# Grasping non-stretchable cloth polygons

Matthew Bell
Department of Computer Science
Dartmouth College
Hanover, NH 03755, USA
mbell@cs.dartmouth.edu

Devin Balkcom
Department of Computer Science
Dartmouth College
Hanover, NH 03755, USA
devin@cs.dartmouth.edu

August 14, 2009

## Abstract

In this paper, we examine non-stretchable 2D polygonal cloth, and place bounds on the number of fingers needed to immobilize it. For any non-stretchable cloth polygon, it is always necessary to pin all the convex vertices. We show that for some shapes, more fingers are necessary. No more than one third of the concave vertices need to be pinned for simple polygons, and no more than one third of the concave vertices plus two fingers per hole are necessary for polygons with holes.

## 1 Introduction

Cloth manipulation is difficult as a result of the flexibility of cloth. When cloth is suspended from one or two points, it develops buckles in a manner that is hard to predict. Grasps that minimize buckling will therefore make it easier to handle a piece of cloth, such as during the flattening or folding of laundry. If we can entirely immobilize a piece of cloth in a flattened configuration, we have full configuration information with which we can plan further actions.

We make a few simple assumptions about the cloth grasping problem. The cloth is non-stretchable, and we will place some number of point fingers on the cloth. These fingers are 'pinned' to the plane; once they are placed, they do not move and directly immobilize the point on the cloth underneath them.

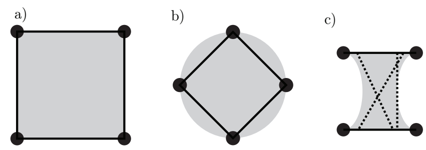



Figure 1: Three flat cloth shapes grasped by fingers. All but b) are immobilized.

The fundamental questions in grasping ask how many fingers are needed for a grasp, and where they should be placed. Figure 1 shows three pieces of cloth, all of which are immobilized except for b).

**Fact 1.** Any line segment with pinned endpoints that is fully contained in a polygon (the endpoints are mutually visible) is immobilized.

First order line segments of this type are indicated by solid lines in Figure 1. If a point somewhere in the polygon lies on a line segment between grasp points or first order lines, then it too will be immobilized, since the endpoints of this second order line are immobilized. A few second order lines are shown as dashed lines in the figure. This process can be repeated as needed with higher order line segments until the entire cloth is immobilized.

There are some cases where we cannot validate a grasp by drawing immobilized lines between fingers. Consider the polygon shown in Figure 2. No finger is visible from another finger. However, no point in the shaded hexagon can move further from any of the

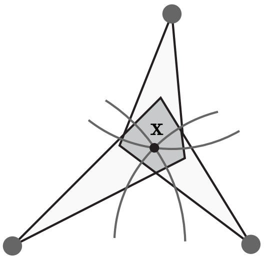
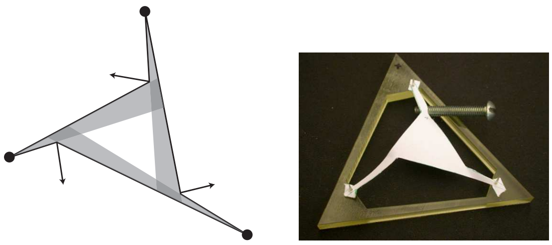



Figure 2: Star grasped with three fingers.



(a) Arrows indicate possible instantaneous velocities.

(b) Paper polygon after motion was applied.

Figure 3: Polygon that cannot be immobilized by pinning convex vertices (closed circles).

fingers, so this region is immobilized; therefore the entire polygon is immobilized.

To immobilize a cloth polygon, there must be a finger at least at each convex vertex; otherwise, that convex vertex will be free to move. In some cases, pinning just the convex vertices is enough. However, the piece of cloth shown in Figure 3 cannot be immobilized by pinning the three convex vertices of the shape. We have verified this result experimentally (Figure 3(b)) and theoretically (Section 5.1).

This polygon is representative of a class of polygons that we call pinwheels. These polygons all require more than $n_{\text{convex}}$ fingers for immobilization. In Theorem 6, we will show that the upper bound is $n_{\text{convex}} + \left\lfloor \frac{1}{3} n_{\text{concave}} \right\rfloor$ fingers for simple polygons. This bound is tight; there exist polygons that require this many fingers for immobilization.

# 2 Related Work

Minimal grasping has always been a challenging problem in robotics, with numerous papers on the subject, as evidenced in Bicchi and Kumar's survey of theoretical work on grasping [Bicchi and Kumar, 2000]. The listing here is meant to be a subset of grasping work that is closest to this paper. Nguyen examined the synthesis of planar force-closure grasps [Nguyen, 1986]. Mishra, Schwartz, and Sharir found bounds on the number of fingers needed to grasp a rigid object [Mishra *et al.*, 1987]. Rimon and Burdick showed that 3 convex fingers suffice to immobilize any smooth or polygonal planar object [Rimon and Burdick, 1995]. Erickson *et al.* examined the use of disc-shaped robots for capturing an arbitrary convex object in the plane [Erickson *et al.*, 2007]. Cheong *et al.* gave bounds for the number of fingers that immobilize a flexible chain of hinged polygons [Cheong *et al.*, 2007]. Rodríguez, Lien, and Amato worked on motion planning in an environment where every object is deformable [Rodríguez *et al.*, 2006]. This type of planning can also be applied to grasping problems.

There are two major types of polygon skeletons that are similar to the support tree that we will construct in Section 6.2. The first is the medial axis [Preparata, 1977], which has the same number of vertices and edges as a support tree. However, medial axes allow for curved edges. The second similar type of skeleton is the straight skeleton [Aichholzer *et al.*, 1995], which has straight edges, but contains more vertices and edges than are needed for a support tree.

This paper also depends on general concepts in visibility, such as those surveyed by Ghosh [Ghosh, 2007], and on triangulation and its applications to the art gallery problem, as explored by O'Rourke [O'Rourke, 1987].

Our problem is similar to that of trying to determine if a structure consisting only of cables is infinitesimally rigid when it is pinned at a set of points. This type of problem is briefly mentioned in Connelly's work on tensegrities and rigidity theory [Connelly, 1999].

There has been significant exploration of cloth behavior in the field of computer graphics. Some examples include Breen's work on building cloth simulations using real world measurements as inputs [Breen *et al.*, 1994] and Choi and Ko's work on cloth buckling [Choi and Ko, 2002].

Cloth manipulation has been used in various laundry folding projects; however, only a few fingers are used for grasping in these projects. Ono *et al.* have worked on a manipulator for cloth handling [Ono *et al.*, 1991], as well as co-operative systems combining touch and vision to unfold cloth [Ono *et al.*, 1995]. Salleh *et al.* have developed a system in which they trace cloth boundaries with grippers to flatten clothes [Salleh *et al.*, 2004]. Hamajima and Kakikura have worked on developing planning strategies for unfolding clothes [Hamajima and Kakikura, 2000].

# 3 Cloth Models and Definitions

Cloth can be modeled in several different ways. In the graphics and simulation worlds, ball and spring models are quite common. However, for our approach, we want the cloth to not stretch, which suggests a developable surface model.

We will use a model that is 'almost' a developable surface model. We assume the cloth cannot stretch, but that the cloth may compress slightly. Our upper bound on the maximum number of fingers needed to grasp cloth ($n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor$) holds for developable surfaces, but we only discuss the existence of polygons requiring this many fingers for the compressible model.

## 3.1 Support Graphs

To discuss polygon immobilization, we use a specific type of polygon skeleton called a support graph; an example is shown with dotted lines in Figure 4.

**Definition 1.** A **support graph** for a polygon is an embedded planar graph contained within the polygon, such that every point of the polygon falls on a line segment (possibly of length zero) that

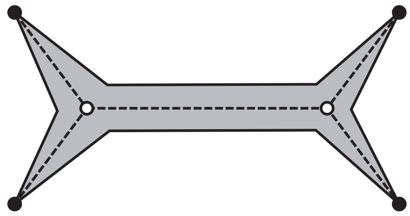


Figure 4: Example of a support graph in a cloth polygon.

- is completely contained within the polygon, and
- has endpoints that are points of the embedded graph (on an edge or at a node).

A **support tree** is a support graph with no cycles.

It is clear that if a support graph for a polygon is immobilized by some set of fingers, every line segment specified in the definition is immobilized, and therefore the polygon is immobilized. We can examine the immobilization of support graphs by placing fingers at vertices.

**Definition 2.** A **pinned vertex** is a graph or polygon vertex that is held in place by a finger. This is indicated in diagrams by a closed circle. (Unpinned vertices have open circles).

**Definition 3.** A **positively-spanned vertex** is a vertex in a graph whose adjacent edges positively span $\mathbb{R}^2$. (For a definition of positive linear spans, see [Davis, 1954].)

There are many ways to construct a support graph for a polygon. Figure 4 shows a support graph constructed by hand, but we can always easily construct a (possibly more complex) support graph by triangulating a polygon. Therefore, if a triangulation of the polygon is immobilized, the polygon is immobilized.

We assume a model of cloth that allows the cloth to compress. In this case, if a triangulation of the cloth is not immobilized by a set of fingers, the cloth is not immobilized.

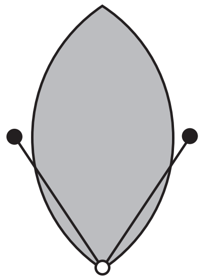


Figure 5: Allowed motion of a non-positively-spanned vertex.

# 4 Immobilizing Trees, Graphs, and Polygons

As an approach to specifying the fingers required to grasp a piece of cloth, we can first describe the fingers needed to immobilize a connected, linear network of non-stretchable string embedded in the plane. If this network is a support graph for a polygon, then that polygon is immobilized in 2D and 3D.

At a minimum, all not positively-spanned vertices must be pinned in order to immobilize a non-stretchable planar graph. The shaded region in Figure 5 illustrates the free motions of an unpinned and non-positively-spanned vertex.

## 4.1 Immobilizing Non-stretchable Graphs

Initially, we consider a non-stretchable tree, and assume that all vertices have degree 1 or 3. Additionally, we will assume that all interior vertices (non-leaves) are positively-spanned vertices. These assumptions will be relaxed later, but they are useful in the first stage of the proof.

In the theorems that follow, we consider only first order constraints on the free motions of vertices, since linear constraints are sufficient for the proofs and simpler to analyze. Using quadratic distance constraints yields the same results. We will use the notation $\overrightarrow{uv}$ to indicate a normalized vector pointing from vertex $u$ to vertex $v$. Figure 6 illustrates the following lemma.

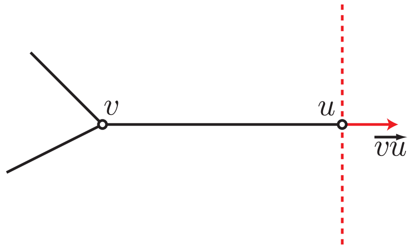



Figure 6: Restriction on allowed motions of $u$.

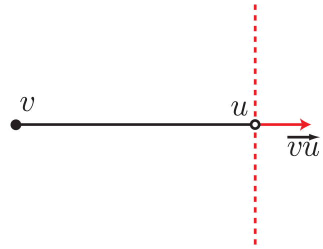



Figure 7: Base case (vertex $v$ is pinned, as indicated by the closed circle).

**Lemma 1.** *Consider a planar non-stretchable tree, with all vertices of degree one or three, and with only positively-spanned interior vertices. Let all the leaves (non-positively-spanned vertices) be pinned, except for one leaf, labeled $u$. Let $v$ be the vertex adjacent to $u$. Vertex $u$ cannot move into the half plane defined by normal $\overrightarrow{vu}$. (This can also be written as a constraint of the form $\dot{u} \cdot \overrightarrow{vu} \leq 0$.)*

*Proof.* **Induction Hypothesis**: Consider a tree subject to the assumptions with all leaves pinned except for $u$, and let $v$ be the vertex adjacent to $u$. $u$ cannot move into the half plane indicated by the constraint $\dot{u} \cdot \overrightarrow{vu} \leq 0$.

**Base Case**: The base case is a tree consisting of only vertices $v$ and $u$ (Figure 7), with vertex $v$ pinned.

**Inductive step**: Given a tree $T$, break it at vertex $v$ into two trees, $T_1$ and $T_2$. Let $a$ be the vertex adjacent to $v$ in $T_1$, and $b$ be the vertex adjacent to $v$ in $T_2$ (Figure 8). By the induction hypothesis, $T_1$ imposes the constraint $\dot{v} \cdot \overrightarrow{av} \leq 0$ (equivalent to $\dot{v} \cdot \overrightarrow{va} \geq 0$), and $T_2$ imposes the constraint $\dot{v} \cdot \overrightarrow{bv} \leq 0$ (equivalent to $\dot{v} \cdot \overrightarrow{vb} \geq 0$).

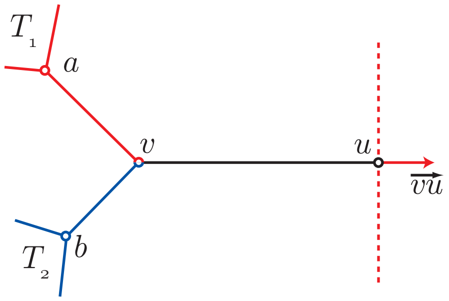



Figure 8: Inductive step.

From our assumptions, we know that $\vec{va}$, $\vec{vb}$, and $\vec{vu}$ positively span $\mathbb{R}^2$. As a result, if both $\dot{v} \cdot \vec{va} \geq 0$ and $\dot{v} \cdot \vec{vb} \geq 0$ are satisfied, then $\dot{v} \cdot \vec{vu} \leq 0$, proving the induction hypothesis. □

This lemma can be extended from restricted motion to immobilization.

**Lemma 2.** *Consider a planar non-stretchable tree, with all vertices of degree 1 or 3 that contains only positively-spanned vertices in its interior. If all the leaves of this tree are pinned, the tree will be immobilized.*

*Proof.* Consider a tree that satisfies Lemma 1, and label its unpinned leaf $u$. Leaf $u$ cannot move away from its adjacent vertex $v$ ($\dot{u} \cdot \vec{vu} \leq 0$, which also implies $\dot{v} \cdot \vec{vu} \leq 0$). If we now pin $u$, we impose a constraint on $v$ of $\dot{v} \cdot \vec{uv} \leq 0$. Combined with the previous constraints at $v$ from the other adjacent edges (which we know positively span $\mathbb{R}^2$ if edge $vu$ is included), this completely immobilizes $v$. The immobilization of $v$ can now be used to show that the vertices adjacent to $v$ are also immobilized. This immobilization can be continued throughout the tree, showing that the entire tree is immobilized. □

This result can be strengthened to any non-stretchable planar tree. The next theorems depend on the concept of splitting vertices of a non-stretchable tree or graph by pinning them. If a vertex $v$ has $k$ adjacent edges, and we pin $v$, then this is equivalent to having $k$ pinned vertices all located at the same point as $v$, with each vertex adjacent to exactly one of the edges adjacent to $v$. Physically, the resulting tree or graph is exactly equivalent to the original tree or graph, as constraints do not propagate past pinned vertices.

**Theorem 3.** *Any planar non-stretchable tree embedded in $\mathbb{R}^2$ (with vertices of any degree) that has its non-positively-spanned vertices pinned is immobilized.*

*Proof.* First, we will remove the assumption that all interior vertices must be spanned vertices, and we allow degree 2 vertices. If any vertex is non-positively-spanned, then it is pinned, as is specified by the theorem statement. Additionally, note that any degree 2 vertices can never have edges positively spanning $\mathbb{R}^2$, and therefore must be pinned. If we break the tree into a forest by splitting it at each non-positively-spanned (and pinned) interior vertex, each component of the forest will be immobilized by Lemma 2. When joined, the resulting complete tree is still immobilized.

Finally, we allow vertices of degree greater than 3. If such a vertex is non-positively-spanned, we can simply use the argument above. If it is positively-spanned, then we need to slightly rework the inductive step of Lemma 1. If vertex $v$ is of degree $d > 3$, it will be split into $d-1$ subtrees (along all edges except $vu$). By the inductive hypothesis, we know there are constraints of the form $\dot{v} \cdot \vec{va_i} \geq 0$ for each subtree $T_i$. We can pick a pair of subtrees $T_i$ and $T_j$, such that $\vec{va_i}$, $\vec{va_j}$, and $\vec{vu}$ positively span $\mathbb{R}^2$. Now, as in the original inductive step, this gives us the desired constraint on $u$. □

If we split a graph into a tree by adding one finger per cycle (and pinning non-positively-spanned vertices), the graph is immobilized.

**Theorem 4.** *Any planar non-stretchable graph with all non-positively-spanned vertices pinned and at least one vertex pinned within each cycle is immobilized.*

*Proof.* Pin one vertex per cycle of the graph. This splits the graph at all of these pinned vertices. Splitting each cycle with one finger converts the graph into a tree, with properties satisfying Theorem 3. □

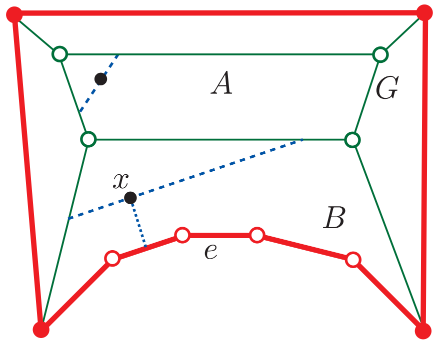



Figure 9: Two types of cells (A and B) in a polygon containing a support graph.

If no vertices in a cycle are pinned, there is no guarantee that the cycle is immobilized. In fact, in general it is very likely that the cycle can move. There are specific cases in which the cycle is immobilized (in particular, if the edges supporting the cycle bisect the exterior angles of the cycle), but these cases are rare.

## 4.2 Grasping Polygons

A tree or graph embedded in a cloth polygon can be used to show that the polygon is immobilized.

**Theorem 5.** *If a cloth polygon contains a planar non-stretchable graph $G$ such that non-positively-spanned vertices of the graph correspond exactly to the convex vertices of the polygon, then the graph is a support graph for the polygon, and immobilizing the graph immobilizes the polygon.*

*Proof.* In order to fit the definition of a support graph (Definition 1), every point in the polygon must lie on a line with endpoints on the support graph.

Consider the polygon and graph shown in Figure 9. Since non-positively-spanned vertices of the graph (thin line) exactly map to all convex vertices, the polygon (thick line) is divided up into two types of cells. Cells that are contained within cycles of the graph are trivial to handle (indicated by A in the figure). For any point within a cycle, any line through the point has endpoints on the graph, and thus is immobilized if the graph is immobilized.

The other type of cell is enclosed by graph edges and a chain of (possibly zero) concave vertices on the polygon boundary (B in the figure). The polygon boundary must consist purely of concave vertices, as a convex vertex would have a non-positively-spanned graph vertex located at it, splitting the cell. Now, consider any point $x$ in the cell. Find the closest polygon edge $e$. Extend a line through $x$ parallel to $e$ until both ends of the line hit the boundary of the cell. The endpoints must both lie on graph edges; if this were not the case, the polygon boundary would contain a convex vertex, and it does not. Therefore, for any point in this type of cell, there exists a line with both endpoints on the graph.

Since both types of cells satisfy the definition of a support graph, $G$ is a support graph. By definition of a support graph, if $G$ is immobilized, the polygon is immobilized.

□

We can now show that $n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor$ fingers are always sufficient to immobilize a polygon. In the following proof, we view a triangulation of a polygon as a graph embedded in the polygon.

**Theorem 6.** *A simple cloth polygon can always be immobilized by pinning $n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor$ vertices.*

*Proof.* Portions of this proof are similar to Fisk's proof that an art gallery requires $\lfloor \frac{n}{3} \rfloor$ guards [Fisk, 1978]. In both proofs, the main problem is placing one item (a guard or a pinned vertex) per triangle.

As in Fisk's proof, we begin by considering a triangulation $T = (V, E)$ of the polygon $P$. We consider the most strict form of a triangulation, in which triangle vertices must also be polygon vertices. In this type of triangulation, concave polygon vertices will be positively-spanned by incident graph edges, and convex vertices will not be. Concave vertices must be positively-spanned because each exterior angle at a concave vertex is less than $\frac{\pi}{2}$, and the interior angle is split into angles of less than $\frac{\pi}{2}$ by the triangulation.

Let all convex vertices of the polygon (and thus all non-positively-spanned vertices of $T$) be pinned. By Theorem 4, $T$ is immobilized if we also pin one vertex per cycle (which, for a triangulation, means one pinned vertex per triangle).

Convex vertices must always be pinned, so we can ignore any edges that are adjacent to them, and we can construct a $T^* = (V^*, E^*)$ that removes these edges. Specifically, $T^*$ contains only the concave vertices of $P$, and only edges that are between pairs of concave vertices. Since any triangulation can be 3-colored [O'Rourke, 1987], and since $T^*$ is a subset of a (3-colorable) triangulation $T$, $T^*$ is also 3-colorable. As in Fisk's proof, one of the three colors must be used no more than $\lfloor \frac{1}{3}|V^*| \rfloor = \lfloor \frac{1}{3} n_{\text{concave}} \rfloor$ times. Now, pin each vertex labeled with the least frequently used color. Since each triangle must have one vertex of each color, each triangle (and therefore cycle) of $T^*$ has one pinned vertex, and therefore each cycle of $T$ has one pinned vertex. As a result, $T$ (and thus $P$) is immobilized. □

The above proof does not hold for non-simple polygons, as triangulations of such polygons are not necessarily 3-colorable. However, we can use the same general idea to give a bound for polygons with holes as well.

**Corollary 7.** *A cloth polygon with* $n_{\text{holes}}$ *holes can always be immobilized by pinning* $n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor + 2n_{\text{holes}}$ *vertices, where both* $n_{\text{concave}}$ *and* $n_{\text{convex}}$ *include the concave and convex vertices in the polygon's holes.*

*Proof.* If we place cuts in the polygon such that each hole is open to the region outside the polygon (either directly through a single cut, or by a chain of cuts through other holes), then we have turned the polygon with holes into a simple polygon with at most $4n_{\text{holes}}$ new vertices (careful cutting can reduce this to $2n_{\text{holes}}$ new vertices if the cuts go between existing vertices). These new vertices will be convex vertices; however, since the two sides of the cut are in the same place, we can use one finger to pin each pair of new convex vertices. As in Theorem 6, we now triangulate the simple polygon, which requires us to pin up to $\lfloor \frac{1}{3} n_{\text{concave}} \rfloor$ concave vertices, plus the original convex vertices, plus two fingers per cut (equivalent to two fingers per hole). Therefore, a polygon with holes will be immobilized with $n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor + 2n_{\text{holes}}$ vertices. □

# 5 When Convex Vertices Are Not Enough

We have shown that $n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor$ fingers are always sufficient to immobilize a simple polygon, but in order to show that this bound is also necessary, we must first show that there are polygons for which a convex vertex grasp is insufficient for immobilization. If we can compute possible free motions of a grasped cloth polygon, then the grasp is clearly insufficient.

## 5.1 Determining Free Motions

We can verify a grasp by constructing an appropriate linear program, and by testing to see if it has any nonzero solutions. This LP is built from distance constraints, which require that the endpoints of an edge cannot move apart beyond their initial stretched distance. We use the standard notion of polygon visibility in this section.

If $x_i$ and $x_j$ are mutually visible, then at every time $t$, the distance between the points must not be greater than the initial (fully stretched) distance:

$$||\overrightarrow{x_i x_j}(t)||^2 \leq ||\overrightarrow{x_i x_j}(0)||^2. \tag{1}$$

At time 0, the time derivative of every distance between pairs of mutually visible points must be nonpositive.

$$\dot{x}_i \cdot \overrightarrow{x_j x_i} + \dot{x}_j \cdot \overrightarrow{x_i x_j} \leq 0 \tag{2}$$

A simple example is a network of points attached by strings as shown in Figure 10. Let $x_1$ and $x_2$ be unpinned points, and let $x_3$ through $x_6$ be pinned. There are five distance constraints, corresponding to the edges. Using the constraints from equation 2, we have

$$\begin{bmatrix} \overrightarrow{x_3 x_1} & 0 \\ \overrightarrow{x_5 x_1} & 0 \\ 0 & \overrightarrow{x_4 x_2} \\ 0 & \overrightarrow{x_6 x_2} \\ \overrightarrow{x_2 x_1} & \overrightarrow{x_1 x_2} \end{bmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} \leq 0. \tag{3}$$

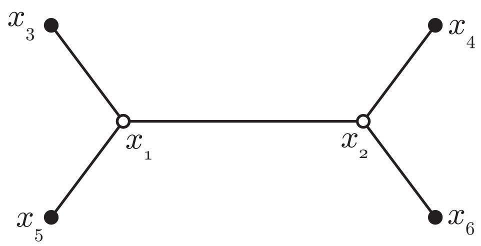



Figure 10: A network of points connected by strings (closed circles are pinned).

We can rewrite this as

$$J\dot{x} \leq 0 \tag{4}$$

This is in the form of constraints for a linear program, and therefore we can use a solver to see if there are any solutions other than $\dot{x} = 0$. If such solutions exist, then the line network can move as described by one of these solutions.

We can extend this easily to an algorithm to verify a grasp for a cloth polygon. To do this, we take any triangulation of the polygon, and consider this as our line network. We then build $J$, which has one row for every edge of the triangulation (with the exception of any edges between pinned points, since the coefficients would all be zero in this case). If $J\dot{x} \leq 0$ only has the solution $\dot{x} = 0$, then the triangulation network is immobilized by the given grasp. We have implemented this algorithm in Matlab, using CGAL [CGAL, 2008] to construct triangulations and lp_solve to check for nonzero solutions given the constraints. An example run of this algorithm for a non-immobilized polygon is shown in Figure 11, with X's indicating one possible set of additional fingers that immobilize the polygon.

If nonzero solutions exist for the lines of a triangulation, we believe that this means that the cloth can move within the given grasp. However, this statement may depend on the cloth model that we use. If we assume that the cloth can simply compress into itself, then it is clear that a nonzero solution will allow movement of the cloth. It is less clear as to what happens if a more realistic model that involves buckling is used, or if the cloth is a developable surface.

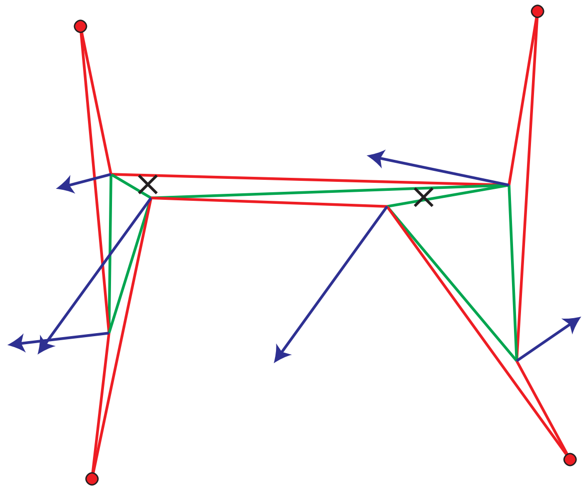


Figure 11: A dual pinwheel, with free motions as shown. Adding fingers at the X's immobilizes the polygon.

## 5.2 Pinwheels

As shown with the example in Figure 11, there are polygons for which a convex vertex grasp is insufficient. All such polygons that we have found fall into a class that we refer to as pinwheels.

**Definition 4.** An ***n*-pinwheel** is a polygon with a cyclic first order visibility structure, where a first order visibility structure is defined as the set of visibility polygons from all of the convex vertices of the polygon. The number $n$ refers to the number of points in the pinwheel.

In an $n$-pinwheel, the visibility polygon from a vertex $v_2$ first intersects its clockwise neighbor's ($v_3$) visibility polygon, followed by its counter-clockwise neighbor's ($v_1$) visibility polygon (see Figure 12 for an example of a 4-pinwheel). The directions can be reversed; if a vertex first sees its counter-clockwise neighbor's visibility polygon, followed by that of its clockwise neighbor, then the polygon also has a pinwheel structure. In order to actually be a pinwheel, this type of visibility intersection must be repeated for all vertices, leading to a cycle of visibility intersections.

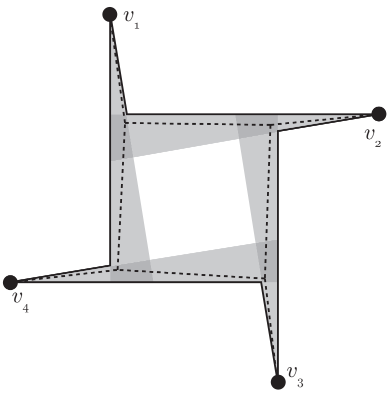



Figure 12: A 4-pinwheel, with its cyclic support graph and first-order visibility polygons.

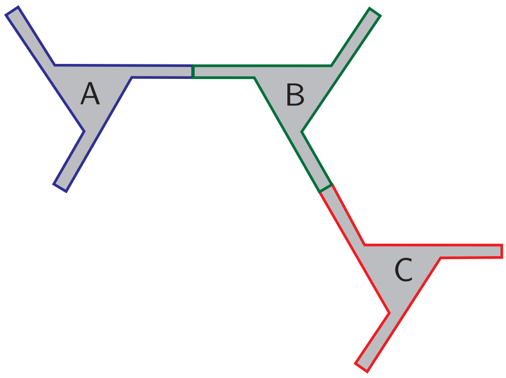



Figure 13: Multiple pinwheels.

**Theorem 8.** *A non-stretchable cloth n-pinwheel can always be immobilized with* $n_{\text{convex}} + 1$ *fingers.*

*Proof.* A support graph with one cycle and $n_{\text{convex}}$ non-positively-spanned vertices located at the convex vertices of the pinwheel can be constructed from the cyclic visibility intersections present in a pinwheel (Figure 12). We already know that all $n_{\text{convex}}$ convex vertices must be pinned. By Theorem 4, pinning any one vertex of the cycle immobilizes the graph, and therefore, pinning the corresponding point in the pinwheel immobilizes the pinwheel. □

We will use pinwheels to show that our upper bound on the number of fingers needed for immobilization is a tight bound.

**Theorem 9.** *There exist non-stretchable cloth polygons that require a grasp of* $n_{\text{convex}} + \left\lfloor \frac{1}{3} n_{\text{concave}} \right\rfloor$ *fingers to be immobilized.*

*Proof.* The class of polygons that we will use to satisfy the statement is based on 3-pinwheels. Consider the triple 3-pinwheel shown in Figure 13. The points have been expanded to two vertices to simplify the edge that is common to pairs of 3-pinwheels. As discussed in Section 5.1, we can build a linear program that gives the possible motions of a 3-pinwheel. From this, we can easily show that only pinning the six convex vertices does not suffice to immobilize one of the modified 3-pinwheels by itself. It is possible to immobilize a single pinwheel by adding one additional finger.

Now, consider attaching pinwheel B to pinwheel A, with all convex vertices pinned. Let us assume that the dual A-B pinwheel can be immobilized with just one additional finger. If this finger is on the boundary between A and B, then neither pinwheel will be immobilized, as this single finger will provide no more support than would have existed had we pinned the convex vertices of each pinwheel. Next, assume that we have placed the extra finger in such a way that all of A is immobilized (note that this is not actually possible). If this is the case, the boundary line between A and B will also be immobilized. However, as we have already stated, this is not enough to immobilize B. The same situation exists in reverse if we put a finger in B that immobilizes B.

Finally, we can extend this chain by adding pinwheel C, followed by another pinwheel attached to C's right point, and so on (Figure 14). There must be one finger per pinwheel to be able to immobilize the entire shape, as fingers outside the boundaries of a pinwheel do not suffice to immobilize it. Since each pinwheel has 3 concave vertices, this means that the overall shape requires $n_{\text{convex}} + \left\lfloor \frac{1}{3} n_{\text{concave}} \right\rfloor$ fingers. □

We are able to make general statements about several classes of polygons. It is possible to place a support tree with non-positively-spanned vertices only

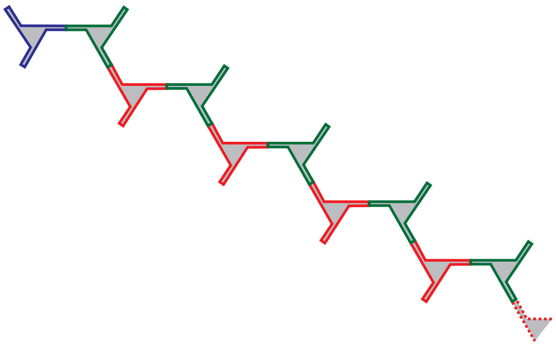
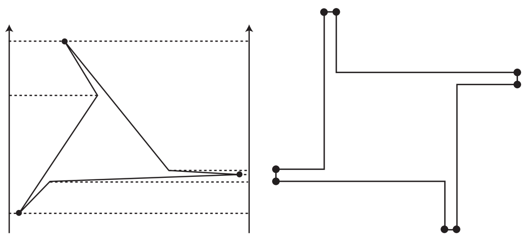


Figure 14: Repeating chain of pinwheels.



(a) Monotone polygon. (b) Orthogonal polygon.

Figure 15: Monotone and orthogonal polygons that cannot be immobilized by a convex vertex grasp.

at convex vertices in all star-shaped and convex polygons; such polygons are thus immobilized by a convex vertex grasp. Pinwheels do not fall into either of these classes. Interestingly, we can construct monotone (Figure 15(a)) and orthogonal (Figure 15(b)) pinwheels.

We have now shown that for a simple non-stretchable cloth polygon, the minimum number of fingers needed to immobilize it is $n_{\text{min}} \in [n_{\text{convex}}, n_{\text{convex}} + \lfloor \frac{1}{3} n_{\text{concave}} \rfloor]$.

# 6 Grasping with Fewer Fingers

A simple algorithm for generating grasps begins by testing a convex vertex grasp using our linear program formulation. If this fails, the triangulation method is used to get a grasp that pins one third of the concave vertices.

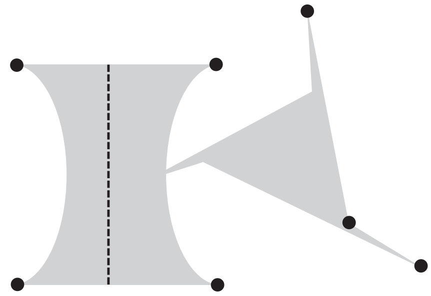
Disregarding the LP step, this algorithm has a running time of $O(n)$. Chazelle showed that triangulation of a simple polygon requires $O(n)$ time [Chazelle, 1991], and the 3-coloring of a triangulation can be implemented with a simple linear time algorithm.

This algorithm is guaranteed to generate a valid grasp; however, the grasp may include unnecessary fingers if there are lengthy chains of concave vertices, as in Figure 16.

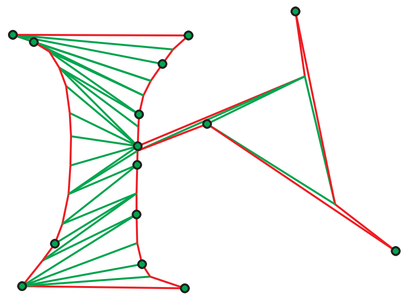
## 6.1 Grasp Reduction

We have developed an algorithm to reduce the size of the grasp, which removes certain fingers by checking to see if they are already immobilized by other portions of the grasp. Consider the example shown in Figure 17. Vertex $v_2$ can be unpinned as long as vertex $v_1$ and edge $e_1$ remain immobilized. Vertices $v_2$ and $v_3$ can be unpinned as long as edges $e_2$ and $e_3$ are immobilized. This grasp reduction algorithm has a running time of $O(n^2)$, as all edges must be scanned for each vertex.

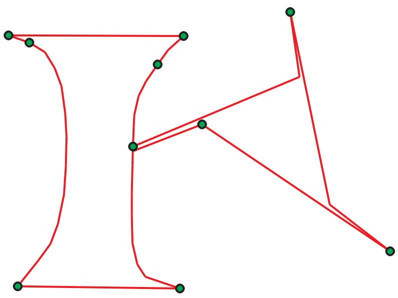
Figure 16 shows example results from our algorithms. Figure 16(a) gives a minimal grasp, which consists of 6 convex vertices, plus one concave vertex. Figure 16(b) shows the results of the grasp building



(a) A valid (minimal) grasp (1 pinned concave vertex).



(b) Grasp built by algorithm (9 pinned concave vertices).



(c) The result of the reduction algorithm (4 pinned concave vertices).

Figure 16: A polygon with $n_{\text{concave}} = 28$, $n_{\text{convex}} = 6$.

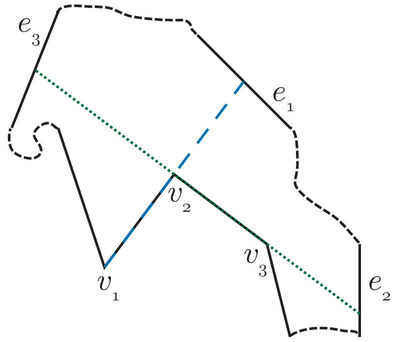



Figure 17: Method for reducing the number of pinned points.

algorithm, and Figure 16(c) shows the grasp after it has been reduced. A few extra vertices still remain; the algorithm could be improved by enabling it to recognize immobilized lines between immobilized edges, such as the dotted line in Figure 16(a).

## 6.2 Graphical Method for Analyzing Immobilization

We have developed a graphical method for determining if a cloth polygon is immobilized by a given grasp. This method relies on embedding a support tree within a polygon. A support tree is fundamentally based on visibility; in particular, adjacent vertices in the support tree must be mutually visible. Visibility is fairly easy to assess visually, and therefore manually placing a support tree in a polygon is a quick method for determining if a polygon is immobilized with a given grasp. We have taken this manual method and expanded it into an algorithm for constructing support trees. Our algorithm repeatedly intersects visibility regions to form a skeleton, and uses an optimizer to try to shift the vertices of the skeleton until the skeleton becomes a support tree.

We have implemented this algorithm in Matlab, using CGAL [CGAL, 2008] and VisiLibity [Obermeyer, 2008] to handle polygon and visibility operations, and OGDF for graph planarity testing. Figure 18 shows the result of running the algorithm on a comb shape.

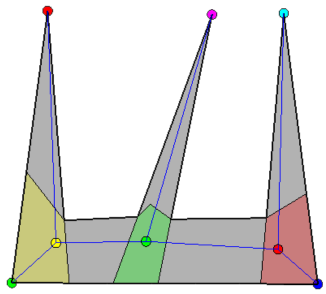


Figure 18: Output from the support tree construction algorithm.

# 7 Conclusion

We have determined that for simple cloth polygons,

$$n_{\text{min}} \in \left[ n_{\text{convex}}, n_{\text{convex}} + \left\lfloor \frac{n_{\text{concave}}}{3} \right\rfloor \right] \quad (5)$$

and for non-simple polygons,

$$n_{\text{min}} \in \left[ n_{\text{convex}}, n_{\text{convex}} + \left\lfloor \frac{n_{\text{concave}}}{3} \right\rfloor + 2n_{\text{holes}} \right] \quad (6)$$

We have shown that both bounds are tight for simple polygons, and that the lower bound is tight for polygons with holes. Additionally, we have developed the geometric method of using support trees to determine if a polygon is immobilized with a given grasp. This method is particularly valuable for visually determining if a polygon is likely to be immobilized with a given grasp.

Our theorems directly led to an algorithm for constructing a valid grasp for any simple cloth polygon. This algorithm does not guarantee a minimal grasp, but it is a significant first step in designing grasps for cloth objects. The algorithm makes use of a simple linear programming method for verifying the validity of a given grasp. We implemented both a linear program based grasp verifier, and a support tree construction algorithm.

Natural extensions of this work include polygons with holes, and 3-dimensional cloth, such as cloth polyhedra. Our results are also applicable to cloth sensing. If the location of any grasp point is unknown, there is no way to show that the cloth is in a flat configuration. Thus, by sensing all of the grasp points, we can determine if a piece of cloth is flat.

# 8 Acknowledgments

This work was supported in part by NSF grants IIS-0643476 and CNS-0708209. We would like to thank Scot Drysdale and Daniela Rus for helping to motivate this work, and for valuable discussions.

# References

[Aichholzer *et al.*, 1995] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761, 1995.

[Bicchi and Kumar, 2000] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, pages 348–353, San Francisco, CA, April 2000.

[Breen *et al.*, 1994] David E. Breen, Donald H. House, and Michael J. Wozny. A particle-based model for simulating the draping behavior of woven cloth. *Textile Research Journal*, 64(11):663–685, November 1994.

[CGAL, 2008] CGAL, Computational Geometry Algorithms Library, v3.3.1, 2008. http://www.cgal.org.

[Chazelle, 1991] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6(1):485–524, December 1991.

[Cheong *et al.*, 2007] Jae-Sook Cheong, A. Frank van der Stappen, Kenneth Y. Goldberg, Mark H. Overmars, and Elon Rimon. Immobilizing hinged polygons. *International Journal of Computational Geometry and Applications*, 17(1):45–70, February 2007.

[Choi and Ko, 2002] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 604–611, San Antonio, TX, 2002.

[Connelly, 1999] Robert Connelly. Tensegrity stuctures: Why are they stable? In M. F. Thorpe and P. M. Duxbury, editors, *Rigidity Theory and Applications*, pages 47–54. Kluwer/Plenum Publishers, New York, NY, 1999.

[Davis, 1954] Chandler Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76(4):733–746, October 1954.

[Erickson *et al.*, 2007] Jeff Erickson, Shripad Thite, Fred Rothganger, and Jean Ponce. Capturing a convex object with three discs. *IEEE Transactions on Robotics*, 23(6):1133–1140, December 2007.

[Fisk, 1978] Steve Fisk. A short proof of Chvátal's Watchman Theorem. *Journal of Combinatorial Theory, Series B*, 24(3):374, 1978.

[Ghosh, 2007] Subir Kumar Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, New York, NY, 2007.

[Hamajima and Kakikura, 2000] Kyoko Hamajima and Masayoshi Kakikura. Planning strategy for task of unfolding clothes. *Robotics and Autonomous Systems*, 32(2–3):145–152, August 2000.

[Mishra *et al.*, 1987] Bhubaneswar Mishra, Jacob T. Schwartz, and Micha Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, November 1987.

[Nguyen, 1986] Van-Duc Nguyen. Constructing stable force-closure grasps. In *Proceedings of 1986 ACM Fall Joint Computer Conference*, pages 129–137, 1986.

[Obermeyer, 2008] Karl J. Obermeyer. The VisiLibity library. http://www.VisiLibity.org, 2008. R-1.

[Ono *et al.*, 1991] Eiichi Ono, H. Ichijo, and N. Aisaka. Robot hand for handling cloth. In *International Conference on Advanced Robotics*, volume 1, pages 769–774, Pisa, Italy, June 1991.

[Ono *et al.*, 1995] Eiichi Ono, Noboyuki Kita, and Shigeyuki Sakane. Strategy for unfolding a fabric piece by cooperative sensing of touch and vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 441–445, Pittsburgh, PA, August 1995.

[O'Rourke, 1987] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, NY, 1987.

[Preparata, 1977] Franco P. Preparata. The medial axis of a simple polygon. In *Mathematical Foundations of Computer Science*, volume 53, pages 443–450, Tatranska Lomnica, Czechoslovakia, September 1977.

[Rimon and Burdick, 1995] Elon Rimon and Joel Burdick. New bounds on the number of frictionless fingers required to immobilize 2D objects. In *IEEE International Conference on Robotics and Automation*, pages 751–757, Nagoya, Aichi, Japan, May 1995.

[Rodríguez *et al.*, 2006] Samuel Rodríguez, Jyh-Ming Lien, and Nancy M. Amato. Planning motion in completely deformable environments. In *IEEE International Conference on Robotics and Automation*, pages 2466–2471, Orlando, FL, May 2006.

[Salleh *et al.*, 2004] Khairul Salleh, Hiroaki Seki, Yoshitsugu Kamiya, and Masatoshi Hikizu. Spreading of clothes by robot arms using tracing method. In *Proc. 5th International Conference on Machine Automation*, pages 63–68, Osaka, Japan, November 2004.