# A New Force-Directed Graph Drawing Method Based on Edge-Edge Repulsion*

Chun-Cheng Lin, Hsu-Chun Yen†
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106, ROC
(sanlin@cobra.ee.ntu.edu.tw, yen@cc.ee.ntu.edu.tw)

## Abstract

*The conventional force-directed methods for drawing undirected graphs are based on either* vertex-vertex *repulsion or* vertex-edge *repulsion. In this paper, we propose a new force-directed method based on* edge-edge *repulsion to draw graphs. In our framework, edges are modelled as charged springs, and a final drawing can be generated by adjusting positions of vertices according to spring forces and the repulsive forces, derived from* potential fields*, among edges. Different from the previous methods, our new framework has the advantage of overcoming the problem of* zero angular resolution*, guaranteeing the absence of any overlapping of edges incident to the common vertex. Given graph layouts probably generated by classical algorithms as the inputs to our algorithm, experimental results reveal that our approach produces promising drawings (especially for trees and hypercubes) not only preserving the original properties of a high degree of symmetry and uniform edge length, but also preventing zero angular resolution. By allowing vertex-vertex overlapping, our algorithm also results in more symmetrical drawings.*

*Keywords:* Force-directed method, potential field, angular resolution.

## 1 Introduction

As graphs are known to be one of the most important abstract models in various scientific and engineering areas, *graph drawing* (or *information visualization* in a broader sense) has naturally emerged as a fast growing research topic in computer science. Among various graph drawing techniques reported in the literature, the so-called force-directed methods (see, e.g., [1, 3, 4, 5, 8]) have received much attention and have become very popular for drawing general, undirected graphs. In such a framework, a graph is viewed as a system of particles with forces acting between the particles, and then a good configuration or drawing of the particles could be generated with locally minimal energy, i.e., the sum of the forces on each particle is zero.

Generally speaking, the notions of repulsions in the setting of conventional force-directed methods fall into two categories, namely, *vertex-vertex repulsion* and *vertex-edge repulsion*. First, Eades [5] presented a *vertex-vertex repulsion* model in which vertices are replaced with charged steel rings and edges with springs to form a mechanical system. The equilibrium configuration, where the sum of repulsive forces due to rings and attractive forces due to springs is zero, normally results in a good drawing. Frunchterman and Reingold [8] subsequently presented an effective modification of the model.

In subsequent studies, *vertex-edge repulsion* models have been proposed to prevent a vertex from being placed too close to an edge, overcoming a potential shortcoming as a result of using the vertex-vertex repulsion model. Davidson and Harel [4] used the paradigm of simulated annealing, suited for combinatorial optimization problems, to draw graphs. Their method tries to find an optimal configuration according to a cost function inclusive of a measure for the distance between each pair of vertex and edge. This measure penalizes the vertex and edge that are too close to each other. In addition, Bertault [1] presented a force-directed method based on vertex-edge repulsion to ensure that two edges cross in the final drawing if and only if they cross in the initial layout as well.

*Aesthetic criteria* specify graphic structures and properties of drawing, such as minimizing number of edge crossings or bends, minimizing area, and so on, but the problem of simultaneously optimizing those criteria is, in many cases, NP-hard. Among important aesthetic criteria, *angular resolution* refers to the smallest angle formed by two neighboring edges incident to the common vertex in straight line drawing, and constructing straight-line drawings of huge graphs with large angular resolution is very important in visualization applications and, in addition, the design of optical communications networks [7]. Formann et al. [7] were the first to study the angular resolution of straight-line drawings, and showed that deciding whether a graph of maximum degree $d$ has an embedding with resolu-

*This work was supported in part by NSC Grant 93-2213-E-002-003.
†Corresponding author

tion $2\pi/d$ (obvious upper bound) is NP-hard.

The main aesthetic criteria concerned in this paper are symmetry, uniform edge length, and maximization of angular resolution. Theoretical and experimental results (see, e.g., [6] and [5]) have suggested that force-directed methods usually enjoy the merit of producing graph layouts with a high degree of symmetry and uniform edge length. However, their graph layouts may have the problem of *zero angular resolution*[1], i.e., there exist at least two of the edges incident to the common vertex overlapping, resulting in a bad drawing with edge-edge and vertex-edge crossings simultaneously. In this paper, a new force-directed method using the concept of *edge-edge* repulsion based upon the theory of *potential fields* is presented to draw graphs without zero angular resolution. The concept of potential fields has already found applications in a variety of areas in computer science and engineering, such as path planning [2] and drawing of graph with nonuniform nodes [3], among others. In our setting, the repulsive forces applied to an edge are caused by its neighboring edge being present in the potential field.[2] Although [2] has derived analytically formulas of repulsive forces between two charged edges respectively, they are unnecessarily complicated to implement practically. Therefore, as we shall see later in our derivation, the formulas of our edge-edge repulsion are very simple and can be implemented easily. Given a nice graph layout probably generated by classical force-directed methods as the input of our algorithm, the experimental results reveal that our approach can produce a drawing not only preserving a high degree of symmetry and uniform edge length but also preventing zero angular resolution. By allowing vertex-vertex overlapping, our method often results in more symmetrical drawings. Finally, because our model is suitable for drawing trees for which local minimal problems are not as critical as for general graphs, our method can be applied to producing dynamical *balloon view* drawings of rooted trees usually used in the field of information visualization.

The rest of this paper is organized as follows. Similar to other force-directed methods, our method involves two parts, i.e., the spring model and the optimal algorithm introduced in Section 2 and Section 3 respectively. Finally, a conclusion is given in Section 5.

## 2 Model of Edge-Edge Repulsion

Our force-directed method with edge-edge repulsion is based upon the idea of replacing edges by charged springs,

[1]Although the simulated annealing method additionally considering a term of angular resolution can be applied, it's not efficient.

[2]Note that if the repulsive forces are considered by all pairs of edges incident to a common vertex, then evenly angular resolution cannot be guaranteed.

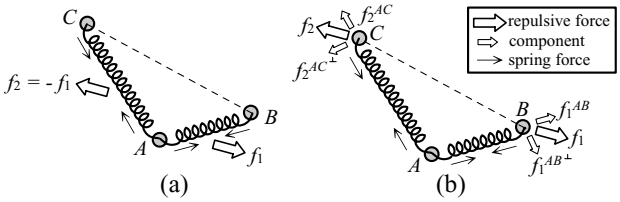


**Figure 1. A graph with three vertices $A$, $B$, and $C$ and two edges modelled by charged springs. (a) The force model where $f_1$ and $f_2$ are repulsive forces acting on $\overline{AB}$ and $\overline{AC}$, respectively. (b) The positions acted by repulsive forces $f_1$ and $f_2$ should be set at the end points $B$ and $C$ of incident edges of vertex $A$.**

as opposed to charging nodes as was done in [5]. The closer two adjacent charged edges are, the stronger the repulsive force between them becomes. Intuitively, larger repulsive forces should make the included angle between two neighboring edges wider. In our design, in addition to stretching the included angle, the repulsive forces also contribute to increasing the lengths of the two edges. Thus the positions acted by repulsive forces are set at the end points of the incident edges as Figure 1 explains. On the other hand, spring forces pull vertices closer when spring lengths are longer than their natural spring lengths. Finally, a drawing without zero angular resolution is generated when the corresponding model reaches an equilibrium between those repulsive forces and spring forces. With a given embedding, two edges are said to be *neighboring edges* if they share a common endpoint, and one is the successor of the other along a clockwise or counter-clockwise rotation. (See $\overline{AB}$ and $\overline{AC}$ in Figure 1.)

In what follows, the formulas for capturing spring forces and repulsive forces are described. The formula of a spring force is based upon the classical spring embedder model [5], which uses the following logarithmic function as the force formula:

$$f_a(d) = C1 \log(d/C2) \tag{1}$$

where $d$ is the spring length, and $C1$ and $C2$ are constants to control the magnitude of the spring force and the natural spring length, respectively.

The generalized formulas of the repulsive force due to two charged edges can be derived from the theory of *potential fields*. The reader is referred to [2] for more about potential fields as well as some of the detailed derivations of exact formulas. However, those formulas derived in [2] appear to be a bit complicated and consequently require special care when implementing such a method. From a practical viewpoint, such a complication may not be needed for the purpose of drawing graphs. Therefore, by observing

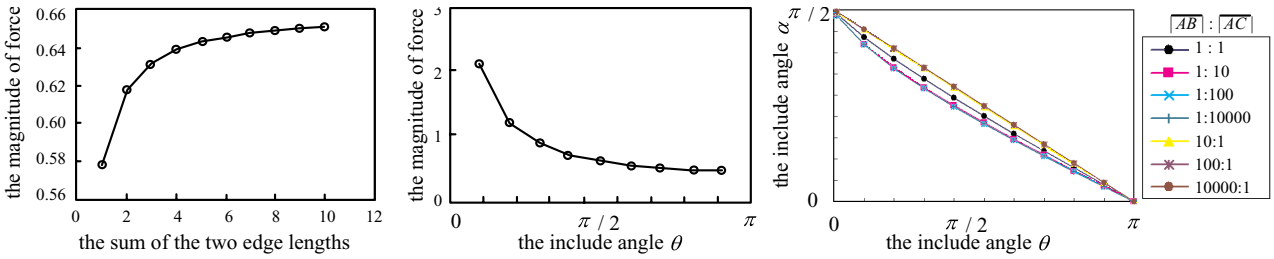


**Figure 2. Curves displaying the relationships between the magnitude of force and (a) the sum of edge lengths (the tendency between the magnitude of force and the length of each edge is similar.); (b) the included angle of two uniformly charged edges. (c) The experimental results designed for measuring the orientation of force. Consider a variety of ratios of $|\overline{AC}|$ to $|\overline{AB}|$ to plot $\alpha$ versus $\theta$.**

some characteristics of edge-edge repulsion and experimental results of [2], we are able to derive a simplified version of repulsive forces. Experiments based on the model reveal encouraging and promising results, as reported in the next section.

The key in our edge-edge repulsion model is to express the repulsive forces between two neighboring edges solely in terms of the lengths of the two edges and the included angle between the two edges. To better explain what this means, consider Figure 1-(b) as an example. It is easy to observe that the magnitudes of the repulsive force due to the two edges $\overline{AB}$ and $\overline{AC}$ are

1. *positively* correlated with the lengths of $\overline{AB}$ and $\overline{AC}$;
2. *negatively* correlated with the angle between $\overline{AB}$ and $\overline{AC}$.

Figures 2-(a) and (b) are curves, based upon the implementation of formulas derived in [2], displaying how the magnitude of the repulsive force between $\overline{AB}$ and $\overline{AC}$ (see Figure 1-(b)) is related to the length (Figure 2-(a)) and the included angle (Figure 2-(b)) of two uniformly charged edges. As Figure 2-(a) indicates, the relationship between the magnitude of force and the total length of edges is asymptotically nondecreasing and concave. Intuitionally, the magnitude should approach to zero as edge lengths approach to zero, and flatten out as edge lengths approach to infinity. It can be captured by an *arctangent* function on $(0, \infty)$, and thus the component $|f|_e$ of magnitude of the repulsive force due to the two edge lengths can be simplified as follows:[3]

$$|f|_e = C3\left(\tan^{-1}(\frac{|\overline{AB}|}{C4}) + \tan^{-1}(\frac{|\overline{AC}|}{C4})\right) \qquad (2)$$

where $C3$ and $C4$ are constants to control the height of the approaching horizontal line and the scale of the horizontal axis, respectively.

[3] $|\overline{AB}|$ and $|\overline{AC}|$ are the lengths of $\overline{AB}$ and $\overline{AC}$, respectively.

Figure 2-(b) shows the relationship between the angle included by $\overline{AB}$ and $\overline{AC}$ and the magnitude of force. It turns out that the curve is asymptotically positive, nonincreasing and convex. The magnitude approaches to infinity as the included angle approaches to zero. On the other hand, the magnitude slowly flattens out as the included angle grows. Such a behavior can be captured by a *cotangent*[4] function on $(0, \pi/2]$, and hence the component $|f|_\theta$ of magnitude of the repulsive force due to the included angle can be set as follows:

$$|f|_\theta = C5\cot(\frac{\theta}{2}) \qquad (3)$$

where $C5$ is a constant to control the scale of the vertical axis, and $\theta$ is the angle included by $\overline{AB}$ and $\overline{AC}$. Note that Figure 2-(b) shows that the magnitude value at $\pi$ is not zero, which, in fact, should be regarded as the contribution to edge lengths. In addition, for avoiding the appearance of values near infinity, $|f|_\theta$ is set to some fixed value when $\theta$ is below certain small cutoff value.

Therefore, the total magnitude $|f|$ can be computed as the sum of (2) and (3) in the following:

$$|f| = |f|_e + |f|_\theta\,. \qquad (4)$$

In what follows, to compute the orientation of repulsive force, consider an angle included by two edges $\overline{AB}$ and $\overline{AC}$ as shown in Figure 3. There exist two angles included by the two edges, and the angle with degree smaller than $\pi$ is denoted as $\theta$. In the process of computing orientation, however, we need to determine which one is $\theta$, and which edge is its right or left included edge. Assume that $u^{AB}$ and $u^{AC}$ are unit vectors of $\overrightarrow{AB}$ and $\overrightarrow{AC}$, respectively, and $u^M$ is the unit vector of the sum of them. Based upon $u^M$ and one of $u^{AB}$ and $u^{AC}$, Proposition 2.1 by using the formula of cross product below allows us to judge which is the right

[4] Intuitively, $|f|_\theta$ should be zero when $\theta$ approaches $\pi$. Figure 2-(b) doesn't display the behavior because the total force involves nonzero $|f|_e$.

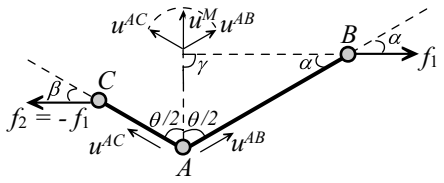


**Figure 3. Illustration of orientation of the repulsive force due to two edges $\overline{AB}$ and $\overline{AC}$ with an included angle $\theta$.**

or left included edge of the smaller included angle $\theta$, and then assign the right (resp. left) edge denoted as $\overline{AB}$ (resp. $\overline{AC}$), and its corresponding unit vector as $u^{AB}$ (resp. $u^{AC}$) accordingly.

**Proposition 2.1** *Assume that $u^{AC} = (a, b)$ and $u^M = (c, d)$, $\overline{AC}$ is the left edge of the included angle $\theta$ if and only if $(c * b - a * d) > 0$.*

Because the angle included by $u^{AB}$ and $u^M$ is equal to $\theta/2$, $\theta/2$ can be uniquely computed by the inner product formula of cosine function as follows:

$$\frac{\theta}{2} = \cos^{-1}\left(\frac{u^M \cdot u^{AB}}{|u^M||u^{AB}|}\right) = \cos^{-1}\left(u^M \cdot u^{AB}\right) \quad (5)$$

where $|u^M|$ and $|u^{AB}|$ are lengths of the unit vectors $u^M$ and $u^{AB}$ respectively, and both equal one.

$f_1$ and $f_2$ are the repulsive forces acting at vertices $B$ and $C$ respectively. $\alpha$ is the acute angle included by $f_1$ and $\overrightarrow{AB}$. In view of the curve shown in Figure 2-(c), it is interesting to observe that $\theta$ and $\alpha$ appear to be correlated with each other according to the following simple linear equation:

$$\alpha + \frac{\theta}{2} = \frac{\pi}{2} \quad (6)$$

where $\alpha$ equals $\pi/2$ when $\theta$ is zero, and $\alpha$ equals zero when $\theta$ is $\pi$. Since, observing (6), $\alpha$ is the complementary angle of $\theta/2$, the orientation $u_{f_1}$ of $f_1$ is perpendicular to $u_M$, i.e., $\gamma = \pi/2$ in Figure 3.

Therefore $u_{f_1}$ can be computed by rotating $u^M$ clockwise with degree of $\pi/2$ as follows:[5]

$$u_{f_1} = f_1/|f| = R(-\pi/2) \cdot u^M = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot u^M \quad (7)$$

Hence, $f_1$ is a vector with magnitude $|f|$ in (4) and orientation $u_{f_1}$ in (7) computed as follows, and $f_2$ equals $-f_1$.

$$f_1 = |f| u_{f_1} \quad (8)$$

Note that the repulsive force in (8) has the advantage that $f_1$ is determined only by three parameters $\overline{AB}$, $\overline{AC}$, and $\theta$, facilitating a simple implementation of our drawing algorithm based upon edge-edge repulsion.

[5] $R(-\pi/2)$ is the rotation matrix of $-\pi/2$.

# 3 Graph Drawing Algorithm

**Algorithm 1** EERepulsion( a nice drawing of graph $G = (V, E)$ )

**Require:** $tmpForce[|V|]$ stores temporary forces of all vertices; $newPos$ and $oldPos$ record the new and old positions of all vertices $V$ respectively;

1: **Input:** a reasonably nice drawing of $G$
**Output:** a nice drawing without zero angular resolution
2: assign initial locations of vertices of $G$
3: determine the neighboring order of adjacency edges of each vertex by using outer product
4: **while** $converged \neq 1$ **do**
5: $converged \leftarrow 1$
6: $oldPosn \leftarrow newPosn$
7: initialize $tmpForce[|V|]$ as zeros matrix
8: **for each** $v$ in $V$ **do**
9: **if** $\exists$ at least two edges incident to $v$ **then**
10: **for all** pair $(e_i,e_j)$ where $e_i = (v, v_i)$, $e_j = (v, v_j)$ are neighboring edges incident to $v$, and $e_i$ is the right edge of their included angle with smaller degree **do**
11: calculate the repulsive force $f_1$ at $e_i$ due to $e_j$ according to (8)
12: $tmpForce[v_i] \leftarrow tmpForce[v_i] + f_1$
13: $tmpForce[v_j] \leftarrow tmpForce[v_j] - f_1$
14: **end for**
15: **end if**
16: **end for**
17: **for each** $e = (v_i, v_j)$ in $E$ **do**
18: calculate the spring force $f_a$ of $e$ according to (1)
19: $tmpForce[v_i] \leftarrow tmpForce[v_i] + f_a$
20: $tmpForce[v_j] \leftarrow tmpForce[v_j] - f_a$
21: **end for**
22: draw graph and simultaneously save new positions to $newPosn$ according to $C6 \times tmpForce[|V|]$ where $C6$ is a constant to control the magnitude of movement in each iteration
23: **if** $\|newPosn - oldPosn\| > \epsilon$ **then**
24: $converged \leftarrow 0$
25: **end if**
26: **end while**

Our drawing algorithm is sketched in Algorithm 1. The algorithm mainly includes three parts: for each vertex, first, lines 8-16 compute the repulsive forces due to each pair of neighboring edges incident to the vertex; second, lines 17-21 compute the spring force of each edge; third, lines 22 draws the graph according to certain scale of the forces acting at each vertex. In addition, the flag $converged$ is used to control the convergence of the algorithm, and the algorithm can reach convergence if appropriate parameters, $C1 - C6$, are given. Assuming that $d_m$ is the maximum degree of vertices, the algorithm in each iteration takes time complexity $O(d_m|V|)$ to compute repulsive forces, $O(|E|)$ to compute spring forces, and $O(|V|)$ to draw graph; hence, the time complexity of the algorithm is $O(d_m|V| + |E|)$. Note that line 3 preprocesses to determine the neighboring order of adjacency edges of each node, and hence costs $O(|V| \times d_m \log d_m)$. The preprocess only runs once since the model of edge-edge repulsion doesn't alter the neighboring order in each iteration as long as each vertex moves slowly under small $C6$.

Conventional force-directed methods cannot guarantee nice drawings for all kinds of graphs because they cannot reach global minimal configuration, i.e., the repulsive forces among some vertices(rings) might be too weak or too strong. Fortunately, some local minimal problems can be resolved by adjusting coefficients of the models. However, the model of edge-edge repulsion doesn't avoid the occurrence of any kind of local minimal because we only locally consider repulsive force of each pair of neighboring edges, and hence the input of our algorithm is restricted to a reasonably nice drawing (see line 1).

## 4 Implementation and Experimental Results

Based on the formulas detailed in the previous sections, we develop a prototype system for drawing undirected graphs. The implementation is in C++ using OpenGL library, and runs on a Pentium IV 3.2GHz PC with memory of size 512MB. Some of experimental results for a variety of graphs using our method and the classical method [5] are presented in Figure 4, and their corresponding statistics are shown in Table 1. About executing time, Table 1 only gives the term of 'average time per iteration' because both the number of iterations and the total running time[6] depend on the settings of different parameters and hence cannot be determined.

For each case in Figure 4, using the initial drawing or the drawing generated by the classical method as the input of our algorithm, our method usually preserves the original properties of a high degree of symmetry and uniform edge length. As shown in (a), our method has the capacity of guaranteeing the drawing of a tree with evenly angular resolution, but the classical method may not. As shown in (b) and (c), given stronger springs, our method may reach the situation of the drawing with the largest angular resolution, but the classical method may not. As shown in (f), given stronger springs, the drawing using our method results in the central star spinning, and hence appears more compact and has more uniform edge lengths.

(g)-(k) draw the $n$-hypercubes [9] with $n = 2$ to 6. Because the model of edge-edge repulsion allows at least two vertices coinciding, our approach may produce drawings with more symmetries. Especially in (g), (j), and (k), although those drawings are improper[7], our method has center symmetry with clear displays, while the classical method only has axial symmetry with somewhat confused displays.

Observing $StdDev/AvgLen$, i.e. the normalized standard deviation of edge lengths, in Table 1, our approach without costing more running time seems to have equal or more uniform edge length than the classical method. As for angular resolution, the classical method may have the problem of zero or few angular resolution (e.g. see (c) and (i)), and our approach normally has larger angular resolution than the classical.

Note that our method may not generate a nice drawing from initial drawing (e.g. see (e) and (i)). Hence two-phase (i.e. first using classical method and then using our method) or hybrid strategies can be applied to improve our method.

[6] Our method sometimes uses the output layout of classical method as input, and, in the case, the total executing time makes no sense.

[7] A drawing is improper if there exist at least two vertices overlapping.

## 5 Conclusion

Different from the conventions of force-directed methods, a new force-directed method based on edge-edge repulsion for generating a straight-line drawing not only preserving the original properties of a high degree of symmetry and uniform edge length but also without zero angular resolution has been proposed and implemented.

A line of future work is to overcome the problem associated with *local minimal* (which also poses difficulties for the conventional force-directed methods) by multilevel techniques or using optimal heuristics, such as simulated annealing, genetic algorithm, etc. It is also of importance and interest to provide more experimental results on graphs of huge size and theoretical results on the power of the model of edge-edge repulsion.

## References


[1] F. Bertault. A force-directed algorithm that preserves edge crossing properties. In *Graph Drawing '99*, pages 351–358. LNCS, Vol. 1731, Springer-Verlag, 1999.

[2] J.-H. Chuang and N. Ahuja. An analytically tractable potential field model of free space and its application in obstacle avoidance. *IEEE Transactions on System, Man, and Cybernetics*, 28(5):729–736, 1998.

[3] J.-H. Chuang, C.-C. Lin, and H.-C. Yen. Drawing graphs with nonuniform nodes using potential fields. In *Graph Drawing 2003*, pages 460–465. LNCS, Vol. 2912, Springer-Verlag, 2004.

[4] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15:301–331, 1996.

[5] P. Eades. A heuristic for graph drawing. *Congress Numerantium*, 42:149–160, 1984.

[6] P. Eades and X. Lin. Spring algorithms and symmetry. *Theoretical Computer Science*, 240(2):379–405, 2000.

[7] M. Formann, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Simvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. *SIAM Journal on Computing*, 22(5):1035–1052, 1993.

[8] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software-Practice and Experience*, 21:1129–1164, 1991.

[9] Mathworld. Introduction to hypercubes. *http://mathworld.wolfram.com/Hypercube.html*.

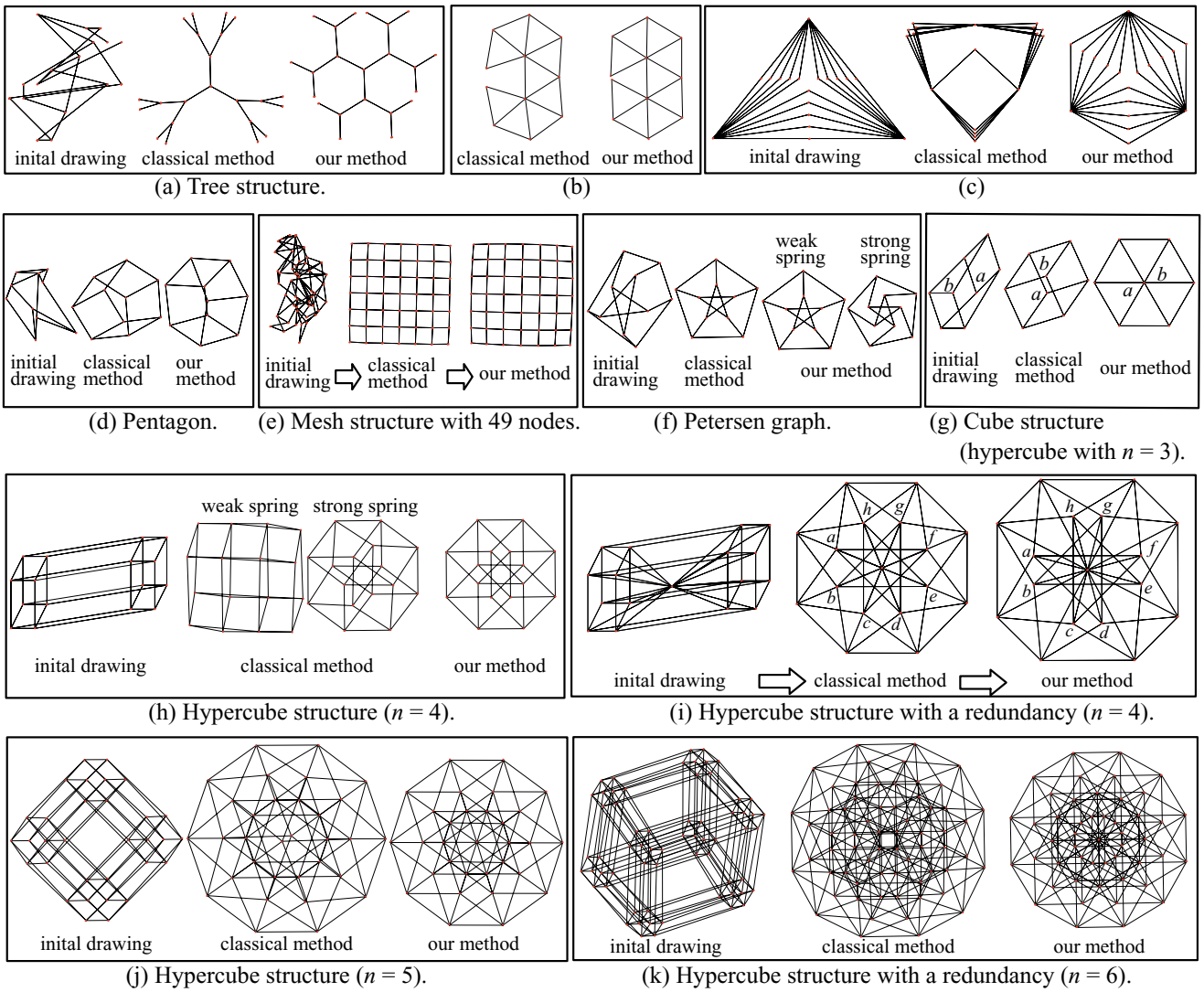


**Figure 4. Experimental results.**

**Table 1. Statistics on the experimental results**

| Graph name | $\|V\|$ | Average time per iteration (sec.) | Average edge length (*AvgLen*) | | Standard deviation of length (*StdDev*) | | $\frac{StdDev}{AvgLen}$ | | Angular resolution (degree) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | both | classic | our | classic | our | classic | our | classic | our |
| Figure 4-(a) | 22 | $< 10^{-3}$ | 1.19 | 1.16 | 0.15 | 0.06 | 0.12 | 0.05 | 25.81 | 119.96 |
| Figure 4-(b) | 11 | $< 10^{-3}$ | 1.01 | 1.01 | 0.02 | 0.01 | 0.02 | 0.01 | 56.87 | 58.86 |
| Figure 4-(c) | 18 | $< 10^{-3}$ | 1.05 | 1.02 | 0.06 | 0.05 | 0.06 | 0.05 | 1.21 | 11.82 |
| Figure 4-(d) | 10 | $< 10^{-3}$ | 1.35 | 1.30 | 0.11 | 0.13 | 0.08 | 0.10 | 23.48 | 48.95 |
| Figure 4-(e) | 49 | $< 10^{-3}$ | 1.26 | 1.03 | 0.08 | 0.01 | 0.06 | 0.01 | 85.17 | 86.64 |
| Figure 4-(f) | 10 | $< 10^{-3}$ | 1.29 | 1.03 | 0.38 | 0.04 | 0.30 | 0.04 | 35.54 | 22.29 |
| Figure 4-(g) | 8 | $< 10^{-3}$ | 1.03 | 1.32 | 0 | 0 | 0 | 0 | 34.68 | 60.00 |
| Figure 4-(h) | 16 | $< 10^{-3}$ | 1.43 | 1.35 | 0.05 | 0.01 | 0.03 | 0.01 | 32.31 | 44.16 |
| Figure 4-(i) | 17 | $< 10^{-3}$ | 1.01 | 1.12 | 0.21 | 0.25 | 0.21 | 0.22 | 0.32 | 12.60 |
| Figure 4-(j) | 32 | $< 10^{-3}$ | 1.70 | 1.38 | 0.09 | 0.01 | 0.05 | 0.01 | 30.23 | 35.35 |
| Figure 4-(k) | 64 | $< 10^{-3}$ | 2.31 | 1.40 | 0.23 | 0.02 | 0.10 | 0.01 | 20.08 | 29.31 |