트리의 삼차원 대칭성 탐지 알고리즘

홍 석희○ Peter Eades
Dept. of Computer Science & Software Engineering, University of Newcastle
{shhong, eades}@cs.newcastle.edu.au

An Algorithm for Detecting Three Dimensional Symmetry in Trees

Seok-Hee Hong○ Peter Eades
Dept. of Computer Science & Software Engineering, University of Newcastle

요 약

대칭성(symmetry)은 그래프를 가시화하여 기하학적 표현을 구축하는 그래프 드로잉 분야에서 그래프의 구조와 특성을 효율적으로 표현해주는 가장 중요한 평가 기준이다. 하지만 현재까지는 이차원 평면에서의 대칭성 문제에 대해서만 기존 연구가 이루어져왔을 뿐 해상도를 증가시키고 대칭성을 보다 풍부하게 표현할 수 있는 그래프의 삼차원 대칭 드로잉에 관한 연구는 아직 제시된 바 없다.

본 논문에서는 그래프 드로잉에서의 삼차원 대칭성 문제를 연구하였다. 먼저 그래프의 삼차원 대칭 드로잉을 구축하기 위해 필요한 삼차원 대칭성 모델을 제시하고 이를 기반으로 하여 트리에서 삼차원 대칭성을 탐지하는 알고리즘을 제시하였다. 이 알고리즘은 트리의 최대의 대칭성을 보여주는 삼차원 드로잉 알고리즘으로 쉽게 확장이 가능하다.

1. 서 론

그래프 드로잉(graph drawing)이란 그래프가 모델링하는 추상적인 정보를 시각적으로 가시화하여 이차원 평면 또는 삼차원 공간상에 기하학적 표현을 구축하는 것을 의미한다[1]. 좋은 드로잉은 그래프 구조에 대한 이해도를 증진시킬 수 있으며 정보의 의미를 명확하게 전달할 수 있고 판독성을 증가시킨다.

대칭성은 그래프 드로잉시 그래프의 구조와 특성을 시각적으로 표현하는 가장 중요한 평가 기준이다. 또한 대칭 드로잉은 전체 그래프가 크기가 작은 동형의 부그래프들로부터 반복적으로 구성됨을 보여줌으로써 전체 그래프에 대한 이해를 쉽게 해준다[6].

그러나 그래프의 대칭 드로잉에 관한 기존 연구는 이차원 평면상의 대칭성 문제에 대해서만 국한되어 연구되어 왔다. 일반 그래프의 대칭성을 탐지하는 문제는 *NP-complete*로 증명되었으며 트리, 외부평면(outerplanar) 그래프 그리고 임베딩된 평면 그래프의 이차원 대칭성을 탐지하는 알고리즘이 제시되었다[5,6,7]. 또한 직병렬 유향 그래프(series-parallel digraph)와 평면 그래프를 대상으로 이차원 대칭성을 탐지하는 알고리즘이 제안되었다[2,3].

본 논문에서는 그래프의 대칭성을 삼차원으로 확장하였다. 먼저 그래프의 삼차원 대칭 드로잉을 위한 삼차원 대칭성 모델을 제시하고 이를 기반으로 하여 트리에서 최대의 삼차원 대칭성을 탐지하는 알고리즘을 제안한다. 삼차원 대칭 드로잉은 해상도를 증가시키며 보다 풍부한 대칭을 표현할 수 있는 장점을 갖는다.

2. 삼차원 대칭성 모델

삼차원 기하학적 물체의 대칭성의 종류는 rotation, reflection, inversion, rotatory inversion의 4가지로 분류가 가능하다[4]. 이차원의 대칭성은 rotation과 reflection의 두 가지만이 존재하므로 삼차원의 대칭성은 이차원의 대칭성과 비교했을 때 그 종류가 더욱 다양하고 풍부하며 따라서 그 수도 크게 증가하는 양상을 띤다.

이차원의 대칭성과 비교했을 때의 또 다른 차이점은 이차원의 경우 rotation은 점을 중심으로 한 회전이었지만 삼차원의 rotation은 축을 중심으로 한 회전이라는 것이다. 또한 reflection 역시 이차원의 경우에는 축에 대한 reflection이었으나 삼차원의 경우에는 면에 대한 reflection이라는 것이 가장 큰 차이점이 된다.

삼차원 기하학적 물체의 대칭성을 기반으로 하여[4] 그래프의 삼차원 대칭 드로잉을 위한 삼차원 대칭성 모델을 다음과 같이 정립할 수 있으며 이는 크게 다음의 세 가지로 구분 가능하다.

1. 밑면이 정다각형인 피라미드 형태
2. 밑면과 윗면이 동일한 정다각형인 프리즘 형태
3. 플라토닉 솔리드

다음 절에서는 본 논문에서 최초로 제시한 그래프 드로잉을 위한 삼차원 대칭성 모델을 기반으로 하여 트리의 삼차원 대칭성을 탐지하는 알고리즘을 제안한다.

3. 트리의 삼차원 대칭성 탐지 알고리즘

삼차원 대칭성의 가능한 종류는 크게 세 가지 종류가 존재하므로 트리의 삼차원 대칭성을 탐지하는 알고리즘은 다음과 같이 기술된다.

알고리즘 3DSymmetry_Tree

1. 트리의 센터를 찾아 이를 루트로 하는 트리로 변환시킨다.
2. 루트에 연결된 부트리들을 동형성 집합으로 분류하여 각 동형성 집합의 크기를 계산하고 다시 각 부트리에 대하여 이 과정을 재귀적으로 적용하여 동형성 집합 트리 ICT를 구성한다(**알고리즘 Construct_ICT**).
3. 루트에서 다음의 형태를 구성하여 각각의 경우의 대칭성을 계산하여 그 중 최대값을 찾는다(단, 센터가 에지인 경우에는 에지에 가상의 정점을 추가하여 루트로 간주한다).
   3.1 **알고리즘 Pyramid**
   3.2 **알고리즘 Prism**
   3.3 **알고리즘 Platonic_Solid**

3.1 알고리즘 Pyramid

피라미드의 경우에는 하나의 회전 대칭축이 존재한다. 그러므로 이 대칭축 상에 최대 두 개의 부트리를 배치하는 것이 가능하다. 따라서 이 경우는 대칭축 상에 부트리가 없는 경우, 하나의 부트리가 존재하는 경우 그리고 두 개의 부트리가 존재하는 경우로 나누어 생각할 수 있다. 또한 두 개의 부트리가 존재하는 경우는 이들이 동일한 동형성 집합에 속하는 경우와 아닌 경우로 나누어 생각할 수 있다.

센터 c에 연결된 부트리들로 구성된 동형성 집합을 X_i라고 하고 각각의 크기를 n_i 라고 하자. 축 상에 부트리가 존재하지 않는 경우의 회전 대칭의 수 g는 $gcd(n_1, n_2, ..., n_k)$가 된다.

하나의 부트리 C_i가 축 상에 배치되는 경우에는 그가 속한 동형성 집합을 X_i라고 하면 회전 대칭의 수 g는 $gcd(n_1, n_2, ..., n_i - 1, ..., n_k)$가 된다. 단 이때 부트리 C_i가 재귀적으로 g개의 회전 대칭을 가져야만 한다. 따라서 전체 회전 대칭의 수는 축 상에 배치되는 부트리가 갖는 회전 대칭의 수를 g_i 라 할 때 $gcd(g, g_i)$가 된다.

또한 두 개의 부트리 C_i, C_j를 축상에 배치하는 경우, 동형인 경우에는 위의 과정에서 $n_i - 2$를 하고, 동형이 아닌 경우에는 $gcd(n_1, n_2, ..., n_i - 1, ..., n_j - 1, ..., n_k)$와 g_i, g_j를 계산하여 $gcd(g, g_i, g_j)$를 구한다. 따라서 이 값을 최대화하도록 부트리를 선정하여 축 상에 배치시켜야만 한다.

축 상에 배치시키는 부트리를 결정하기 위해서는 좀 더 복잡한 계산이 요구되므로 이를 위해 동형성 집합 트리 (Isomorphism Class Tree) ICT를 정의한다.

ICT의 루트는 센터가 된다. 그에 연결된 노드 v_i는 각 동형성 집합 X_i를 의미하며 각 노드에는 각 동형성 집합의 크기 n_i가 저장되어 있다. 또한 그 동형성 집합에 속한 부트리 C_i가 자체적으로 가질 수 있는 가능한 대칭 드로잉의 대칭의 수로 구성된 리스트 Lv_i를 유지한다. 이 값은 하위 레벨부터 계산되어 진다. 다시 각 v_i는 그 동형성 집합에 속한 부트리 C_i에서 루트를 제거하여 얻어지는 부트리들로 구성되는 동형성 집합들을 나타내는 노드들을 그 자식 노드로 갖는다. 이런 방식으로 ICT는 재귀적으로 구성되며 최하위 레벨에는 하나의 노드로만 구성된 부트리를 나타내는 단말 노드들로 구성된다.

각 레벨마다 노드 vi의 n_i의 값은 동형성 집합으로 분류할 때 계산되며 Lv_i는 최하위 레벨부터 레벨 단위로 계산된다. 최하위 레벨의 노드 v_i의 Lv_i는 $\{n_i, n_i - 1\}$이 된다. 만일 n_i가 1인 경우에는 ∞를 지정한다.

ICT의 내부 노드 v_i의 Lv_i는 다음과 같이 계산된다. v_i의 자식 노드들을 $u_1, u_2, ..., u_k$ 라고 하자. $g_0 = gcd(n_1, n_2, ..., n_k)$, $g_i = gcd(nu_1, nu_2, ..., nu_i - 1, ..., nu_k)$를 계산한다. 각 u_i마다 Lu_i안의 각 원소 e_i에 대하여 $g_i' = gcd(g_i, e_i)$를 계산하여 그 결과가 1이 아니고 동일한 값이 Lv에 없는 경우 g_i'을 Lv에 첨가한다. 그러므로 **알고리즘 Construct_ICT**는 다음과 같이 기술된다.

알고리즘 Construct_ICT

1. ICT를 구성하고 각 노드 v_i마다 n_i를 계산한다.
2. ICT의 단말 노드 v_i의 Lv_i를 $\{nv_i, nv_i - 1\}$로 지정한다(단 n_i가 1인 경우에는 ∞를 지정한다).
3. 최하위 레벨 다음 레벨부터 루트 레벨 이전까지 각 레벨의 노드 v_i에 대하여 다음을 반복한다.
   (노드 v_i의 자식 노드들을 $u_1, u_2, ..., u_k$ 라고 하자.)
   3.1 $g = gcd(nu_1, nu_2, ..., nu_k)$
   if ($g \neq 1$) then $Lv \leftarrow g$
   3.2 for $i = 1$ to k
   $g_i = gcd(nu_1, nu_2, ..., nu_i - 1, ..., nu_k)$
   3.3 for $i = 1$ to k
   for each $e_j \in Lu_i$
   $g_i' = gcd(g_i, e_j)$
   if ($g_i' \neq 1$) and ($g_i' \notin Lv$)
   then $Lv \leftarrow g_i'$

ICT를 구성한 후 루트 레벨에서 루트 r과 그의 자식들 $s_1, s_2, ..., s_k$들로 위에 기술한 4가지 경우로 부트리를 배치시켜 피라미드 형태를 구성하고 각각의 대칭성을 분석하여 그 중에서 최대값을 구한다. 따라서 **알고리즘 Pyramid**는 다음과 같이 기술된다.

알고리즘 Pyramid

1. $g_1 = gcd(s_1, s_2, ..., s_k)$
2. for $i = 1$ to k
   $g_i = gcd(s_1, s_2, ..., s_i - 1, ..., s_k)$
   for $i = 1$ to k
   for each e_j in Ls_i
   $g_i' = gcd(g_i, e_j)$
   if ($g_i' \neq 1$) and ($g_i' \notin Lr$)
   then $Lr \leftarrow g_i'$
   g2 = max(Lr)

3. for i = 1 to k
   $g_i = gcd(s_1, s_2, ..., s_i - 2, ..., s_k)$
   for i = 1 to k
   for each e_j in Ls_i
   $g_i' = gcd(g_i, e_j)$
   if ($g_i' \neq 1$) and ($g_i' \notin Lr$)
   then $Lr \leftarrow g_i'$
   $g_3 = max(Lr)$
4. for i = 1 to k
   for $j = i + 1$ to k
   $g_{ij} = gcd(s_1, s_2, ..., s_i - 1, s_j - 1, ..., s_k)$
   for i = 1 to k
   for $j = i + 1$ to k
   for each e_m in Ls_i, each e_n in Ls_j
   $g_i' = gcd(g_{ij}, e_m, e_n)$
   if ($g_i' \neq 1$) and ($g_i' \notin Lr$)
   then $Lr \leftarrow g_i'$
   $g_4 = max(Lr)$
5. $g = max(g_1, g_2, g_3, g_4)$

3.2 알고리즘 Prism

프리즘의 경우에는 피라미드와 동일한 z축 이외에도 xy 평면상에 최대 두 종류의 회전 대칭축이 존재 가능하다. 따라서 대칭축 상에 위치하는 부트리의 수가 증가하게 된다. 단 xy 평면상의 회전 대칭축 상에 위치하는 부트리의 경우 그 개수는 z축의 회전 대칭축이 k-fold 인 경우 역시 k 개의 copy가 필요하다. 또한 z축 상에 부트리가 위치하는 경우 이들은 반드시 동형이어야 한다.

따라서 전체의 대칭이 최대가 되도록 z축과 xy축 상에 존재 가능한 두 종류 부트리를 결정해야 한다. 이들 세가지 종류에 대해 각각 존재하거나 존재하지 않는 경우로 나누어 생각할 수 있으며 존재하는 경우는 다시 동일한 부트리이거나 아닌 경우로 나누어진다.

각 경우에 대하여 프리즘 형태를 구성하여 대칭성을 분석하는 **알고리즘 Prism**은 **알고리즘 Pyramid**와 유사하므로 자세한 알고리즘의 기술은 생략한다.

3.3 알고리즘 Platonic_solid

플라토닉 솔리드에는 정4면체, 정6면체, 정8면체, 정12면체 그리고 정20면체 다섯 가지가 존재한다. 그러나 정6면체와 정8면체, 그리고 정12면체와 정20면체는 서로 듀알(dual)이므로 플라토닉 솔리드의 대칭성은 크게 세 가지로 구분된다.

정4면체에는 3-fold 회전축이 4개 그리고 2-fold 회전축이 3개 존재하고 정6면체와 정8면체는 4-fold 회전축이 3개, 3-fold 회전축이 4개, 2-fold 회전축이 6개 존재한다. 그리고 정12면체와 정20면체에는 5-fold 회전축이 6개, 3-fold 회전축이 10개, 2-fold 회전축이 15개 존재한다[4].

그러므로 각 경우에 부트리를 배치시키는 경우를 고려해야 한다. 즉 이는 크게 세가지 종류의 부트리를 축 상에 배치하는 것을 의미하므로 프리즘과 유사한 방법으로 계산이 가능하며 자세한 알고리즘의 기술은 생략한다.

플라토닉 솔리드는 가장 풍부한 대칭성을 갖는다. 따라서 트리를 플라토닉 솔리드의 대칭성을 유지할 수 있도록 구성하여 배치할 수 있다면 이 경우 삼차원 대칭성이 가장 극대화된다.

3.4 full symmetry group

지금까지는 회전 대칭만을 계산해 왔다. 그러나 full symmetry group에는 reflection과 inversion, rotatory inversion을 포함한다.

k-fold 회전 대칭을 갖는 피라미드에는 k개의 reflection이 존재한다. 프리즘의 경우에는 z축에 k-fold 회전 대칭을 갖는 경우 $k+1$개의 reflection과 $k-1$ 개의 inversion을 포함한 rotatory inversion을 갖는다.

정4면체의 경우에는 24개, 정6면체의 경우는 48개, 정12면체의 경우에는 120개의 대칭이 존재한다. 따라서 full symmetry group의 크기가 최대가 되도록 구성을 선택해야 한다.

[정리 1] 트리의 삼차원 대칭성을 탐지하는 알고리즘은 $O(n^3)$ 시간이 소요된다(단 n은 정점의 개수이다).

4. 결 론

대칭성은 그래프 드로잉시 그래프의 구조와 특성을 시각적으로 표현하는 가장 중요한 미적 기준이다.

본 논문에서는 그래프의 삼차원 대칭 드로잉을 구축하기 위해 필요한 삼차원 대칭성 모델을 제시하고 이를 기반으로 하여 트리에서 삼차원 대칭성을 탐지하는 알고리즘을 제안하였다. 이 알고리즘은 트리의 최대의 대칭성을 보여주는 삼차원 드로잉 알고리즘으로 쉽게 확장이 가능하다.

향후 연구과제로는 다양한 그래프를 대상으로 그의 삼차원 대칭 드로잉을 구축하는 알고리즘을 개발하는 것이다.

참고문헌

[1] G. Di Battista, P. Eades, R. Tamassia and I. G. Tollis, Graph Drawing : Algorithms for the Visualization of Graphs, Prentice-Hall, 1998.

[2] S-H. Hong, P. Eades and S-H. Lee, "Finding Planar Geometric Automorphisms in Planar Graphs," Algorithms and Computation(Proc. ISAAC'98), LNCS vol. 1533, pp 277-286, Springer Verlag, 1998.

[3] S-H. Hong, P. Eades, A. Quigley and S-H. Lee, "Drawing Algorithms for Series-Parallel Digraphs in Two and Three Dimensions," Graph Drawing (Proc. GD'98), LNCS vol. 1547, pp. 198-209, Springer Verlag, 1998.

[4] E. H. Lockwood and R. H. Macmillan, Geometric Symmetry, Cambridge University Press, Cambridge, 1978.

[5] J. Manning and M. J. Atallah, "Fast Detection and Display of Symmetry in Trees," Congressus Numerantium, 64, pp. 159-169, 1988.

[6] J. Manning, "Geometric Symmetry in Graphs," Ph.D. Thesis, Purdue Univ., 1990.

[7] J. Manning and M. J. Atallah, "Fast Detection and Display of Symmetry in Outerplanar Graphs," Discrete Applied Mathematics, 39, pp. 13-35, 1992.