이진트리의 굴곡점이 없는 수직 도시 알고리즘

허혜정, 이정민, 이주영
덕성여자대학교 전산학과
{hjhur, j020026, jylee}@namhae.duksung.ac.kr

The Orthogonal Drawing Algorithm with no bend for Binary Trees

Hyejung Hur, Jung-Min Lee & Ju-Young Lee
Dept. of Computer Science Duksung Women's University

요 약

본 논문에서는 n개의 정점들로 이루어진 이진트리를 상향과 비상향에 대한 수직 도시 방법으로 기존의 연구에서의 최소 면적인 $O(n\log\log n)$영역에 굴곡점이 없이 도시하는 알고리즘을 개발하였다.

1. 서론

그래프 도시(drawing)란 주어진 그래프를 정의된 미적 기준에 따라 평면이나 공간과 같은 기하학적 대상에 포함시키거나 연관시키는 방법과 체계를 의미한다. 이것은 주어진 상황에서의 연관관계의 파악과 이해를 쉽게 도와주기 위해서 가시적으로 표현해주는 역할을 통해서 그래프에 대한 이해도와 판독성을 증가시킨다.

본 논문에서는 수직(orthogonal) 도시 방법을 사용해서 이진트리를 도시한다. 수직 도시란 각각의 정점(vertex)은 점(point)으로 매핑하고, 각각의 에지(edge)는 수평(horizontal)과 수직(vertical) 라인(line) 요소로서 그려지는 것을 말한다. 이러한 수직 드로잉은 회로 스키마(circuit schematics), 자료 흐름도(data flow diagrams), 엔티티 관계도(entitiy relationship diagrams)등의 많은 응용분야를 갖기 때문에 관심을 받고 있다. 특히, 굴곡점(bend) 수를 최소한으로 줄이면서 수직 도시를 하는 알고리즘을 발견하려는 연구가 많이 이루어지고 있다[1, 2, 3, 4]. 기존에 발표된 연구 중 이진트리를 수직 도시 방법으로 굴곡점을 줄이면서 도시한 대표적인 알고리즘인 Tamassia의 알고리즘에서는 $O(n\log\log n)$영역에 $O(n)$만큼의 굴곡점을 가지며[1], 같은 면적인 $O(n\log\log n)$에 현재까지 연구된 최소한의 굴곡점은 $O(\frac{n}{\log n})$만큼이다[2]. 본 논문에서는 이진트리를 수직 도시 방법으로 $O(n\log\log n)$영역에 굴곡점이 없이 도시하는 알고리즘을 개발하였다.

2. 용어 정의

이 장에서는 본 논문에서는 필요로 하는 기본적인 정의들에 대해서 살펴 볼 것이다.

루트가 있는 어느 한 트리(tree)를 T라 할 때 T에 속하는 한 정점을 v라 하면 이 v의 높이(height)는 정점 v에서부터 마지막 잎(leaf) 정점들까지의 path들의 최대한의 길이라 할 수 있다. 또한 어느 한 트리의 사이즈를 나타내는 $size(T)$는 T가 가지고 있는 정점들의 갯수라 할 수 있다. T의 서브트리를 t라 할 때 t의 면적은 t를 포함하는 최소의 사각형 면적이라 할 수 있으며 이때 t의 너비(width)와 높이(height)는 그 사각형의 가로길이와 세로길이이다.

수직 도시(orthogonal drawing)란 각각의 정점(vertex)은 점(point)으로 매핑하고, 각각의 에지(edge)는 수평(horizontal)과 수직(vertical) 라인(line) 요소로서 그려지는 것을 말한다.

상향(Upward) 도시는 비대칭적인 방향성 그래프(digraph)이고 각각의 에지들은 모두 다 비증가적으로(nonincreasing) 그려지며 자식노드의 y좌표가 부모노드의 y좌표와 같을 수 있지만 클 수는 없다는 특징을 가지는 도시를 말한다. 이 도시는 순서가 있는 방향성 그래프를 포함하는 계층적인 관계를 효과적으로 보여줄 수 있다는 장점을 지니고 있다. 이에 반해 비상향(non-upward) 도시는 항상 에지의 방향의 순서가 비증가적으로 고정되어질 필요가 없이 자유로운 드로잉 기법이라 할 수 있다.

굴곡점(bend)란 수평, 수직 라인이 서로 만나는 곳에 정점이 없을 때 그 만나는 점을 말한다. 각 굴곡점은 일종의 가상적인 정점으로 생각할 수 있다.

3. 수직 도시 알고리즘

이진트리를 수직 도시 방법으로 O($n\log\log n$)영역에 굴곡점(bend)이 없이 도시하는 알고리즘을 개발하였다.

Tamassia의 알고리즘에서는 O($n\log\log n$)영역에 O(n)만큼의 굴곡점을 가지며, 같은 면적에 현재까지 연구된 최소한의 굴곡점은 O($\frac{n}{\log n}$)만큼이다.

우선, 트리를 조각으로 나누는 방법에 대해서 알아보자. 본 논문에 기술된 알고리즘은 트리를 정의 3.1에 따라 조각(fragment)과 분리자(seperator)로 나누고 조각들을 각각 도시한 것을 분리자로 조각들을 연결하는 것을 원칙으로 한다.

정의 3.1

트리의 전체 정점(vertex)의 개수를 n이라고 할 때, 각 정점을 루트로 하는 서브트리가 가지고 있는 전체 정점의 수를 $\log n$으로 나누어서 내림한 것을 k하고 할 때, k가 자노드들의 k보다 크다면 그 노드를 *critical*하다고 말한다. *critical*한 정점 주변의 모든 에지들을 분리자들로 정하고, 분리자들을 제거하면 $\frac{n}{\log n}$개의 조각들로 나누어진다.

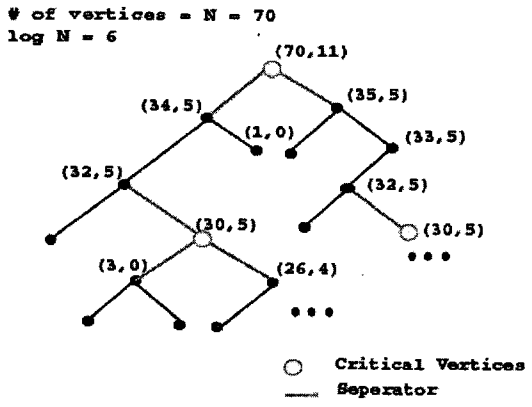
정리 3.1

정의 3.1에 따라 O($\log n$)만큼의 정점들로 이루어진 조각들이 전체 트리에 대해서 O($\frac{n}{\log n}$)만큼 생성된다.

증명

O($\frac{n}{\log n}$)의 조각들이 전체 트리에 대해서 생성된다는 위 사실에 따르면 분리자들의 수는 적어도 $\frac{n}{\log n}$개가 생기며 그에 따른 조각들의 수는 O($\frac{n}{\log n}$)이 생성된다. 따라서 조각들은 적어도 O($\log n$)만큼의 정점을 가지고 있다. ■

(그림 1)은 정의 3.1에 정의된 critical vertices와 분리자들을 예를 통해 보여주고 있다.

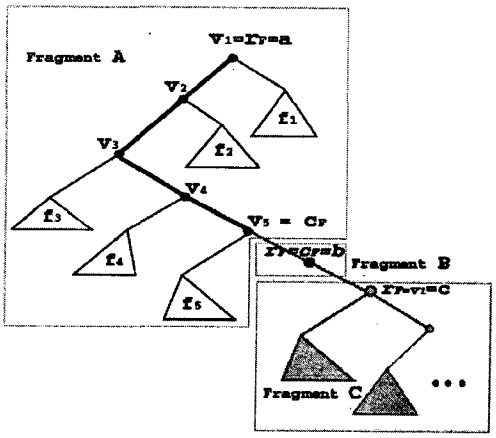


(그림 1) 조각 트리 만들기

3.1 상향 수직 도시 알고리즘

n개의 정점들로 구성되어진 이진트리를 평면 상향 수직 도시를 하기 위한 알고리즘은 다음과 같다.

(i) 정의 3.1에 따라 critical vertices와 분리자들을 찾아서 조각트리로 만든다. (그림 2)는 조각트리로 이루어진 트리의 예이다.

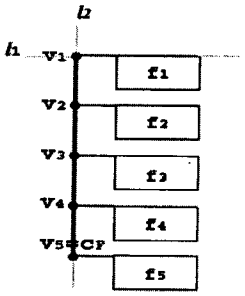


(그림 2) 조각들로 이루어진 트리의 예

정의 3.2

각 조각내에서 가장 상위레벨에 있는 정점을 r_F로서 정의하고, 다음 조각으로 연결되어지는 정점을 c_F로서 정의한다. 다음 조각으로 연결되어지는 정점이 없다면 임의의 정점을 c_F로서 정의한다.

(ii) (그림 2)에 있는 Fragment A를 r_F와 c_F사이의 패스(path)를 설정한 후 패스 위에 존재하는 노드들을 v_i (1 ≤i≤ h)라 한다. r_F는 v_1이 되며, c_F는 v_h가 된다. v_i의 자식 서브 트리를 f_i라 한다. 각 조각들은 (그림 3)과 같은 원칙에 따라 배치한다. 각 서브트리는 수직(vertical)알고리즘을 적용해서 도시한다.



(그림 3) 조각 배치

(iii) 각 조각들과 분리자로 구성되어진 트리를 left-heavy 트리로 변경한 후, 각 조각들을 수직적으로(vertical) 배치한다.

정리 3.2

상향 수직 도시 알고리즘은 $O(n)$ 시간 복잡도를 가지고 이진 트리를 $O(n\log\log n)$ 영역에 굴곡점 없이 도시한다.

증명

각 조각은 $\log n$만큼의 정점들로 구성되어지므로 (그림 3)과 같은 방법의 알고리즘을 통해서 각 조각의 너비(width)는 $O(size(f_i)) = O(\log n)$이고, 높이(height)는 $O(\Sigma_i size(f_i)) = O(size(F)) = O(\log\log n)$가 된다. 각 조각들의 내부를 도시하기 위한 영역은 $O(\log n\log\log n)$이다. 각 조각들을 분리자를 통해 연결할 때 너비는 조각의 너비에 자식 조각이 두 개 있을 때 분리자만큼만 더해지므로 전체 너비의 변화 없이 $O(\log n)$이고, 높이는 조각의 높이와 전체 조각의 수를 곱한 것으로 $O(\log\log n)\times O(\frac{n}{\log n}) = O(\frac{n\log\log n}{\log n})$이다. 따라서 전체 면적은 $O(\log n)\times O(\frac{n\log\log n}{\log n}) = O(n\log\log n)$이다. 그리고 트리를 조각트리로 만들었을 때 여러개의 정점을 가진 조각 다음에는 반드시 하나의 정점을 가지는 하나의 조각(critical vertex)이 이어진다. 임의의 조각에서의 c_F가 다음 조각의 r_F와 수직적으로 이어질 수 있도록 c_F의 아래쪽(south open)을 비워놓았기 때문에 각 조각들을 분리자로 연결할 때 굴곡점이 생성되지 않는다. ■

3.2 비상향 수직 도시 알고리즘

n개의 정점들로 구성되어진 이진트리를 평면 비상향 수직 도시를 하기 위한 알고리즘은 다음과 같다.

(i)과 **(iii)** 과정은 앞 3.1에서 설명한 상향 수직 도시 알고리즘과 같다.

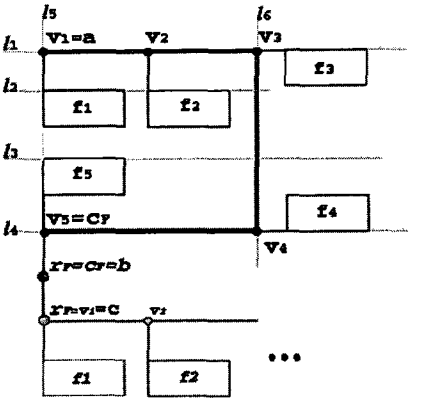
(ii) l_1과 l_5는 조각이 시작하는 y, x 좌표이다. l_2는 l_1+1, 이고, l_3는 $l_2+\max height(f_h,\ldots,f_{h-2})+1$, l_4 는 $l_3+\max height(f_{h-1},f_h+1)$이고, l_6은 $l_5+\sum_{i=1}^{h-3}(width(f_i)+1)$ 이다. (그림 4)와 같이 v_1에서 v_{h-3}까지는 v_5부터 v_6까지 배치하고 각각의 서브트리는 l_2에서 시작하도록 배치한다. v_{h-2}는 l_1과 l_6의 교차점에 위치하도록 하고 그의 서브트리는 오른쪽에 붙인다. v_{h-1}은 l_4와 l_6의 교차점에 배치하고 그의 서브트리는 오른쪽에 배치한 후 180도 회전시킨 위치에 배치한다. v_h은 l_4와 l_5의 교차점에 배치하고 그의 서브트리는 f_1과 같이 아래쪽으로 배치한 후 180도 회전 시켜서 재배치시킨다.

정리 3.3

비상향 수직 도시 알고리즘은 $O(n)$ 시간 복잡도를 가지고 이진트리를 $O(n\log\log n)$ 영역에 굴곡점 없이 도시한다.

증명

각 조각은 $\log n$만큼의 정점들로 구성되어지므로 (그림 3)과 같은 방법의 알고리즘을 통해서 각 조각의 너비(width)는 $O(\Sigma_i size(f_i)) = O(size(F)) = O(\log n)$이고, 높이(height)는 $O(size(f_i)) = O(\log\log n)$가 된다. 각 조각들의 내부를 도시하기 위한 영역은 $O(\log n\log\log n)$이다. 전체 트리의 영역은 앞 3.2에서 증명된 것과 같고, $O(n\log\log n)$이다. 굴곡점이 생성되지 않는다는 증명도 정리3.2에서의 증명과 같다. ■



(그림 4) 비상향에서의 조각내 배치

4. 결론

본 논문에서는 n개의 정점으로 이루어진 이진 트리를 평명 그리드에 그리는 한 방법인 평면 상향 수직 그리드(planar upward orthogonal grid drawing) 방법을 이용하여 너비는 $O(\log n)$, 높이는 $O(\frac{n\log\log n}{\log n})$, 면적 $O(n\log\log n)$은 유지하면서 굴곡점의 수를 없애는 방법을 제시하였고, 또 다른 방법인 평명 비상향 수직 그리드(planar non-upward orthogonal grid drawing) 방법을 이용하여 너비는 $O(\frac{n\log\log n}{\log n})$, 높이는 $O(\log n)$, 면적 $O(n\log\log n)$은 유지하면서 굴곡점의 수를 없애는 방법을 제시하였다.

참고문헌

[1] A. Garg, M. T. Goodrich, and R. Tamassia, "Planar upward tree drawings with optimal area", Internat. J. Comput. Geom. Appl., 6:333-356, 1996.

[2] C.S. Shin, S.K. Kim, and K.Y. Chwa, "Area-efficient algorithms for upward straight-line tree drawings", Proc 2nd International Computing & Combinatorics Conferences (COCOON'96) LNCS, vol. 1090, pp. 106-116, 1996

[3] 김성권, "적은 굴곡점을 가진 이진트리를 그리는 알고리즘", 한국정보과학회논문지:시스템 및 이론, 27권 2호, 페이지 209-215, 2000

[4] M. S. Rahman, S. Nakano and T. Nishizeki, "A Linear Algorithm for Bend Optimal Orthogonal Drawings of Triconnected Cubic Plane Graphs", J. of Graph Algorithms and Application, vol. 3, no. 4, pp.31-64, 1999