Optimal Net Surface Problems with Applications*

Xiaodong Wu** and Danny Z. Chen

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
FAX: 219-631-9260
{xwu,chen}@cse.nd.edu

Abstract. In this paper, we study an interesting geometric graph called *multi-column graph* in the d-D space ($d \geq 3$), and formulate two combinatorial optimization problems called the *optimal net surface problems* on such graphs. Our formulations capture a number of important problems such as surface reconstruction with a given topology, medical image segmentation, and metric labeling. We prove that the optimal net surface problems on general d-D multi-column graphs ($d \geq 3$) are NP-hard. For two useful special cases of these d-D ($d \geq 3$) optimal net surface problems (on the so-called *proper ordered multi-column graphs*) that often arise in applications, we present polynomial time algorithms. We further apply our algorithms to some surface reconstruction problems in 3-D and 4-D, and some medical image segmentation problems in 3-D and 4-D, obtaining polynomial time solutions for these problems. The previously best known algorithms for some of these applied problems, even for relatively simple cases, take at least exponential time. Our approaches for these applied problems can be extended to higher dimensions.

Keywords: Geometric Graphs, Algorithms, NP-hardness, Surface Reconstructions, 3-D Image Segmentations

1 Introduction

We study an interesting geometric graph $G = (V, E)$ in the d-D space ($d \geq 3$), defined as follows. Given any undirected graph $B = (V_B, E_B)$ embedded in $(d-1)$-D (called the *base graph*) and an integer $K > 0$, $G = (V, E)$ is an undirected graph in d-D *generated* by B and K. For each vertex $i = (x_0, x_1, \ldots, x_{d-2}) \in V_B$, there is a set V_i of K vertices in G corresponding to i; $V_i = \{(x_0, x_1, \ldots, x_{d-2}, k) : k = 0, 1, \ldots, K-1\}$, called the *i-column* of G. We

* This research was supported in part by the National Science Foundation under Grant CCR-9988468 and the 21st Century Research and Technology Fund from the State of Indiana.

** The work of this author was supported in part by a fellowship from the Center for Applied Mathematics, University of Notre Dame, Notre Dame, Indiana, USA.

P. Widmayer et al. (Eds.): ICALP 2002, LNCS 2380, pp. 1029–1042, 2002.

denote the vertex $(x_0, x_1, \ldots, x_{d-2}, k)$ of V_i by i_k. If an edge $(i, j) \in E_B$, we say that the i-column and j-column are *adjacent* to each other in G. The edges in G can only connect pairs of vertices in adjacent columns. We call G thus defined a d-D *multi-column graph* generated by its $(d-1)$-D base graph B with a *height* K. A *net surface* in G (briefly called a *net*) is a subgraph of G defined by a function $\mathcal{N}$: $V_B \rightarrow \{0, 1, \ldots, K-1\}$, such that for each edge $(i, j) \in E_B$, $(i_{\mathcal{N}(i)}, j_{\mathcal{N}(j)})$ is an edge in E. For simplicity, we denote a net by its function $\mathcal{N}$. Intuitively, a net $\mathcal{N}$ in G is a special mapping of the $(d-1)$-D base graph B to the d-D space, such that $\mathcal{N}$ "intersects" each i-column at exactly one vertex and $\mathcal{N}$ preserves the topology of B. Let $V(H)$ and $E(H)$ denote the vertices and edges of a graph H. We consider two optimal net surface problems.

Optimal V-weight net surface problem: Given a d-D multi-column graph $G = (V, E)$, each vertex $v \in V$ having a real-valued weight $w(v)$, find a net $\mathcal{N}$ in G such that the weight $\alpha(\mathcal{N})$ of $\mathcal{N}$, with $\alpha(\mathcal{N}) = \sum_{v \in V(\mathcal{N})} w(v)$, is minimized.

Optimal VE-weight net surface problem: Given a d-D multi-column graph $G = (V, E)$, each vertex $v \in V$ having a real-valued weight $w(v)$ and each edge $e \in E$ having a real-valued cost $c(e)$, find a net $\mathcal{N}$ in G such that the cost $\beta(\mathcal{N})$ of $\mathcal{N}$, with $\beta(\mathcal{N}) = \sum_{v \in V(\mathcal{N})} w(v) + \sum_{e \in E(\mathcal{N})} c(e)$, is minimized.

These optimal net surface problems find applications in several areas such as medical image analysis, computational geometry, computer vision, and data mining [34]. For example, we model a 3-D/4-D image segmentation problem as computing an optimal V-weight net surface, and a surface reconstruction problem in $\Re^d$ with a given underlying topology as an optimal VE-weight net problem, motivated by the deformable model in image analysis [25,32]. See Section 4 for more details of these problems.

We are particularly interested in d-D multi-column graphs ($d \geq 3$) with a special property called *proper ordering.* Let $V_B(i)$ denote the set of vertices adjacent to a vertex i in the base graph B. A multi-column graph G is said to be *proper ordered* if the following two conditions hold on every i-column V_i of G. (1) For each vertex $i_k \in V_i$, i_k is connected to a non-empty sequence of consecutive vertices in every adjacent j-column V_j of V_i (i.e., $j \in V_B(i)$), say $j_{k'}, j_{k'+1}, \ldots, j_{k'+s}$ ($s \geq 0$); we call $(j_{k'}, j_{k'+1}, \ldots, j_{k'+s})$, in this order, the *edge interval* of i_k on V_j, denoted by $I(i_k, j)$. (2) For any two consecutive vertices i_k and i_{k+1} in V_i, $First(I(i_k, j)) \leq First(I(i_{k+1}, j))$ and $Last(I(i_k, j)) \leq Last(I(i_{k+1}, j))$ for each $j \in V_B(i)$, where $First(I)$ (resp., $Last(I)$) denotes the d-th coordinate of the first (resp., last) vertex in an edge interval I. We call such a graph G a *proper ordered multi-column graph* (briefly, a proper ordered graph). Many multi-column graphs for applied problems are proper ordered (e.g., medical image segmentation and surface reconstruction in 3-D and 4-D, as shown in Section 4).

We are also interested in a useful special case of the optimal VE-weight net surface problem, called the **optimal VCE-weight net surface problem**. The optimal VCE-weight net surface problem is defined on a d-D proper ordered graph $G = (V, E)$ such that the cost of each edge $(i_k, j_{k'}) \in E$ is $f_{ij}(|k - k'|)$, herein $f_{ij}(\cdot)$ is a convex non-decreasing function associated with each edge $(i, j) \in E_B$.

These d-D optimal net surface problems ($d \geq 3$), even on proper ordered graphs, might appear to be computationally intractable at first sight. In fact, even for a very restricted case of the optimal V-weight net problem on a 3-D proper ordered graph arising in 3-D image segmentation, the previously best known exact algorithms [33,15] take at least exponential time. (The 2-D case of this image segmentation problem can be solved in polynomial time by computing shortest paths [13,33].) Interestingly, we are able to develop polynomial time algorithms for the optimal V-weight net problem and optimal VCE-weight net problem on proper ordered graphs. Let $n_B = |V_B|, m_B = |E_B|, n = |V|, m = |E|$, and K be the height of G. Denote by $T(n', m')$ the time for finding a minimum s-t cut in an edge-weighted directed graph with $O(n')$ vertices and $O(m')$ edges [3]. Our main results are summarized as follows.

– Proving that in d-D ($d \geq 3$), the optimal V-weight net problem on a general multi-column graph is NP-hard , and the optimal VE-weight net problem is NP-hard even on a proper ordered graph.
– A $T(n, m_B K)$ time algorithm for the optimal V-weight net problem on a proper ordered graph G.
– A $T(n, \kappa m_B K)$ time algorithm for the optimal VCE-weight net problem, where $\kappa = \max_{(i_k, j_{k'}) \in E} |k - k'|$.
– Modeling a surface reconstruction problem in $\Re^d$ ($d = 3, 4$) with a given topology of the underlying surface as a geometric optimization problem, and solving it as an optimal VCE-weight net problem in polynomial time (depending on a tolerance error $\epsilon > 0$).
– Modeling 3-D and 4-D medical image segmentation of smooth objects as an optimal V-weight net problem on a proper ordered graph, and solving it in $T(n, n)$ time, where n is the number of input image voxels. Our solutions for these problems can be extended to higher dimensions.

In fact, our work on optimal nets can be viewed as a rather general theoretical framework, which is likely to be interesting in its own right. This framework captures a number of other problems, such as metric labeling problems [26,17,12] and a class of integer optimization problems [22,19,21]. All those problems can be transformed to a V-weight or VE-weight net problem on a special proper ordered graph. Our polynomial time algorithms are inspired by Hochbaum's solutions for the minimum closure problem [30,20] and s-excess problem [21], and use a technique for integer optimization over some special monotone constraints [19]. Hochbaum [19] defined a class of integer programs with some special monotone constraints and cast that problem as a minimum cut problem on graphs. We generalize the graph construction in [19] to proper ordered graphs through a judicious characterization of the structures of the proper ordering.

Kleinberg and Tardos [26], Gupta and Tardos [17], and Chekuri *et al.* [12] studied the metric labeling problem, formulating the problem as optimizing a combinatorial function consisting of *assignment costs* and *separation costs*. We are able to reduce this metric labeling problem to the optimal VE-weight net problem on a special proper ordered graph. The metric labeling problem is known to be NP-hard when the separation costs are non-convex [26]. Boykov *et al.* [11]

and Ishikawa and Geiger [24] showed a direct reduction to the minimum s-t cuts for a special case of metric labeling when the label set is $L = \{0, 1, \ldots, l\}$ and the separation costs are linear; but, their algorithms cannot handle real-valued vertex weights.

We omit the proofs of the lemmas due to the space limit.

2 Hardness of the Net Surface Problems on Multi-column Graphs

This section presents some hardness results for the two optimal net problems. We show that the optimal V-weight net problem and optimal VE-weight net problem on a general d-D multi-column graph are both NP-hard ($d \geq 3$). Actually, the optimal VE-weight net problem is NP-hard even on a 3-D proper ordered graph.

Section 3.1 shows that finding an optimal V-weight net in a d-D proper ordered graph is polynomially solvable ($d \geq 3$). The proper ordering property is crucial to our polynomial time algorithms. Here, we prove that without the proper ordering, the optimal V-weight net problem on a d-D multi-column graph is NP-hard ($d \geq 3$), by a sophisticated reduction from the 3SAT problem [16]. In fact, even finding any net in such a graph in 3-D is NP-complete.

Lemma 1. *Deciding whether there exists a net in a 3-D multi-column graph (EXNET) is NP-complete.*

The NP-hardness of the optimal V-weight net problem thus follows from Lemma 1.

Theorem 1. *The optimal V-weight net problem on a d-D multi-column graph is NP-hard ($d \geq 3$).*

Now, we show the NP-hardness of the optimal VE-weight net problem on a proper ordered graph (whose edge costs need not form convex non-decreasing functions). Kleinberg and Tardos studied the metric labeling problem and pointed out its NP-completeness [26]. We can transform the metric labeling problem to an instance of the optimal VE-weight net problem on a special proper ordered graph with any two adjacent columns forming a complete bipartite graph.

Theorem 2. *The optimal VE-weight net problem on a proper ordered graph is NP-hard.*

3 Algorithms for Net Surface Problems on Proper Ordered Graphs

This section presents our polynomial time algorithms for computing an optimal V-weight net and an optimal VCE-weight net on any proper ordered graph G.

We use a unified approach which formulates both the optimal V-weight and VCE-weight net problems as computing a minimum s-t cut [3] in another graph transformed from G. Our approach is inspired by Hochbaum's solutions for the minimum closure problem [30,20] and minimum s-excess problem [21]. Note that the minimum s-excess problem is an extension of the minimum closure problem.

Without loss of generality (WLOG), we assume that the base graph B is connected. Otherwise, we can solve these optimal net problems with respect to each connected component of B separately.

3.1 The Optimal V-Weight Net Surface Problem

Consider an n_B-vertex and m_B-edge base graph $B = (V_B, E_B)$ and an arbitrarily generated proper ordered graph $G = (V, E)$ of height K, each vertex v of G having a real-valued weight $w(v)$. Let $n = |V| = n_B * K$ and $m = |E|$. This subsection presents a $T(n, m_B K)$-time algorithm for finding an optimal V-weight net in G, where $T(n', m')$ denotes the time bound for finding a minimum s-t cut in an edge-weighted directed graph with $O(n')$ vertices and $O(m')$ edges [3]. Interestingly, the time bound of our algorithm is independent of the number of edges in G.

We begin with reviewing the minimum closure problem [30,20]. A *closed set* $\mathcal{C}$ in a directed graph is a set of vertices such that all successors of any vertex in $\mathcal{C}$ are also contained in $\mathcal{C}$. Given a directed graph $G' = (V', E')$ with real-valued vertex weights, the minimum closure problem seeks a closed set in G' with the minimum total vertex weight.

We now discuss our construction. First, we build a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ from G as follows. $\tilde{V} = \{\tilde{i}_k : i_k \in V\}$. The weight $\tilde{w}(\tilde{i}_k)$ of each vertex $\tilde{i}_k$ in $\tilde{G}$ is assigned in the following way: For each $i \in V_B$, the weight of vertex $\tilde{i}_0$ is set to $w(i_0)$, and for every $k = 1, 2, \ldots, K-1$, $\tilde{w}(\tilde{i}_k) = w(i_k) - w(i_{k-1})$. Next, we define the edges for $\tilde{G}$. For each $i \in V_B$ and every $k = 1, 2, \ldots, K-1$, vertex $\tilde{i}_k$ has a directed edge $(\tilde{i}_k, \tilde{i}_{k-1})$ to vertex $\tilde{i}_{k-1}$ with a cost $+\infty$ (note that the edge costs of $\tilde{G}$ are for computing a minimum closed set later). We call the directed path $\tilde{i}_{K-1} \to \tilde{i}_{K-2} \to \cdots \to \tilde{i}_0$ in $\tilde{G}$ the *i-chain*. For any ordered pair of adjacent columns in G, say (i-column, j-column), and each $k \in \{0, 1, \ldots, K-1\}$, let the edge interval of the vertex i_k on V_j be $I(i_k, j) = (j_p, j_{p+1}, \ldots, j_{p+s})$, with $p \geq 0$, $s \geq 0$, and $p + s < K$. Then, let a directed edge with a cost $+\infty$ go from vertex $\tilde{i}_k$ to vertex $\tilde{j}_p$ in $\tilde{G}$. Note that the same construction is also done for the ordered pair (j-column, i-column). Figure 1(a) shows part of the graph G that is associated with an edge $(i, j) \in E_B$, and Figure 1(b) illustrates the corresponding constructions in $\tilde{G}$. Let $\tilde{V}_0 = \{\tilde{i}_0 : i \in V_B\}$, and $\tilde{G}_0 \subseteq \tilde{G}$ be the induced subgraph of the vertex set $\tilde{V}_0$ in $\tilde{G}$. The following lemma characterizes several properties of $\tilde{G}_0$, which are useful to our construction.

Lemma 2. *(1) $\tilde{G}_0$ is a strongly connected component of $\tilde{G}$. (2) $\tilde{V}_0$ is a closed set in $\tilde{G}$. (3) For any non-empty closed set $\mathcal{S}$ in $\tilde{G}$, $\tilde{V}_0 \subseteq \mathcal{S}$. (4) $\tilde{G}_0$ corresponds to a net in G.*

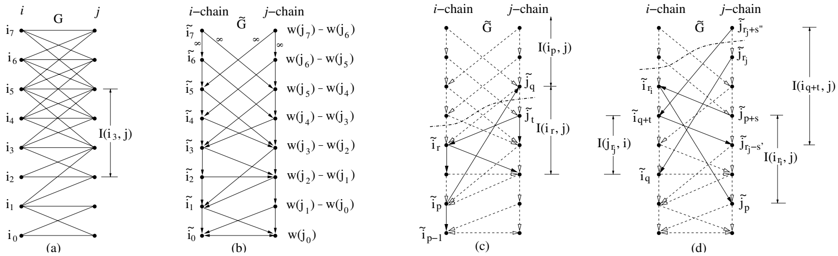

Fig. 1. (a) Illustrating the proper ordering of G. (b) Constructing graph $\tilde{G}$ from G. (c) Illustrating the proof of Lemma 3. (d) Illustrating the proof of Lemma 4.

The next two lemmas show the relations between a V-weight net in G and a closed set in $\tilde{G}$.

Lemma 3. *A V-weight net $\mathcal{N}$ in G corresponds to a non-empty closed set $\mathcal{S}$ in $\tilde{G}$ with the same weight.*

Lemma 4. *A non-empty closed set $\mathcal{S}$ in $\tilde{G}$ corresponds to a V-weight net $\mathcal{N}$ in G with the same weight.*

Based on Lemmas 3 and 4, we have the following lemma.

Lemma 5. *The net $\mathcal{N}^*$ corresponding to a non-empty minimum closed set $\mathcal{S}^*$ in $\tilde{G}$ is an optimal V-weight net in G.*

Note that the characterizations in the above lemmas are all concerned with non-empty closed sets in $\tilde{G}$. However, the minimum closed set $\mathcal{S}^*$ in $\tilde{G}$ can be empty (with a weight zero), and when this is the case, $\mathcal{S}^* = \emptyset$ gives little useful information on $\tilde{G}$. Fortunately, our careful construction of $\tilde{G}$ still enables us to overcome this difficulty. If the minimum closed set in $\tilde{G}$ is empty, then it implies that the weight of every non-empty closed set in $\tilde{G}$ is non-negative. To obtain a minimum *non-empty* closed set in $\tilde{G}$, we do the following: Let M be the total weight of vertices in $\tilde{V}_0$; pick an arbitrary vertex $\tilde{i}_0 \in \tilde{V}_0$ and assign a new weight $\tilde{w}(\tilde{i}_0) - M - 1$ to $\tilde{i}_0$. We call this a *translation operation* on $\tilde{G}$. From Lemma 2, $\tilde{V}_0 \neq \emptyset$ is a closed set in $\tilde{G}$ and is a subset of any non-empty closed set in $\tilde{G}$. Further, observe that the total weight of vertices in the closed set $\tilde{V}_0$ (after a translation operation on $\tilde{G}$) is negative. This implies that any minimum closed set in $\tilde{G}$ (after a translation operation on $\tilde{G}$) cannot be empty. Also based on Lemma 2, we have the following lemma.

Lemma 6. *For a non-empty closed set $\mathcal{S}$ in $\tilde{G}$, let $\tilde{w}(S)$ denote the total weight of $\mathcal{S}$ before any translation operation on $\tilde{G}$. Then after a translation operation, the weight of $\mathcal{S}$ is $\tilde{w}(S) - M - 1$.*

Now, we can simply find a minimum closed set $\mathcal{S}^*$ in $\tilde{G}$ after performing a translation operation on $\tilde{G}$. Based on Lemma 6, $\mathcal{S}^*$ is a minimum *non-empty* closed set in $\tilde{G}$ before the translation.

As in [30,20], we compute a minimum non-empty closed set in $\tilde{G}$, as follows. Let $\tilde{V}^+$ and $\tilde{V}^-$ denote the set of vertices in $\tilde{G}$ with non-negative and negative weights, respectively. Define a new directed graph $\tilde{G}_{st} = (\tilde{V} \cup \{s,t\}, \tilde{E} \cup E_{st})$. The vertex set of $\tilde{G}_{st}$ is the vertex set $\tilde{V}$ of $\tilde{G}$ plus a source s and a sink t. The edge set of $\tilde{G}_{st}$ is the edge set $\tilde{E}$ of $\tilde{G}$ plus a new edge set E_{st}. E_{st} consists of the following edges: The source s is connected to each vertex $v \in \tilde{V}^-$ by a directed edge of cost $-\tilde{w}(v)$; every vertex $v \in \tilde{V}^+$ is connected to the sink t by a directed edge of cost $\tilde{w}(v)$. Let $(\mathcal{S}, \bar{\mathcal{S}})$ denote a finite-cost s-t cut in $\tilde{G}_{st}$ with $s \in \mathcal{S}$ and $t \in \bar{\mathcal{S}}$, and $C(\mathcal{S}, \bar{\mathcal{S}})$ denote the total cost of the cut. Note that the directed edges in the cut $(\mathcal{S}, \bar{\mathcal{S}})$ are either in $(\mathcal{S} \cap \tilde{V}^+, t)$ or in $(s, \bar{\mathcal{S}} \cap \tilde{V}^-)$. Let $\tilde{w}(V')$ denote the total weight of vertices in a subset $V' \subseteq \tilde{V}$. Then, we have $C(\mathcal{S}, \bar{\mathcal{S}}) = -\tilde{w}(\tilde{V}^-) + \sum_{v \in \mathcal{S}} \tilde{w}(v)$.

Note that the term $-\tilde{w}(\tilde{V}^-)$ is fixed and is the sum over all vertices with negative weights in $\tilde{G}$. The term $\sum_{v \in \mathcal{S}} \tilde{w}(v)$ is the total weight of all vertices in the source set $\mathcal{S}$ of $(\mathcal{S}, \bar{\mathcal{S}})$. But, $\mathcal{S} - \{s\}$ is a closed set in $\tilde{G}$ [30,20]. Thus, the cost of a cut $(\mathcal{S}, \bar{\mathcal{S}})$ in $\tilde{G}_{st}$ and the weight of the corresponding closed set in $\tilde{G}$ differ by a constant, and the source set of a minimum cut in $\tilde{G}_{st}$ corresponds to a minimum closed set in $\tilde{G}$. Since the graph $\tilde{G}$ has n vertices and $O(m_B K)$ edges, we have the following result.

Theorem 3. *Given an n_B-vertex, m_B-edge base graph B and a generated proper ordered vertex-weight graph $G = (V, E)$ with n vertices, m edges, and a height K, an optimal V-weight net in G can be computed in $T(n, m_B K)$ time.*

3.2 The Optimal VCE-Weight Net Surface Problem

In this section, we study the optimal VCE-weight net problem. Besides each vertex v in the proper ordered graph $G = (V, E)$ having a real-valued weight $w(v)$ as for the optimal V-weight net problem, each edge $e = (i_k, j_{k'}) \in E$ has a cost $c(e) = f_{ij}(|k - k'|)$, where $f_{ij}(\cdot)$ is a convex and non-decreasing function associated with the edge $(i, j) \in E_B$. We give a $T(n, \kappa m_B K)$-time algorithm for the optimal VCE-weight net problem, where $\kappa = \max_{(i_k, j_{k'}) \in E} |k - k'|$, and $T(n', m')$ is the time bound for finding a minimum s-t cut in an edge-weighted directed graph with $O(n')$ vertices and $O(m')$ edges [3].

Hochbaum [18,21] studied the minimum s-excess problem, which is a relaxation of the minimum closure problem [30,20]. Given a directed graph $G' = (V', E')$, each vertex $v' \in V'$ having an arbitrary weight $w'(v')$ and each edge $e' \in E'$ having a cost $c'(e') \geq 0$, the problem seeks a vertex subset $\mathcal{S}' \subseteq V'$ such that the cost of $\mathcal{S}'$, $\gamma(\mathcal{S}') = \sum_{v' \in \mathcal{S}'} w'(v') + \sum_{\substack{(u',v') \in E' \\ u' \in \mathcal{S}', v' \in \bar{\mathcal{S}}'}} c'(u', v')$, is minimized, where $\bar{\mathcal{S}}' = V' - \mathcal{S}'$. Instead of forcing all successors of each vertex in $\mathcal{S}'$ to be in $\mathcal{S}'$, the s-excess problem charges a penalty onto the edges leading to such immediate successors that are not included in $\mathcal{S}'$.

Below we construct a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ from the graph G, and argue the equivalence between the optimal VCE-weight net problem on G and the minimum s-excess problem on $\tilde{G}$.

As in Section 3.1, $\tilde{V} = \{\tilde{i}_k : i_k \in V\}$. The graph $\tilde{G}$ includes the same construction of the i-chains for all $i \in V_B$ and the same weight assignment $\tilde{w}(\cdot)$ for each vertex in $\tilde{G}$. The directed edges of cost $+\infty$ associated with the edge intervals in G, $I(i_k, j)$ and $I(j_k, i)$ for every $(i,j) \in E_B$ and $0 \le k < K$, are put into $\tilde{G}$ in the same way as well. The main difficulty here is how to enforce the edge penalty in $\tilde{G}$ such that when an edge is on a net $\mathcal{N}$ in G, its cost is charged appropriately to the corresponding cut in $\tilde{G}$. We overcome this difficulty by using a novel edge cost penalty embedding scheme below.

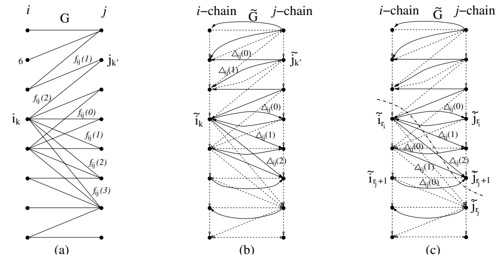

Fig. 2. (a) Part of a proper ordered graph G (associated with an edge $(i,j) \in E_B$) with weighted edges. (b) Constructing the graph $\tilde{G}$ from G (the dashed arrows are for edges with cost $+\infty$, and the solid arrows enforce the edge penalty for charging the costs of the corresponding edges that are on an arbitrary net $\mathcal{N}$ in G). (c) Illustrating the proof of Lemma 9.

For every ordered pair of adjacent columns in G, say (i-column, j-column), consider each edge interval $I(i_k, j) = (j_p, j_{p+1}, \ldots, j_{p+s})$ of $i_k \in V_i$ on V_j, with $0 \le k < K$, $p \ge 0$, $p + s < K$, and $s \ge 0$. If $k > p$, then for every $k' = 0, 1, \ldots, k - p - 1$, there is a directed edge in $\tilde{G}$ from $\tilde{i}_k$ to $\tilde{j}_{k-k'}$, called a *cost-penalty edge*, whose cost is $\tilde{c}(\tilde{i}_k, \tilde{j}_{k-k'}) = \Delta_{ij}(k')$ (see Figure 2). Herein, the functions $\Delta_{ij}(\cdot)$ are defined as follows.

$$\Delta_{ij}(0) = f_{ij}(1) - f_{ij}(0)$$
$$\Delta_{ij}(x+1) = [f_{ij}(x+2) - f_{ij}(x+1)] - [f_{ij}(x+1) - f_{ij}(x)], \; x = 0, 1, \ldots$$

Lemma 7. *Every function $\Delta_{ij}(\cdot)$ is non-negative.*

WLOG, we assume that each $f_{ij}(0) = 0$ (otherwise, we can subtract $f_{ij}(0)$ from the cost of each edge between every two adjacent i-column and j-column in G, without affecting essentially the costs of the VCE-weight nets in G). Let $\tilde{V}_0 = \{\tilde{i}_0 : i \in V_B\}$, and $\tilde{G}_0 \subseteq \tilde{G}$ be the induced subgraph of the vertex set $\tilde{V}_0$ in $\tilde{G}$. Then $\tilde{G}_0$ has several nice properties.

Lemma 8. *(1) $\tilde{G}_0$ is a strongly connected component of $\tilde{G}$. (2) $\tilde{V}_0$ is an s-excess set in $\tilde{G}$ with a finite cost. (3) For any non-empty s-excess set $\mathcal{S}$ in $\tilde{G}$ with a finite cost $\gamma(\mathcal{S})$, $\tilde{V}_0 \subseteq \mathcal{S}$. (4) $\tilde{G}_0$ corresponds to a net in G.*

Next, we demonstrate that computing an optimal VCE-weight net $\mathcal{N}^*$ in G is equivalent to finding a minimum non-empty s-excess set $\mathcal{S}^*$ in $\tilde{G}$.

Lemma 9. *A VCE-weight net $\mathcal{N}$ in G corresponds to an s-excess set $\mathcal{S} \neq \emptyset$ in $\tilde{G}$ with the same cost.*

Lemma 10. *A non-empty s-excess set $\mathcal{S}$ with a finite cost $\gamma(\mathcal{S})$ in $\tilde{G}$ defines a VCE-weight net $\mathcal{N}$ in G with $\beta(\mathcal{N}) = \gamma(\mathcal{S})$.*

Following Lemmas 9 and 10, a minimum non-empty s-excess set $\mathcal{S}^*$ in $\tilde{G}$ corresponds to an optimal VCE-weight net $\mathcal{N}^*$ in G, as stated in Lemma 11.

Lemma 11. *A minimum s-excess set $\mathcal{S}^* \neq \emptyset$ in $\tilde{G}$ defines an optimal VCE-weight net $\mathcal{N}^*$ in G.*

In the case when the minimum s-excess set in $\tilde{G}$ is empty, we apply a similar translation operation onto $\tilde{G}$ as in Section 3.1, in which $M = \tilde{w}(\tilde{V}_0)$. Thus, from Lemma 8, we obtain the lemma below.

Lemma 12. *A minimum s-excess set $\mathcal{S}^* \neq \emptyset$ in $\tilde{G}$ can be found after a translation operation on $\tilde{G}$.*

A minimum s-excess set in $\tilde{G}$ can be computed by using a minimum s-t cut algorithm, as in [21]. If the minimum s-excess set in $\tilde{G}$ thus obtained is empty, then by Lemma 12, we can first perform a translation operation on $\tilde{G}$, and then apply the s-t cut based algorithm to obtain a minimum *non-empty* s-excess set in $\tilde{G}$. As in Section 3.1, we define a directed graph $\tilde{G}_{st}$ from $\tilde{G}$. Note that the costs of the directed edges from the source s in $\tilde{G}_{st}$ are the negatives of the weights of the corresponding vertices. Let $C(\mathcal{A}, \mathcal{B})$ denote the sum of costs of all directed edges in a cut $(\mathcal{A}, \mathcal{B})$ in $\tilde{G}_{st}$ with their heads in $\mathcal{A}$ and tails in $\mathcal{B}$. Let $\mathcal{S}$ be an s-excess set in $\tilde{G}$ with a finite cost $\gamma(\mathcal{S})$. Then, we have $\gamma(\mathcal{S}) = -\tilde{w}(\tilde{V}^-) + C(\{s\} \cup \mathcal{S}, \{t\} \cup \bar{\mathcal{S}})$.

Note that the term $-\tilde{w}(\tilde{V}^-)$ is fixed. Hence, the set $\mathcal{S} \subseteq \tilde{V}$ is a set with the minimum s-excess cost in the graph $\tilde{G}$ if and only if $\mathcal{S} \cup \{s\}$ is the source set of a minimum cut in $\tilde{G}_{st}$. Since the graph $\tilde{G}_{st}$ has $O(n)$ vertices and $O(\kappa m_B K)$ edges, where $\kappa = \max_{(i_k, j_{k'}) \in E} |k - k'|$, we have the following result.

Theorem 4. *An optimal VCE-weight net in a proper ordered graph can be found in $T(n, \kappa m_B K)$ time.*

4 Applications

This section discusses two application problems for our optimal net results on proper ordered graphs: surface reconstruction and medical image segmentation. For $d = 3, 4$, we present efficient algorithms for a surface reconstruction problem in $\Re^d$ with a given topology and for d-D medical image segmentation. Our algorithms for these problems can also be extended to higher dimensions.

4.1 Surface Reconstruction in $\Re^d$ with a Given Topology

Surface reconstruction is an important problem in many applications, such as CAD, computer graphics, computer vision, and mathematical modeling. It often involves computing a piecewise linear approximation to a desired surface from a set of sample points.

Surface reconstruction problems in $\Re^3$ have been intensively studied in computer graphics and computer vision (e.g., see [8,23,27]), and computational geometry (e.g., see [5][4][9][14]). In this paper, we study a surface reconstruction problem in $\Re^d$ ($d = 3, 4$) with a given topology of the underlying surface. Considerable work has been done on surface reconstruction with some given topological constraints or structures (e.g., see [10][7][29][6]).

Our focus is on computing a piecewise linear ϵ-approximation to a "smooth" surface from sample data. Agarwal and Desikan [1] and Agarwal and Suri [2] studied the problem of computing a piecewise linear function with the minimum complexity to approximate a 3-D xy-monotone surface within a tolerance error $\epsilon > 0$. However, our criterion for measuring the quality of the output ϵ-approximate surface is different from those in [1,2].

Let a set of n sample points in $\Re^d$ ($d = 3, 4$), $\mathcal{P} = \{(x, \mathcal{P}(x)) : x \in \mathcal{D} \subset \Re^{d-1}, \mathcal{P}(x) \in \Re\}$, of some underlying smooth surface $\mathcal{S}$ be given, where $\mathcal{D}$ is an n-point set in $\Re^{d-1}$. The topology of $\mathcal{S}$ in $\Re^d$ is completely specified by a neighborhood system $\mathcal{H}$ on $\mathcal{D}$ (i.e., $\mathcal{H}$ is the edge set of a graph defined on the vertex set $\mathcal{D}$). For example, $\mathcal{H}$ is for a triangulated planar graph embedded in the plane $z = 0$ for $\mathcal{S}$ in $\Re^3$ (e.g., a Delaunay triangulation of n planar points). For two points $x, y \in \mathcal{D}$, if $(x, y) \in \mathcal{H}$, then we say the points $(x, \mathcal{P}(x))$ and $(y, \mathcal{P}(y))$ in $\Re^d$ are *adjacent* on $\mathcal{S}$. We seek a mapping for an approximation surface (whose topology has already been specified by $\mathcal{H}$ on $\mathcal{D}$), $\mathcal{F}$: $\mathcal{D} \to \Re$, that satisfies the following conditions. (For simplicity, we let $\mathcal{F}$ denote the approximation surface.)

- $\mathcal{F}$ is *ϵ-approximate*: Given a tolerance error $\epsilon > 0$, for any $x \in \mathcal{D}$, $|\mathcal{F}(x) - \mathcal{P}(x)| \leq \epsilon$. This means that the approximation surface $\mathcal{F}$ should not deviate too much from the sample data.
- $\mathcal{F}$ is *δ-smooth*: Given a smoothness system $\delta = \{\delta_{ij} \geq 0 : (x_i, x_j) \in \mathcal{H}\}$ for all pairs of adjacent points in $\mathcal{P}$, $|\mathcal{F}(x_i) - \mathcal{F}(x_j)| \leq \delta_{ij}$ holds for every $(x_i, x_j) \in \mathcal{H}$.

Such a surface is said to be *ϵ-approximate and δ-smooth.* Our goal is to compute a piecewise linear ϵ-approximate and δ-smooth surface $\mathcal{F}$ (if it exists) such that the *energy cost* $\mathcal{E}(\mathcal{F})$ of the surface, with

$$\mathcal{E}(\mathcal{F}) = \sum_{x \in \mathcal{D}} g(\mathcal{F}(x), \mathcal{P}(x)) + \sum_{(x_i,x_j) \in \mathcal{H}} f((x_i, \mathcal{F}(x_i)),\ (x_j, \mathcal{F}(x_j))),$$

is minimized, where $g(\cdot, \cdot)$ is a *data force* function that attracts $\mathcal{F}$ to the sought surface $\mathcal{S}$, and $f(\cdot, \cdot)$ represents the *internal tension* of the surface $\mathcal{F}$. In fact, the "smoother" $\mathcal{F}$ is, the smaller its internal tension becomes. We use the Euclidean distance as the internal tension function (of course, other convex functions are applicable). This model, which captures both the local and global information in determining a surface based on sample points, is inspired by the deformable model in [25][32] for image analysis. We also assume an accuracy factor τ for $\mathcal{F}$ on all points $x_i \in \mathcal{D}$, with $0 < \tau < \epsilon$.

We solve this surface reconstruction problem as an optimal *VCE*-weight net problem. Our algorithm for this problem has some details different from those in Section 3.2, which we leave to the full version due to the space limit. Let $K = 2\lceil \frac{\epsilon}{\tau} \rceil + 1$ and $\kappa = \max_{\delta_{ij} \in \delta} \frac{\delta_{ij}}{\tau}$. We have the theorem below.

Theorem 5. *Given an n-point set* $\mathcal{P} = \{(x, \mathcal{P}(x)) : x \in \mathcal{D} \subset \Re^{d-1}, \mathcal{P}(x) \in \Re\}$ *in* $\Re^d$ $(d = 3, 4)$*, a neighborhood system* $\mathcal{H}$ *defined on* $\mathcal{D}$*, a tolerance* $\epsilon > 0$*, and a smoothness system* δ*, we can compute an optimal* ϵ-approximate*,* δ-smooth *surface for* $\mathcal{P}$ *within an accuracy of* τ *in* $T(nK, \kappa|\mathcal{H}|K)$ *time.*

4.2 Medical Image Segmentation in 3-D and 4-D

Segmentation is one of the most challenging problems in image analysis. Medical image segmentation in 3-D and 4-D is essentially useful [28]. However, little work has been done on 4-D medical image segmentation. In this section, we study some 3-D and 4-D medical image segmentation problems and give efficient algorithms based on our optimal V-weight net solution.

Segmenting a 3-D volumetric image is to identify 3-D surfaces representing object boundaries in the 3-D space [31]. Let $\mathcal{I}(X, Y, Z) = \{(x, y, z) : 0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z\}$ be a 3-D image of size $X \times Y \times Z$, with each voxel having a density value. Here, Y is the number of 2-D slices in the image, X is the number of columns in each slice, and Z is the number of voxels in each column. Note that an image $\mathcal{I}(X, Y, Z)$ can be viewed as representing a cylindrical object such as a vessel, or a spherical object such as a left ventricle [33]. 3-D images for such objects can be transformed such that the sought surface $\mathcal{S}$ in $\mathcal{I}(X, Y, Z)$ for the object boundary contains exactly one voxel in each column of every xz-slice [33]. Further, since many anatomical structures (e.g., human brains, iliac arteries, and ventricles) are smooth, one may desire the sought surface $\mathcal{S}$ to be sufficiently "smooth". Precisely, given two integers M_1 and M_2, for any (x_1, y_1, z_1) and (x_2, y_2, z_2) on $\mathcal{S}$, if $|x_1 - x_2| = 1$ and $y_1 = y_2$, then $|z_1 - z_2| \leq M_1$, and if $x_1 = x_2$ and $|y_1 - y_2| = 1$, then $|z_1 - z_2| \leq M_2$. We call such a surface in $\mathcal{I}(X, Y, Z)$ an (M_1, M_2)-*smooth surface*. Note that M_1 characterizes the smoothness within each xz-slice, and M_2 specifies the smoothness across the neighboring xz-slices. A weight $w(x, y, z)$ is assigned to each voxel (x, y, z) such that the weight is inversely related to the likelihood in that the voxel may appear at the desired object boundary, which is usually determined by using simple low-level image features [33]. Thus, a segmentation problem on $\mathcal{I}(X, Y, Z)$ is to obtain an (M_1, M_2)-smooth surface $\mathcal{S}$ such that the total weight $\mathcal{W}(\mathcal{S})$ of $\mathcal{S}$, $\mathcal{W}(\mathcal{S}) = \sum_{(x,y,z) \in \mathcal{S}} w(x, y, z)$, is minimized. The previously best known exact algorithms for this problem take at least exponential time [15,31,33].

We solve the problem of segmenting an optimal (M_1, M_2)-smooth surface from $\mathcal{I}(X, Y, Z)$ by transforming it to the optimal V-weight net problem on a proper ordered graph.

Theorem 6. *Given a 3-D image* $\mathcal{I}(X, Y, Z)$ *and two integers* $M_1 \geq 0$ *and* $M_2 \geq 0$*, an optimal* (M_1, M_2)*-smooth surface in* $\mathcal{I}(X, Y, Z)$ *can be computed in* $T(n, n)$ *time, with* $n = X \times Y \times Z$.

Next, we consider segmenting 4-D medical images, which are crucial for dynamic anatomical structure studies and pathology monitoring through time [28].

Let $\mathcal{I}$ denote a 4-D image representing a sequence of T 3-D images $(\mathcal{I}_0, \ldots, \mathcal{I}_{T-1})$ acquired at T different time points $t_0 < \cdots < t_{T-1}$ with each $\mathcal{I}_i = \mathcal{I}_i(X, Y, Z)$ being (M_1, M_2)-smooth. In addition, along the time dimension, the movement of the 3-D object is assumed to be continuous and smooth, which is specified by the third smoothness parameter M_3. A sequence of 3-D surfaces in $\mathcal{I}$, $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_{T-1})$, that satisfies all three smoothness constraints is called an (M_1, M_2, M_3)-*smooth* surface in 4-D. Our goal is to find an (M_1, M_2, M_3)-smooth surface $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_{T-1})$ whose total weight $\mathcal{W}(\mathcal{S}) = \sum_{t=0}^{T-1} \sum_{(x,y,z,t) \in \mathcal{S}_t} w(x, y, z, t)$ is minimized, where $w(x, y, z, t)$ is the weight of a voxel $(x, y, z, t) \in \mathcal{I}$, assigned in a similar way as in the 3-D case. We are not aware of any previous exact algorithm for this 4-D segmentation problem.

We reduce this 4-D segmentation problem to the optimal V-weight net problem on a 4-D proper ordered graph. The base graph is a 3-D $X \times Y \times T$ grid.

Theorem 7. *Given an n-voxel 4-D image $\mathcal{I} = \{\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_{T-1}\}$, with each $\mathcal{I}_t$ being a 3-D image at a time point t, and smoothness parameters M_1, M_2 and M_3, an (M_1, M_2, M_3)-smooth surface $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{T-1}\}$ in $\mathcal{I}$ whose total weight $\mathcal{W}(\mathcal{S})$ is minimized can be computed in $T(n, n)$ time.*

Acknowledgments. The authors are very grateful to Shuang Luan for helpful discussions on the hardness of the optimal net surface problems.

References

1. P.K. Agarwal and P. Desikan, An Efficient Algorithm for Terrain Simplification, *Proc. 8th ACM-SIAM SODA*, 1997, 139-147.
2. P.K. Agarwal and S. Suri, Surface Approximation and Geometric Partitions, *SIAM J. Comput.*, 19(1998), 1016-1035.
3. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Inc., 1993.
4. N. Amenta, S. Choi, T.K. Dey, and N. Leekha, A Simple Algorithm for Homeomorphic Surface Reconstruction, *Proc. 16th ACM Symp. Comp. Geom.*, 2000, 213-222. Also *International J. of Computational Geometry and Applications*, to appear.
5. N. Amenta, S. Choi, and R.K. Kolluri, The Power Crust, Unions of Balls, and the Medial Axis Transform, *International J. of Computational Geometry and Applications*, to appear.
6. C. Bajaj and G. Xu, Modeling, and Visualization of C^1 and C^2 Scattered Function Data on Curved Surfaces, *Proc. of 2nd Pacific Conference on Computer Graphics and Applications*, 1997, 19-29.
7. R.E. Barnhill, K. Opitz, and H. Pottman, Fat Surfaces: A Trivariate Approach to Triangle-based Interpolation on Surface, *CAGD*, 9(5)(1992), 365-378.
8. F. Bernardini, C. Bajaj, J. Chen, and D. Schikore, Automatic Reconstruction of 3-D CAD Models from Digital Scans, *International J. of Computational Geometry and Applications*, 9(4-5)(1999), 327-361.

9. J.D. Boissonnat and F. Cazals, Smooth Surface Reconstruction via Natural Neighbor Interpolation of Disance Functions, *Proc. 16th ACM Symp. Comput. Geom.*, 2000, 223-232.
10. R.M. Bolle and B.C. Vemuri, On Three-dimensional Surface Reconstruction Methods, *IEEE Trans. Pat. Anal. Mach. Intell.*, 13(1)(1991), 1-13.
11. Y. Boykov, O. Veksler, and R. Zabih, Markov Random Fields with Efficient Approximations, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.
12. C. Chekuri, A. Khanna, J. Naor, and L. Zosin, Approximation Algorithms for the Metric Labeling Problem via a New Linear Programming Formulation, *Proc. 12th ACM-SIAM SODA*, 2001, 109-118.
13. D.Z. Chen, J. Wang, and X. Wu, Image Segmentation with Monotonicity and Smoothness Constraints, *Proc. 12th Annual International Symp. on Algorithms and Computation*, New Zealand, December 2001, 467-479.
14. T.K. Dey and J. Giesen, Detecting Undersampling in Surface Reconstruction, *Proc. 17th Sympos. Comput. Geom.*, 2001, 257–263.
15. R.J. Frank, D.D. McPherson, K.B. Chandran, and E.L. Dove, Optimal Surface Detection in Intravascular Ultrasound Using Multi-dimensional Graph Search, *Computers in Cardiology, IEEE*, Los Alamitos, CA, 1996, 45-48.
16. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, NY, 1979.
17. A. Gupta and E. Tardos, Constant Factor Approximation Algorithms for a Class of Classification Problem, *Proc. 32nd ACM STOC*, 2000, 652-658.
18. D.S. Hochbaum, The Pseudoflow Algorithm and the Pseudoflow-Based Simplex for the Maximum Flow Problem, Lecture Notes in Computer Science, Vol. 1412, *Proc. 6th International IPCO Conf.*, 1998, 325-337.
19. D.S. Hochbaum, Instant Recognition of Half Integrality and 2-Approximations, Lecture Notes in Computer Science, Vol. 1444, *Proc. APPROX'98*, 1998, 99-110.
20. D.S. Hochbaum, A New-old Algorithm for Minimum Cut in Closure Graphs, *Networks*, 37(4)(2001), 171-193.
21. D.S. Hochbaum, An Efficient Algorithm for Image Segmentation, Markov Random Fields and Related Problems, *J. of the ACM*, 48(2001), 686-701.
22. D.S. Hochbaum and J.G. Shanthikumar, Convex Separable Optimization Is Not Much Harder Than Linear Optimization, *J. of the ACM*, 37(4)(1990), 843-862.
23. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, Surface Reconstruction from Unorganized Points, *SIGGRAPH'92*, 1992, 71-78.
24. H. Ishikawa and D. Geiger, Segmentation by Grouping Junctions, *IEEE Conf. on Computer Vision and Pattern Recognition*, 1998, 125-131.
25. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, *Int. J. Comput. Vision*, 1(4)(1988), 321-331.
26. J. Kleinberg and E. Tardos, Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields, *Proc. 40th IEEE Symp. on Foundations of Computer Science*, 1999, 14-23.
27. R. Mencl and H. Muller, Interpolation and Approximation of Surfaces from 3-D Scattered Data Points, Research Report No. 662, Fachbereich Informatik, Lehrstuhl VII, University of Dortmund, Germany, 1997.
28. J. Montagnat and H. Delingette, Space and Time Shape Constrained Deformable Surfaces for 4D Medical Image Segmentation, Lecture Notes in Computer Science, Vol. 1935, *Proc. Medical Image Computing and Computer-Assisted Intervention*, 2000, 196-205.

29. G.M. Nielson, T.A. Foley, B. Hamann, and D. Lane, Visualizing and Modeling Scattered Multivariate Data, *IEEE CG&A*, 11(3)(1991), 47-55.
30. J.C. Picard, Maximal Closure of a Graph and Applications to Combinatorial Problems, *Management Science*, 22(1976), 1268-1272.
31. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd edition, Brooks/Cole Publishing Company, Pacific Grove, CA, 199-205.
32. D. Terzopoulos, A. Witkin, and M. Kass, Constraints on Deformable Models: Recovering 3-D Shape and Nonrigid Motion, *Artif. Intelligence*, 36(1)(1988), 91-123.
33. D.R. Thedens, D.J. Skorton, and S.R. Fleagle, Methods of Graph Searching for Border Detection in Image Sequences with Applications to Cardiac Magnetic Resonance Imaging, *IEEE Trans. on Medical Imaging*, 14(1)(1995), 42-55.
34. T. Tokuyama, personal communication, June 2001.