Draining a Polygon
–or–
Rolling a Ball out of a Polygon*

Greg Aloupis[†] Jean Cardinal[†] Sébastien Collette[‡†]
Ferran Hurtado[§] Stefan Langerman[¶] Joseph O'Rourke[‖]

September 15, 2008

Abstract

We introduce the problem of draining water (or balls representing water drops) out of a punctured polygon (or a polyhedron) by rotating the shape. For 2D polygons, we obtain combinatorial bounds on the number of holes needed, both for arbitrary polygons and for special classes of polygons. We detail an $O(n^2 \log n)$ algorithm that finds the minimum number of holes needed for a given polygon, and argue that the complexity remains polynomial for polyhedra in 3D. We make a start at characterizing the *1-drainable* shapes, those that only need one hole.

1 Introduction

Imagine a closed polyhedral container P partially filled with water. How many surface point-holes are needed to entirely drain it under the action of gentle rotations of P? It may seem that one hole suffices, but we will show that in fact sometimes $\Omega(n)$ holes are needed for a polyhedron of n vertices. Our focus is on variants of this problem in 2D, with a brief foray in Section 4 into 3D. We address the relationship between our problem and injection-filling of polyhedral molds [BvKT98] in Section 2.1.

A second physical model aids the intuition. Let P be a 2D polygon containing a single small ball. Again the question is: How many holes are needed

*This is a revised and expanded version of [ACC+08].

[†] [greg.aloupis,jcardin,secollet,slanger]@ulb.ac.be Université Libre de Bruxelles (ULB), CP212, Bld. du Triomphe, 1050 Brussels, Belgium.

[‡] Chargé de Recherches du FRS-FNRS.

[§] ferran.hurtado@upc.edu Universitat Politècnica de Catalunya, Jordi Girona 1–3, E-08034 Barcelona, Spain.

[¶] Chercheur Qualifié du FRS-FNRS.

[‖] orourke@cs.smith.edu Smith College, Northampton, MA 01063, USA.

to ensure that the ball, regardless of its initial placement, will escape to the exterior under gentle rotation of P? Here the ball is akin to a single drop of water. We will favor the ball analogy, without forgetting the water analogy.

Models. We consider two models, the (gentle) *Rotation* and the *Tilt* models. In the first, P lies in a vertical xy-plane, and gravity points in the $-y$ direction. The ball B sits initially at some convex vertex v_i; vertices are labeled counterclockwise (ccw). Let us assume that v_i is a local minimum with respect to y, i.e., both v_{i-1} and v_{i+1} are above v_i. Now we are permitted to rotate P in the vertical plane (or equivalently, alter the gravity vector). In the Rotation model, B does not move from v_i until one of the two adjacent edges, say $e_i = v_iv_{i+1}$, turns infinitesimally beyond the horizontal, at which time B rolls down e_i and falls under the influence of gravity until it settles at some other convex vertex v_j. For example, in Figure 1, B at v_4 rolls ccw when v_4v_5 is horizontal, falls

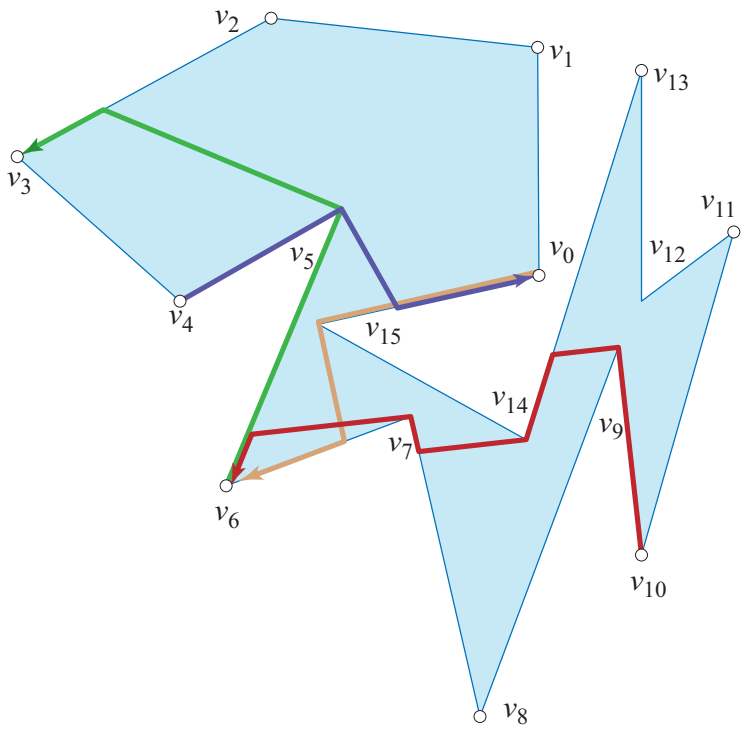

Figure 1: Polygon with several ball paths.

to edge $v_{15}v_0$, and comes to rest at v_0. Similarly, B at v_{10} rolls clockwise (cw) to v_6 after three falls. Note that all falls are parallel, and (arbitrarily close to) orthogonal to the initiating edge (in the Rotation model). After B falls to an edge, it rolls to the endpoint on the obtuse side of its fall path.

Notice that the exact gravity vector is ill-defined in this Rotation model, in that falls are "infinitesimally beyond" 90° with respect to the current edge. To

overcome this issue, we could assume that the angle is either exactly 90°, or equal to $90° + \epsilon$. Our proofs apply in both models, and so we opt to follow the convention that the angle is exactly 90° for simplicity of exposition.

The only difference in the Tilt model is that any gravity vector may be selected. Any gravity vector $\overrightarrow{g}$ within the "wedge" from 90° before (cw) $u =$

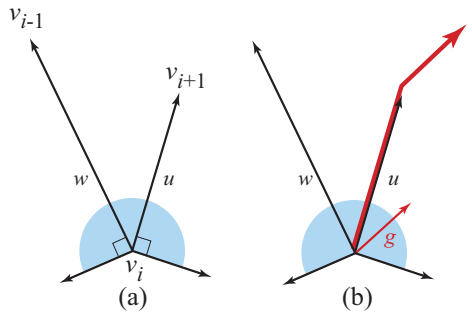

Figure 2: Gravity vectors in the indicated wedge of angles cause the ball to roll from v_i.

$v_{i+1} - v_i$, ccw around to 90° after (ccw) $w = v_{i-1} - v_i$, will initiate a departure of B from v_i; see Figure 2(a). Those $\overrightarrow{g}$ between u and v cause the ball to depart along $\overrightarrow{g}$, whereas those $\overrightarrow{g}$ within the ±90° ranges at the extremes of the wedge cause the ball to depart along one of the incident edges, as in Figure 2(b). For example, in Figure 1, B at v_4 rolls to $\{v_0, v_3\}$ in the Rotation model, but can roll to $\{v_0, v_1, v_2, v_3\}$ in the Tilt model. The Rotation model more accurately represents physical reality, for rain drops or for balls. The Tilt model mimics various ball-rolling games (e.g., Labyrinth) that permit quickly "tilting" the polygon/maze from the horizontal so that any departure vector from v_i can be achieved. We emphasize that, aside from this departure difference, the models are identical. In particular, inertia is ignored, and rotation while the ball is "in-flight" is forbidden (otherwise we could direct B along any path).

There are two "degenerate" situations that can occur. If B falls exactly orthogonal to an edge e, we arbitrarily say it rolls to the cw endpoint of e. If B falls directly on a vertex, both of whose edges angle down with respect to gravity, we stipulate that it rolls to the cw side.

Questions. Given P, what is the minimum number of point-holes needed to guarantee that any ball, regardless of starting position, may eventually escape to the exterior of P under some sequence of rotations/tilts? Our main result is that this number can be determined in $O(n^2 \log n)$ time. In terms of combinatorial bounds, we show that some polygons require $\lfloor n/6 \rfloor$ and $\lfloor n/7 \rfloor$ holes (in the Rotation/Tilt models respectively), but $\lfloor n/4 \rfloor$ holes always suffice. We make a start at characterizing the *1-drainable* polygons, those that only need one hole. Finally we argue that the minimum number of holes can be computed for a 3D polyhedron in polynomial time.

2 Traps: Combinatorial Bounds

We start by exhibiting polygons that need $\Omega(n)$ holes to drain. The basic idea is shown in Figure 3(a) for the Rotation model. We create *traps* with 5 vertices

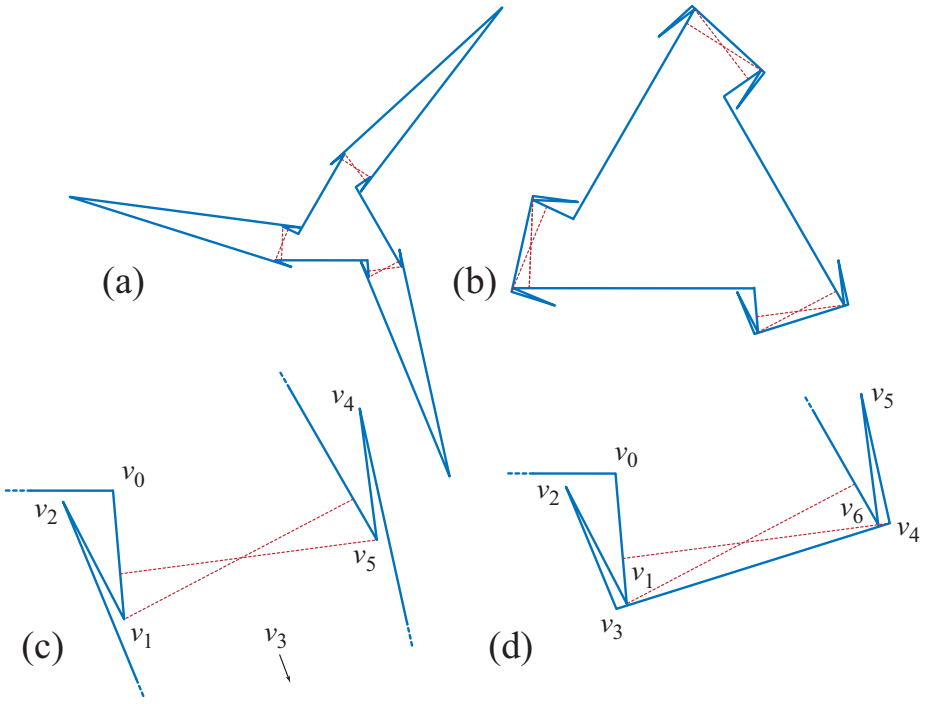

Figure 3: (a) Trap for Rotation model. (b) Trap for Tilt model. (c,d) Details of traps.

forming an "arrow" shape, connected together around a convex polygonal core so that 6 vertices are needed per trap. A ball in v_4 rolls to fall on edge v_0v_1, but because of the slightly obtuse angle of incidence, rolls to v_2; and symmetrically, v_2 leads to v_4. So there is a cycle (defined precisely in Section 3) that "traps" the ball among $\{v_2, v_3, v_4\}$ and isolates it from the other two traps. Therefore three holes are required to drain this polygon. In the Tilt model, Figure 3(a) only needs one hole, because B could roll directly from v_3 through the $v_1{-}v_5$ "gap." However, the polygon in Figure 3(b) requires 3 holes. Here the range of effective gravity tilt vectors from v_4 is so narrow that the previous analysis holds. These examples establish necessity for general polygons:

Proposition 1 *In the Rotation model, $\lfloor n/6 \rfloor$ holes are sometimes necessary to drain an n-vertex polygon. In the Tilt model, $\lfloor n/7 \rfloor$ are sometimes necessary.*

We establish an upper bound of $\lfloor n/4 \rfloor$ in Section 3.1 (Lemma 12) after introducing the necessary machinery. Before moving to that machinery, we explore

a number of specialized polygonal shapes to continue developing intuition.

Orthogonal Polygons.

Proposition 2 *In the Rotation model, $\lfloor n/12 \rfloor$ holes are sometimes necessary to drain an n-vertex orthogonal polygon, and in the Tilt model, $\lfloor n/28 \rfloor$ are sometimes necessary. In either model, $\lfloor (n+4)/8 \rfloor$ holes suffice.*

Proof: The example that establishes the Rotation lower bound of $\lfloor n/12 \rfloor$ is shown in Figure 4(a), which can be viewed as an orthogonal version of Figure 3(a). For example, a ball at v_1 will roll to v_2, "skipping over" the corridor C. The polygon is completed by connecting k traps to a rectangular "bus," forming a polygon of $n = 12k + 4$ vertices, which needs k holes.

The Tilt lower bound of $\lfloor n/28 \rfloor$ is established by a similar but more complex trap, shown in Figure 4(b). A ball at v_1 can only reach v_2 with a narrow range of angles, and that range of angles skips over the corridor C. A ball on e_1 can only move under leftward gravity vectors, and so cannot exit from the left "lobe" of the shape. Similar reasoning from e_2 and e_3 shows that a ball anywhere in the trap can never reach C. Again connecting k traps to a rectangle creates a polygon of $n = 28k + 4$ vertices that needs k holes.

The upper bounds follow by classifying all convex vertices as one of four types $\{q_0, q_1, q_2, q_3\}$, depending on the quadrant that includes the polygon interior in a neighborhood of the vertex. An orthogonal polygon of n vertices has exactly $(n+4)/2$ convex vertices, so there is one quadrant q^* type that occurs $\leq (n+4)/8$ times. We claim that a ball at any q_i-vertex can roll to some q_j-vertex, for each $j \neq i$, in either model. It clearly suffices to establish this in the Rotation model, for we can always choose a tilt to mimic the Rotation behavior. Let q_0 be the quadrant including the angles from 0 to $\pi/2$ (measured from the $+x$-axis), and q_1 including $\pi/2$ to π. A ball at a q_0-vertex rolling rightward (along the ccw incident edge) will either immediately reach a q_1-vertex at the other end of that edge, or will fall to a lower q_0-vertex (because of the cw convention for orthogonal falls; see Figure 4(a)). Eventually a lowest edge must be reached where it can roll directly to a q_1-vertex. Repeating the argument, q_1 can reach q_2, etc., so q_i can reach any q_j.

The upper bound is completed by piercing each of the $\lfloor (n+4)/8 \rfloor$ vertices of type q^*, which suffices to drain P. □

Star-Shaped Polygons.

Proposition 3 *In the Rotation model, $\lfloor n/20 \rfloor$ holes are sometimes necessary to drain an n-vertex star-shaped polygon.*

Proof: We first show that, for any $k > 1$, there is a star-shaped n-gon that requires k holes to drain. Figure 5(a) shows the construction for k=2. A ball in "well" labeled 0 rolls ccw to well $2, 4, \ldots$, and cw also to an even-numbered well. A ball in well 1 rolls ccw and cw to odd-numbered wells. Thus, two holes

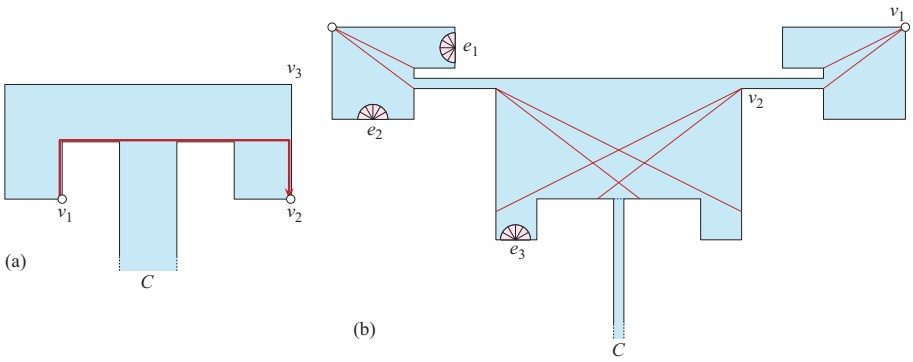

Figure 4: Orthogonal polygon traps: (a) Rotation model; (b) Tilt model. Note that, in (a), the ball rolls to v_2 rather than v_3 because orthogonal falls roll cw.

are needed when there are an even number of wells. The construction can be generalized to any $k > 1$.

The example in (a) of the figure has $5 \times 12 = 60$ vertices and so only establishes a $\lfloor n/30 \rfloor$ lower bound. Reducing each well to four vertices and arranging them around a decagon (Figure 5(b)) permits the same type of even/odd trap with only $n = 40$ vertices, establishing the claimed $\lfloor n/20 \rfloor$ bound. □

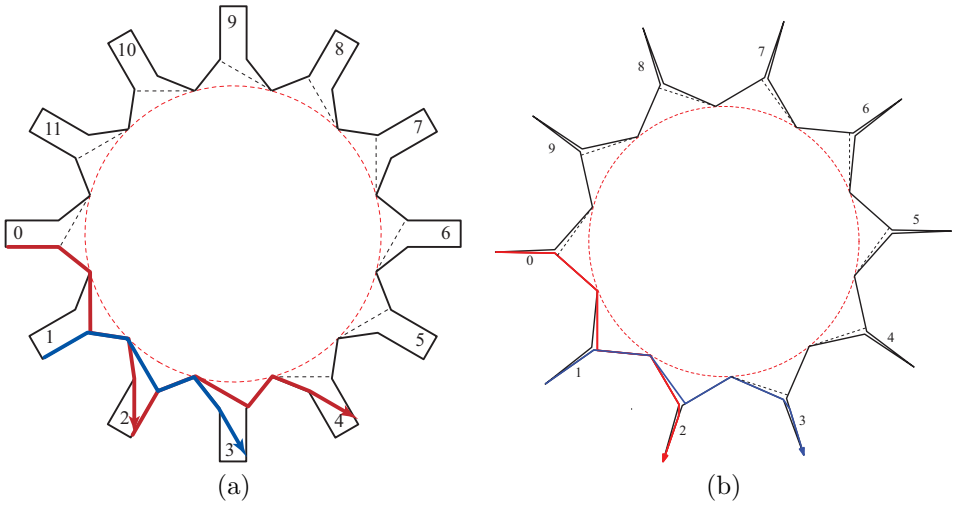

Figure 5: Star-shaped polygons that requires two holes to drain in the Rotation model. (a) Based on a dodecagon; (b) Based on a decagon.

We have no bounds for star-shaped polygons under the Rotation model, and we leave establishing nontrivial upper and lower bounds as open problems (Section 5).

2.1 1-Drainable Shapes

Define a *k-drainable* polygon as one that can be drained with k holes but not with $k-1$ holes. For example, Figure 1 is 1-drainable with a hole at v_6. The 1-drainable shapes are especially interesting. Note that these shapes do depend on the model: Figure 3(a) is 1-drainable in the Tilt model but 3-drainable in the Rotation model.

Our definition of k-drainable polygons is inspired by the k-fillable polygons of [BT94][BvKT98], those mold shapes that can be filled with liquid metal poured into k pin holes. Despite the apparent inverse relationship between filling and draining, the two concepts are rather different. In particular, there are star-shaped polygons k-drainable in the rotation model (Proposition 3), but Theorem 7.2 of [BvKT98] shows that these are all "2-fillable with re-orientation." Also, there are 1-drainable polygons that are k-fillable (with or without re-orientation). In Figure 6, a hole at v_0 suffices to drain each spiral piece, because a ball at, say, v_i, can roll to v_0 (in several moves). However, the tip of each spiral needs to be filled separately, and so the shape is only k-fillable.

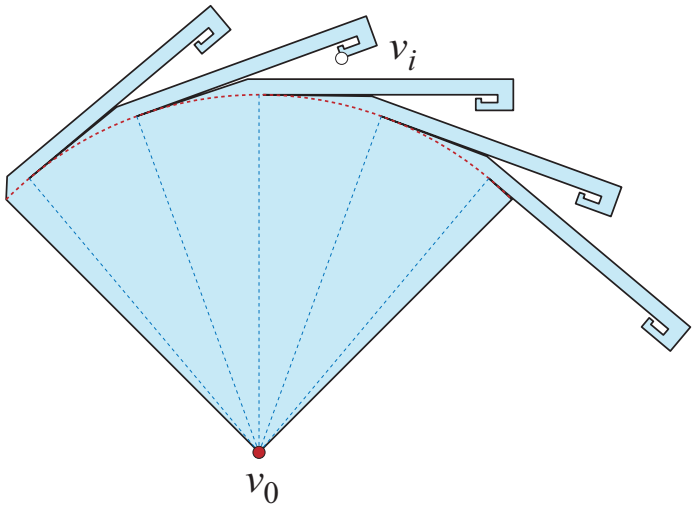

Figure 6: A polygon 1-drainable (in either the Rotation or Tilt model), but k-fillable, with or without re-orientation.

The essence of the reason for the difference between filling and draining is that, without re-orientation, k-filling fixes the orientation throughout, and the model of k-filling with re-orientation in [BvKT98] demands that the liquid metal harden after a partial filling, before turning to a new orientation. In contrast, our draining model permits considerably more rotational freedom.

We have only made limited progress in classifying the 1-drainable shapes. We defer to Section 3.2 establishing that "fans," special star-shaped polygons, are 1-drainable, as that proof uses concepts in Lemma 6 below. Monotone

Model	Shape	Number of Holes	
		Lower bound (Necessary)	Upper Bound (Sufficient)
Rotation	Polygons	$\lfloor n/6 \rfloor$	$\lfloor n/4 \rfloor$
	Orthogonal Polygons	$\lfloor n/12 \rfloor$	$\lfloor (n+4)/8 \rfloor$
	Star-shaped Polygons	$\lfloor n/20 \rfloor$	$\lfloor n/4 \rfloor$
	Fans, Monotone Polygons	1	1
Tilt	Polygons	$\lfloor n/7 \rfloor$	$\lfloor n/4 \rfloor$
	Orthogonal Polygons	$\lfloor n/28 \rfloor$	$\lfloor (n+4)/8 \rfloor$
	Star-shaped Polygons	?	$\lfloor n/4 \rfloor$
	Fans, Monotone Polygons	1	1

Table 1: Combinatorial bounds on the number of holes that will drain various shape classes of polygons.

polygons are easily seen to be 1-drainable:

Proposition 4 *Monotone polygons are 1-drainable, in either model.*

Proof: Let P be a polygon monotone with respect to the y-axis. We claim that in either model, a hole at either y-extreme vertex suffices to drain P. For the Tilt model, select a gravity vector $\overrightarrow{g}$ to point in the $-y$ direction. Then the lowest vertex v_0 is the only local y-minimum, and no ball can stop before arriving at v_0. In the Rotation model, each move lowers the ball with respect to the original y-axis orientation. So again a hole at the lowest vertex v_0 suffices. □

Table 1 summarizes our combinatorial bounds, including the few that are established later in the paper.

3 The Pin-Ball Graph

Let G be a directed graph whose nodes are the convex vertices of P, with v_i connected to v_j if B can roll in one "move" from v_i to v_j. Here, a move is a complete path to the local y-minimum v_j, for some fixed orientation of P. We conceptually label the arcs of G with the sequence of vertices and edges along the path $\rho(v_i, v_j)$. Thus, the (v_{10}, v_6) arc in Figure 1 is labeled $(v_9, e_{13}, v_{14}, e_7, v_7, e_5)$. We use G_R and G_T to distinguish the graphs for the Rotation and Tilt models respectively, and G when the distinction is irrelevant.

We gather together a number of basic properties of G in the following lemma.

Lemma 5 (G Properties)

1. *Every node of G_R has out-degree 2; a node of G_T has out-degree at least 2 and at most $O(n)$.*

2. *Both G_R and G_T have $O(n)$ nodes (one per convex vertex). G_R has at most $2n$ arcs, while G_T has $O(n^2)$ arcs, and sometimes $\Omega(n^2)$ arcs.*
3. *Each path label has length $O(n)$ (in either model).*
4. *The total number of path labels on the arcs of G_R is $O(n^2)$, and sometimes $\Omega(n^2)$.*
5. *The total number of labels in G_T is $O(n^3)$, and sometimes $\Omega(n^3)$.*

Proof: We only mention those properties which are not obvious.

1. Assume for the purposes of contradiction that there is a node of G_R that has out-degree 1, i.e., that both cw and ccw departures from v_i lead to the same v_j. A ray from v_i toward v_{i-1} until ∂P is hit bounds a subpolygon on the left of the ray that contains the termination vertex v_j. See Figure 7. If v_{i-1} is reflex, this subpolygon is nondegenerate; if v_{i-1} is convex, then $v_{i-1} = v_j$ and the subpolygon degenerates to this single vertex. A second ray from v_i toward v_{i+1} until ∂P is hit bounds a subpolygon on the right of the ray that contains v_j.

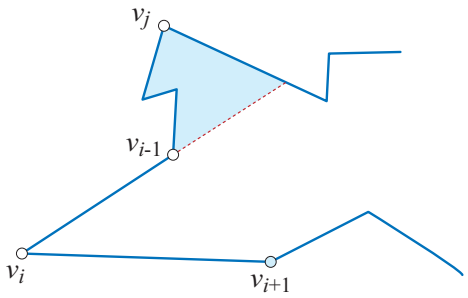

Figure 7: Rolling from v_i in the Rotation model. The left and right subpolygons (shaded) are disjoint.

   These two subpolygons are necessarily disjoint, so v_j cannot lie in both.

2. G_T has $\Omega(n^2)$ arcs when P is a convex polygon.
3. A single path travels monotonically with respect to gravity, and so can only touch $O(n)$ edges and vertices.
4. Figure 8 shows that the total number of path labels for G_R is sometimes $\Omega(n^2)$.
5. A modification of this figure establishes an $\Omega(n^3)$ bound for the total number of labels in G_T.

□

We will see below that G_T can be constructed more efficiently than what the cubic total label size in Lemma 5(5) might indicate.

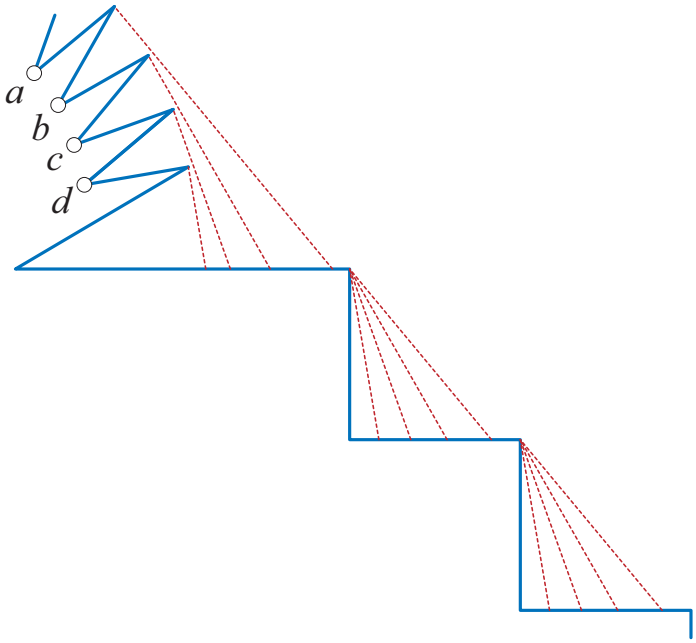

Figure 8: The paths from each of $\Omega(n)$ vertices $a, b, c, d, \ldots$ could touch $\Omega(n)$ edges.

Noncrossing Paths. A ball path corresponding to one arc of G is a polygonal curve, monotone with respect to gravity $\overrightarrow{g}$. The path is composed of subsegments of polygon edges, as well as *fall* segments, each of which is parallel to $\overrightarrow{g}$ and incident to a reflex vertex. A directed path ρ naturally divides P into a "left half" $L = L(\rho)$ of points left of the traveling direction, and a "right half" $R = R(\rho)$, where L and R are disjoint, and $L \cup R \cup \rho = P$. Two ball paths ρ_1 and ρ_2 (properly) *cross* if ρ_2 contains points in both $L(\rho_1)$ and $R(\rho_1)$. For example, in Figure 1, $\rho(v_0, v_6)$ crosses $\rho(v_{10}, v_6)$. Let $\overline{L} = L(\rho) \cup \rho$ be the closure of $L(\rho)$, and similarly define $\overline{R}$.

Two paths can only cross at a reflex vertex (as do $\rho(v_4, v_0)$ and $\rho(v_6, v_3)$ in Figure 1) or on fall segments of each (as do $\rho(v_0, v_6)$ and $\rho(v_{10}, v_6)$).

Lemma 6 (Noncrossing) *Two paths ρ_1 and ρ_2 from the same source vertex v_0 never properly cross (in either model).* See Figure 9.

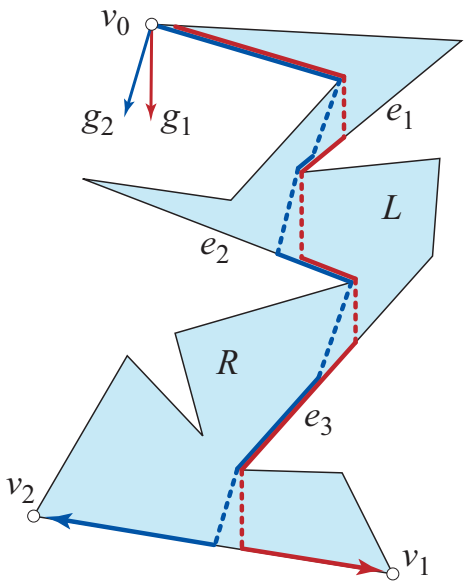

Figure 9: Paths from the same source v_0 do not cross. Fall segments are dashed. L and R indicate polygon "halves" left and right of the directed paths.

Proof: Let $\overrightarrow{g_1}$ and $\overrightarrow{g_2}$ be the gravity vectors for ρ_1 and ρ_2. Orient P so that $\overrightarrow{g_1}$ is vertically downward, as in Figure 9. Now we argue that $\rho_2 \subset \overline{R_1}$.

We first assume that the angular separation between the two gravity vectors is $\leq \pi$ (as illustrated in the figure). The path ρ_2 is in $\overline{R_1}$ at v_0, and it cannot cross into $L(\rho_1)$ along an edge of P. So it could only cross into $L(\rho_1)$ at either the upper endpoint of a fall segment (a reflex vertex), or in the interior of a fall segment. But both are impossible because $\overrightarrow{g_2}$ is pointing right-to-left.

If the angular separation between the two gravity vectors is $> \pi$, then the same result is obtained by reversing the roles of $\overrightarrow{g_1}$ and $\overrightarrow{g_2}$. □

Lemma 7 (Label Intervals) *In the Tilt model, a particular label λ appears on the arcs of G_T originating at one particular v_i within an interval $[\overrightarrow{g_1}, \overrightarrow{g_2}]$ of gravity directions.*

Proof: Suppose to the contrary that λ appears on the paths ρ_1 and ρ_2 for $\overrightarrow{g_1}$ and $\overrightarrow{g_2}$ but not on the path ρ' for $\overrightarrow{g'}$, $\overrightarrow{g_1} < \overrightarrow{g'} < \overrightarrow{g_2}$. Then, because ρ' cannot cross ρ_1 by Lemma 6, the element corresponding to λ is inside a region Q between ρ_1 and ρ'. If we let x be the first point at which ρ_1 and ρ' deviate, and let the arcs of G_T induced by $\overrightarrow{g_1}$ and $\overrightarrow{g_2}$ be (v_i, v_j) and (v_i, v_k) respectively, then Q is bounded by the portion of ρ_1 from x to v_i, the boundary ∂P from v_i to v_j, and the portion of ρ' from x to v_j. Now, for ρ_2 to include λ, it must cross into Q, and since it cannot cross ∂P, it must cross either ρ_1 or ρ', contradicting Lemma 6. □

Cycles and Strongly Connected Components. The directed path from any vertex v_i of G leads to a cycle in G, because every node has at least two outgoing edges by Lemma 5(1). Any maximal cycle in G has length at least 3. Anything less would involve a pair (v_i, v_j) connecting only to each other, which would contradict Lemma 5(1). Note that any pair of convex vertices adjacent on ∂P form a non-maximal cycle of length 2.

A cycle is a particular instance of a *strongly connected component* (SCC) of G, a maximal subset $C \subset G$ in which each node has a directed path to all others.

Define a graph G^* as follows. Let $C_1, C_2, \ldots$ be the SCC's of G. Contract each C_k to a node c_k of G^*, while otherwise maintaining the connectivity of G. (This graph is known as the *condensation* or *strong component graph* of G.) Then G^* is a DAG (because all cycles have been contracted).

Lemma 8 (Sinks) *The minimum number m of holes needed to drain P is the number of sinks of G^*.*

Proof: Certainly m holes suffice, for, by the construction of G^*, we can roll any ball to the SCC containing its terminal cycle. Piercing each c_k with one hole ensures that the ball can escape.[1] Leaving any sink unpierced constitutes a trap from which balls in the sink's connected component cannot escape. So m holes are necessary. □

Lemma 9 *The locations of the minimum number m of holes needed to drain P can be found in linear time in the size $|G|$ of G, once G has been constructed.*

Proof: Finding the SCC's of a graph is linear in $|G|$ (via, e.g., two depth-first searches [CLRS01, p. 552ff]). Both contracting to G^*, and finding the sinks of G^*, are again linear in $|G|$. □

We summarize the overall algorithm below.

[1] We should note that, after the ball exits the polygon, it still might be "trapped" in the exterior of P, unable to roll to ∞. We do not pursue this possibility.

Minimum Holes (Rotation or Tilt Model)
1. Construct G in $O(n^2 \log n)$ time.
2. Find Strongly Connected Components (SCCs) of G, in $O(\|G\|) = O(n^2)$ time.
3. Contract each SCC to a node of a new graph G^*, in $O(\|G\|)$ time.
4. Find sinks of G^* in $O(\|G\|)$ time.
5. Select one convex vertex in each sink for a hole.

We now turn to the first step of this algorithm: efficient construction of G.

Construction of G. Our goal is to construct the unlabeled G. Labels merely represent the paths that realize each arc of G. The example in Figure 8 seems to require $\Omega(n^2)$ ray-shooting queries in the Rotation model, and as we do not know how to avoid this, we set as our goal an $O(n^2 \log n)$ algorithm. This is straightforward for G_R, so we focus on G_T, which by Lemma 5(5) is potentially cubic.

We first preprocess P for efficient ray-shooting queries, using fractional cascading to support ray shooting in a polygonal chain. This takes $O(n \log n)$ preprocessing time and supports $O(\log n)$ time per query ray [CEG$^+$94]. Next we construct the visibility polygon from each vertex of P in linear time per vertex, and so overall $O(n^2)$ time [JS87]. From these visibility polygons, for each v_i we construct a *gravity diagram* D_i. This partitions all gravity vectors $\overrightarrow{g}$ into angular intervals labeled with the next vertex to which B will roll from v_i with tilt $\overrightarrow{g}$. For example, Figure 10(a) shows the gravity diagram for v_4 in Figure 1. Note that D_i only records the next vertex encountered in the path, not the ultimate destination of the edge of G_T. We maintain each diagram in a data structure that permits any $\overrightarrow{g}$ to be located in $O(\log n)$ time.

We now argue that we can construct all paths with source v_i, and therefore all arcs of G_T leaving v_i, in $O(n \log n)$ time. Compute the path ρ_0 for the cw extreme gravity vector $\overrightarrow{g_0}$ that leaves v_i (perpendicular to $v_i v_{i+1}$). This uses $O(n)$ ray-shooting queries for the fall segments of ρ_0, totaling $O(n \log n)$ time. Let $\rho_0 = (v_i, v_{i_1}, v_{i_2}, \ldots, v_j)$. During its construction, we locate $\overrightarrow{g_0}$ within each diagram D_{i_k}. Now we find the minimum angle between $\overrightarrow{g_0}$ and the next ccw event over all diagrams. This can be accomplished in $O(\log n)$ time using a priority queue. Call the next event $\overrightarrow{g_1}$, and suppose it occurs at v_{i_k} in diagram D_{i_k}. We now construct the path ρ_1 from v_{i_k} onward, until it terminates at a new vertex, or rejoins ρ_0 (recall from Figure 9 that paths might rejoin, i.e., the suffixes from v_{i_k} are not necessarily disjoint). In our "update" from ρ_0 to ρ_1, let V_0 be the set of vertices lost from ρ_0, and V_1 those gained in ρ_1. The priority queue of minima is updated by deleting those for V_0 and inserting those for V_1. The angular sweep about v_i continues in the same manner until the full gravity vector range about v_i is exhausted. Figure 10(b) illustrates one step of this process, where v_{i_1} determines the transition event between g_0 and g_1, at which point the path changes from $\rho_0 = (v_i, v_{i_1}, v_{i_2}, v_j)$ to $\rho_1 = (v_i, v_{i_1}, v'_{i_2}, v_k)$.

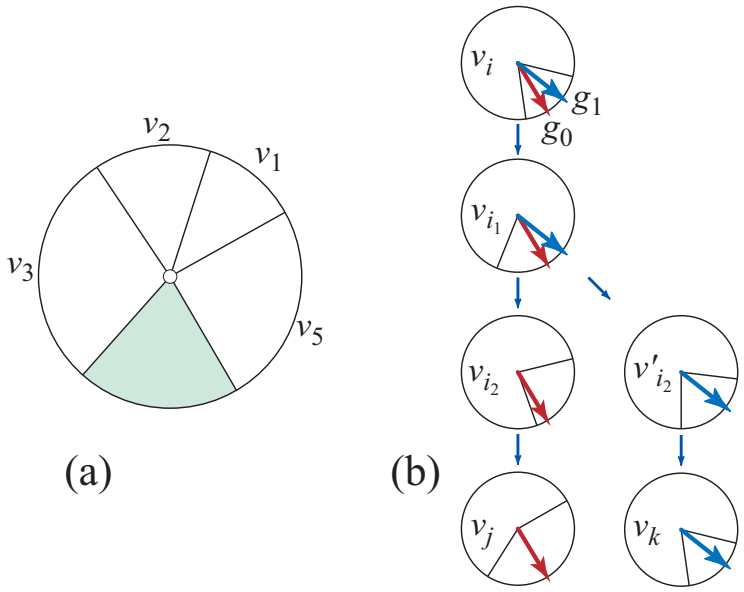

Figure 10: (a) Gravity diagram for v_4 in Fig. 1. (b) Gravity diagrams for paths ρ_0 and ρ_1.

By Lemma 7, each diagram abandoned in this sweep is never revisited. Thus the number of invocations of the minimum operation to find the next event is $O(n)$, or $O(n \log n)$ overall. Repeating for each v_i we obtain:

Lemma 10 *G can be constructed in $O(n^2 \log n)$ time.*

The algorithm summarized below leads, in combination with the overall algorithm, to Theorem 11.

Construction of G_T
1. Preprocess P for ray-shooting in $O(n \log n)$ time.
2. Construct visibility polygon from each vertex, in $O(n^2)$ time.
3. Construct gravity diagrams data structure in $O(n^2)$ time.
Initialize priority queue of minimum gravity angle events.
4. For each vertex v_i:
Sweep $\overrightarrow{g}$ about v_i, maintaining ball path ρ, in $O(n \log n)$ time.
Output G_T edges from v_i to the last vertex of ρ.

Theorem 11 *The locations of the minimum number of holes needed to drain P can be found in $O(n^2 \log n)$.*

For orthogonal polygons, all fall segments are (arbitrarily close to) parallel to the two directions determined by the polygon edges, and the situation illustrated

in Figure 8 cannot occur. We leave it as a claim that this permits G_R to be constructed (and the holes located) in $O(n \log n)$ time.

3.1 General Upper Bound

Figure 3(a) can be generalized to show that, for any number of traps $t \geq 3$, 6 vertices are enough to create a trap in the Rotation model, leading to the lower bound of $\lfloor n/6 \rfloor$ for the number of holes needed to drain. (For one trap, n=3 vertices are enough, and for t=2, n=11 vertices are enough.) Although we suspect that traps cannot be made with fewer than 5 vertices each on average (for $t \geq 3$), the only upper bound we can establish is $\lfloor n/4 \rfloor$:

Lemma 12 *Any polygon of n vertices can be drained with at most $\lfloor n/4 \rfloor$ holes (in either model).*

Proof: We establish the result for the Rotation model, which then establishes it for the Tilt model (because Rotation can be simulated by Tilt). Lemma 5(1) establishes that the out-degree of a node of G_R is 2. This implies that the shortest cycle, and the smallest sink SCC in G_R, contains at least 3 (convex vertices), for a cycle of length 2 would require each vertex to have out-degree 1. Let b be the number of *blocks* of consecutive convex vertices around P. Note that, for $b > 1$, P has at least b reflex vertices, one between each pair of adjacent blocks. Let t be the number of traps (sink SCCs). We know that the number of convex vertices of P is at least $3t$. Thus, the number of vertices of P satisfies $n \geq 3t+b$. Now notice that all the vertices in one block must belong to the same SCC, because each vertex in the chain can roll to its neighbor. This implies that the number of traps cannot exceed the number of blocks, $b \geq t$. Thus $n \geq 4t$, so $t \leq \frac{1}{4}n$, and the claimed upper bound follows. □

3.2 Fans

We claimed in Table 1 that fans are 1-drainable. We now establish this result, using the $L(\rho)$ notion introduced for Lemma 6.

Let the *ccw roll* from v_i be the roll toward v_{i+1} in the Rotation model, or equivalently, the tilt according to $\overrightarrow{g}$ perpendicular to v_iv_{i+1} in the Tilt model.

Lemma 13 (Kernel) *Let P be star-shaped with kernel K. Then for each arc $(v_i, v_j) \in G$ corresponding to the ccw roll path ρ from v_i, K is in $\overline{L(\rho)}$, i.e., K is on or to the left of ρ.*

Proof: Let $\overrightarrow{g}$ be the gravity vector for the roll, perpendicular to v_iv_{i+1}. Then K must be in the closed halfplane H left of v_iv_{i+1}, because P is star-shaped from K. On the other hand, $\rho(v_i, v_j)$ is monotonic with respect to $\overrightarrow{g}$, and so lies in the closure of the complementary halfplane $\overline{H}$. Therefore, $v_j \in \overline{H}$, and thus K is left or on the path ρ, and so $K \subset \overline{L(\rho)}$. See Figure 11. □

A *fan* is a star-shaped polygon whose kernel includes a convex vertex.

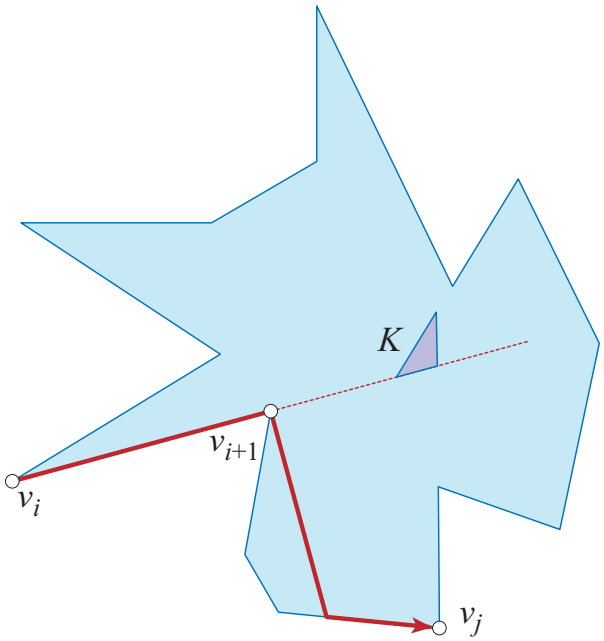

Figure 11: Star-shaped polygon.

Proposition 14 *Fans are 1-drainable. (in either model).*

Proof: Let $v \in K$ be the convex vertex in the kernel. Starting at any vertex v_i, let $C = (v_i, \ldots, v_j, v_k, \ldots, v_i)$ be the cycle generated by successive ccw rolls. If $v \notin C$, then there must be an arc (v_j, v_k) of G whose path ρ "skips over" v, which, because these are ccw rolls, must have $v \in R(\rho)$ in the right region for ρ. This contradicts Lemma 13. □

As we know from Figure 5, Proposition 14 cannot be extended to star-shaped polygons in the Rotation model.

4 3D

A number of models are possible for 3D polyhedra. We confine ourselves to a Tilt model that permits departure from a vertex v at a direction vector lying in any of the faces of P incident to v. We do not see how to mimic the efficient construction of G_T previously described, so we content ourselves with sketching how it can be accomplished in polynomial time: $O(n^7 \log n)$.

For the start vertex v_0, define the gravity diagram D_0 to partition the unit sphere S into regions that indicate the next edge crossed by the ball as it rolls from v_0. Now let e_i be some edge. We define a gravity diagram D_i over the Cartesian product of all the possible crossing positions on e_i, with all the possible approach vectors. This is, topologically, a segment crossed with a circle—a

cylinder. We partition this diagram into regions each of which identifies the next edge to be encountered by the ball after it leaves e_i. Each D_i has complexity $O(n^2)$.

Overlay all of the $O(n)$ cylindrical gravity edge diagrams. This can be viewed as an arrangement of $O(n^3)$ arcs on a cylinder, and so partitions it into $O(n^6)$ regions. Any gravity vector within a particular region r leads to the same complete ball path and therefore the same arc of G_T. For each region, compute the path as before, using $O(n \log n)$ time. This will construct G_T in $O(n^7 \log n)$ time. From here on the logic of Lemmas 8 and 9 applies as before.

5 Open Problems

1. Can the upper bound of $\lfloor n/4 \rfloor$ in Lemma 12 be improved? Can it be improved for special shapes, e.g., star-shaped polygons?
2. Are star-shaped polygons 1-drainable in the Tilt model (the unfilled cell in Table 1)? More generally, characterize 1-drainable polygons.
3. Suppose the ball B has finite radius r. Perhaps a retraction of P by r would suffice to lead to a similar algorithm to that in Section 3.[2]
4. Suppose m balls are present in P at the start, and P is k-drainable. What is the computational complexity of finding an optimal schedule of rotations, say, in terms of the total absolute angle turn, or in terms of the number of angular reversals?
5. We phrased our problem (in Section 1) as draining water/balls to the exterior of P, and noted in the proof of Lemma 8 that this is not the same as draining to infinity. What is the analog of our combinatorial bounds (Table 1) for the draining-to-∞ model?
6. We handled the degenerate situation of a ball/droplet falling directly on a vertex by stipulating that it rolls to the clockwise side. A natural alternative is to split the droplet into equal halves, which roll in opposite directions. In this split model, it is at least conceivable that a polygon could never be fully drained. Can this happen, in 2D or in 3D?

References

[ACC+08] Greg Aloupis, Jean Cardinal, Sébastien Collette, Ferran Hurtado, Stefan Langerman, and Joseph O'Rourke. Draining a polygon–or–rolling a ball out of a polygon. In *Proc. 20th Canad. Conf. Comput. Geom.*, pages 79–82, August 2008.

[BT94] Prosenjit Bose and Godfried T. Toussaint. Geometric and computational aspects of manufacturing processes. *Comput. & Graphics*, 18:487–497, 1994.

[2] We owe this idea to a referee.

[BvKT98] Prosenjit Bose, Mark van Kreveld, and Godfried T. Toussaint. Filling polyhedral molds. *Comput. Aided Design*, 30(4):245–254, April 1998.

[CEG+94] Bernard Chazelle, Herbert Edelsbrunner, Michelangelo Grigni, Leonidas J. Guibas, John Hershberger, Micha Sharir, and Jack Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ron L. Rivest, and Cliff Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[JS87] Barry Joe and R. B. Simpson. Correction to Lee's visibility polygon algorithm. *BIT*, 27:458–473, 1987.