Drawing Venn Triangles

Jeremy J. Carroll
Publishing Systems and Solutions Laboratory
HP Laboratories Bristol
HPL-2000-73
December 21st , 2000*

venn diagram,
triangles,
corners

We use a straight-line graph-drawing algorithm to draw overlapping triangles. The graphs and extended co-linearity constraints are chosen to be graphs of 6-Venn diagrams, using a graph theoretic brute-force search. A method for identifying the corners in a family of intersecting simple closed curves is described. Hence Grunbaum's problem of drawing a Venn diagram of six triangles is solved. Two of the 126 distinct solutions are presented.

* Internal Accession Date Only
Approved for External Publication

Drawing Venn Triangles

Jeremy J. Carroll
Hewlett-Packard Laboratories Bristol, UK
jjc@hpl.hp.com

Abstract

We use a straight-line graph-drawing algorithm to draw overlapping triangles. The graphs and extended co-linearity constraints are chosen to be graphs of 6-Venn diagrams, using a graph theoretic brute-force search. A method for identifying the corners in a family of intersecting simple closed curves is described. Hence Grünbaum's problem of drawing a Venn diagram of six triangles is solved. Two of the126 distinct solutions are presented.

1. Introduction

This paper presents an extended example of how to use the straight-line graph drawing algorithm presented in Carroll [2]. This consists of a solution to Grünbaum's Venn diagram of six triangles problem.

We find all 126 graph theoretic solutions using an exhaustive graph-theoretic brute force search, and then draw them using the straight-line graph-drawing algorithm.

Issues that occur in this problem that may reoccur in other straight line drawing problems are:

- Corners, where are they?
- Which constraints are really part of our problem?
- Specifying the relationship of the corners to other lines in the figures.

2. Venn Diagrams and Venn Triangles

In 1975, Grünbaum [3] first posed the problem of drawing a simple Venn diagram of six triangles. This is six triangles such that:

- No three lines are concurrent
- The triangles divide the plane into 64 regions each of which is inside its own unique subset of the set of triangles.

The clearest such diagram, produced by the work described in this paper, is shown in

Figure 1.

Ruskey [5] provides a very helpful survey of work in Venn diagrams; and we should note Grünbaum and Winkler's 5-Venn Triangles [4].

From a graph theoretic point of view a simple n-Venn diagram is a labelled plane graph G with labels taken from $\{1,2,\cdots,n\}$ such that:

- The edges with label i form a cycle C_i that we refer to as the i-th curve of the Venn diagram.
- Every vertex of G lies on exactly two of these curves.
- G has 2^n faces and for each subset of $\{C_1,C_2,\cdots,C_n\}$ there is exactly one face inside the selected curves and outside the other curves.

A somewhat more general type of graph is a FISC or family of intersecting simple closed curves, described by Bultena et al.[1]

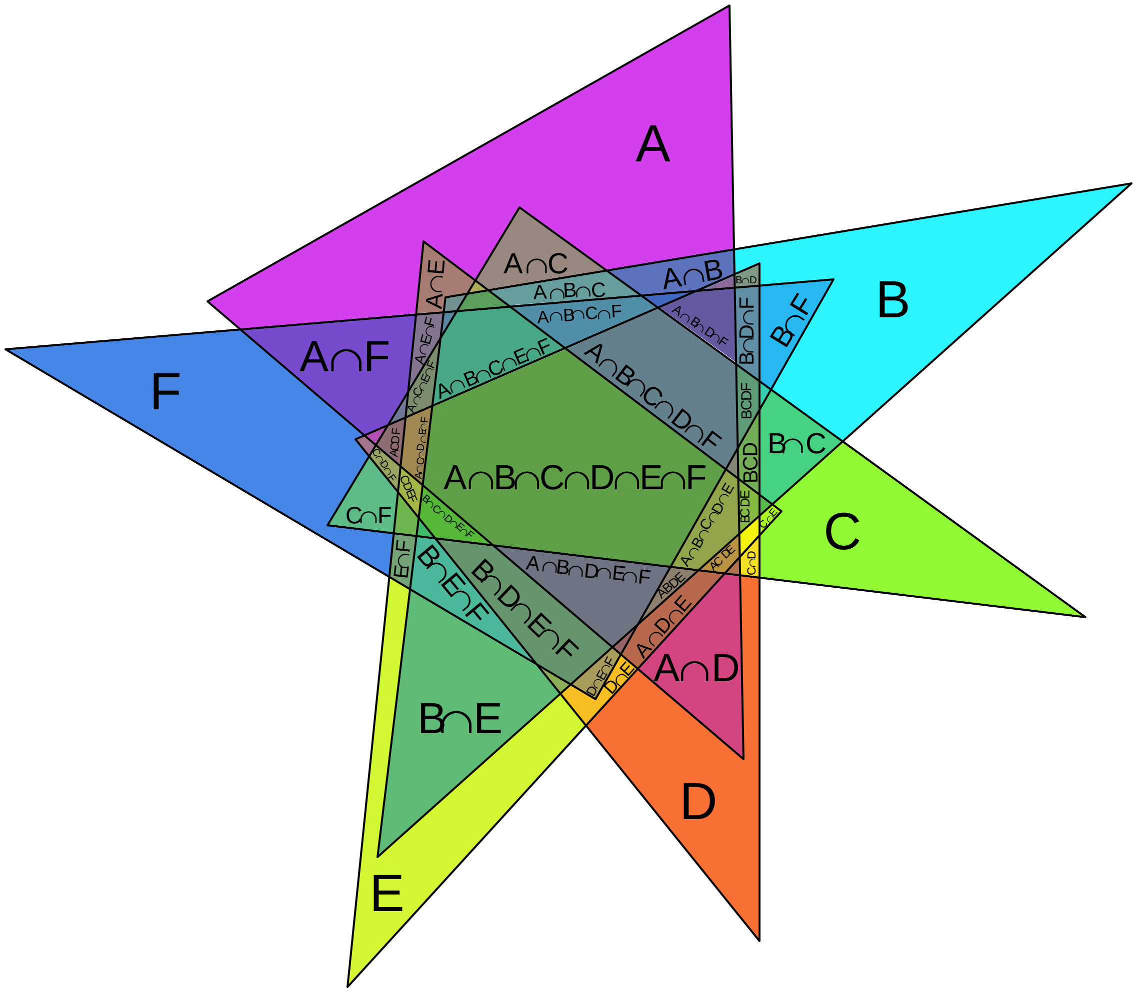


Figure 1 A Venn Diagram of Six Triangles

3. Straight-line drawing

The algorithm presented in Carroll [2] draws graphs subject to co-linearity constraints. It requires a planar graph and specified “straight paths” as input. It outputs the polar coordinates of a line for each of the straight paths such that the lines together form a drawing of the graph. Additional input required is facial boundary information and a detailed description of the relationship between the end-points of each straight path and some other straight paths that are otherwise unrelated. Further, an aesthetic function measuring the beauty of a solution (in terms of its polar coordinates) is required.

The basic structure of the algorithm is to convert all the graph theoretic information into inequalities over polar coordinates of a solution and then to solve the inequalities using non-linear programming techniques. The inequalities are of two types linear angle inequalities, expressing basic Euclidean constraints on the angle ordinates, and the betweenness inequalities that relate to triangles formed by any three straight paths in the figure.

4. Finding Venn Triangles

The approach we take to the 6-Venn Triangle problem is to first conduct a brute force search to find all 6-Venn diagrams, and then to filter the results to give only those diagrams that plausibly can be drawn as triangles. Some of these filtering considerations were moved into the search algorithm itself in order to speed things up. The exhaustive search took approximately 0.5e15 machine instructions (i.e. 2 months on the hardware used).[1]

4.1 Graph Theoretic Venn Diagrams

From a graph theoretic point of view an n-Venn diagram is a plane graph with edges labelled from $\{1,2,\ldots n\}$, such that for each label the corresponding set of edges forms a cycle known as a curve. The interiors of these curves correspond to the traditional “sets” of a Venn diagram. We note that a plane graph has faces, and each face has a cycle of edges going around it, called its facial cycle. For a Venn diagram there are 2^n faces, each being the intersection of the interior of a unique subset of the set of curves.

4.2 Facial Cycle Search

The underlying idea behind the search is that a 6-Venn diagram has 64 faces by definition; and that each of these faces has a facial cycle with between three and six edges with no one curve appearing more than once. This gives ${}_6C_3 2! + {}_6C_4 3! + {}_6C_5 4! + {}_6C_6 5! = 394$ different possible facial cycles per face; and bounds the total search space from above by $394^{64} \approx 10^{166}$. Standard search techniques and considerations of symmetry allow pruning of the search space down to a tractable number.

As the facial cycle for each face is chosen this constrains the possible facial cycles of adjacent faces. Whenever the search algorithm considers a partial solution in which one of the partial curves of the diagram is in two or more disconnected components, one of which is a cycle then that solution and all its children are rejected. This is because a 6-Venn diagram consists of 6 curves, and a cycle cannot have a proper subcycle.

[1] Reflecting upon the results it was clear that further substantial performance improvements could have been achieved by a more thorough use of convexity constraints.

4.3 Graph-theoretic Triangles

Given that we want to draw each curve as a triangle we have a number of simple graph theoretic observations:

- Each pair of curves intersects at most 6 times.
- Each triangle is a convex figure hence the theorem of Bultena et al. [1] applies and the Venn diagram is monotone (i.e. its directed dual graph has exactly one source and one sink).

Filtering using these criteria is unproblematic.

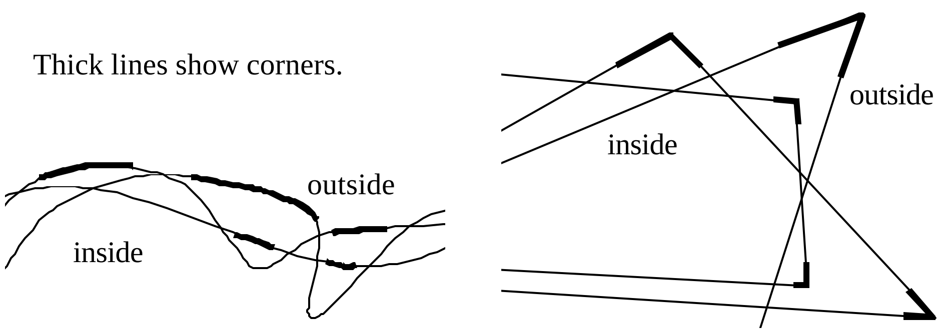
More complex is that we need to identify three edges on each curve such that we can add an additional subdividing vertex on each of these edges. We then form three paths along each curve between the subdividing vertices.

We now have 18 paths. If any pair of these paths intersect two or more times, then it is not possible to draw the graph with each path as a single straight line.

Hence we use the following cornering algorithm to count the minimal number of corners on each curve. If any curve requires four or more corners the Venn diagram is rejected.

4.4 Cornering Algorithm

The key observation behind the cornering algorithm is that if we have two piecewise linear intersecting convex curves, and one curve passes from inside the other to outside, and then back inside, then it must pass through a corner.



For a convex Venn diagram we can identify the central face that lies inside all the curves. Moreover, this face has an edge from each of the curves. (Both these follow from Bultena et al.'s convexity theorem [1]).

For each curve C, we start with its edge on the central face, and proceed around the curve in one direction.

We keep track of two sets:

- a set Out of curves outside of which we lie.

- a set *Passed* of curves which we have recently crossed from the inside to the outside.

Both sets are initialised to empty. On our walk around C, as we pass the vertex v we look at the other curve C' passing through that vertex.

If C' is in *Out* then:

- We remove C' from *Out*.
- If C' is in *Passed* then we set *Passed* as the empty set and add v to the result set. The idea is that there must be a corner between any two vertices in the result set.

Otherwise, C' is not in *Out* and:

- We add C' to *Out*.
- We add C' to *Passed*.

At the end of the walk we look at the cardinality of the result set. This tells us the minimum number of corners required on this curve.

By conducting a similar walk in the opposite direction around the curve we get a corresponding result set. We can align these two result sets, and find sub-paths along which a corner must lie. For each sub-path one end lies in one result set and the other end in the other.

We arbitrarily choose one edge in each of these subpaths and subdivide it with an additional vertex.

If any curve has fewer than three corners found with this algorithm then additional corners are added arbitrarily.

5. Flexibility in Cornering

During the cornering analysis there is an arbitrary choice of which of the edges in the corner is subdivided. There are further choices about how to place a third corner when the cornering analysis only finds two corners.

These choices can be thought of as uncertainty about the betweenness properties in the straight line drawing. i.e. in what order does one of the lines cross the other lines, noting in particular that line with which it forms a corner. The geometric concept of betweenness is expressed in the graph-drawing algorithm by the betweenness inequalities involving triangles in the figure (see Carroll [2]). We can express this uncertainty about the drawing by removing some of the betweenness inequalities from the set to be solved in the algorithm. In particular, for the Venn triangles we remove any betweenness inequality involving two sides of one of the triangles and a third line that crosses one but not both of the first two lines.

6. Corner information

We need to give the graph-drawing algorithm a clear idea of the relationship between the corners of the triangles (i.e. the end-points of the straight paths) and other straight paths in the figure.

To do this, for each pair of paths P and Q that form two sides of a triangle, and for each R that does not intersect either P or Q, we may specify an unrelated vertex v for which there are two v-R straight paths. v lies on the same side of R as $P \cap Q$. In this way we have specified which side of R the corner should lie.

For Venn Triangles the following suffices:

> For each Venn triangle[2] t;
>
> for each pair of sides P and Q of t;
>
> for each Venn triangle t' containing $P \cap Q$;
>
> for each side R of t' that does not intersect P or Q;
>
> > then $P \cap Q$ lies on the same side of R as the corner of t' opposite to R.

This statement is necessary. Sufficiency is less clear. It has proved sufficient in practice; in that each of the drawings of the 126 6-Venn triangles do have the Venn property.

7. The radius and inner radius

Before invoking the graph-drawing algorithm we need to specify an aesthetic objective.

This can be any function of the polar coordinates of the lines that is locally differentiable and convex in the vicinity of a solution. We can also combine many such objectives with max or min.

For the Venn triangle problem we choose two different aesthetic functions. Both are linked to the distance of the corners of the triangles from the origin in the central face. If l_1 and l_2 are two sides of a triangle with polar coordinates (d_1, q_1) and (d_2, q_2); then the distance between the origin and the corner l_1 n l_2 is given by[3]:

$$\frac{d_1}{\left|\cos\left(q_1 - \tan^{-1}\left(\frac{d_2\cos(q_1) - d_1\cos(q_2)}{d_1\sin(q_2) - d_2\sin(q_1)}\right)\right)\right|}$$

---

[2] The term Venn triangle indicates one of the curves of the Venn diagram that is to be drawn as three straight paths. This is to distinguish it from some other triangle in the figure consisting of three vertices linked by three straight paths (which may be subpaths of the sides of the Venn triangles).

[3] I am not wholly convinced that there isn't a simpler formula!

The first objective function is to minimise the maximum value of this distance over all the corners in the Venn diagram[4]. This produces diagrams where, for a given clarity of diagram, the piece of paper needed to draw the diagram is as small as possible. Figure 1 and Figure 2 use this objective.

Following Grünbaum and Winkler [4], it is often preferable to omit the outer corners of Venn triangles, leaving them to the readers' imagination. We hence use a second aesthetic function that minimizes the inner radius, i.e. where we omit the corners on the outer face from consideration. (We then need to specify some minimum angle for these corners, e.g. 0.1°. Such minimums can be added to the non-linear program used by the graph-drawing algorithm.) Figure 3 uses this objective, but is the same diagram as Figure 2.

We have now found all the required inputs to the graph drawing algorithm and so we invoke it. The polar origin used for these diagrams lies in the central face.

---

[4] In practice the non-linear programming system used found it easier to maximise the minimum value of:

$$\frac{\left|\cos\left(q_1 - \tan^{-1}\left(\frac{d_2\cos(q_1) - d_1\cos(q_2)}{d_1\sin(q_2) - d_2\sin(q_1)}\right)\right)\right|}{d_1}$$

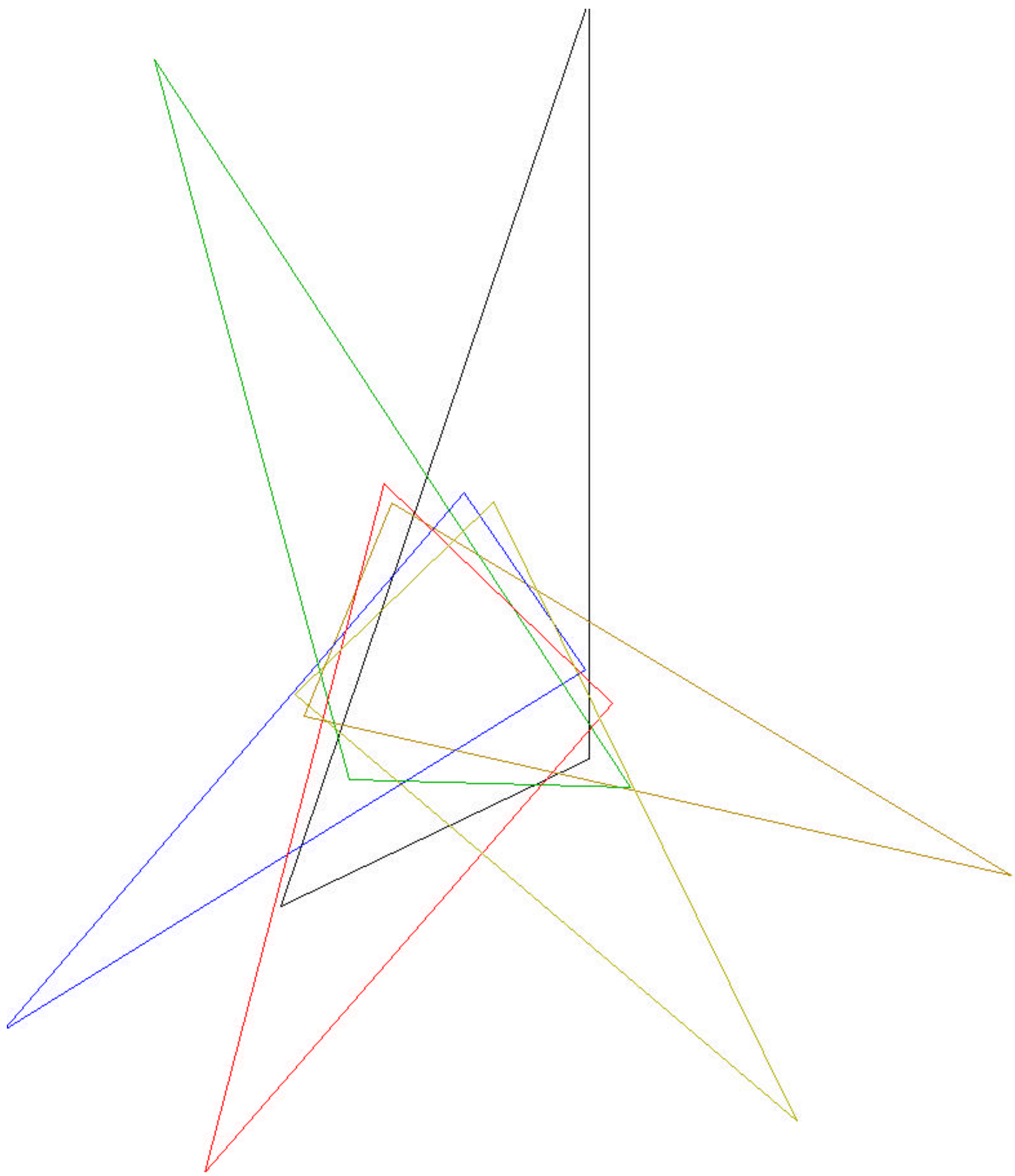


Figure 2 Tighter 6-Venn Triangles

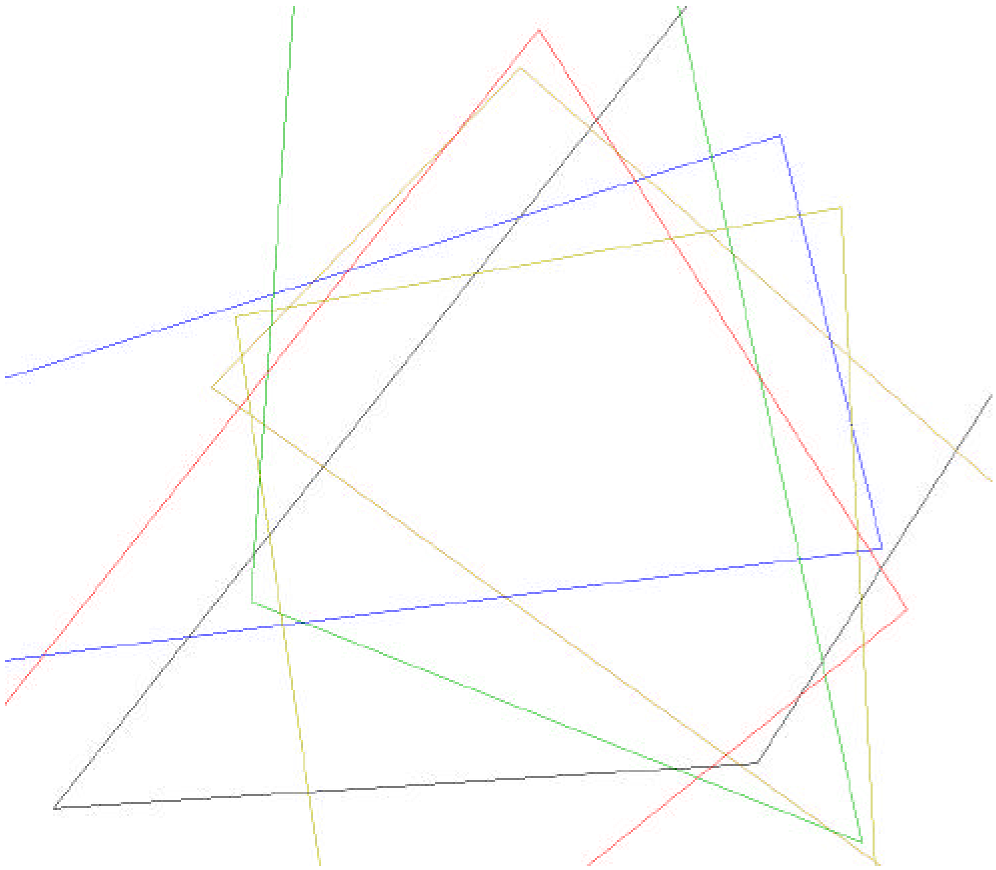


Figure 3 The distorted centre of Figure 2

8. Notes on the Results

126 sets of 6-Venn triangles have been found. There are no others.

Two measures of size or difficulty are found, the radius (i.e. the maximum distance of a corner from the origin) and the inner radius (i.e. excluding corners on the outer face). The smallest solution Figure 1 has radius 49 and inner radius 38. The unit for all these figures arises as the minimal value of $\frac{4\Delta^2}{abc}$, which is very approximately the height of the smallest triangle in the figure (a b and c are the sides of any triangle formed in the figure, Δ the area).

For diagrams formed by minimising the inner radius, the inner radius ranged from 20 to 532, with the (outer) radius ranging from 183 and up. For diagrams formed by minimising the radius it ranged from 49 to 3065, with inner radius ranging from 38 up to 826.

These figures suggest that to draw Venn triangles when millimetre precision is available then Figure 1 can be drawn in a 5 centimetre square piece of paper whereas the worst-case figure would require 3 meters.

9. Future Directions

The same techniques could be used for finding higher order Venn diagrams formed by convex polygons with more sides. More care would be needed to take advantage of the convexity of the curves to prune the search space, or we could limit ourselves to a non-exhaustive search.

Further work on angles is needed to answer questions like can any of these 6-Venn triangles be drawn with equilateral triangles, or can we form a Venn diagram from 7 rectangles.

An alternative approach to limiting the search space size at higher orders would be to restrict the search to symmetric Venn diagrams.

10. References

[1] B. Bultena, B. Grünbaum, and F. Ruskey, *Convex Drawings of Intersecting Families of Simple Closed Curves*, 1998. Presented at the 11th Canadian Conference on Computational Geometry, (1999), 18-21.

[2] J.J. Carroll, *Drawing Straight Lines*. Hewlett-Packard Laboratories Report, No. HPL-2000-72, Bristol, U.K., 2000.

[3] B. Grünbaum, *Venn diagrams and independent families of sets*. Mathematics Magazine 48, 1975. pp12-22.

[4] B. Grünbaum and P. Winkler, *A Venn diagram of 5 triangles*. Mathematics Magazine 55, 1982. p311.

[5] Frank Ruskey, *A Survey of Venn Diagrams*, The Electronic Journal of Combinatorics 4, DS#5, 1997, http://www.combinatorics.org/Surveys/ds5/VennEJC.html.