Boundary-Optimal Triangulation Flooding*

Richard J. Nowakowski[1] and Norbert Zeh[2]

[1] Department of Mathematics and Statistics, Dalhousie University, Halifax, NS B3H 3J5, Canada.
rjn@mathstat.dal.ca

[2] Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 1W5, Canada.
nzeh@cs.dal.ca

Abstract. Given a planar triangulation all of whose faces are initially white, we study the problem of colouring the faces black one by one so that the boundary between black and white faces as well as the number of connected black and white regions are small at all times. We call such a colouring sequence of the triangles a flooding. Our main result shows that it is in general impossible to guarantee boundary size $\mathcal{O}(n^{1-\epsilon})$, for any $\epsilon > 0$, and a number of regions that is $o(\log n)$, where n is the number of faces of the triangulation. We also show that a flooding with boundary size $\mathcal{O}(\sqrt{n})$ and $\mathcal{O}(\log n)$ regions can be computed in $\mathcal{O}(n \log n)$ time.

1 Introduction

We study the following *triangulation flooding problem* posed by Hurtado [8]: Given a planar triangulation all of whose triangles are initially white, find an ordering of the triangles such that, when colouring the triangles black in this order, the number of connected monochromatic regions and the boundary between white and black regions are small at all times.

In this paper, we provide an $\mathcal{O}(n \log n)$-time algorithm that finds an order of colouring the triangles such that, at all times, the boundary between the black and the white regions has size $\mathcal{O}(\sqrt{n})$, there are $\mathcal{O}(\log n)$ white regions, and there is only one black region. We also show that the boundary size as well as the number of white regions are best possible up to constant factors. In particular, there cannot be any floodings with boundary size $o(\sqrt{n})$ in general because, at the time when half the faces are black and half the faces are white, we would have a $\frac{1}{2}$-separator of size $o(\sqrt{n})$; but Lipton and Tarjan [11] show that the minimal size of such a separator is $\Omega(\sqrt{n})$ in the worst case. As for the number of white regions, we show that, for any $\epsilon > 0$, there exists a family of triangulations such that, if a boundary size of $\mathcal{O}(n^{1-\epsilon})$ is desired, there have to be $\Omega(\log n)$ monochromatic regions at some point during the colouring process.

The triangulation flooding problem can be rephrased as a layout problem of the dual of the triangulation: Given a 3-regular planar graph, find a linear layout $(v_1, v_2, \ldots, v_n)$ of small cutwidth and such that for every cut $(V_i, V \setminus V_i)$, the

* This work was partially supported by NSERC.

number of connected components of $G - E_i$ is small, where $V_i = \{v_1, v_2, \ldots, v_i\}$ and E_i is the set of edges with exactly one endpoint in V_i; that is, the edges in E_i cross the cut.

Layouts of small cutwidth have applications in network reliability [9], graph drawing [15], and rendering and stream processing of triangular meshes [1, 2, 7]. The cutwidth of a graph is also closely related to its search number, which is relevant in VLSI design [10] and network security [3, 6, 16]. See [4] for a comprehensive survey of graph layout problems and their applications. Next we discuss some results in this area that are closely related to our work.

It is well-known that every planar graph of maximal degree d has cutwidth $\mathcal{O}(\sqrt{dn})$ [5]; for 3-regular planar graphs, this implies that the cutwidth is $\mathcal{O}(\sqrt{n})$. Our result shows that 3-regular planar graphs have linear layouts of cutwidth $\mathcal{O}(\sqrt{n})$ that are "well-behaved" in the sense that their cuts do not partition the graph into too many connected components.

In [2], the following model for rendering a triangular mesh is studied: Vertices are pushed onto (and popped from) a vertex stack. To render a triangle, it is necessary that all three vertices of the triangle are on the stack. The question studied in [2] is to determine the stack size required to render any triangular mesh while pushing every vertex onto the stack only once. It is shown that a stack of size $\mathcal{O}(\sqrt{n})$ suffices. In order to obtain this result, a recursive separator decomposition similar to the one used in [5] for computing a layout of small cutwidth is used. Our approach for finding a colouring sequence of the triangles also uses a recursive separator decomposition. The main difference is that we use simple-cycle separators [14] in our partition and that we combine these separators carefully to keep the number of monochromatic regions small.

While keeping the number of monochromatic regions small at all times is of limited importance for mesh rendering (apart from aesthetic concerns if the rendering process is slow), it is important for encoding triangular meshes in a minimal number of bits [1, 7], which is desirable for applications that need to stream triangular meshes over communication channels of limited bandwidth such as the internet. The encoding schemes of [1, 7] achieve a compression of between 0.2 and 3.5 bits per vertex, depending on the regularity of the mesh. Similar to our triangulation flooding problem, the idea is to start at a triangle of the mesh and encode the remaining triangles one at a time so that the next triangle to be encoded is chosen from among the triangles that share an edge with a previously encoded triangle. If the third vertex of the triangle is a new vertex, the degree of the vertex is encoded. If the vertex has been seen before, the addition of the triangle may split the set of triangles yet to be encoded into two regions. This needs to be recorded using a so-called split code whose length is logarithmic in the length of the current boundary cycle. Thus, split codes are much more expensive than the encoding of new vertices and should be avoided. Using our approach, the boundary size is guaranteed to be $\mathcal{O}(\sqrt{n})$, which helps to bound the length of the split codes. We also guarantee that the number of lists on the boundary list stack in the algorithm of [1] is $\mathcal{O}(\log n)$; each list on this stack represents a boundary cycle of the already encoded region. Unfortunately,

the total number of split operations may still be linear and, other than in the algorithm of [1], the order in which the triangles are encoded cannot easily be determined from the degrees of the boundary vertices and, thus, needs to be explicitly encoded.

Finally, we want to point out the relationship between our results and the graph search problem where a number of searchers try to clean the edges of a graph all of whose edges are contaminated. In order to clean an edge, a searcher has to move along the edge; if there exists a path between a contaminated edge and a clean edge that is free of searchers, the edge is recontaminated. The search number of a graph is the minimal number of searchers that are sufficient to clean the graph. The connected search number is the minimal number of searchers that are sufficient to clean the graph and, in addition, guarantee that the graph spanned by clean edges is connected at all times. Determining the search number of a graph is NP-complete [13]. For graphs of maximal degree 3, it is known that the search number is equal to the graph's cutwidth [12]. Using the result of [5], this implies that the search number of a 3-regular planar graph is $\mathcal{O}(\sqrt{n})$. In fact, the proof of [5] immediately implies that the *connected* search number of a 3-regular planar graph is $\mathcal{O}(\sqrt{n})$. Our result proves that, for 3-regular planar graphs, $\mathcal{O}(\sqrt{n})$ searchers suffice to clean the graph while keeping the clean subgraph connected and keeping the number of maximal connected contaminated subgraphs small.

2 Preliminaries

In this section, we introduce the terminology used in this paper and discuss previous results that are used in our algorithms.

Let T be a given planar triangulation with n faces (triangles). A *colouring* c of T is an assignment of a colour $c(f) \in \{\text{black}, \text{white}\}$ to every face f of T. Colouring c is *trivial* if it colours all faces black or all faces white. The *boundary* ∂c of a non-trivial colouring c is the set of edges in T whose two adjacent triangles have different colours. We say that a colouring c is $t(n)$-*bounded* if $|\partial c| \leq t(n)$. Consider the connected components of $\mathbb{R}^2 \backslash \partial c$. Each of these components contains either only black or only white faces. Accordingly, we call such a region black or white. We define W and B to be the union of all white and black faces, respectively. We say that colouring c is $(b(n), w(n))$-*scattered* if B consists of at most $b(n)$ connected regions and W consists of at most $w(n)$ connected regions. The colouring is $s(n)$-*scattered* if the total number of monochromatic regions is at most $s(n)$; that is, a $(b(n), w(n))$-scattered colouring is also $(b(n) + w(n))$-scattered. The *black count* of a colouring c is the number of faces f of T such that $c(f) = \text{black}$. A *flooding* is a sequence $F = (c_0, c_1, \ldots, c_n)$ of colourings such that, for all $0 \leq i \leq n$, the black count of colouring c_i is i and $c_i(f) = \text{black}$ implies that $c_{i+1}(f) = \text{black}$. We say that a flooding F is $t(n)$-bounded, $(b(n), w(n))$-scattered, or $s(n)$-scattered if every non-trivial colouring in F is $t(n)$-bounded, $(b(n), w(n))$-scattered, or $s(n)$-scattered, respectively.

If both regions W and B are connected, the boundary of colouring c is a simple cycle C. We call this cycle a *simple-cycle separator* of T. Given an assignment of weights to the faces of T such that the weights of all faces sum to 1, we call cycle C a *simple-cycle ϵ-separator* if the total weight of the faces in each of the regions W and B is at most ϵ. The *size* of a simple-cycle separator is the number of edges on the cycle. Miller shows the following result.

Theorem 1 (Miller [14]). *Given a planar triangulation T with n vertices and a weight function that assigns non-negative weights to the faces of T such that the total weight of all faces is at most 1 and no face has weight more than $\frac{2}{3}$, a simple-cycle $\frac{2}{3}$-separator of size at most $\sqrt{8n}$ for T can be found in linear time.*

3 New Results

In Section 4, we prove that every planar triangulation T has an $\mathcal{O}(\sqrt{n})$-bounded $(1, \mathcal{O}(\log n))$-scattered flooding. More strongly, given a face f_0 of T, there exists such a flooding $F = (c_0, c_1, \ldots, c_n)$ such that $c_1(f_0) =$ black; that is, f_0 is the first triangle coloured black and every subsequent triangle is coloured black only after at least one of its neighbours has been coloured black. We show that such a flooding can be computed in $\mathcal{O}(n \log n)$ time.

In Section 5, we prove that the result from Section 4 is best possible in a very strong sense: For any $0 < \epsilon < 1$, there exists a family of triangulations that do not have $\mathcal{O}(n^{1-\epsilon})$-bounded $o(\log n)$-scattered floodings.

4 Floodings with Small Boundary and Low Scatter

In this section, we show how to efficiently find a flooding for a given triangulation that maintains a small boundary at all times, keeps the black region connected, and creates only few white regions at any time. The following theorem states this formally:

Theorem 2. *For every planar triangulation T with n faces and every face f_0 of T, there exists an $\mathcal{O}(\sqrt{n})$-bounded $(1, \mathcal{O}(\log n))$-scattered flooding $F = (c_0, c_1, \ldots, c_n)$ such that $c_1(f_0) =$ black. Such a flooding can be computed in $\mathcal{O}(n \log n)$ time.*

To prove Theorem 2, we first compute a recursive partition $\mathcal{P}$ of T using simple-cycle and multi-path separators and then use an ordering of the leaves of $\mathcal{P}$ to compute the order in which to colour the faces of T. The subgraphs of T in this partition are *partial triangulations* defined as follows: A partial triangulation is an embedded planar graph G with the following properties:

(i) Every face of G is marked as either *interior* or *exterior*.
(ii) All interior faces are triangles.
(iii) The union of all interior faces forms a connected region.
(iv) No two exterior faces share an edge.

The *boundary* ∂G of G is the set of edges on the boundaries of the exterior faces. To compute the partition $\mathcal{P}$, we make use of the following lemma.

Lemma 1. *Every partial triangulation G with $n \geq 2$ interior faces contains a separator of one of the following two types:*

(i) A simple cycle C of length at most $\sqrt{8n+16}$ such that at most $2n/3$ interior faces are on either side of C.
(ii) A set of simple paths $P_1, P_2, \ldots, P_k$ of total length at most $\sqrt{8n+16}$ such that each path has its endpoints, but no internal vertices, in ∂G and each of the regions into which the interior of G is partitioned by these paths contains at most $2n/3$ interior faces.

Such a separator can be found in linear time.

Proof. We triangulate the exterior faces of G and give weight $1/n$ to every interior face of G and weight 0 to every triangle produced by triangulating the exterior faces. We use Theorem 1 to find a simple-cycle separator C of the resulting triangulation T. Since the interior of T is connected, T has at most $n+2$ vertices. Hence, by Theorem 1, the cycle C has length at most $\sqrt{8n+16}$. If C contains only interior edges of G, we have Case (i). Otherwise, let $P_1, P_2, \ldots, P_k$ be the maximal sub-paths of C that consist of only interior edges of G. Every path P_i is simple because C is simple. Every region in the partition of G is completely on one side of C and thus has size at most $2n/3$. Thus, we have Case (ii). The complexity of all three steps of this procedure is linear. □

Given a partial triangulation G, a separator as in Lemma 1 partitions G into partial triangulations $G_1, G_2, \ldots, G_h$ as follows: Let $R_1, R_2, \ldots, R_h$ be the connected regions into which the interior of G is divided by the separator given by the lemma. Then the partial triangulation G_i has the faces of R_i as its interior faces. The exterior faces of G_i are bounded by the boundary edges of R_i. The following observation is an immediate consequence of the simplicity of C.

Observation 1. *If G has an exterior face, then at most one of the partial triangulations $G_1, G_2, \ldots, G_h$ does not share an edge with an exterior face of G.*

Next we define a recursive partition $\mathcal{P}$ of T into partial triangulations by repeated application of Lemma 1. The root of $\mathcal{P}$ is T. For every non-leaf partial triangulation G, its children are obtained by applying Lemma 1 to G. Every non-leaf partial triangulations has more than $\sqrt{n}$ interior faces; every leaf partial triangulation has at most $\sqrt{n}$ interior faces. To construct the desired flooding F, we compute an ordering of the leaf partial triangulations of $\mathcal{P}$ and flood each triangulation in turn. We obtain this ordering as the left-to-right ordering of the leaves of $\mathcal{P}$ obtained by ordering the children of every internal node of $\mathcal{P}$, from the root toward the leaves. Given that the children of the ancestors of a partial triangulation G in $\mathcal{P}$ have already been ordered, we give colours black and white to the boundary edges of G as follows: Let e be such an edge, let f be the face in G incident to e, let f' be the other face incident to e, and let G' be the leaf

triangulation that contains f'. Let H and H' be the ancestors of G and G' that are children of the LCA H'' of G and G'. If H' precedes H in the order of the children of H'', edge e is black; otherwise, edge e is white. To order the children of G, we now apply the following lemma.

Lemma 2. *If a partial triangulation G in $\mathcal{P}$ has f_0 as an interior face or has at least one black boundary edge, then the children of G can be arranged in an order $G_1, G_2, \ldots, G_h$ that satisfies the following two conditions:*

(i) If f_0 is an interior face of G, then f_0 is an interior face of G_1 and every child G_i, $i > 1$, shares an edge with a child G_j, $j < i$. If f_0 is not an interior face of G, then every child G_i of G shares an edge with a child G_j, $j < i$, or has a black boundary edge of G on its boundary.
(ii) There is at most one child G_i of G that does not share an edge with a child G_j, $j > i$, and does not have a white boundary edge of G on its boundary.

Proof. We call a boundary edge of a child G_i of G black, white, or internal depending on whether it is a black or white boundary edge of G or an edge between two children of G. We partition the children of G into three groups: If f_0 is an interior face of G, then $\mathcal{G}_2$ contains the child G_1 that contains f_0 and all remaining faces are in $\mathcal{G}_3$. $\mathcal{G}_1$ is empty in this case. If f_0 is not an interior face of G, then $\mathcal{G}_3$ contains all children whose boundary edges are white or internal. $\mathcal{G}_2$ contains all children that have at least one black boundary edge and at least one boundary edge that is white or is shared with a child in $\mathcal{G}_3$. Group $\mathcal{G}_1$ contains the remaining children, that is, those whose boundary edges are either black or shared with other children in $\mathcal{G}_1$ and $\mathcal{G}_2$. If groups $\mathcal{G}_2$ and $\mathcal{G}_3$ are empty, we move an arbitrary child from group $\mathcal{G}_1$ to group $\mathcal{G}_2$.

In the ordering of the children of G, the children in $\mathcal{G}_1$ precede the children in $\mathcal{G}_2$, which in turn precede the children in $\mathcal{G}_3$. The children in $\mathcal{G}_2$ are arranged in no particular order. The children in $\mathcal{G}_3$ are arranged so that each such child G_i shares an edge with a child G_j, $j < i$. Since $\mathcal{G}_1 \cup \mathcal{G}_2$ is non-empty, the connectivity of the interior of G implies the existence of such an ordering. The children in $\mathcal{G}_1$ are arranged so that each such child G_i shares an edge with a child G_j, $j > i$. Again, the non-emptiness of $\mathcal{G}_2 \cup \mathcal{G}_3$ and the connectivity of the interior of G imply the existence of such an ordering.

If f_0 is in G, then Condition (i) is trivially satisfied by the constructed ordering. So assume that f_0 is not in G. Then every face in $\mathcal{G}_1 \cup \mathcal{G}_2$ has a black boundary edge and every child G_i in $\mathcal{G}_3$ shares an edge with a child G_j, $j < i$. To see that Condition (ii) is satisfied, observe that, by Observation 1, at most one child in $\mathcal{G}_3$ does not have a white boundary edge. Every child G_i in $\mathcal{G}_1$ shares an edge with a child G_j, $j > i$; every child in $\mathcal{G}_2$ shares an edge with a child G_j in $\mathcal{G}_3$ or has a white boundary edge. □

Given the ordering of the leaf partial triangulations of $\mathcal{P}$ produced by top-down application of Lemma 2, we flood these partial triangulations in this order. The first leaf partial triangulation contains f_0 and is flooded starting from f_0. Every subsequent triangulation G is flooded starting from a face that has a black

edge of G on its boundary. A simple inductive argument shows that such a face always exists. (Details appear in the full paper.) The following lemma shows how to flood leaf triangulations.

Lemma 3. *A leaf triangulation G can be flooded starting from any face f of G so that the boundary between black and white interior faces of G has size at most $\sqrt{n}+2$, the number of black regions interior to G is one at all times, and the number of white regions interior to G never exceeds $\log\sqrt{n}$.*

Proof. We keep the black region connected by colouring f black and subsequently colouring a face f' black only if at least one of its adjacent faces is black. The bound on the boundary size follows immediately from the fact that G has at most $\sqrt{n}$ interior faces. To guarantee that there are never more than $\log\sqrt{n}$ white regions, we choose the order in which to colour triangles using the following recursive procedure: Let R be the current region to be coloured. After colouring f black, R is the interior of G minus f. We choose a face f' in R adjacent to a black face and colour it black. Face f' divides R into at most two connected regions. We flood each of them recursively, first the smaller one, then the bigger one. If we consider this recursive partition of the interior of G into white regions, then at any time only $\log\sqrt{n}$ ancestors of the current region can have siblings waiting to be flooded because each such sibling is of at least the same size as the corresponding ancestor of the current region. Every such sibling is completely contained in a white region of the current colouring of G, and every white region consists of at least one such sibling. Hence, there are at most $\log\sqrt{n}$ white regions at any time. □

The next three lemmas establish that the computed flooding F has the desired properties.

Lemma 4. *Every colouring in F defines only one black region.*

Proof. This is obvious, once we observe that F floods the children of any partial triangulation in $\mathcal{P}$ in left-to-right order. Hence, for every leaf partial triangulation G, a black boundary edge is one that already has an incident black triangle at the time when G is being flooded. The flooding of G starts at a triangle incident to such an edge and keeps the black region connected, by Lemma 3. □

Lemma 5. *The boundary size of every colouring in F is $\mathcal{O}(\sqrt{n})$.*

Proof. Consider the leaf partial triangulation G currently being flooded. Only ancestors of G can be bichromatic. Thus, the boundary of the black region is part of the separators computed at these ancestors. Since the separator of an ancestor with n' internal faces has size $\mathcal{O}(\sqrt{n'})$ and the sizes of these ancestors are geometrically decreasing, the proper ancestors of G contribute $\mathcal{O}(\sqrt{n})$ to the boundary size of the colouring. By Lemma 3, G itself also contributes $\mathcal{O}(\sqrt{n})$. □

Lemma 6. *Every colouring in F defines at most $\mathcal{O}(\log n)$ connected white regions.*

Proof. Consider the leaf partial triangulation G currently being flooded. As already observed, only ancestors of G can be bichromatic. It suffices to prove that every ancestor contributes at most one white region to the current colouring. This is sufficient because there are only $\mathcal{O}(\log n)$ such ancestors and the flooding of G itself contributes only $\mathcal{O}(\log n)$ white regions to the current colouring, by Lemma 3.

Our claim follows from Lemma 2. In particular, the second condition implies that at most one of the white children of a partial triangulation G is not included in the same white region as some white sibling of an ancestor of G. □

Partition $\mathcal{P}$ can be computed in $\mathcal{O}(n \log n)$ time by repeated application of Lemma 1. In particular, every face of G is contained in exactly one partial triangulation per level of $\mathcal{P}$, so that the cost of computing every level of $\mathcal{P}$ is $\mathcal{O}(n)$. There are at most $\log_{3/2} n = \mathcal{O}(\log n)$ levels in $\mathcal{P}$. Once partition $\mathcal{P}$ is given, the remainder of the algorithm is easily carried out in linear time. (Details appear in the full paper.)

We have shown that an $\mathcal{O}(\sqrt{n})$-bounded $(1, \mathcal{O}(\log n))$-scattered flooding of a planar triangulation can be computed in $\mathcal{O}(n \log n)$ time. This completes the proof of Theorem 2.

5 Families of Hard Triangulations

As already mentioned in the introduction, it is in general impossible to obtain an $o(\sqrt{n})$-bounded flooding for a given triangulation. In this section, we prove that the scatter of the flooding in Theorem 2 is also optimal, even if we relax the bound on the boundary size to be $\mathcal{O}(n^{1-\epsilon})$, for any $0 < \epsilon < 1$, and we allow more than one black region, that is, we are only interested in the total number of monochromatic regions. The following theorem states this formally:

Theorem 3. *For any $0 < \epsilon < 1$, there exists a family of triangulations that do not have $\mathcal{O}(n^{1-\epsilon})$-bounded $o(\log n)$-scattered floodings.*

To prove Theorem 3, we show how to construct, for any pair of parameters n and ϵ, a triangulation T of size $\mathcal{O}(n)$ as in Theorem 3. Then we define a tree X that captures the structure of T; prove that the scatter of any $\mathcal{O}(n^{1-\epsilon})$-bounded flooding of T cannot be less than the minimal scatter of a flooding of X, defined below; and finally prove that X does not have an $o(\log n)$-scattered flooding.

To construct triangulation T, we place triangles in the plane and then triangulate the regions bounded by these triangles. Every such triangle Δ is said to be at a level (i, j). We compare levels lexicographically; that is, $(i, j) < (i', j')$ if either $i < i'$ or $i = i'$ and $j < j'$.

The first triangle we place is a bounding triangle at level $(0, 0)$. All subsequent triangles are placed inside this triangle. After placing the bounding triangle, we iteratively place two level-$(i + 1, 0)$-triangles into every level-$(i, 0)$ triangle until the last level $(\ell, 0)$ we produce contains between $n^{\epsilon'}$ and $2n^{\epsilon'}$ triangles, where $\epsilon' = \epsilon/2$. Observe that $\ell \geq \log(n^{\epsilon'}) = \epsilon' \cdot \log n$. We call levels $(0, 0), (1, 0), \ldots, (\ell, 0)$

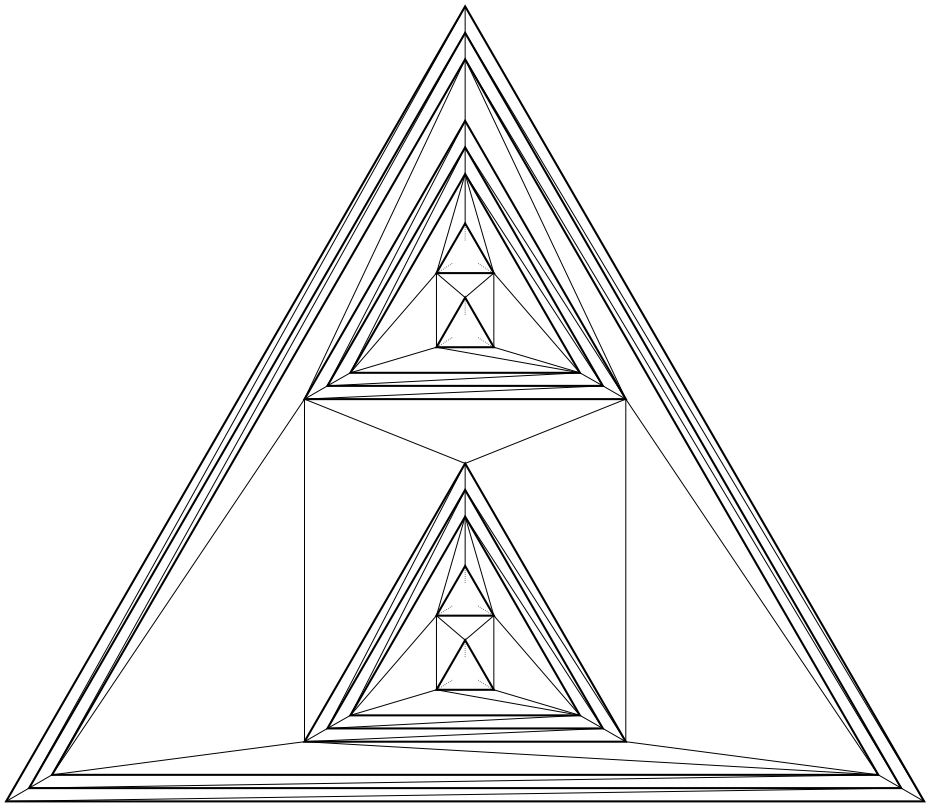

Fig. 1. The construction of a triangulation as in Theorem 3. The nested triangles in the hierarchy are shown in bold. Thin edges show a possible triangulation obtained from this hierarchy of triangles.

branching levels. Now we continue by placing triangles at *non-branching levels* (i, j), $j > 0$. Let $m = \lceil n^{1-\epsilon'} \rceil$. Then we place one level-(i, j) triangle Δ, for $0 \leq i \leq \ell$ and $1 < j < m$, into every level-$(i, j-1)$ triangle Δ' so that the level-$(i+1, 0)$ triangles contained in Δ' are also contained in Δ. We obtain our final triangulation T by triangulating the regions between triangles at consecutive levels (see Figure 1). To avoid confusion, we refer to the nested triangles we place during the construction of T as *triangles* and to the triangular faces of T as *faces*. Our first observation proves that T has the desired number of faces.

Observation 2. *Triangulation T has $\mathcal{O}(n)$ faces.*

Proof. The number of triangles at level $(\ell, 0)$ is at most $2n^{\epsilon'}$. The number of triangles at all branching levels is at most twice that. Every triangle at a branching level $(i, 0)$ contains $m = \lceil n^{1-\epsilon'} \rceil$ triangles at non-branching levels (i, j). Hence, the total number of triangles is $\mathcal{O}(n^{\epsilon'} \cdot n^{1-\epsilon'}) = \mathcal{O}(n)$. Since every triangle con-

tributes 3 vertices to the vertex set of T, T has $\mathcal{O}(n)$ vertices and hence, by Euler's formula, $\mathcal{O}(n)$ faces. □

We call the region bounded by two triangles Δ and Δ' such that Δ' is contained in Δ and these triangles are at levels (i, j) and $(i, j + 1)$, for some i and j, a *ring*. If Δ and Δ' are at levels $(i, 0)$ and $(i, m - 1)$, respectively, we call the region bounded by Δ and Δ' a *tube*. The next observation establishes that every $\mathcal{O}(n^{1-\epsilon})$-bounded colouring of T has to contain a sufficient number of monochromatic rings, which is the key to lower-bounding the scatter of any $\mathcal{O}(n^{1-\epsilon})$-bounded flooding of T by the scatter of floodings of certain trees.

Observation 3. *Every $\mathcal{O}(n^{1-\epsilon})$-bounded colouring c of T colours at least one ring in every tube completely white or completely black.*

Proof. Assume that there is a tube in T none of whose rings is monochromatic; that is, every ring contains at least one black and at least one white face. Then there is at least one boundary edge between black and white faces in every ring in this tube. Since there are $\lceil n^{1-\epsilon'} - 1 \rceil = \omega(n^{1-\epsilon})$ rings per tube, this contradicts the assumption that c is $\mathcal{O}(n^{1-\epsilon})$-bounded. □

Next we define a tree X whose floodings lower-bound the scatter of any $\mathcal{O}(n^{1-\epsilon})$-bounded flooding of T. We begin by constructing a tree X_0: Tree X_0 contains one node per region that is bounded by a set of triangles and does not contain any triangle. There is an edge between two nodes if the corresponding regions have a common triangle on their boundaries. Tree X is obtained from X_0 by replacing every maximal path whose internal nodes have degree two with an edge. The nodes of X represent the exterior triangle of T, the regions that are bounded by three triangles, and the triangles that do not contain any other triangles. The edges of X represent the tubes of T.

For every $\mathcal{O}(n^{1-\epsilon})$-bounded colouring c of T, we define a colouring c' of the edges of X as follows: By Observation 3, every tube of T contains either a black or a white ring, or both. We colour the corresponding edge of X black if there is a black ring in the tube and white if all rings in the tube are either white or bichromatic.

A *monochromatic subtree* of X under colouring c' is a maximal subtree all of whose edges have the same colour. We call a colouring of X $s(n)$-scattered if it defines at most $s(n)$ monochromatic subtrees of X. A flooding of X is defined analogously to a flooding of T, the only difference being that we colour edges. A flooding is $s(n)$-scattered if all its colourings are $s(n)$-scattered.

The next lemma proves that the number of monochromatic subtrees of X defined by c' is a lower bound on the number of monochromatic regions of T defined by colouring c.

Lemma 7. *For every $\mathcal{O}(n^{1-\epsilon})$-bounded colouring c of T with k monochromatic regions, the corresponding colouring c' of X defines at most k monochromatic subtrees of X.*

Proof. Let k' be the number of monochromatic subtrees of X under colouring c'. We prove that $k \geq k'$. If there are at most two monochromatic subtrees, the lemma is trivial because c cannot define less than one region of T and if X has two monochromatic subtrees, then T has two regions of different colours. So assume that there are at least three monochromatic subtrees in X. We prove that, for each of these subtrees, there is at least one monochromatic region in T.

Consider two black subtrees X_1 and X_2 and two edges $e_1 \in X_1$ and $e_2 \in X_2$. Since $X_1 \neq X_2$, there has to be at least one white edge e' on the path connecting e_1 and e_2. Since edges e_1 and e_2 are black, there is at least one black face in each of the tubes represented by these edges. Call these faces f_1 and f_2. Edge e' is white because there is no black ring in the tube represented by e'. Hence, by Observation 3, this tube contains at least one white ring. Since e' is on the path from e_1 to e_2 in X, this ring separates f_1 from f_2. Therefore, f_1 and f_2 belong to different black regions of T. This proves that, for every black subtree of X, there is at least one black region in T. A symmetric argument shows that the number of white regions of T is at least the number of white subtrees of X. □

The following is an easy consequence of Lemma 7.

Corollary 1. *If T has an $\mathcal{O}(n^{1-\epsilon})$-bounded $s(n)$-scattered flooding, then X has an $s(n)$-scattered flooding.*

We have to show that X does not have an $o(\log n)$-scattered flooding. Observe that, by the construction of T and X, X has 2^h nodes, for some integer h.

Lemma 8. *Let X have 2^h nodes. Then X does not have an $\lfloor h/2 \rfloor$-scattered flooding.*

Proof. Observe that, if we root X at the node representing the exterior face, X is a complete binary tree with 2^{h-1} leaves whose root has an extra parent. We call such a tree a *hanger*. Next we use induction on h to prove that a hanger with 2^h nodes does not admit an $\lfloor h/2 \rfloor$-scattered flooding.

If $h \leq 3$, the claim holds trivially because every colouring is at least 1-scattered and every flooding of a tree with more than one edge is at least 2-scattered. So assume that $h > 3$ and that the claim holds for $h - 2$. Let $F = (c_0, c_1, \ldots, c_{2^h-1})$ be a flooding of a hanger H with 2^h nodes. By removing the root of H, its child, and the three edges incident to these vertices, we partition H into two complete subtrees; both subtrees can be partitioned into two hangers with 2^{h-2} nodes. We denote these hangers as H_1, H_2, H_3, H_4. Since the restriction of F to any H_j is a flooding of H_j with duplicate consecutive colourings, there have to be colourings $c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}$, $i_1 < i_2 < i_3 < i_4$, such that the restriction of c_{i_j} to H_j defines at least $\lfloor h/2 \rfloor$ monochromatic subtrees of H_j. If, for any colouring c_{i_j}, $H - H_j$ is not monochromatic, c_{i_j} defines at least $\lfloor h/2 \rfloor + 1$ monochromatic subtrees of H. Now we observe that, for colouring c_{i_2}, $H - H_2$ cannot be monochromatic. Indeed, c_{i_2} succeeds c_{i_1}, so H_1 contains at least one edge that is coloured black by c_{i_2}; c_{i_2} precedes c_{i_3}, so H_3 contains at least one edge that is coloured white by c_{i_2}. Hence, H does not have an $\lfloor h/2 \rfloor$-scattered flooding. □

To complete the proof of Theorem 3, it suffices to observe that X has $\Theta(n^{\epsilon'})$ nodes. By Lemma 8, this implies that X does not have an $o(\log n)$-scattered flooding and, hence, by Corollary 1, that T does not have an $\mathcal{O}(n^{1-\epsilon})$-bounded $o(\log n)$-scattered flooding.

Acknowledgements. We would like to thank two anonymous referees for pointing out the relationship of the triangulation flooding problem to the large number of problems discussed in the introduction.

References

1. P. Alliez and M. Desbrun. Valence-driven connectivity encoding for 3d meshes. In *Proceedings of Eurographics*, pages 480–489, 2001.
2. R. Bar-Yehuda and C. Gotsman. Time/space tradeoffs for polygon mesh rendering. *ACM Transactions on Graphics*, 15(2):141–152, 1996.
3. L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro. Capture of an intruder by mobile agents. In *Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures*, pages 200–209, 2002.
4. J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34(3):313–356, 2002.
5. K. Diks, H. N. Djidjev, O. Sykora, and I. Vrto. Edge separators of planar and outerplanar graphs with applications. *Journal of Algorithms*, 14:258–279, 1993.
6. M. Franklin, Z. Galil, and M. Yung. Eavesdropping games: A graph theoretic approach to privacy in distributed systems. *Journal of the ACM*, 47(2):225–243, 2000.
7. C. Gotsman. On the optimality of valence-based connectivity coding. *Computer Graphics Forum*, 22(1):99–102, 2003.
8. F. Hurtado. Open problem posed at the 15th Canadian Conference on Computational Geometry. 2003.
9. D. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM Journal on Computing*, 29(2):492–514, 1999.
10. T. Lengauer. Black-white pebble games and graph separation. *Acta Informatica*, 16(4):465–475, 1981.
11. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
12. F. Makedon and H. Sudborough. Minimizing width in linear layout. In *Proceedings of the 10th International Colloquium on Automata, Languages, and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 478–490. Springer-Verlag, 1983.
13. N. Megiddo, S. Hakimi, M. Garey, D. Johnson, and C. Papadimitriou. The complexity of searching a graph. *Journal of the ACM*, 35(1):18–44, 1988.
14. G. L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *Journal of Computer and System Sciences*, 32:265–279, 1986.
15. P. Mutzel. A polyhedral approach to planar augmentation and related problems. In *Proceedings of the 3rd Annual European Symposium on Algorithms*, volume 979 of *Lecture Notes in Computer Science*, pages 497–507, 1995.
16. E. H. Spafford and D. Zamboni. Intrusion detection using autonomous agents. *Computer Networks*, 34(4):547–570, 2000.