Manifold Reconstruction from Unorganized Points

Daniel Freedman
Rensselaer Polytechnic Institute
Department of Computer Science
Troy, NY, USA 12180
freedman@cs.rpi.edu

Abstract

A new algorithm for manifold reconstruction is presented. The goal is to take samples drawn from a finite-dimensional manifold, and to reconstruct a manifold, based only on the samples, which is a good approximation to the true manifold; nothing of the true manifold's geometry or topology is known a priori. The algorithm constructs a simplicial complex based on approximating tangent hyperplanes to the manifold, and does so efficiently. Successful examples are presented for curve reconstruction in the plane, curve reconstruction in space, and surface reconstruction in space.

1 Introduction

This paper is concerned with the problem of *manifold reconstruction*. This is a problem in computational geometry/topology which may be posed as follows:

Manifold Reconstruction Problem

Given a dense sampling X of some K-dimensional compact C^1 manifold F which is embedded in a Hilbert space Z, construct a K-dimensional compact manifold R which approximates F well.

Such a formulation is not complete, in the sense that many concepts such as "good approximation" remain vague; these issues will be cleared up shortly. For the moment, let us focus on practical applications of the manifold reconstruction problem, as well as some of the difficulties inherent in finding a solution.

Manifold reconstruction is important in several areas. First, curve reconstruction schemes are often useful in computer vision and image processing, where one is presented with a series of unorganized edge-points, and wishes to join these points into a curve. Surface reconstruction is useful in CAD and computer graphics, where points in $\mathbb{R}^3$ are given, and the desired output is a surface. An application is "reverse engineering," where a machine part is scanned with a laser range finder, and a model of the object's geometry is automatically generated. Finally, the general problem of manifold reconstruction may be construed as a particular form of learning. For example, one may wish to learn the set of all possible curves corresponding to the outlines of a particular object; in many cases, this set will correspond to a manifold in curve-space. Thus, the goal is to learn a finite-dimensional manifold (embedded, in this case, in an infinite-dimensional space) from training examples.

Having posed the problem, some of its challenges may now be discussed. The key problem faced by the designer of a manifold reconstruction algorithm is that the sample points X are *unorganized*. That is, no relationship amongst the points is known beforehand. This difficulty is seen most easily in the context of what might seem, at first blush, a simple problem: trying to reconstruct a curve embedded in $\mathbb{R}^2$ from its samples. If the points are organized, i.e., the adjacency relationship between samples is known, then reconstruction is completely straightforward. This situation isn't particularly interesting because it is quite obvious that the more finely sampled the true manifold is, the closer the reconstruction will be to the original; this is just a matter of calculus. However, suppose that the adjacency relationship amongst the points is *not* known. In this case, the issue of how to reconstruct the curve is not at all clear. Should we attach all samples to their nearest neighbours? Should we instead begin by connecting a single point to its nearest neighbour, and then connect that point to is nearest neighbour except for the one already attached, and so on? It is fairly easy to think of examples which expose the flaws of these simple greedy schemes; and since we don't know anything about the true manifold F, except for its dimension, then it seems foolhardy to employ algorithms which are known not to work on some cases. Furthermore, we have been looking at the "relatively straightforward" case provided by curves embedded in two dimensions. What of more complex cases, such as 7-manifolds embedded in $\mathbb{R}^{82}$; or K-manifolds embedded in function spaces? In this case, it is not at all easy to think of heuristic algorithms like the one mentioned above. As a result, sophisticated algorithms, which possess mathematical properties which are amenable to analysis, must be elaborated if we are to have a chance of solving the manifold reconstruction problem.

The outline of this paper is as follows. Section 2 will review the existing literature in the field of manifold reconstruction. While the problem has not been solved, several special cases have been, and these will be discussed. Section 3 outlines the new approach. The algorithm's properties, including those related to complexity, are discussed in section 4. Results are shown and conclusions are drawn in 5.

2 Review of Existing Literature

Very little work has been done in terms of solving the problem of manifold reconstruction in a rigourous way. Principal components analysis is a very simple stab at this kind of problem, but it makes the highly unsatisfactory assumption that the manifold is linear, which will not generally be the case. Bregler and Omohundro [5] attack a related type of problem, but do not give any provable results. Furthermore, their method is problematic in that the object it generates is not truly a manifold, as it may have varying dimension in different parts. The graphics literature [8, 7, 6] presents several examples of attempts to reconstruct surfaces (2-manifolds) embedded in $\mathbb{R}^3$; these results, while often effective practically, are neither provable, nor extensible to the general case. More recently, Amenta [2, 1] has solved the manifold reconstruction problem, as posed above, for the special cases of reconstructing curves (1-manifolds) in $\mathbb{R}^2$ and surfaces in $\mathbb{R}^3$. Her solutions have used machinery from computational geometry, both in terms of construction and proofs. Other papers in a similar vein have followed suit [3, 4].

3 The Algorithm

Before outlining the algorithm, several clarifications must be made to the problem formulation given on page 1. First, we say that the manifold R is a *good approximation* to the manifold F if (1) R is homeomorphic to F and (2) $d(R, F)$ is sufficiently small, where $d(\cdot,\cdot)$ is an appropriate metric (such as the symmetric Hausdorff distance). Second, the notion of density is not spelled out explicitly here; however, the relevant notion is probably that of local feature size, as used by Amenta [2, 1]. Third, the reconstruction R will be a C^0, rather than a C^1 manifold; in particular, R will be a simplicial complex. Diffeomorphism between the original and reconstructed manifolds is a goal of future research.

Let us begin by establishing the basic notation. The embedding space is Z; the manifold is F; the sampling of the manifold is X. $X \subset F \subset Z$, with $|X| < \infty$. Recall that the embedding space is restricted to be a Hilbert space. The goal of the algorithm is to construct a simplicial complex based only on X, denoted $SC(X)$. The algorithm consists of four steps.

Step 1: Finding Neighbourhoods

The goal is to find the neighbourhood $NBD(x)$ for each point $x \in X$, where $NBD(x) \subset X$. This step is quite simple: out of the entire sampling X, $NBD(x)$ is defined to be the rK closest points to x, where $r \geq 1$ and K is the manifold dimension. r is the single parameter in the algorithm; it is expected that the validity of the algorithm will depend on the choice of r and the sampling density.

Step 2: Finding Tangent Hyperplanes

The goal is to find an approximation to the tangent hyperplane at each point $x \in X$. In fact, however, this will often only be possible at a subset of points in X; the reasons for this, and its significance will become clear shortly. In order to find the **approximating tangent hyperplane** at x, denoted $ATH(x)$, we must make use of the following definition.

Definition: A point x is said to be **enclosed** by the $K+1$ points $\{x_i\}_{i=0}^K$ if the projection of x onto the K-dimensional hyperplane running through $\{x_i\}_{i=0}^K$ is contained within the K-dimensional simplex defined by $\{x_i\}_{i=0}^K$.

It will be useful to have a simple method for testing the enclosure property on a given set of points. Note that the hyperplane running through $\{x_i\}_{i=0}^K$ is given by

$$\text{span}\{x_1 - x_0, \ldots, x_K - x_0\} + x_0$$

Indeed, any point on the hyperplane may be given by

$$x_0 + \sum_{i=1}^{K} \lambda_i (x_i - x_0) = x_0 + \sum_{i=1}^{K} \lambda_i \Omega_i$$

where $\Omega_i = x_i - x_0$ and λ is a $K \times 1$ column vector which can take on any value in $\mathbb{R}^K$. Now, the projection problem is specified as

$$\min_{\lambda \in \mathbb{R}^K} \left\| x_0 + \sum_{i=1}^{K} \lambda_i \Omega_i - x \right\|$$

This can be solved as follows:

$$\left\| x_0 + \sum_{i=1}^{K} \lambda_i \Omega_i - x \right\|^2 = \lambda^T \Psi \lambda - 2\psi^T \lambda + ||\zeta||^2$$

where $\zeta = x - x_0$, Ψ is a Hermitian $K \times K$ matrix given by $\Psi_{ik} = \langle \Omega_i, \Omega_k \rangle$ and ψ is a $K \times 1$ column vector given by $\psi_i = \frac{1}{2}(\langle \Omega_i, \zeta \rangle + \langle \zeta, \Omega_i \rangle)$. This may be now be minimized to yield

$$\lambda^* = \Psi^{-1}\psi$$

Finally, if we define $\theta_0 = 1 - \sum_{i=1}^{K} \lambda_i^*$ and $\theta_i = \lambda_i^*$, $i = 1, \ldots, K$, then the condition that x is enclosed by $\{x_i\}_{i=0}^{K}$ is equivalent to the condition

$$0 \leq \theta_i \leq 1, \quad i = 0, \ldots, K$$

which is quite easy to test.

Given the enclosure property, an algorithm for calculating $ATH(x)$ may now be specified.

Definition: A set of $K+1$ points $\{x_i\}_{i=0}^{K}$ which are drawn from the neighbourhood of x, $NBD(x)$, is said to be a **tangent basis** for the point $x \in X$ if (a) x is enclosed by $\{x_i\}_{i=0}^{K}$ and (b) no other point in $NBD(x)$ is enclosed by $\{x_i\}_{i=0}^{K}$. A shifted version of the hyperplane defined by $\{x_i\}_{i=0}^{K}$, namely span$\{x_1 - x_0, \ldots, x_K - x_0\} + x$, is referred to as x's **approximate tangent hyperplane**, or $ATH(x)$. Note that $x \in ATH(x)$.

It may be the case that a point x has no approximating tangent hyperplane; in this case, we declare x to belong to the boundary.

Definition: Any point x which has no tangent basis is said to belong to the **boundary**, denoted BD.

Let $\rho : NBD(x) \to \{1, \ldots, rK\}$ be the ordering of the points in the $NBD(x)$ in terms of their proximity to x; i.e., if $\rho(\hat{x}) = 1$, then $\hat{x}$ is the closest point to x. Then the algorithm is as in figure 1.

Note that this algorithm for calculating the approximate tangent hyperplane, as based on the enclosure property, is an intuitive one. The idea is that the ATH is best approximated by a nearby secant, where the correct notion of closeness is given by enclosure; this is essentially inspired by the intermediate value theorem.

Step 3: Calculating the Local Region and Edge-Set

The goal is to find the little patch of the manifold which contains x. As stated, this is, of course, ill-defined; however, a particular definition of the local region will be given which is useful in calculating the simplicial complex $SC(X)$.

Definition: The **perpendicular bisecting halfspace** of the point $x \in X$ with respect to the point $x' \in X$ is denoted by

```
**Let** $z = 0$.
**Do**
        $(i_0, \ldots, i_K) = \underset{1 \leq i_k \leq rK : i_0 < \cdots < i_K, \sum_{k=0}^{K} i_k > z}{\text{argmin}} \sum_{k=0}^{K} i_k$
        $x_k = \rho^{-1}(i_k) \quad k = 0, \ldots, K$
        $TB = \{x_k\}_{k=0}^{K}$
        $OTH = NBD(x) - \{x\} - TB$
        $z = \sum_{k=0}^{K} i_k$
**Until** ($x$ is enclosed by $TB$) and ($\nexists \hat{x} \in OTH$ which
  is enclosed by $TB$)
**If** $i_k = r - k,\ k = 0, \ldots, K$
        $x \in BD$
**else**
        $ATH(x) = \text{span}\{x_1 - x_0, \ldots, x_K - x_0\} + x$.
**end**
```

Figure 1. The algorithm for calculating the approximating tangent hyperplane at x, $ATH(x)$.

$PBH(x, x')$ and is given by

$$PBH(x, x') = \{z \in Z : ||z - x|| \leq ||z - x'||\}$$

Definition: The **local region** of $x \in X - BD$, denoted $LR(x)$, is given by

$$LR(x) = \left[\bigcap_{x' \in X} PBH(x, x') \right] \bigcap ATH(x)$$

Note that the local region is only defined for points that are not in the boundary; and that the local region is K-dimensional, as $ATH(x)$ is K-dimensional, and that $x \in LR(x)$ (since $x \in ATH(x), PBH(x, x')$).

Calculating the local region is a matter of deciding which of the constraints provided by intersecting the halfspaces $PBH(x, x')$ are binding. That is, we could write

$$LR(x) = \left[\bigcap_{x' \in ES(x)} PBH(x, x') \right] \cap ATH(x)$$

where $ES(x) \subset X$, and is the smallest such set. $ES(x)$ is referred to as the **edge-set** of x, for reasons that will be clear shortly. In order to find the edge-set, a convex hull technique may be used.

In particular, note that

$$ATH(x) = \{z \in Z : z = \Omega_x \lambda + x, \lambda \in \mathbb{R}^K\}$$

where $\Omega_x \lambda$ is a short form for $\sum_{i=1}^{K} \Omega_{x,i} \lambda_i$, and $\Omega_{x,i}$ are the vectors found in step 2, and

$$PBH(x, x') = \{z \in Z : \langle \alpha_{x'}, x - \beta_{x'} \rangle \geq 0\}$$

where $\alpha_{x'} = x - x'$ and $\beta_{x'} = \frac{1}{2}(x + x')$. Note that

$$LR(x) = \bigcap_{x' \in X} (PBH(x, x') \cap ATH(x))$$

and

$$PBH(x, x') \cap ATH(x) = \{z \in Z : \gamma_{x,x'}^T \lambda \geq \delta_{x,x'}, \lambda \in \mathbb{R}^K\}$$

where $\gamma_{x,x'}$ is a $K \times 1$ column vector whose i^{th} entry is $\langle \Omega_{x,i}, \alpha_{x'} \rangle$, and $\delta_{x,x'} = \langle \alpha_{x'}, \beta_{x'} - x \rangle$ is a scalar. Thus, the problem is reduced to looking for the binding set of inequalities out of the entire set

$$\gamma_{x,x'}^T \lambda \geq \delta_{x,x'} \quad x' \in X - \{x\}$$

It may be shown that this problem is equivalent to finding the convex hull in $\mathbb{R}^K$ of the points

$$\epsilon_{x,x'} = \frac{\gamma_{x,x'}}{\delta_{x,x'}} \quad x' \in X - \{x\}$$

The constraints which bind are exactly the same as the points which lie on the exterior of the convex hull. In terms of complexity considerations, finding the convex hull J points in $\mathbb{R}^K$ is $O(J^{\lceil K/2 \rceil})$. Finding the set of binding constraints gives us $ES(x)$.

Step 4: From the Edge-Set to the Simplicial Complex

The goal is to take the edge-sets $ES(x)$ $\forall x \in X - BD$ and to find a simplicial complex. A K-dimensional simplex consists of edges between all pairs of the $K + 1$ points. Thus, an algorithm to convert edge-sets into simplices may proceed as follows. For each x, and for each combination of K points culled from $ES(x)$ labelled $x_1, \ldots, x_K$, verify for each $i = 1, \ldots, K$ whether $\{x\} \cup \{x_1, \ldots, x_K\} - \{x_i\} \subset ES(x_i)$. If so, then the simplex corresponding to the points $x, x_1, \ldots, x_K$ belongs to $SC(X)$. (Otherwise, it does not.)

4 Discussion of the Algorithm

The algorithm which was specified in the previous section is fairly involved. We will try to explain aspects of it on two independent fronts: why it works, and its complexity.

4.1 Motivation of the Algorithm

The following result is proven in [1]. Let $V(X)$ be the Voronoi Diagram of the set of sample points X of manifold F, where the manifold is of dimension 2, and is embedded in $\mathbb{R}^3$; note that by $V(X)$ we mean the entire diagram, rather than just the vertices. Let $\Upsilon = V(X) \cap F$; intersecting the Voronoi Diagram, which is a partition of $\mathbb{R}^3$, with the manifold F has the effect of partitioning the manifold. Now, let Dual(Υ) be the dual of Υ; that is, if two samples x_1 and x_2 are contained in cells of Υ which border each other, let Dual(Υ) contain an edge between x_1 and x_2. It can be shown that if X is a dense enough sampling of F (in a formal sense not to be described here), then Dual(Υ) will be a simplicial complex which is homeomorphic to F and sufficiently close to it.

Of course, this theorem cannot be used for constructing $SC(X)$, since the construction of Dual(Υ) relies on knowledge of F, which we do not possess. All we know, of course, is the sampling $X \subset F$. However, if a local knowledge of F was available, this might prove sufficient to construct $SC(X)$. Now, F is locally approximated at a point x by the tangent hyperplane to F at x; so if F can be thought of as a collection of tangent hyperplanes, then perhaps some progress can be made.

In fact, this is exactly what is done. In step 2, the approximating tangent hyperplane is calculated at each point. In step 3, this information about the local behaviour of the manifold is used to find the edge-set in a manner which is exactly parallel to intersecting the Voronoi Diagram of the points with this local approximation of the manifold. However, the advantage of doing things in the way described in the previous section is in terms of the complexity, as discussed in the next section.

4.2 Complexity

To analyze the complexity of the overall algorithm, it will be simplest to break it down step by step.

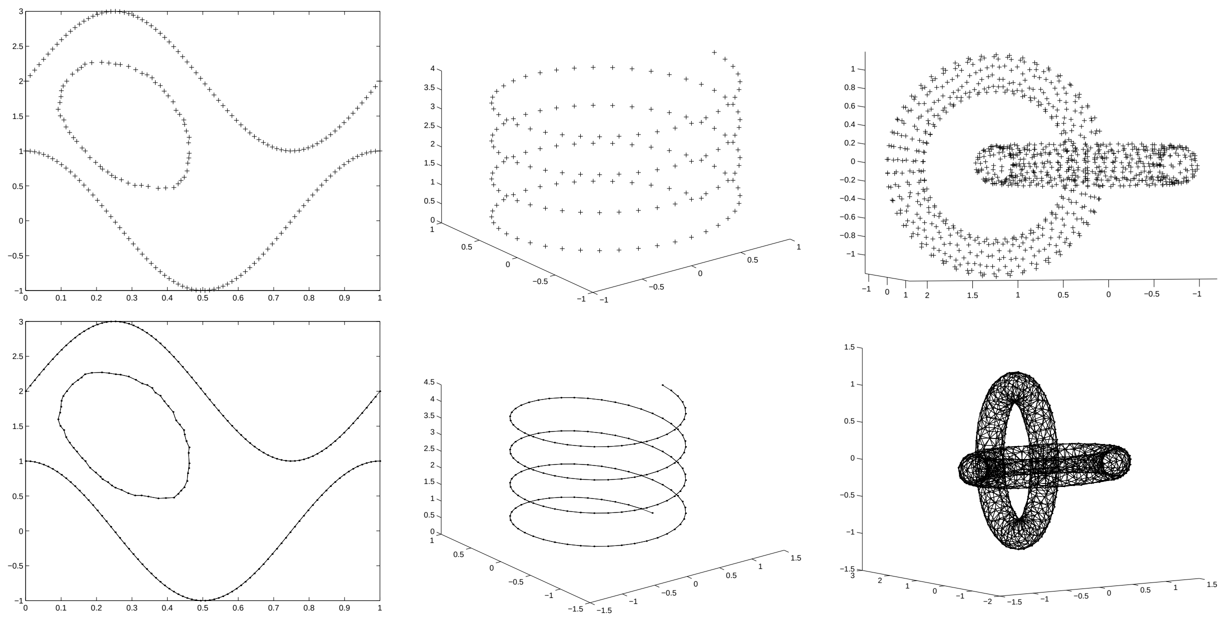
Step 1: For a given x, this step is $O(rKJ)$ since the neighbourhood is of size rK. (In fact, the step could be $O(J \log J)$: all of the points could be sorted in terms of their distance with respect to a fixed point; however, we will assume that $rK < \log J$.) Thus, for all J points, this step is $O(rKJ^2)$.

Step 2: In the worst case scenario, all combinations of $K + 1$ points out of the neighbourhood, which is of size rK must be searched. As a result, for a single point the complexity is $\binom{rK}{K+1} < (rK)^{K+1}$; for all points, it is $O(J(rK)^{K+1})$.

Step 3: As was mentioned in section 3, the calculation of the local region amounts to a convex hull calculation of J points in K dimensions; this has complexity $O(J^{\lceil K/2 \rceil})$.

Step 4: In principle, an edge-set $ES(x)$ may have as many as J (or really $J - 1$) elements; in practice, of course, this is highly unlikely. However, assuming that this is the case, then the complexity of dealing with a single edge-set is $\binom{J}{K} < J^K$; so the complexity of constructing the simplicial complex from all of J edge-sets is $O(J^{K+1})$.

If J is large, the complexity of step 4 dominates, so that the



Curves embedded in $\mathbb{R}^2$. Curves embedded in $\mathbb{R}^3$. Surfaces embedded in $\mathbb{R}^3$.

Figure 2. Examples of manifold reconstruction. The samples are shown above and the reconstructed manifold below.

complexity of the entire algorithm is $O(J^{K+1})$. However, as has been noted, in a typical problem, $ES(x)$ should be $O(1)$ for all $x \in X$; as a result, the complexity from step 3 dominates, and the complexity is $O(J^{\lceil K/2 \rceil})$. In fact, however, we may lower the complexity further if we are willing to sacrifice a little bit of certainty.

5 Results and Conclusions

In order to demonstrate the efficacy of the reconstruction algorithm, three examples are shown in figure 2. Owing to the nature of what can be visualized, only three types of examples are here reproduced: one-manifolds (curves or disconnected curves) embedded in $\mathbb{R}^2$, one-manifolds embedded in $\mathbb{R}^3$, and two-manifolds (surfaces or disconnected surfaces) embedded in $\mathbb{R}^3$. (Of course, in cases of interest we are often looking for, say, three-manifolds embedded in $\mathbb{R}^{40}$; however, there is no good way of visualizing this.) In all cases, $r = 10$ is used.

In each case, the samples are shown above and the reconstructed manifold is shown below. Despite the varying geometries and topologies, the algorithm is correct in all of its reconstructions. It should be noted, however, that the algorithm does not work in all cases. In particular, the algorithm seems to fail in cases where the samples are close to non-generic, that is, nearly regular samplings. For example, the algorithm consistently fails to reconstruct a small piece of a plane embedded in $\mathbb{R}^3$ which is sampled in a gridlike fashion. Future research will focus on ensuring that the algorithm works in such cases (assuming the sampling is dense enough), as well as on proving results which establish the validity of the algorithm.

References

[1] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.

[2] N. Amenta, M. Bern, and D. Eppstein. The crust and the beta-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60(2):125–135, 1998.

[3] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, pages 415–421, 1998.

[4] N. Amenta and S. Choi. One-pass delaunay filtering for homeomorphic 3d surface reconstruction. Technical Report TR99-08, University of Texas at Austin, 1999.

[5] C. Bregler and S. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 494–499, 1995.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH '96*, pages 303–312, 1996.

[7] H. Hoppe, T. DeRose, T. Duchamp, H. Jin, J. McDonald, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proc. SIGGRAPH '94*, pages 19–26, 1994.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstructions from unorganized points. In *Proc. SIGGRAPH '92*, pages 71–78, 1992.