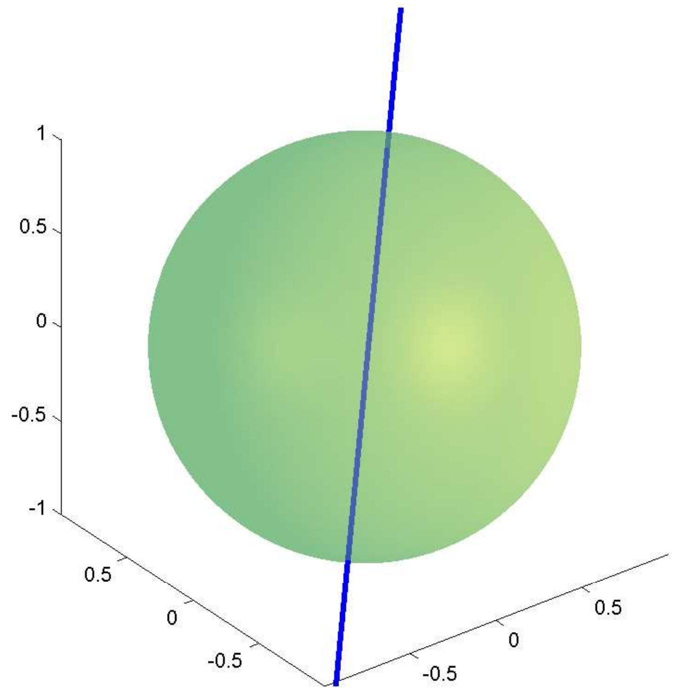
It can be shown that I is a regular value of F. Thus, the ***group of $n \times n$ orthogonal matrices*** is a manifold of dimension

$$\boldsymbol{n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}}.$$

Remark. $O(n)$ has two components formed by the orthogonal matrices with positive and negative determinants, resp. The ***positive component*** is denoted $\boldsymbol{SO(n)}$.

• Lines in $\mathbb{R}^n$ Through the Origin

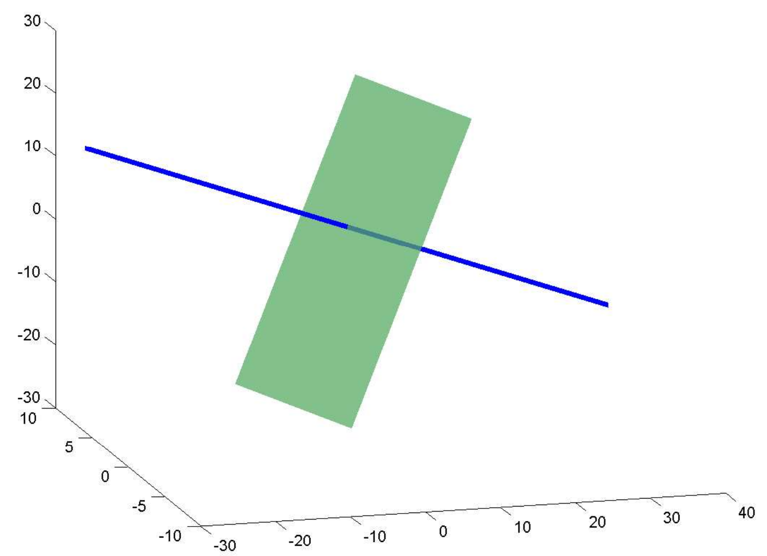
A line in $\mathbb{R}^n$ through the origin is completely determined by its intersection with the unit sphere $\mathbb{S}^{n-1}$. This intersection always consists a pair of antipodal points $\{x, -x\} \subset \mathbb{S}^{n-1}$. Thus, the ***space of lines through the origin*** can be modeled on the sphere $\mathbb{S}^{n-1}$ with antipodal points identified. This is an $(n-1)$-dimensional manifold known as the ***real projective space*** $\mathbb{RP}^{n-1}$.



• Grassmann Manifolds

The spaces $\boldsymbol{G(n,k)}$ of all k-planes through the origin in $\mathbb{R}^n$, $k \leq n$, generalize real projective spaces. $G(n,k)$ is a manifold of dimension $k(n-k)$ and is known as a ***Grassmann manifold***.

A k-plane in $\mathbb{R}^n$ determines an $(n-k)$-plane in $\mathbb{R}^n$ (namely, its orthogonal complement), and vice-versa. Thus, $\boldsymbol{G(n,k) \cong G(n,n-k)}$.

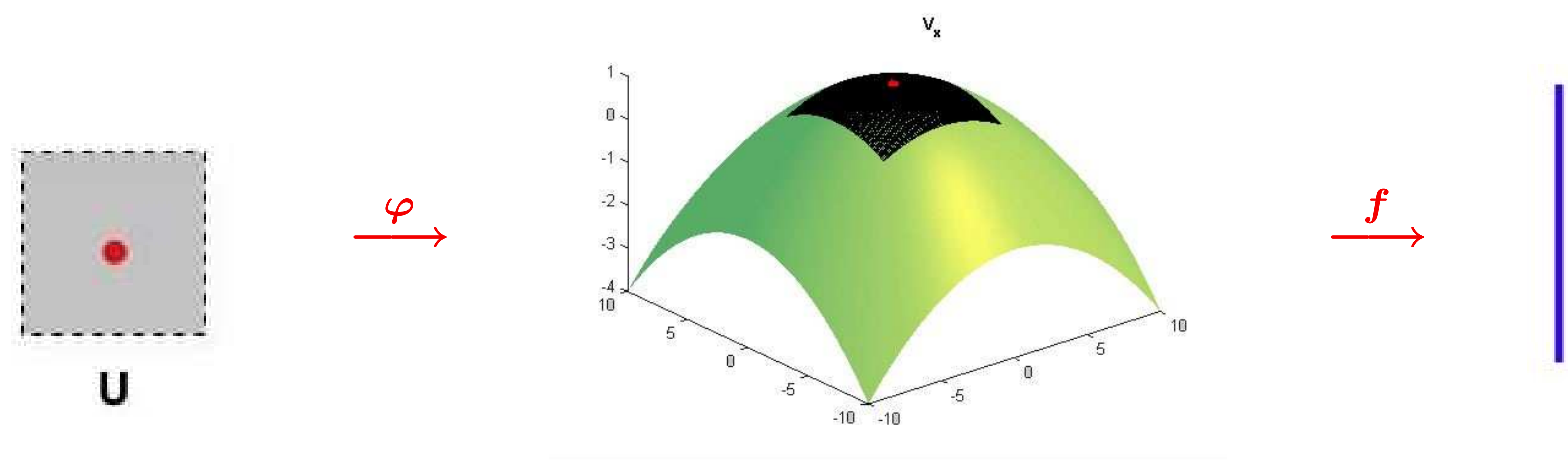


What is a Differentiable Function?

Let $f\colon M^n \subset \mathbb{R}^m \longrightarrow \mathbb{R}$ be a function. Given $u_0 \in M$, let $\varphi\colon U \longrightarrow \mathbb{R}^m$ be a parameterization such that $\varphi(x_0) = u_0$. The map f is said to be ***differentiable at the point u_0*** if the composite map

$$f \circ \varphi\colon U \subset \mathbb{R}^n \longrightarrow \mathbb{R},$$

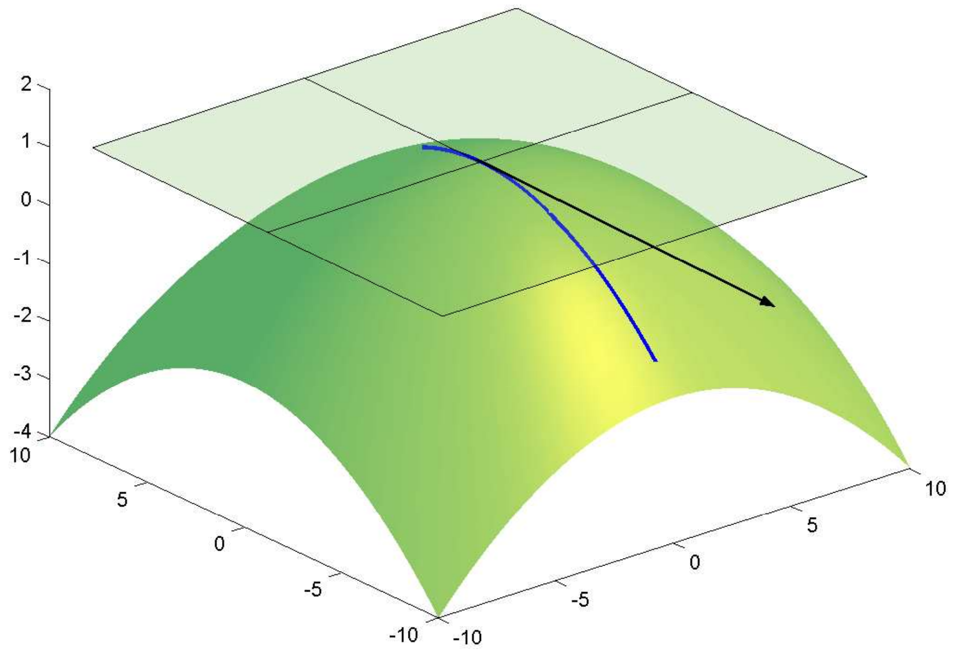
given by $f \circ \varphi(x) = f(\varphi(x))$, is differentiable at x_0.



Tangent Vectors

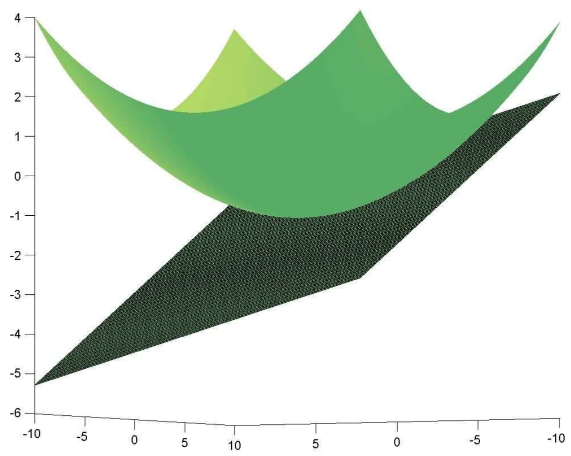
A ***tangent vector*** to a manifold $M^n \subset \mathbb{R}^m$ at a point $p \in M$ is a vector in $\mathbb{R}^m$ that can be realized as the velocity at p of a curve in M.

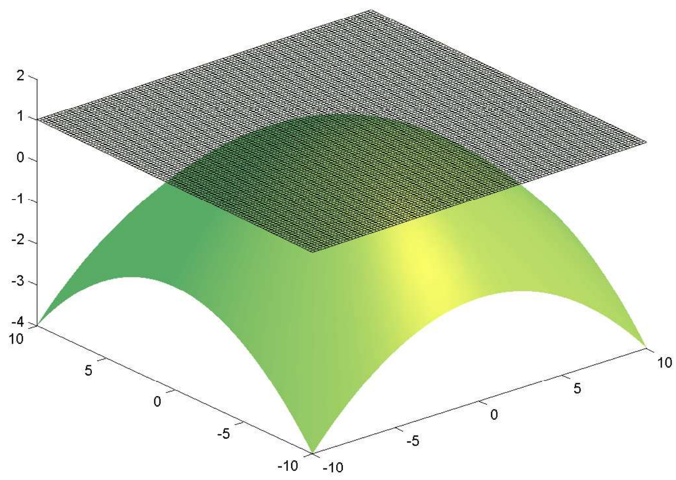
More precisely, a vector $\boldsymbol{v} \in \mathbb{R}^{\boldsymbol{m}}$ is tangent to M at p if there is a smooth curve $\alpha\colon (-\varepsilon, \varepsilon) \to M \subseteq \mathbb{R}^m$ such that $\boldsymbol{\alpha(0) = p}$ and $\boldsymbol{\alpha'(0) = v}$.



Tangent Spaces

The collection of all tangent vectors to an n-manifold $M^n \subset \mathbb{R}^m$ at a point p forms an n-dimensional real vector space denoted $\boldsymbol{T_pM}$.





If $\varphi\colon U \subset \mathbb{R}^n \to M$ is a parameterization with $\varphi(a) = p$, then the set

$$\left\{ \frac{\partial \varphi}{\partial x_1}(a), \ldots, \frac{\partial \varphi}{\partial x_n}(a) \right\}$$

is a basis of T_pM.

Example

Let $F : \mathbb{R}^m \longrightarrow \mathbb{R}$ be a differentiable function, and let $a \in \mathbb{R}$ be a regular value of F. The manifold $\boldsymbol{M = F^{-1}(a)}$ is $(n-1)$-dimensional and the tangent space to M at p consists of all vectors ***perpendicular to the gradient vector*** $\nabla F(x)$; that is,

$$\boldsymbol{T_x M = \{v \in \mathbb{R}^m \ : \ \langle \nabla F(x), v \rangle = 0\}\,.}$$

Here, $\langle \ , \ \rangle$ denotes the usual inner (or dot) product on $\mathbb{R}^n$.

- **The Sphere**

If $F(x_1, \ldots, x_n) = x_1^2 + \ldots + x_n^2$, the level set $\mathbb{S}^{n-1} = F^{-1}(1)$ is the $(n-1)$-dimensional ***unit sphere***. In this case, $\nabla F(x) = 2(x_1, \ldots, x_n) = 2x$.

Thus, the tangent space to the $\mathbb{S}^{n-1}$ at the point x consists of vectors orthogonal to x; i.e.,

$$\boldsymbol{T_x \mathbb{S}^{n-1} = \{v \in \mathbb{R}^n \ : \ \langle x, v \rangle = 0\}\,.}$$

Derivatives

Let $f\colon M \longrightarrow \mathbb{R}$ be a differentiable function and $\boldsymbol{v \in T_pM}$ a tangent vector to M at the point p.

Realize v as the velocity vector at p of a parametric curve $\alpha\colon (-\varepsilon, \varepsilon) \longrightarrow M$. The ***derivative of f*** at p ***along v*** is the derivative of $f(\alpha(t))$ at $t = 0$; that is,

$$\boldsymbol{df_p(v) = \frac{d}{dt}\,(f \circ \alpha)\,(0).}$$

The derivatives of f at $p \in M$ along vectors $v \in T_pM$ can be assembled into a single linear map

$$\boldsymbol{df_p\colon T_pM \longrightarrow \mathbb{R} \qquad v \longmapsto df_p(v),}$$

referred to as the ***derivative*** of f at p.

Differential Geometry on M

Given a manifold $M^n \subset \mathbb{R}^m$, one can ***define geometric quantities*** such as the length of a curve in M, the area of a 2D region in M, and curvature.

The length of a parametric curve $\alpha\colon [0,1] \to M$ is defined in the usual way, as follows. The velocity vector $\alpha'(t)$ is a tangent vector to M at the point $\alpha(t)$. The ***speed*** of α at t is given by

$$\|\alpha'(t)\| = [\langle \alpha'(t), \alpha'(t) \rangle]^{1/2},$$

and the ***length*** of α by

$$L = \int_0^1 \|\alpha'(t)\| \, dt.$$

Geodesics

The study of geodesics is motivated, for example, by the question of finding the ***curve of minimal length*** connecting two points in a manifold M. This is important in applications because minimal-length curves can be used to define an intrinsic ***geodesic metric*** on M, which is useful in a variety of ways. In addition, geodesics provide interpolation and extrapolation techniques, and tools of statistical analysis on non-linear manifolds.

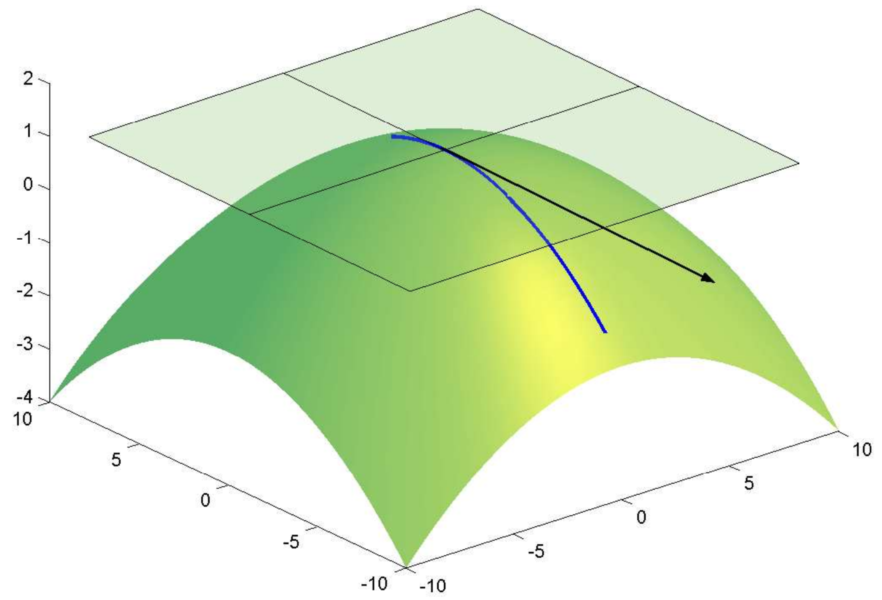
In order to avoid further technicalities, think of geodesics as length-minimizing curves in M (locally, this is always the case). A more accurate way of viewing geodesics is as ***parametric curves with zero intrinsic acceleration***.

A curve $\alpha\colon I \to M \subset \mathbb{R}^m$ is a geodesic if and only if the ***acceleration vector*** $\boldsymbol{\alpha''(t)} \in \mathbb{R}^{\boldsymbol{m}}$ ***is orthogonal*** to $\boldsymbol{T_{\alpha(t)}M}$, for every $t \in I$.

Numerical Calculation of Geodesics

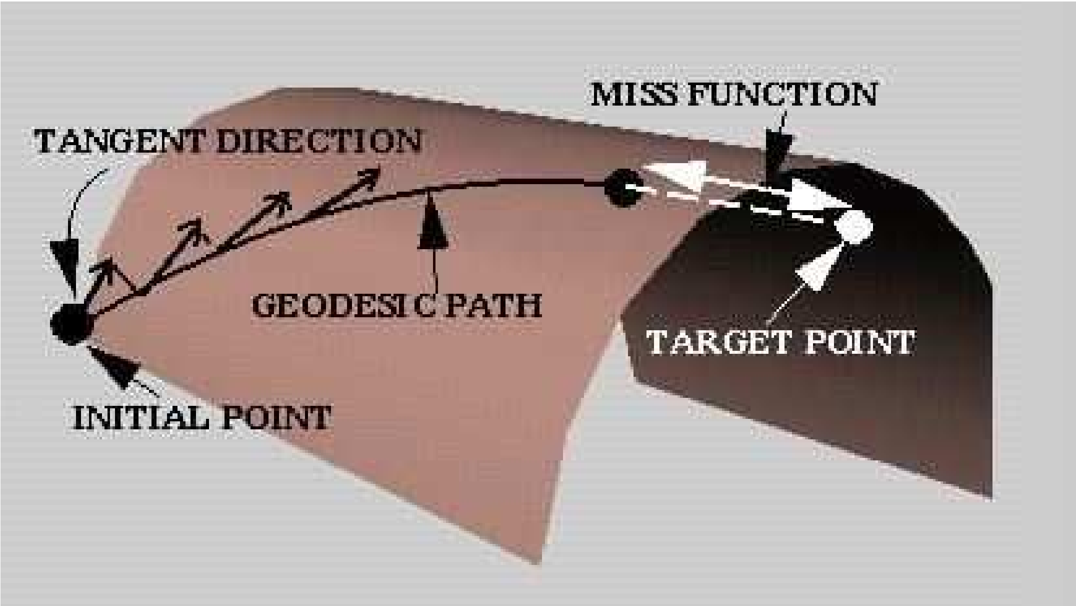
1. Geodesics with prescribed initial position and velocity

For many manifolds that arise in applications to computer vision, one can write the differential equation that governs geodesics explicitly, which can then be integrated numerically using classical methods.

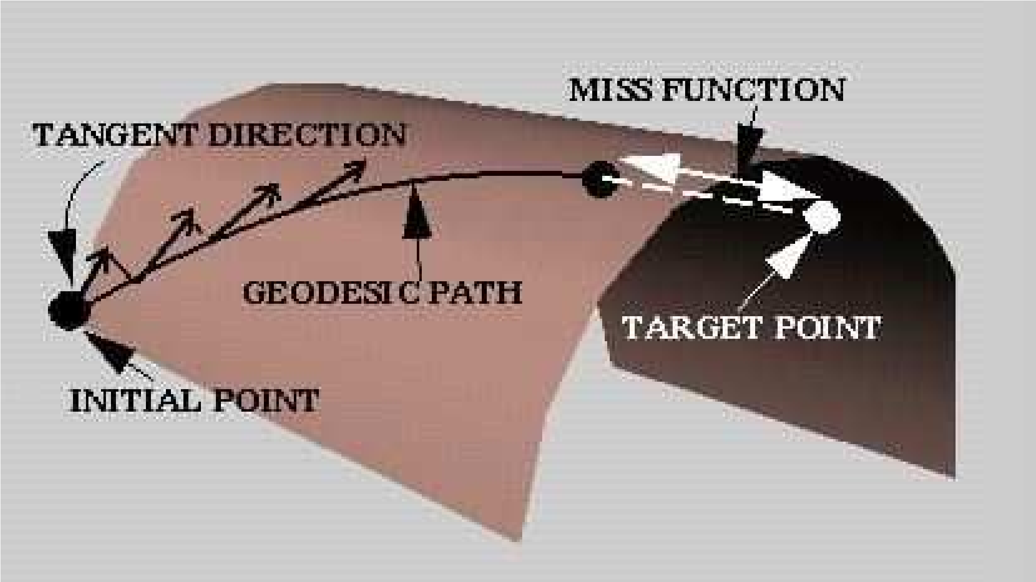


2. Geodesics between points $a, b \in M$

One approach is to use a ***shooting strategy***. The idea is to shoot a geodesic from a in a given direction $v \in T_a M$ and follow it for unit time. The terminal point $\exp_a(v) \in M$ is known as the ***exponential of v***. If we **hit the target** (that is, $\exp(v) = b$), then the problem is solved.



Otherwise, we consider the ***miss function*** $\boldsymbol{h(v) = \| \exp(v) - b\|^2}$, defined for $v \in T_a M$.



The goal is to minimize (or equivalently, annihilate) the function h. This ***optimization problem*** on the linear space $T_a M$ can be approached with standard gradient methods.

Optimization Problems on Manifolds

How can one find ***minima*** of functions defined on nonlinear manifolds ***algorithmically***? How to carry out a ***gradient search***?

Example. Let $x_1, \ldots, x_k$ be sample points in a manifold M. How to make sense of the ***mean*** of these points?

The arithmetic mean $\boldsymbol{\mu = (x_1 + \ldots + x_k)/k}$ used for data in linear spaces does not generalize to manifolds in an obvious manner. However, a simple calculation shows that $\boldsymbol{\mu}$ can be viewed as the ***minimum*** of the ***total variance function***

$$\boldsymbol{V(x) = \frac{1}{2} \sum_{i=1}^{k} \|x - x_i\|^2.}$$

This minimization problem can be posed in more general manifolds using the geodesic metric.

Given $x_1, \ldots, x_k \in M$, consider the total variance function

$$V(x) = \frac{1}{2} \sum_{i=1}^{k} d^2(x, x_i),$$

where d denotes geodesic distance. A ***Karcher mean*** of the sample is a local minimum $\mu \in M$ of V.

Back to Optimization

Gradient search strategies used for functions on $\mathbb{R}^n$ can be adapted to ***find minima*** of functions $f\colon M \longrightarrow \mathbb{R}$ defined on nonlinear manifolds. For example, the calculation of Karcher means can be approached with this technique. We discuss gradients next.

What is the Gradient of a Function on M?

The ***gradient*** of a function $F\colon M \longrightarrow \mathbb{R}$ at a point $p \in M$ is the unique vector $\nabla_M F(p) \in T_pM$ such that

$$dF_p(v) = \langle \nabla_M F(p), v \rangle \,,$$

for every $v \in T_pM$. ***How can it be computed in practice?*** Oftentimes, F is defined not only on M, but on the larger space $\mathbb{R}^m$ in which M is embedded. In this case, one first calculates

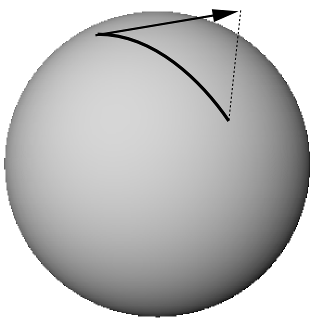
$$\nabla F(p) = \left(\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_1} \right) \in \mathbb{R}^m.$$

The orthogonal projection of this vector onto T_pM gives $\nabla_M F(p)$.

Gradient Search

At each $p \in M$, calculate $\nabla_M f(p) \in T_pM$. The goal is to search for minima of f by integrating the negative gradient vector field $-\boldsymbol{\nabla}_{\boldsymbol{M}} \boldsymbol{f}$ on M.

- **Initialize** the search at a point $p \in M$.
- **Update** p by infinitesimally **following the unique geodesic** starting at p with initial velocity $-\nabla_M f(p)$.
- **Iterate** the procedure.



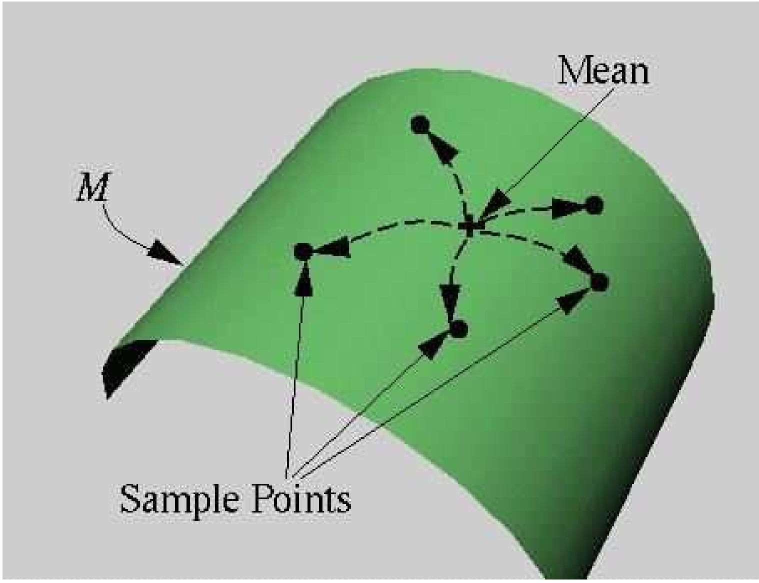
Remark. The usual convergence issues associated with gradient methods on $\mathbb{R}^n$ arise and can be dealt with in a similar manner.

Calculation of Karcher Means

If $\{x_1, \ldots, x_n\}$ are sample points in a manifold M, the negative gradient of the total variance function V is

$$\nabla_M V(x) = v_1(x) + \ldots + v_n(x),$$

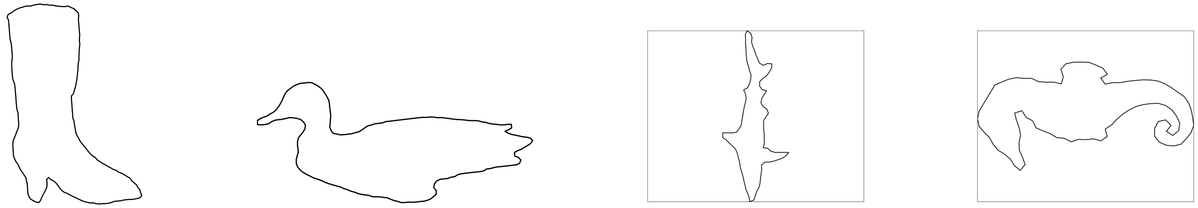
where $v_i(x)$ is the initial velocity of the geodesic that connects x to x_i in unit time.



Applications to the Analysis of Planar Shapes

To illustrate the techniques discussed, we apply them to the study of shapes of planar contours. We consider two different approaches, namely:

- The ***Procrustean Shape Analysis*** of Bookstein and Kendall.
- The ***Parametric-Curves Model*** of Klassen, Srivastava, Mio and Joshi.



In both cases, a ***shape*** is viewed as an ***element of a shape space***. ***Geodesics*** are used to quantify shape similarities and dissimilarities, interpolate and extrapolate shapes, and develop a statistical theory of shapes.

Procrustean Shape Analysis

A contour is described by a finite, ordered sequence of ***landmark points*** in the plane, say, $p_1, \ldots, p_n$. Contours that differ by ***translations, rotations and uniform scalings*** of the plane are to be viewed as having the same shape. Hence, shape representations should be insensitive to these transformations.

If μ be the centroid of the given points, the vector

$$x = (p_1 - \mu, \ldots, p_n - \mu) \in \mathbb{R}^{2n}.$$

is invariant to translations of the contour. This representation places the centroid at the origin. To account for uniform scaling, we normalize x to have unit length. Thus, we adopt the preliminary representation

$$y = x / \|x\| \in \mathbb{S}^{2n-1}.$$

In this y-representation, rotational effects reduce to ***rotations about the origin***. If $y = (y_1, \ldots, y_n)$, in complex notation, a $\boldsymbol{\theta}$-rotation of $y \in \mathbb{S}^{2n-1}$ about the origin is given by

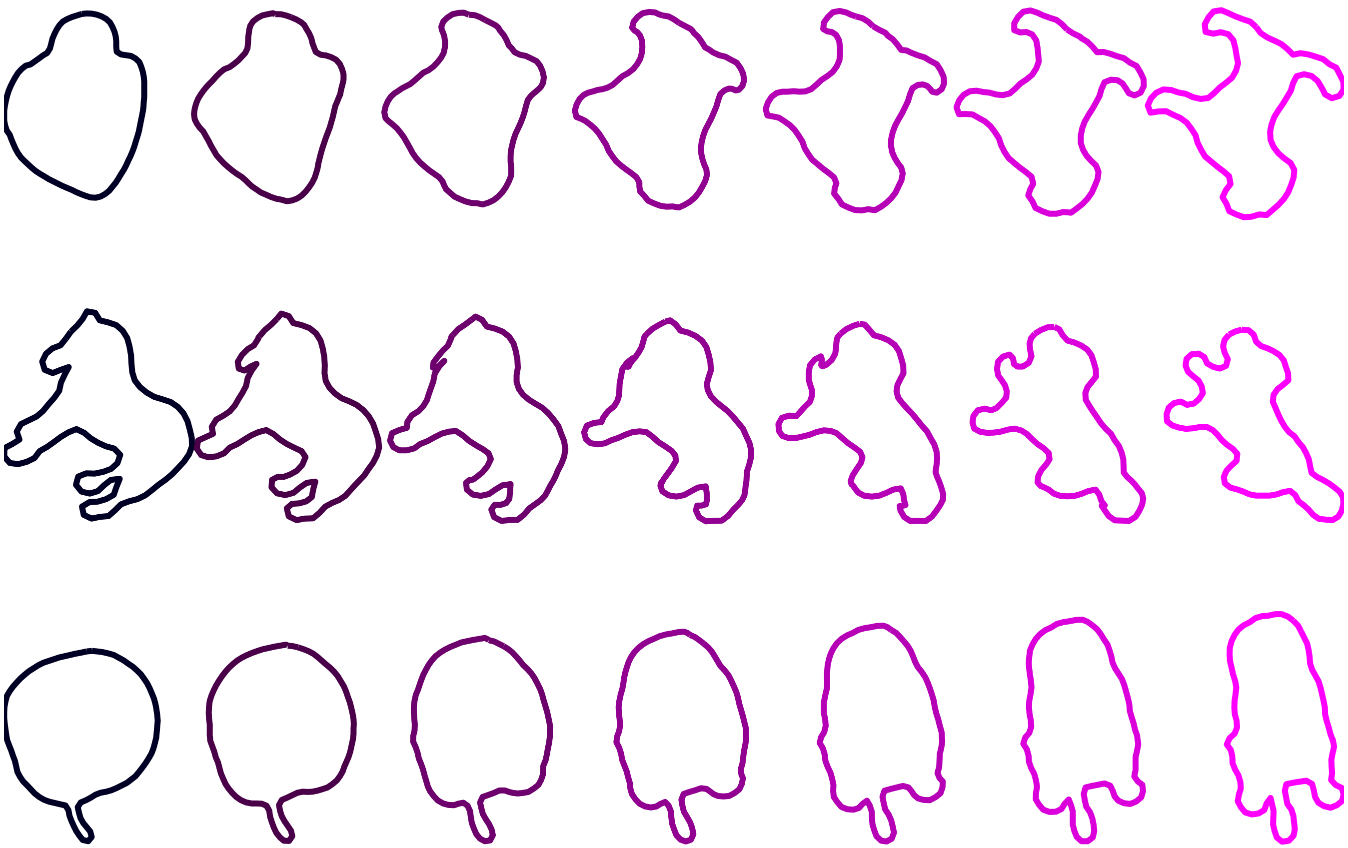
$$\boldsymbol{y \mapsto \lambda y = (\lambda y_1, \ldots, \lambda y_n),}$$

where $\lambda = e^{j\theta}$, where $j = \sqrt{-1}$.

In other words, vectors $\boldsymbol{y \in \mathbb{S}^{2n-1}}$ that differ by a unit complex scalar represent the same shape. Thus, the ***Procrustean shape space*** is the space obtained from $\mathbb{S}^{2n-1}$ by identifying points that differ by multiplication by a unit complex number. It is manifold of dimension $(2n-2)$ known as the ***complex projective space*** $\mathbb{C}\boldsymbol{P(n-1)}$.

Geodesics in $\mathbb{C}P(n-1)$ can be used to study shapes quantitatively and develop a statistical theory of shapes. This will be discussed in more detail in Part III.

Examples of Procrustean Geodesics



Parametric-Curves Approach

As a preliminary representation, think of a planar shape as a parametric curve $\boldsymbol{\alpha : I \longrightarrow \mathbb{R}^2}$ traversed with constant speed, where $I = [0, 1]$.

To make the representation invariant to uniform scaling, fix the length of α to be 1. This is equivalent to assuming that $\boldsymbol{\|\alpha'(s)\| = 1}$, $\forall s \in I$.

Since α is traversed with unit speed, we can write $\boldsymbol{\alpha'(s) = e^{j\theta(s)}}$, where $j = \sqrt{-1}$.

A function $\theta \colon I \longrightarrow \mathbb{R}$ with this property is called an ***angle function*** for α. Angle functions are insensitive to translations, and the effect of a rotation is to add a constant to θ.

To obtain shape representations that are ***insensitive to rigid motions*** of the plane, we fix the average of θ to be, say, π. In other words, we choose representatives that satisfy the constraint

$$\boldsymbol{\int_0^1 \theta(s)\, ds = \pi.}$$

We are only interested in closed curves. The ***closure condition***

$$\alpha(1) - \alpha(0) = \int_0^1 \alpha'(s)\, ds = \int_0^1 e^{j\theta(s)}\, ds = 0$$

is equivalent to the real conditions

$$\boldsymbol{\int_0^1 \cos\theta(s)\, ds = 0} \quad \text{and} \quad \boldsymbol{\int_0^1 \sin\theta(s)\, ds = 0}.$$

These are nonlinear conditions on θ, which will make the geometry of the shape space we consider interesting.

Let $\mathcal{C}$ the collection of all θ satisfying the three constraints above. We refer to an element of $\mathcal{C}$ as a pre-shape.

- Why call $\theta \in \mathcal{C}$ a **pre-shape instead of shape**?

This is because a shape may admit multiple representations in $\mathcal{C}$ due to the choices in the placement of the initial point ($s = 0$) on the curve. For each shape, there is a circle worth of initial points. Thus, our ***shape space*** $\mathcal{S}$ is the quotient of $\mathcal{C}$ by the action of the re-parametrization group $\mathbb{S}^1$; i.e., the space

$$\boldsymbol{\mathcal{S} = \mathcal{C}/\,\mathbb{S}^1},$$

obtained by identifying all pre-shapes that differ by a reparameterization.

The manifold $\mathcal{C}$ and the shape space $\mathcal{S}$ are ***infinite dimensional***. Although these spaces can be analyzed with the techniques we have discussed, we consider finite-dimensional approximations for implementation purposes.

We discretize an angle function $\theta\colon I \to \mathbb{R}$ using a uniform sampling

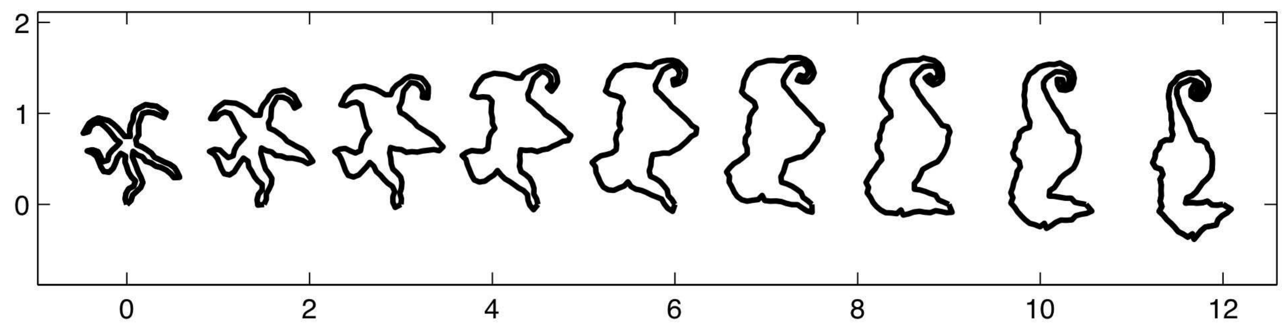
$$\boldsymbol{x = (x_1, \ldots, x_m) \in \mathbb{R}^m.}$$

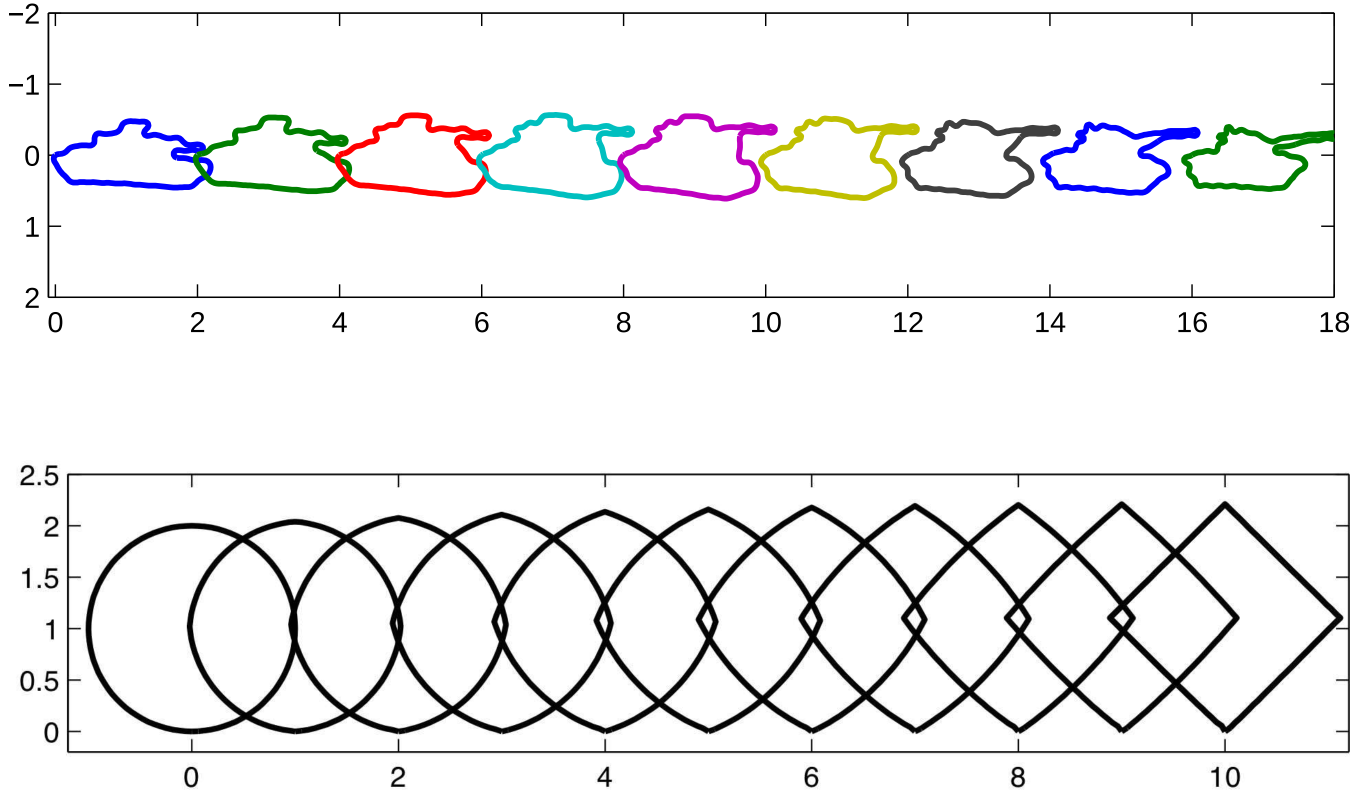
The ***three conditions*** on θ that define $\mathcal{C}$ can be rephrased as:

$$\begin{aligned} \int_0^1 \theta(s)\, ds = \pi \quad &\Longleftrightarrow \quad \frac{1}{m}\sum_{i=1}^m x_i = \pi \\ \int_0^1 \cos\theta(s)\, ds = 0 \quad &\Longleftrightarrow \quad \sum_{i=1}^m \cos(x_i) = 0 \\ \int_0^1 \sin\theta(s)\, ds = 0 \quad &\Longleftrightarrow \quad \sum_{i=1}^m \sin(x_i) = 0 \end{aligned}$$

Thus, the ***finite-dimensional analogue*** of $\mathcal{C}$ is a manifold $\mathcal{C}_m \subset \mathbb{R}^m$ of dimension $(m-3)$. Modulo 2π, a reparameterization of θ corresponds to a ***cyclic permutation*** of $(x_1, \ldots, x_m)$, followed by an adjustment to ensure that the first of the three conditions is satisfied. The quotient space by reparameterizations is the finite-dimensional version of the shape space $\mathcal{S}$.

Examples of Geodesics in $\mathcal{S}$




−2
−1
0
1
2
0
2
4
6
8
10
12
14
16
18
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10

CVPR 2004 TUTORIAL ON

NONLINEAR MANIFOLDS IN COMPUTER VISION

III. STATISTICS ON MANIFOLDS

Anuj Srivastava

Washington Mio and Xiuwen Liu

Center for Applied Vision and Imaging Sciences
@ Florida State University

Sunday, June 27^{th}, 1:00-5:00pm, Washington DC.

STATISTICS – OUTLINE

We are interested in extending usual techniques from statistics onto nonlinear manifolds: learning, estimation, clustering, testing.

We will cover the following topics:

- Probability distributions on manifolds of interest.
- Definitions and computations of intrinsic means and covariances. Extrinsic versus intrinsic analysis.
- Sampling and Monte Carlo estimation on manifolds. Statistical bounds on estimation.
- Learning and hypothesis testing on manifolds.

PROBABILITY DISTRIBUTIONS ON MANIFOLDS

- Probability densities on $\mathbb{R}^n$ are defined with respect to the Lebesgue measure. Similarly, probability measures on a manifold M is defined with respect to an invariant measure, e.g. Haar measure for Lie groups.
- (Radon-Nikodym) Derivatives of probability measures with respect to underlying invariant measures provide probability densities.
- Some examples of probability densities:
  - Unit circle S^1: Analog of Gaussian density:

$$f(\theta; \theta_0, a) = \frac{1}{I_0(a)} \exp(a \cos(\theta - \theta_0))$$

    is the von Mises density. θ_0 is the mean angle, and a is a measure of the variance. I_0 is a modified Bessel function of zeroth order.

- Special Orthogonal group $SO(n)$:

$$f(O|A) = \frac{1}{Z}\exp(trace(OA^T))$$

for some $A \in \mathbb{R}^{n\times n}$. Let $A = \bar{O}P$ be the polar decomposition of A, then $\bar{O}$ is the mean of f and P relates to the variance.

- Grassmann manifold $\mathcal{G}_{n,d}$: Let $U \in \mathcal{G}_{n,d}$ be represented by an $n \times d$ (tall-skinny) orthogonal matrix.

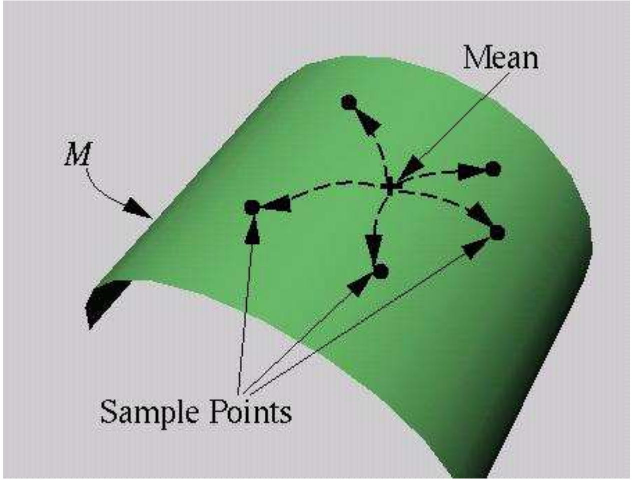
$$f(U) = \frac{1}{Z}\exp(trace(UAU^T))$$

for some symmetric, positive definite matrix $A \in \mathbb{R}^{n\times n}$. SVD of A gives mean and variance of f.

A more general form is given later.

STATISTICS ON MANIFOLDS

Definition and estimation of mean on a manifold.



For a Riemannian manifold M, let $d(p_1, p_2)$ be the Riemannian distance between p_1 and p_2 in M. Also, let M be embedded inside $\mathbb{R}^n$ for some n, and let $\|p_1 - p_2\|$ denote the Euclidean distance after embedding.

Mean under a probability density function $f(p)$:

- Intrinsic mean:

$$\hat{p} = \underset{p \in M}{\operatorname{argmin}} \int_M d(p,u)^2 f(u)\gamma(du) \; ,$$

- Extrinsic mean:

$$\hat{p} = \underset{p \in M}{\operatorname{argmin}} \int_M \|p-u\|^2 f(u)\gamma(du) \; ,$$

  This can also be viewed as computing the mean in $\mathbb{R}^n$ and then projecting it back to M. Extrinsic analysis implies emedding the manifold in a larger Euclidean space, computing the estimate there, and projecting the solution back on the manifold.

  The solution may depend upon the choice of embedding.

EXAMPLE OF MEAN ESTIMATION ON A CIRCLE

Given $\theta_1, \theta_2, \ldots, \theta_n \in S^1$.

1. Intrinsic Mean:

$$d(\theta_1, \theta_2) = min\{|\theta_1 - \theta_2|, |\theta_1 - \theta_2 + 2\pi|, |\theta_1 - \theta_2 - 2\pi|$$

   Then, find the mean

$$\hat{\theta} = \mathrm{argmin}\,\theta \sum_{i=1}^{n} d(\theta, \theta_i)^2 \ .$$

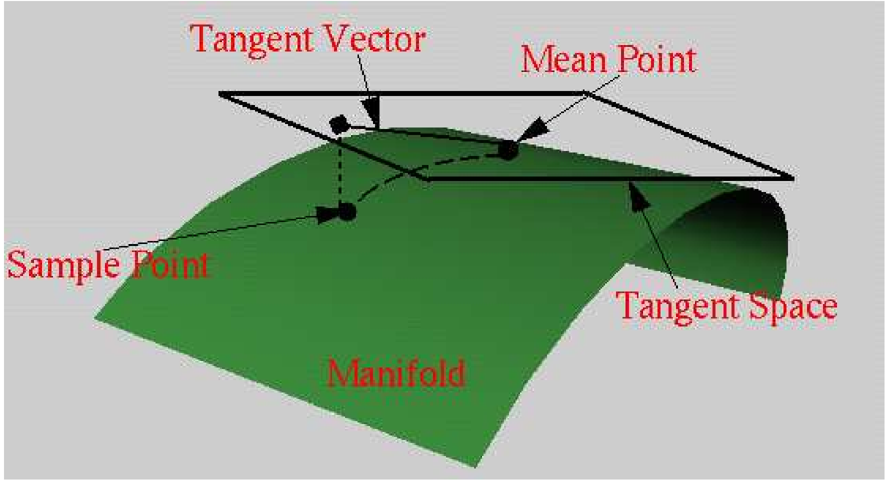
   Solve this problem numerically.

2. An Extrinsic Mean: Let $z_i = e^{j\theta_i} \in \mathbb{C}$, $i = 1, \ldots, n$. Then, $\hat{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$, and set $\hat{\theta} = arg(\hat{z})$.

STATISTICS ON MANIFOLDS

Covariance under a probability density function $f(p)$:

- Let $\exp_p : T_P(M) \mapsto M$ be the exponential map, computed using geodesic defined earlier.
- Let $\hat{p}$ be the mean, and $T_{\hat{p}}(M)$ be the tangent space at $\hat{p}$. Then, let $\tilde{f}$ be the probability density induced on $T_{\hat{p}}(M)$ using the inverse of exponential map.



- Similarly one can compute higher monments.

Define the covariance Σ:

$$\Sigma = \int x x^T \tilde{f}(x) dx \quad \in \mathbb{R}^{m \times m}$$

where $m = dim(M)$ and $x = \exp_{\hat{p}}^{-1}(p)$.

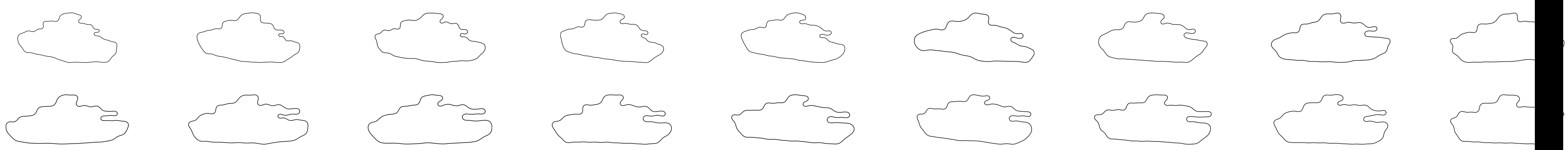
(Essentially, one computes the covariance matrix in the tangent space.)

- A simple example of $\tilde{f}$ is multivariate normal in the tangent space at $\hat{p}$.

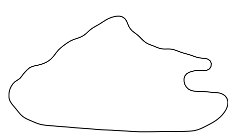
LEARNING PROBABILITY MODELS FROM OBSERVATIONS

Example: Assume a multivariate normal model in the tangent space at mean.

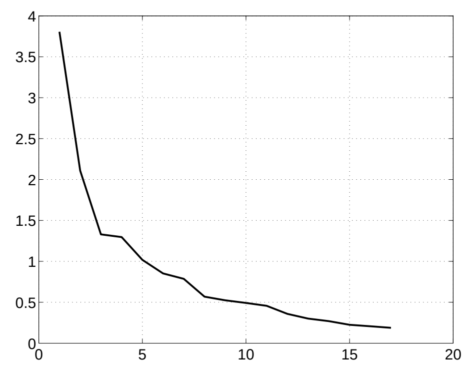
Observed shapes:



Sample statistics of observed shapes:



Mean Shape



Eigen Values of Covariance matrix

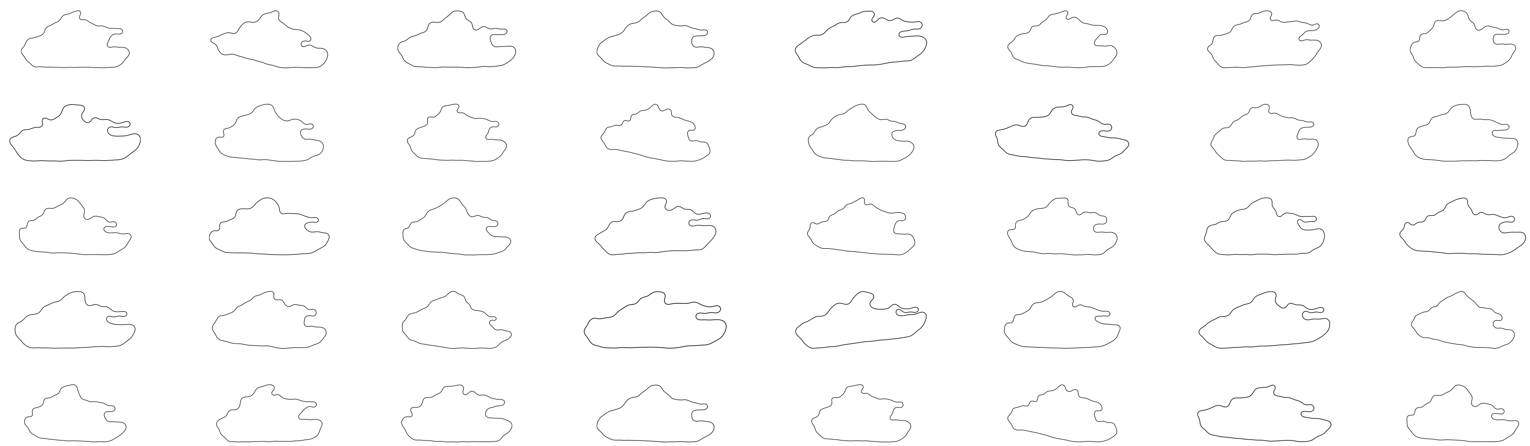
LEARNING PROBABILITY MODELS FROM OBSERVATIONS

Synthesis:

(i) Generate samples of x, multivariate normal on the tangent space, and

(ii) Use $\exp_{\hat{p}}(x)$ to generate samples on M.

Examples: Samples from multivariate normal model on tangent shape space.



CVPR 2004 TUTORIAL ON

NONLINEAR MANIFOLDS IN COMPUTER VISION

IV. ALGORITHMS AND EXAMPLES

Xiuwen Liu

Anuj Srivastava and Washington Mio

CAVIS
Center for Applied Vision and Imaging Sciences
@ Florida State University

Algorithms and Examples

Here we focus on computational issues of implementing algorithms based on nonlinear manifolds. Live demonstrations will be given to show the feasibilities, effectiveness and efficiency.

We use two examples for illustrations.

- Optimal Component Analysis.
- Computation of geodesics on shape manifolds and shape clustering

Optimal Component Analysis

Optimal component analysis (OCA) is a framework to formulate computer vision problems based on linear representations on proper manifolds

- Define optimization criteria.
- Identify the underlying manifolds, such as Grassmann or Stiefel manifold.
- Design optimization algorithm utilizing the geometric properties of the manifold.

While there exist several standard component analysis techniques, a critical difference between OCA and these standard techniques is that OCA provides a general procedure in that it can be used to solve different kinds of problems

Commonly Used Linear Representation Manifolds

Paramete-rization	Manifold	Search space	Examples
Subspace	Grassmann $\mathcal{G}_{n,d}$	$d(n-d)$	Optimal subspace/ Principal/minor components
Orthonormal basis	Stiefel $\mathcal{S}_{n,d}$	$d(n-\frac{d+1}{2})$	Orthonormal filters
Directions	Direction manifold (d Grassmann $\mathcal{G}_{n,1}$)	$d(n-1)$	Optimal filters / Independent components
Vectors	Euclidean	dn	Linear basis with weights

Optimal Linear Subspaces

When the performance does not depend on the choice of basis but only the subspace, the underlying solution space is Grassmann.

To be concrete, here we use recognition for illustration. Let there be C classes to be recognized from the images; each class has k_{train} training images (denoted by $I_{c,1}, \ldots, I_{c,k_{train}}$) and k_{cr} cross validation (denoted by $I'_{c,1}, \ldots, I'_{c,k_{cr}}$).

Optimal Linear Subspaces -cont.

For recognition, we define the following criterion,

$$\rho(I'_{c,i}, U) = \frac{\min_{c' \neq c, j} D(I'_{c,i}, I_{c',j}; U)}{1/|\tau(I'_{c,i}, K)| \sum_{j \in \tau(I'_{c,i}, K)} D(I'_{c,i}, I_{c,j}; U) + \epsilon_0}, \qquad (1)$$

where $\tau(I'_{c,i}, K)$ is the set of images of K closest to $I'_{c,i}$ but in class c $D(I_1, I_2; U) = \|a(I_1, U) - a(I_2, U)\|$, and $\epsilon_0 > 0$ is a small number to avoid division by zero. Then, define R according to:

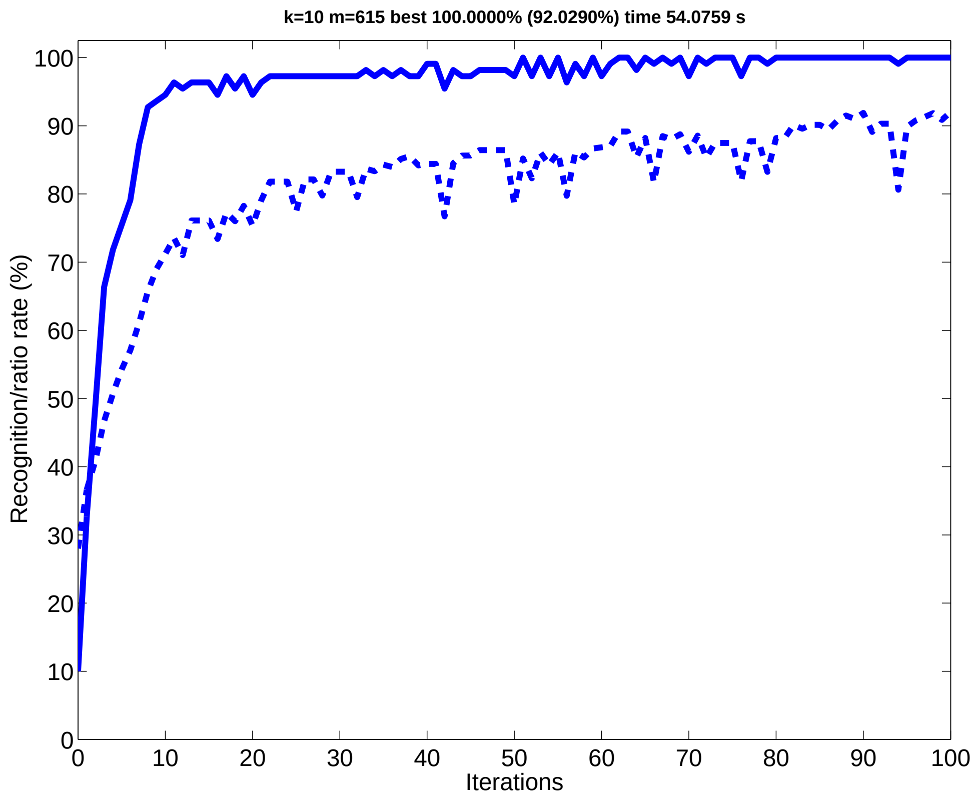
$$R(U) = \frac{1}{Ck_{cr}} \sum_{c=1}^{C} \sum_{i=1}^{k_{cr}} h(\rho(I'_{c,i}, U) - 1), \quad 0 \leq R \leq 1.0 \qquad (2)$$

where $h(\cdot)$ is a monotonically increasing and bounded function. In our experiments, we have used $h(x) = 1/(1 + \exp(-2\beta x))$, where β controls the degree of smoothness of R.

Optimal Linear Subspaces -cont.

Live Demonstration One: Recognition using part of the CMPU PIE dataset

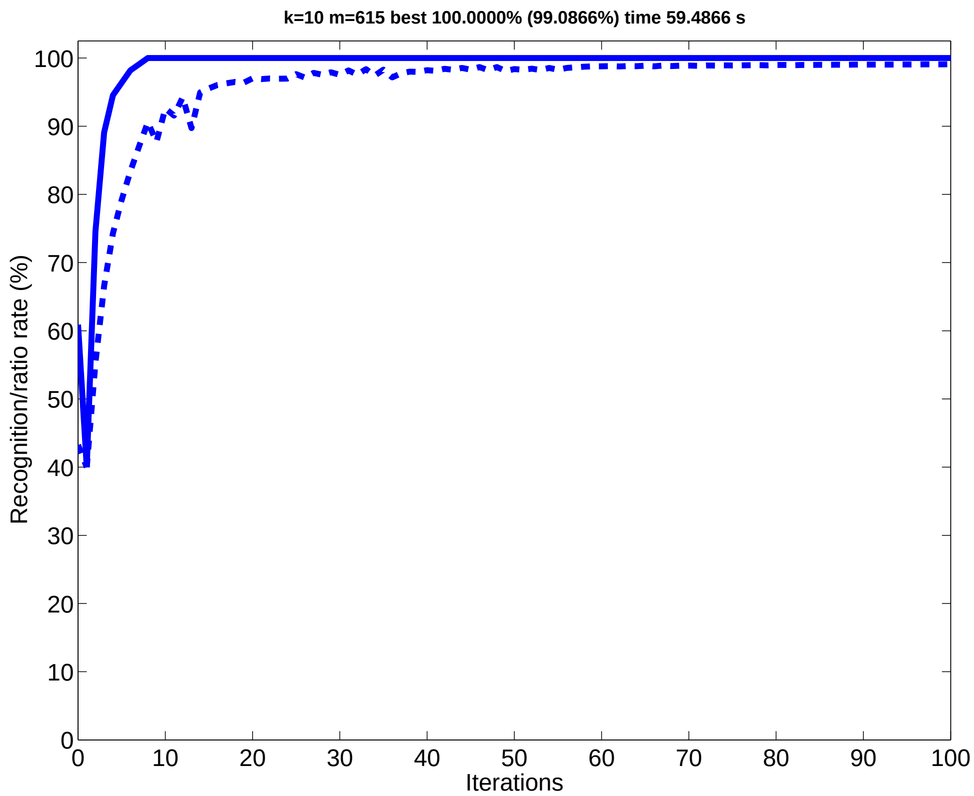
- $C = 10.$
- $n = 25 \times 25.$
- $d = 10.$
- Initial condition: ICA.
- 54s for 100 iterations



Optimal Linear Subspaces -cont.

Live Demonstration Two: Recognition using part of the CMPU PIE dataset

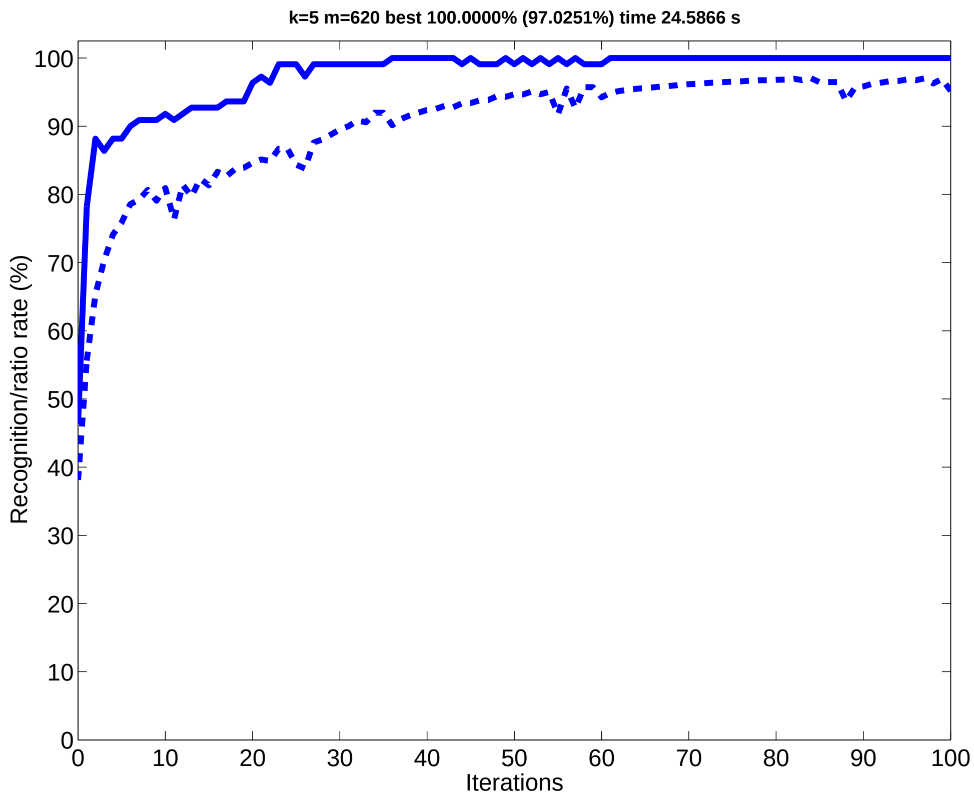
- $C = 10$.
- $n = 25 \times 25$.
- $d = 10$.
- Initial condition: Axis.
- 1m for 100 iterations



Optimal Linear Subspaces -cont.

Live Demonstration Three: Recognition using part of the CMPU PIE dataset

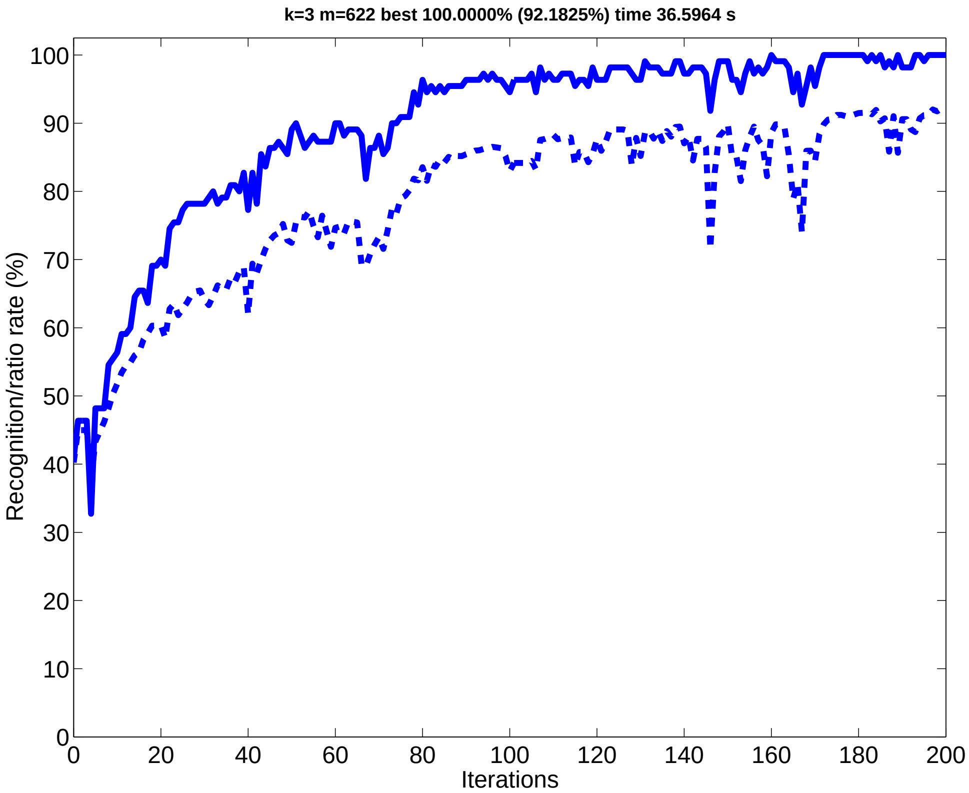
- $C = 10$.
- $n = 25 \times 25$.
- $d = 5$.
- Initial condition: Axis.
- 25s for 100 iterations



Optimal Linear Subspaces -cont.

Live Demonstration Four: Recognition using part of the CMPU PIE dataset

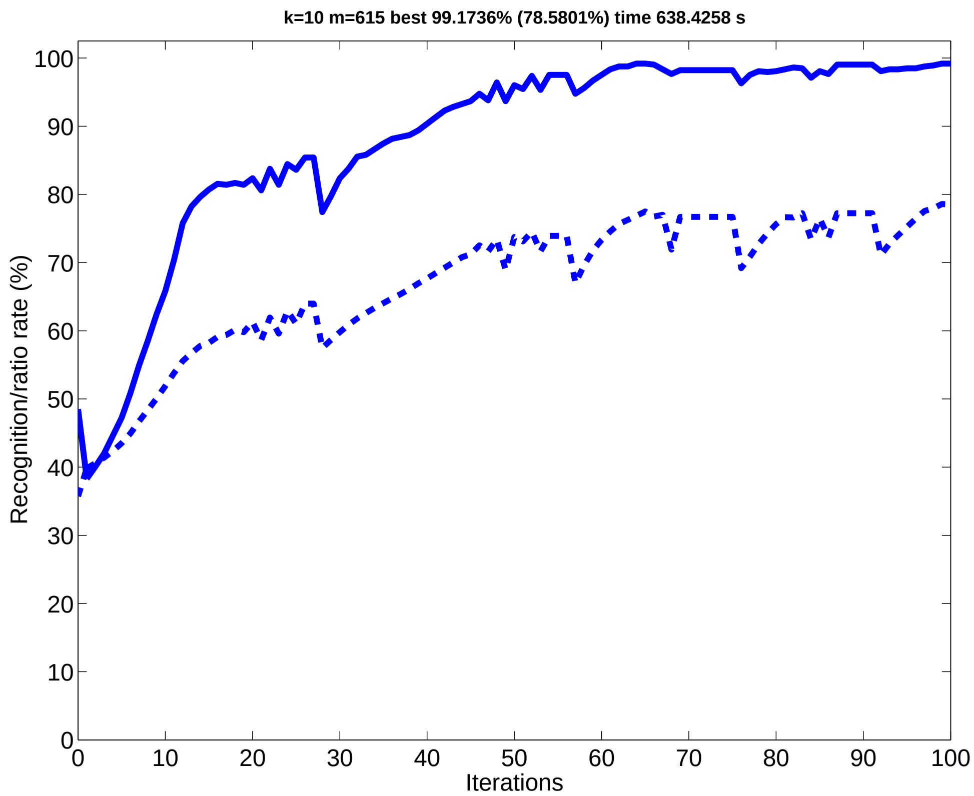
- $C = 10$.
- $n = 25 \times 25$.
- $d = 3$.
- Initial condition: Axis.
- 36.5s for 200 iterations



Optimal Linear Subspaces -cont.

Offline Demonstration: Recognition using part of the CMPU PIE dataset

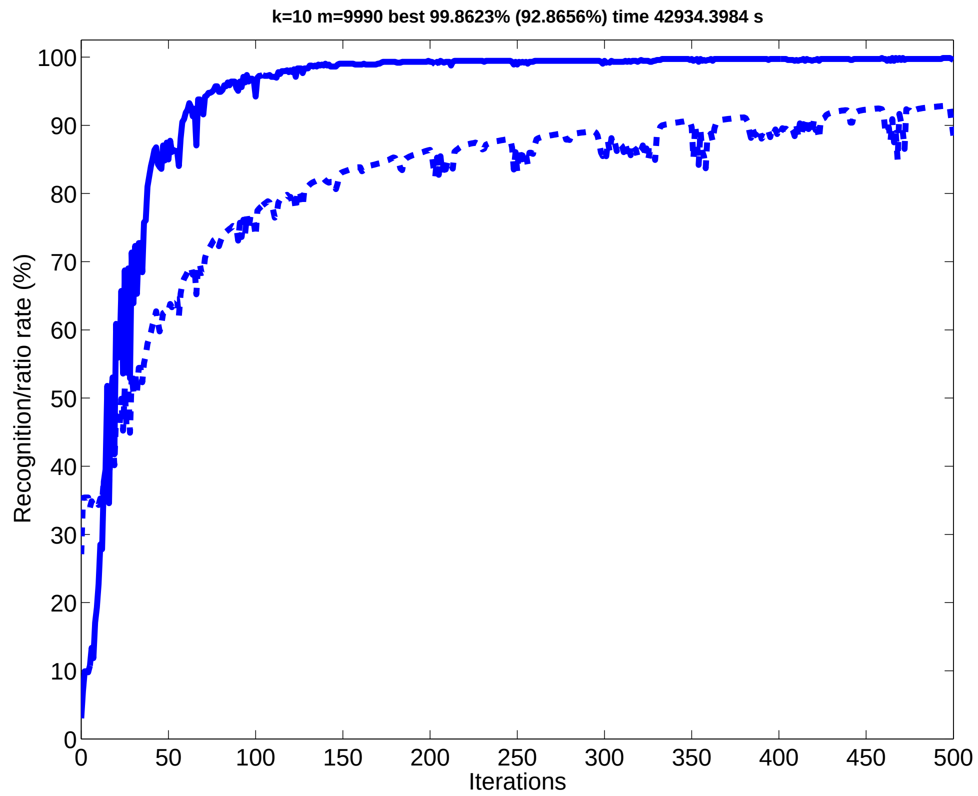
- $C = 66$.
- $n = 25 \times 25$.
- $d = 10$.
- Initial condition: Axis.
- 10m for 100 iterations



Optimal Linear Subspaces -cont.

Offline Demonstration: Recognition using part of the CMPU PIE dataset

- $C = 66$.
- $n = 100 \times 100$.
- $d = 10$.
- Initial condition: Axis.
- 42934s for 500 iterations

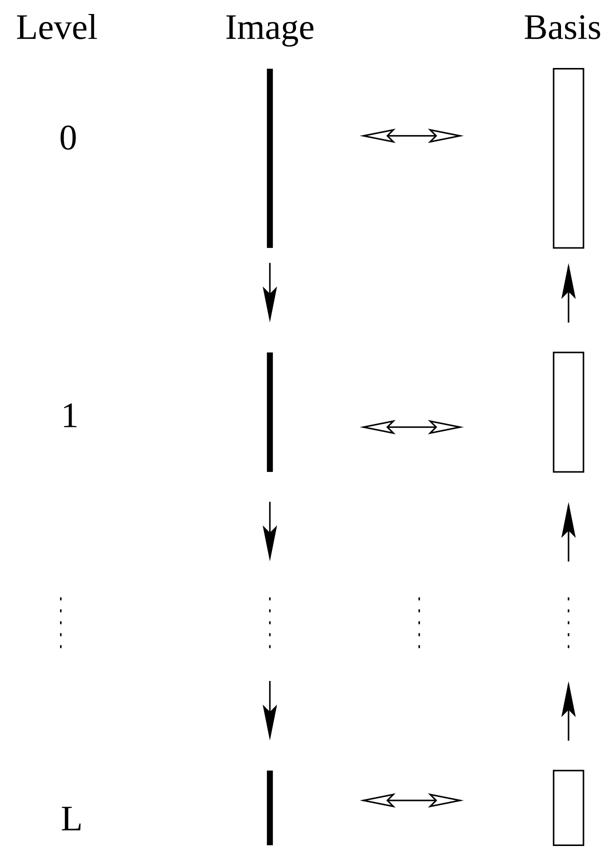


Further Speedup Techniques

We have developed techniques to further speed up the optimization process.

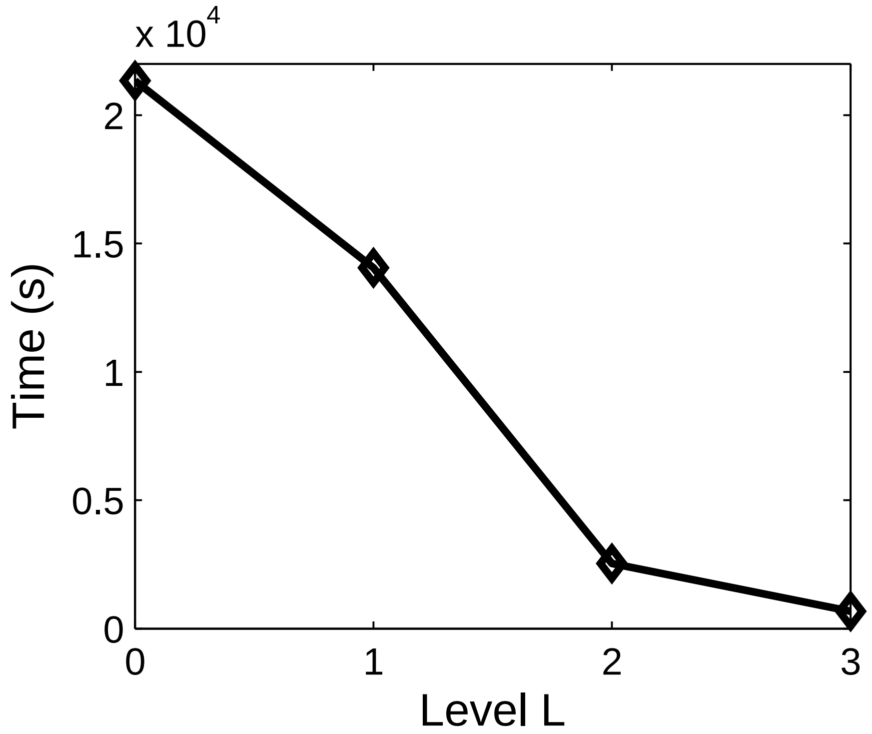
A particularly effective technique is called hierarchical learning based on some heuristics to decompose the process into layers.

Hierarichal Learning



Hierarichal Learning -cont.

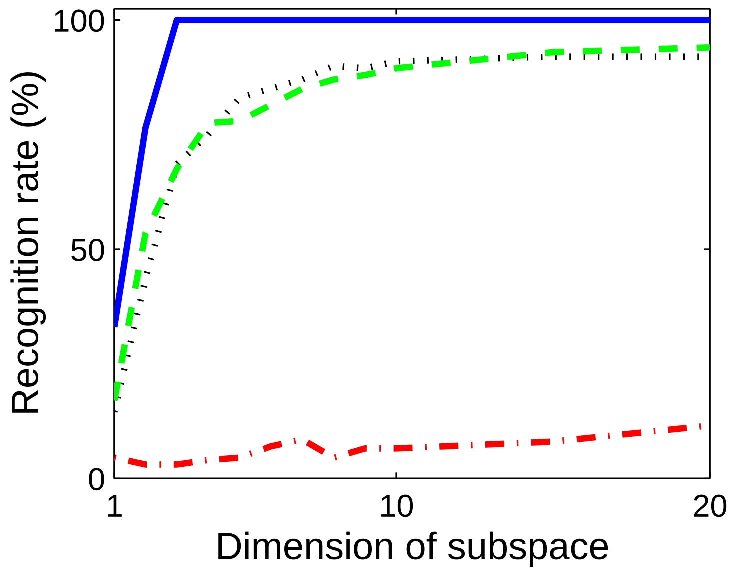
We have applied hierarchical learning on CMPU PIE and ORL datasets. We typically speed up the process by a factor of 30.

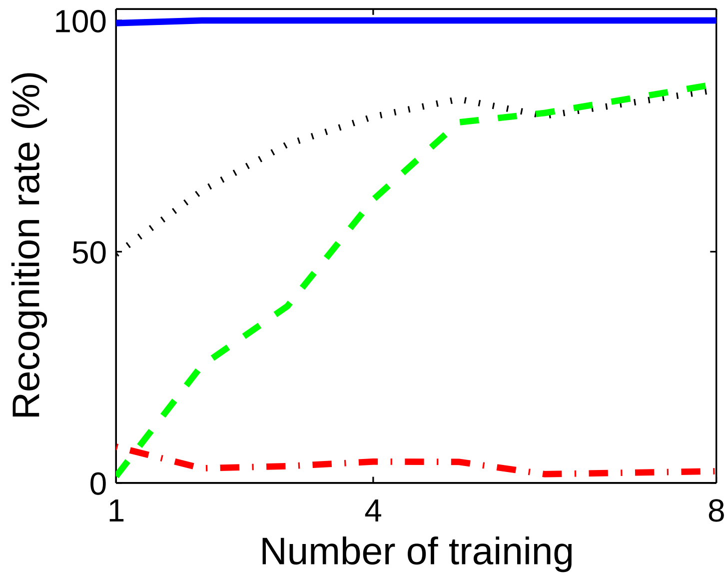


Optimal Linear Subspaces -cont.

Some experimental results

PCA - Black; ICA - Red; FDA - Green; OCA - Blue



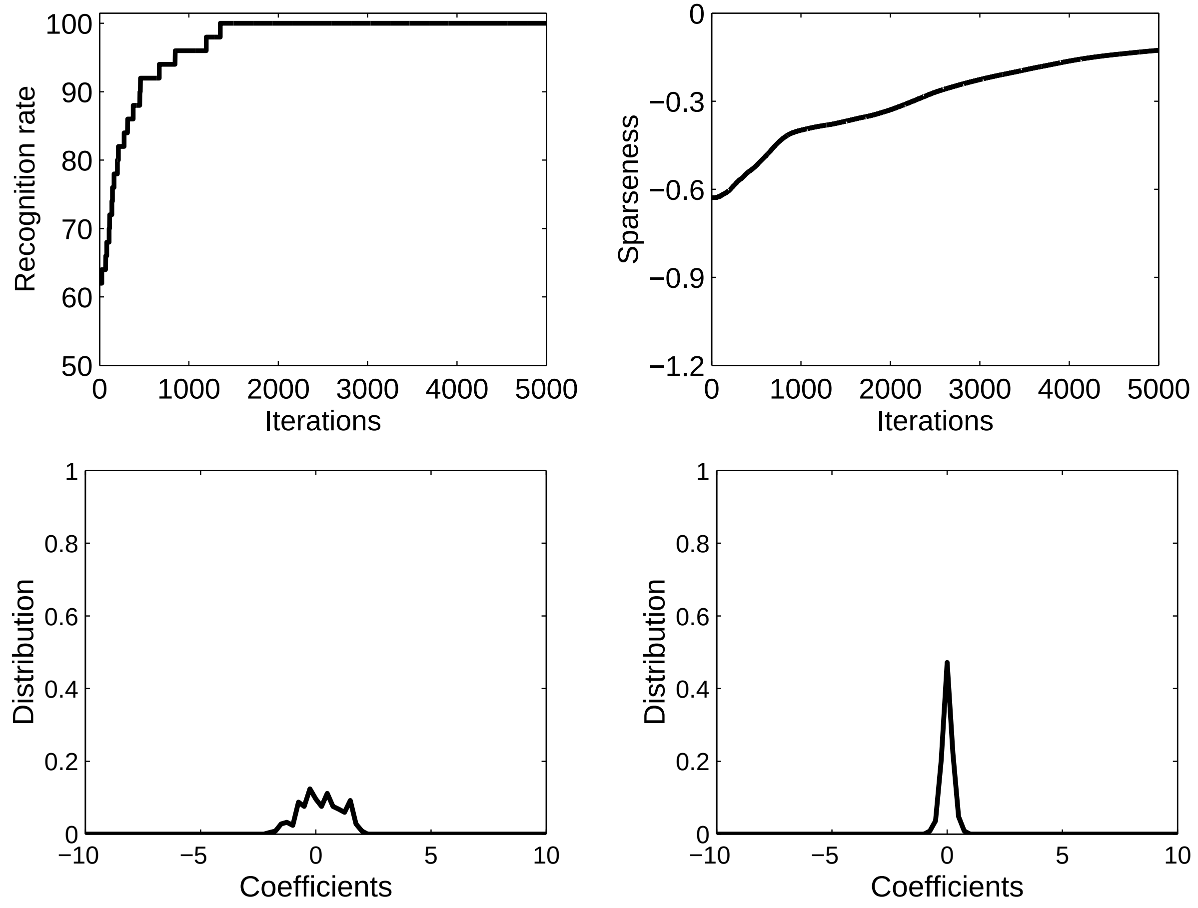


Other Performance Functions

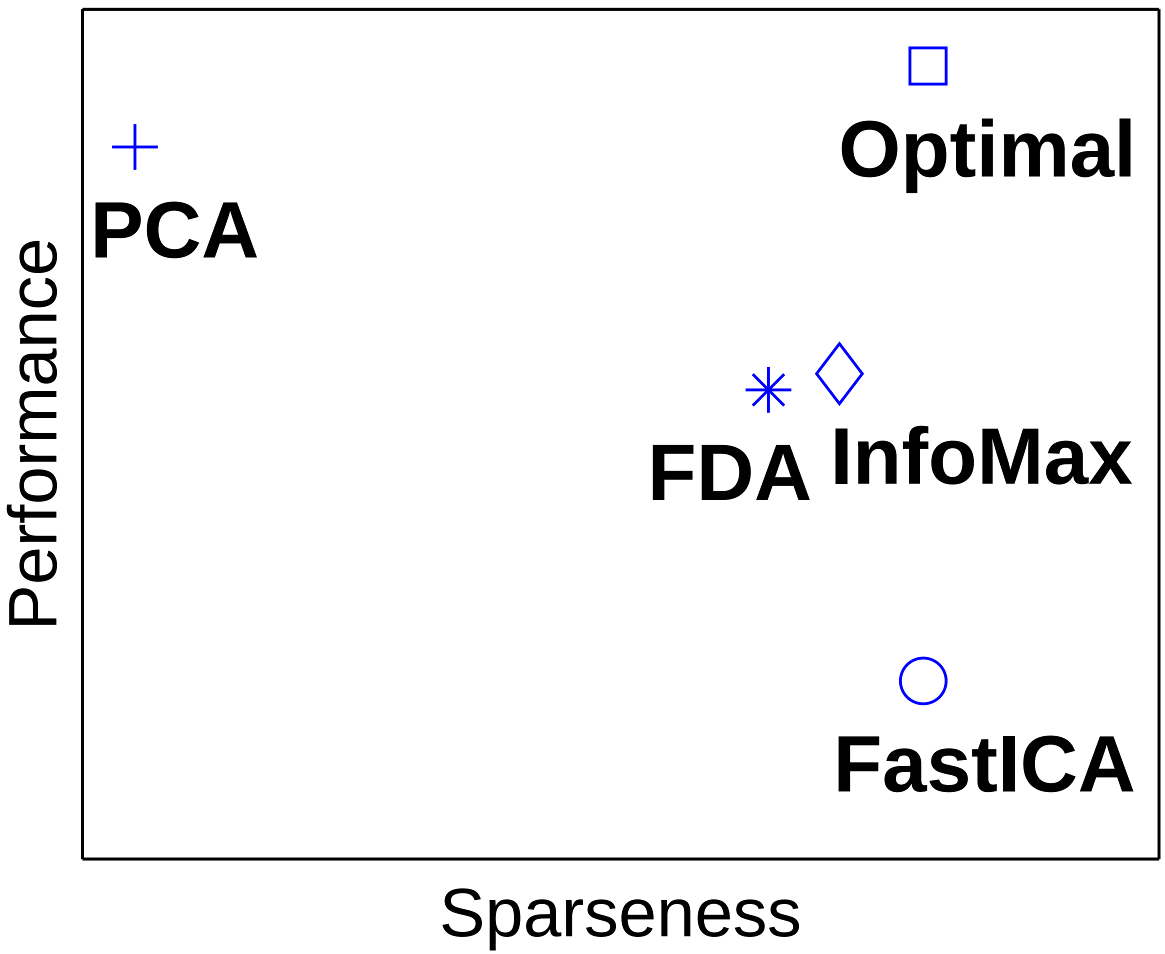
Note that this framework applies to any performance functions.

To derive linear representations that are sparse and effective for recognition, we have used a linear combination of sparseness and recognition performance as the performance function.

Sparse Linear Representations for Recognition

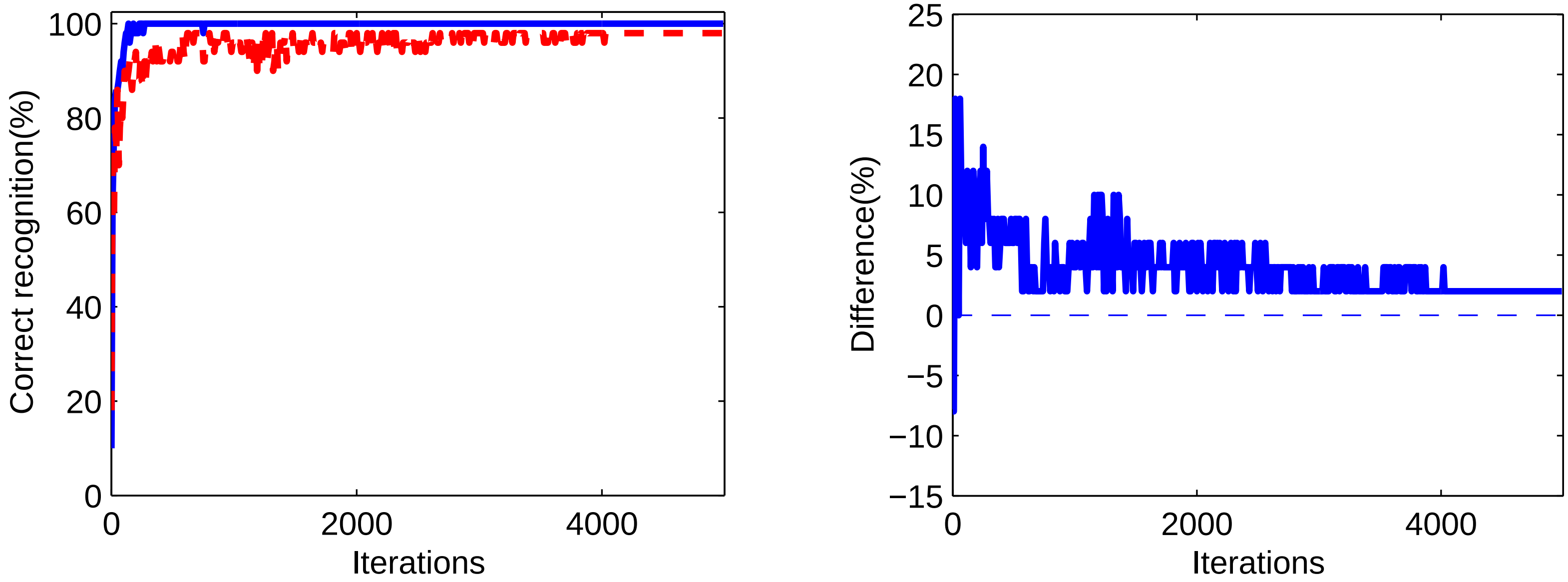


Comparison of Sparseness and Recognition

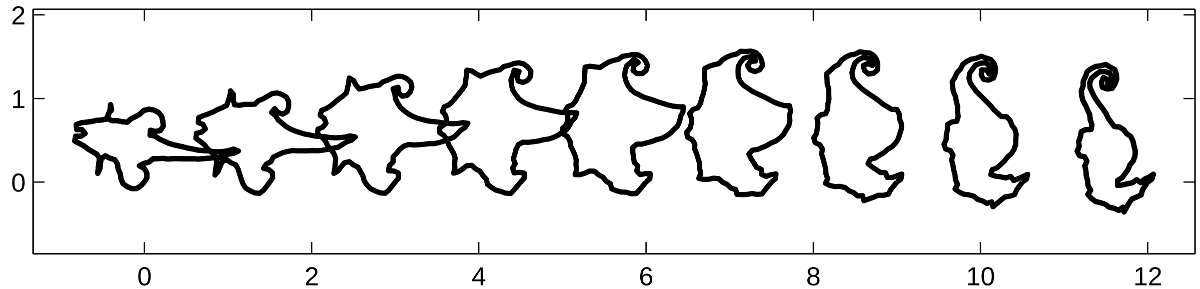


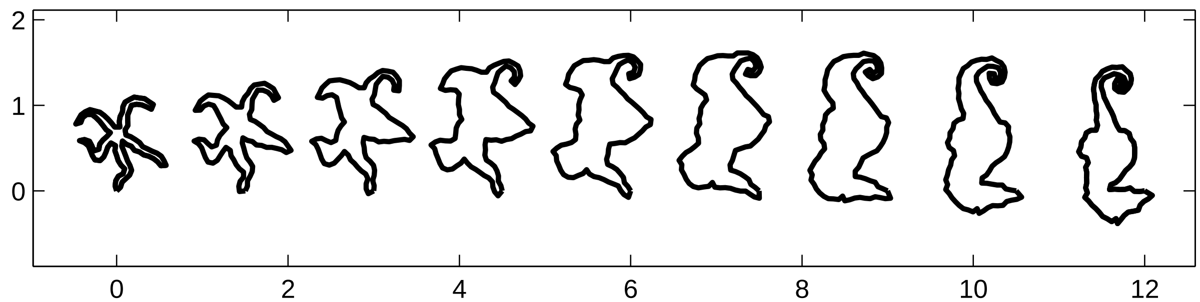
Discussion

Note that OCA provides a framework to formulate and solve problems. In computer vision, a central issue is generalization, which is determined by the performance function you use. The performance we use here is loosely connected to large margin idea and often give good generalization performance.



Geodesic Path Examples on Shape Manifold





Geodesic Computation and Shape Clustering

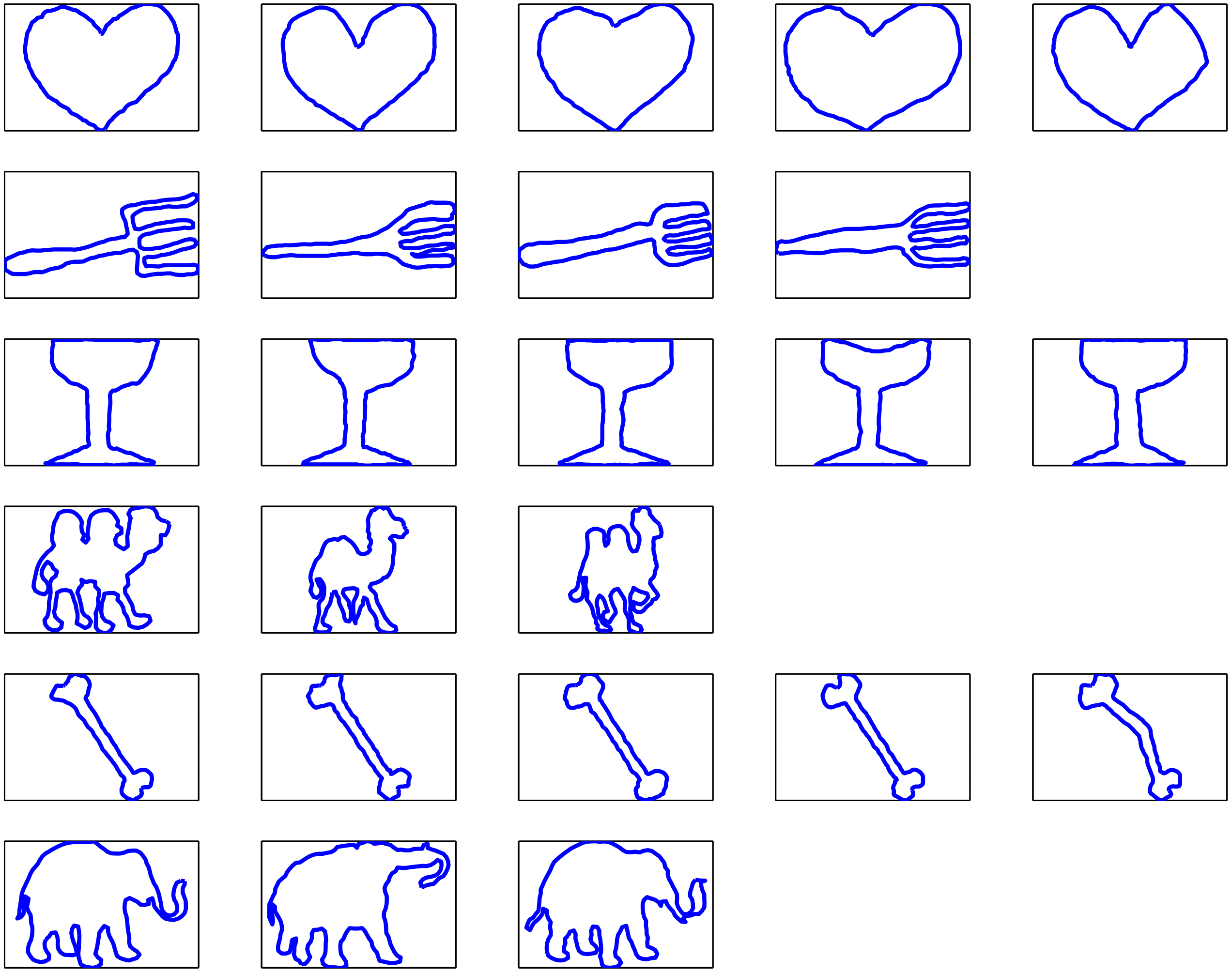
Live Demonstration Five: Geodesic Computation and Shape Clustering.

Here we show how fast one can compute geodesics among shapes and then perform clustering based on the pair-wise geodesic distances.

For a small database of 25 shapes, the pair-wise geodesic (300) computation takes 13 seconds in C and the clustering takes 10 seconds in Matlab (A C implementation would improve by a factor 5-10).

For a larger database of 300 shapes, the pair-wise geodesic (44850) computation takes 600 seconds and the clustering takes 170 seconds in Matlab.

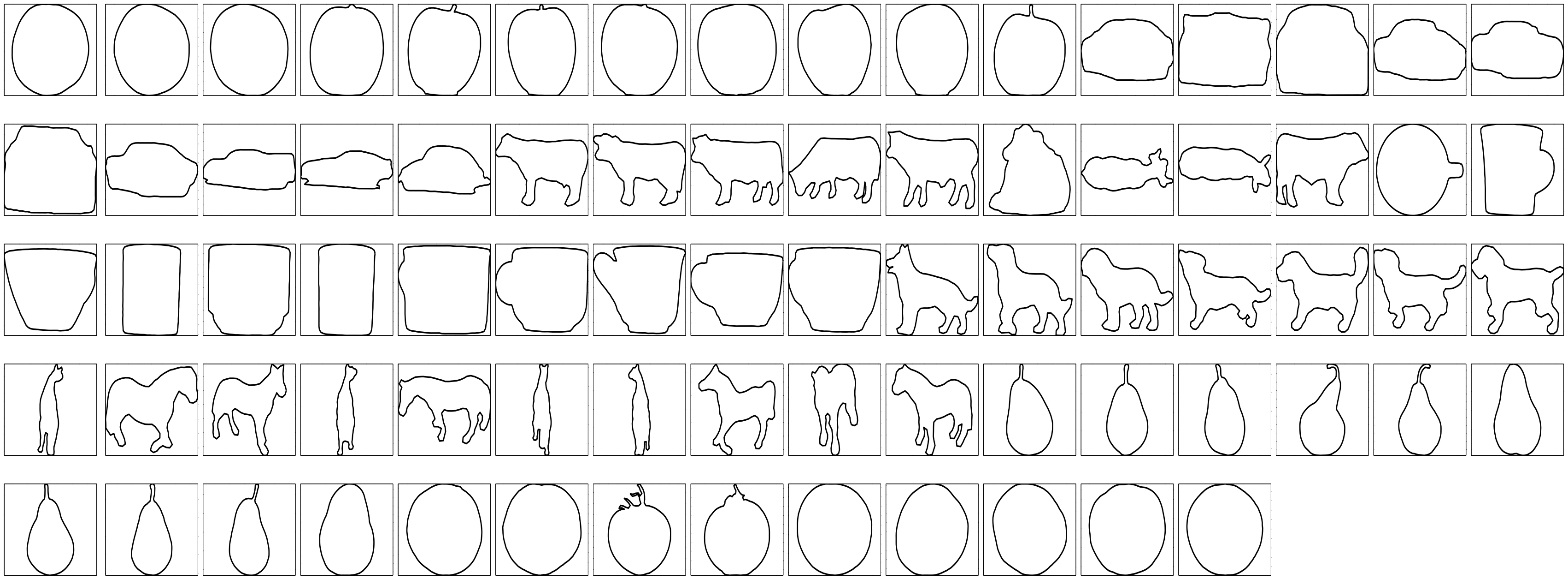
Shape Clustering -cont.

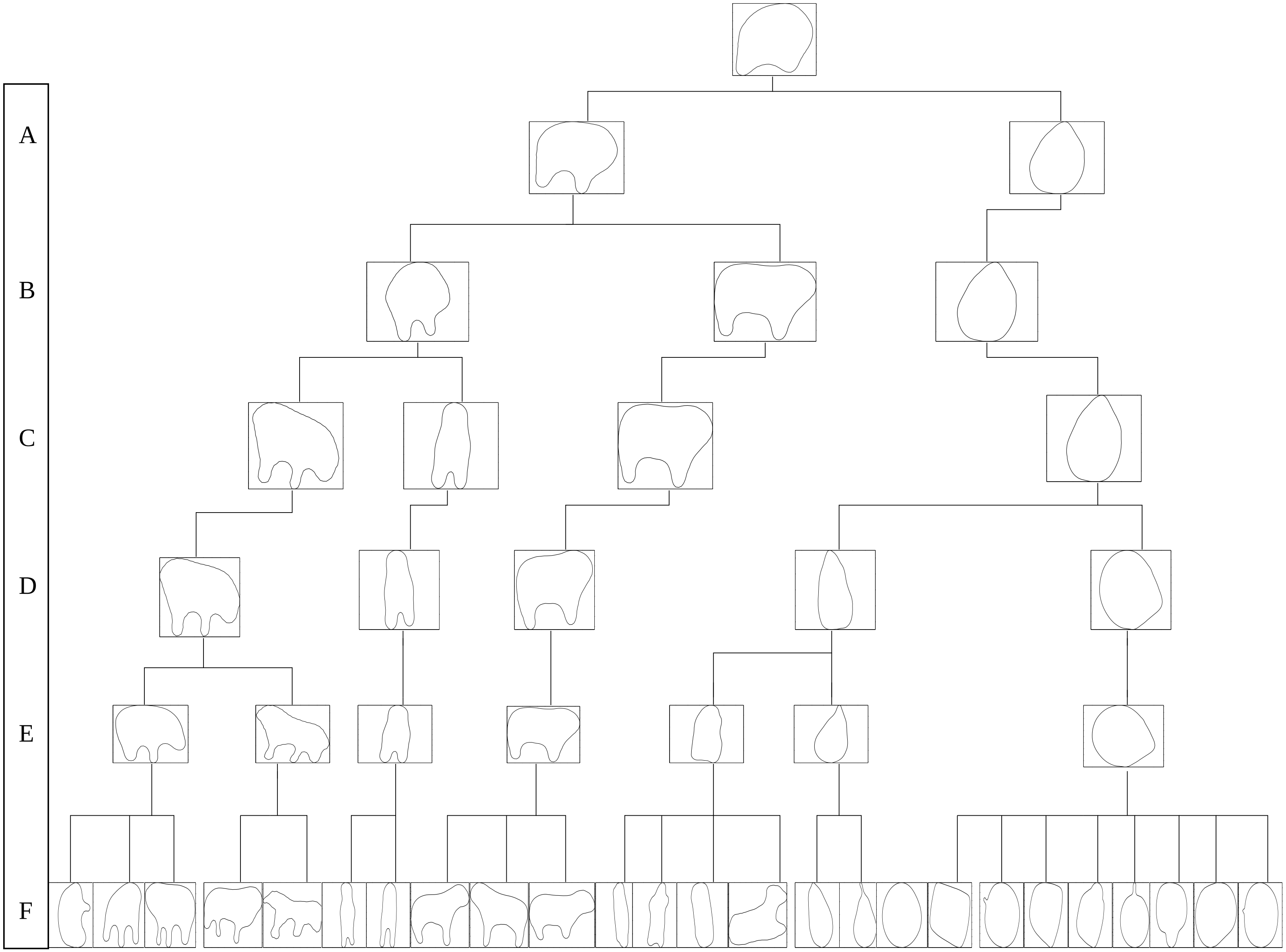


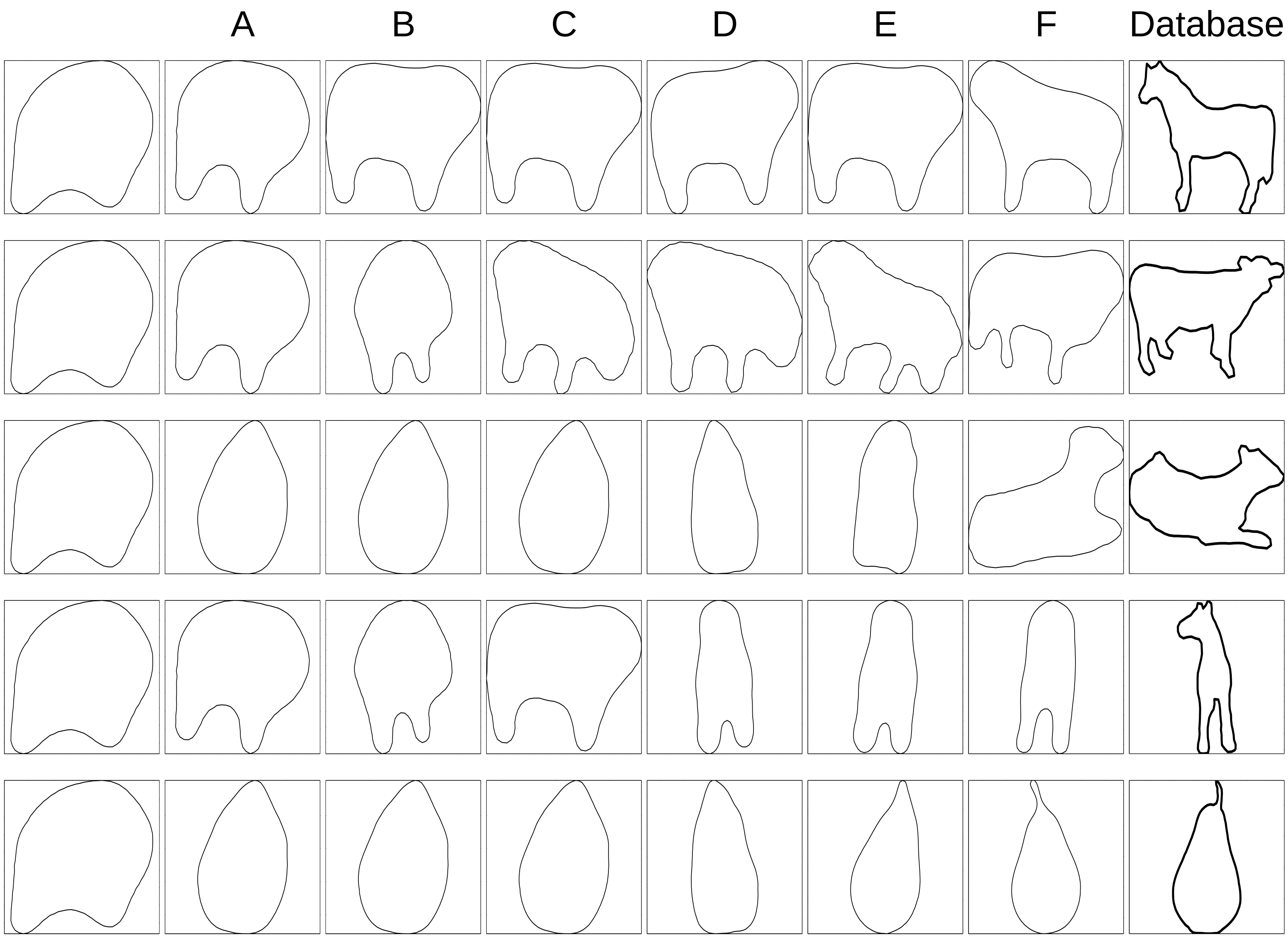
Shape Clustering -cont.

Hierarchical Shape Clustering

We have clustered 3000 shapes using this method. This figure shows the first few layers in the resulting shape hierarchy.




A
B
C
D
E
F


A
B
C
D
E
F
Database

A Brief Summary of Part IV

- By utilizing geometric structures of underlying manifolds, we have developed effective optimization algorithms that can be implemented efficiently.
- Note that the computation here can be done offline.
- We hope techniques along this line of research will be developed for many other computer vision problems.

THANK YOU!

References

[1] W. M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, Inc., 1986.

[2] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Son, 1998.

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

[4] U. Grenander, M. I. Miller, and A. Srivastava. Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR. *IEEE Transactions on PAMI*, 20(8):790–802, 1998.

[5] H. Hendricks. A Cramer-Rao type lower bound for estimators with values in a manifold. *Journal of Multivariate Analysis*, 38:245–261, 1991.

[6] K. Kanatani. *Geometric Computation for Machine Vision*. Clarendon Press, Oxford, 1993.

[7] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machiner Intelligence*, 26(3):372–383, March, 2004.

[8] X. Liu, A. Srivastava, and K. Gallivan. Optimal linear representations of images for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):662–666, 2004.

[9] K. V. Mardia. *Statistics of Directional Data*. Academic Press, 1972.

[10] J. M. Oller and J. M. Corcuera. Intrinsic analysis of statistical estimation. *Annals of Statistics*, 23(5):1562–1581, 1995.

[11] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[12] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290:2268–2269, 2000.

[13] C. G. Small. *The Statistical Theory of Shape*. Springer, 1996.

[14] Michael Spivak. *A Comprehensive Introduction to Differential Geometry, Vol I & II*. Publish or Perish, Inc., Berkeley, 1979.

[15] A. Srivastava and E. Klassen. Monte Carlo extrinsic estimators for manifold-valued parameters. *IEEE Trans. on Signal Processing*, 50(2):299–308, February 2001.

[16] A. Srivastava and E. Klassen. Bayesian, geometric subspace tracking. *Journal for Advances in Applied Probability*, 36(1):43–56, March 2004.

[17] J. B. Tenenbaum, V. Silva, , and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.