

영상처리와 그래픽스 기술의 접목을 위한 자료구조의 설계 및 구현

이태환^o, 채옥삼
경희대학교 전자계산공학과

Design and Implementation of Data Structure for Combination Image Processing and Graphics

Tae-Hwan Lee^o and Ok-Sam Chae
Dept of Computer Engineering, Kyunghee Univ.

요 약

영상처리의 결과로 얻어지는 예지나 선분과 같은 영상특성들은 그래픽의 기본 요소인 프리미티브들로 표현될 수 있다. 영상처리 결과를 그래픽 프리미티브로 표현하게 되면 보다 효과적으로 visualization할 수 있고 그래픽 환경의 편집기능을 이용하여 새로운 그래픽스를 생성할 수 있다. 반대로 그래픽 기술을 이용하여 생성된 영상은 패턴인식이나 Model Based 삼차원 물체인식 등에 사용될 수 있다. 이처럼 영상처리와 그래픽스는 서로 밀접한 관계를 가지고 있지만 이 두 분야를 지원하는 자료구조는 이를 반영하지 않고 있다. 영상처리란 위한 자료구조는 신속한 접근을 목표로 개발된 반면 그래픽스를 위한 자료구조는 공간절약과 빠른 display를 목표로 설계되었다. 본 연구에서는 영상처리와 그래픽스를 동시에 효과적으로 지원할 수 있는 자료구조를 설계하고 구현한다.

1. 서 론

최근 컴퓨터를 이용하는 분야는 게임, 영상/비디오, 가상현실, 입체영화, TV광고 등 다양하다. 이러한 응용 분야에서는 카메라를 통해서 입력된 영상이 컴퓨터에서 생성된 그래픽스와 결합됨으로써 보다 현실감 있고 이해도가 높은 그림을 만들어내고 있다. 영상처리 측면에서도 컴퓨터그래픽스와 영상처리 기술의 결합은 매우 큰 의미를 갖고 있다. 영상처리 결과로 얻어지는 선분, 예지, 영역(region)들과 같은 영상특징들(image features)은 그래픽의 기본 요소들로 간주될 수 있다. 만약 이들을 그래픽 환경에서 사용할 수 있는 자료구조로 표현한다면 영상처리 결과를 효과적으로 보여줄 수 있고 이들을 여러 용도로 편집될 수 있다. 예를 들어서 영역분할 결과로 얻어진 물체의 경계를 그래픽 도메인에서 arc나 ellipse와 같은 도형으로 근사화시켜서 새로운 그래픽 객체를 만들거나 이것을 다시 영상인식에 사용할 수도 있다. 따라서 그래픽 생성과 영상처리를 상호보완적으로 지원할 수 있는 개발환경의 구축은 멀티미디어 기술의 진보와 함께 필연적인 과제라 할 수 있다. 이를 위해서는 영상처리 결과를 그래픽 영역에서 사용할 수 있고 그래픽 영역에서 생성된 결과를 영상처리 영역에서 효과적으로 사용할 수 있는 자료구조의 설계가 필요하다. 그러나 지금까지의 연구는 각 분야별로 각자의 특성에 맞게 독자적으로 진행되었기 때문에 두 분야를 동시에 지원하기 위해서는 새로운 자료구조의 설계가 필요하다. 예를 들어서 영상처리를 위해서 개발된 IUE[1], PIKS[2], Khoros[3] 같은 영상처리 시스템에서는 영상과 영

상분할 결과 얻어지는 영상특징들을 저장할 자료구조를 설계함에 있어서 신속한 접근을 최우선으로 하였고[4][5], 대부분의 그래픽 시스템들의 자료구조는 저장 공간의 절약과 빠른 rasterize를 목적으로 개발되었다.[6][7] 이러한 구조는 영상처리 영역과 그래픽스 영역을 위한 자료 형식들 사이의 호환성을 고려하지 않고 개발되었기 때문에 상호간의 자료 공유나 전달이 어렵다. 따라서 본 연구에서는 영상처리와 그래픽스 영역을 동시에 지원할 수 있는 자료구조를 설계하고 구현하여 적합성을 시험하고자 한다. 또한 구현된 자료 구조 및 코드의 최적화 및 재사용성을 위해 객체-지향 기법을 이용하였다.

2. 자료 구조 설계시 고려점

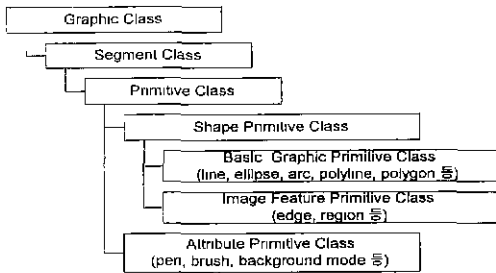
본 연구에서는 영상처리 및 그래픽스 영역을 지원하는 자료 구조의 설계를 목적으로 한다. 영상처리는 자료에 빠르게 접근하고 쉽게 표현하여 영상에서 추출한 특성값들을 효율적으로 표현하는 것이 목적이고 그래픽스는 빠르게 출력하고 쉽게 그리며 rasterize를 목적으로 한다. 따라서, 영상처리와 그래픽스를 모두 지원하기 위해서는 다음과 같은 것이 고려되어야 한다.

- 영상처리 결과를 저장하고 입출력하기 위한 자료 구조와 함수들을 포함하며 자료 구조를 빠르게 액세스하고 재사용할 수 있어야 한다.

- 영상처리를 위한 자료 구조와 그래픽스를 위한 자료 구조를 체계적인 계층으로 분리하여 필요한 정보만을 사용함으로써 수행속도와 메모리의 효율을 증대시킬 수 있어야 한다.
- 영상처리 결과와 그래픽스의 프리미티브를 동일하게 처리할 수 있어야 한다.
- 그래픽스 영역의 기본 기능은 유지되어야 한다

3. 자료 구조 설계

그래픽스의 오브젝트(프리미티브) 특성상 서로 독립된 객체들이기 때문에 객체-지향 개념이 적합하다. 따라서 자료 구조 설계시 “클래스”를 사용하였다. 본 연구의 목표는 영상처리와 그래픽스 영역을 지원하는 자료 구조의 설계이다. 그래픽스의 기능을 유지하기 위해 그래픽스의 기본 계층 구조를 유지하면서 또한 영상처리를 지원하기 위해서 영상처리의 결과들(영상특성)도 그래픽스의 프리미티브로 표현하여 쉽게 저장하고 재사용될 수 있도록 하였다. 그림 1을 보면 Shape Primitive Class 아래에 기본 그래픽 프리미티브와 영상처리 결과를 저장하기 위한 프리미티브들이 있는 것을 볼 수 있다

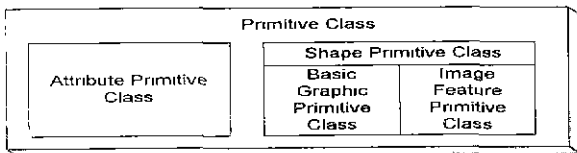


[그림 1. 계층도]

이것은 영상처리의 결과를 그래픽스의 프리미티브들과 구분하지 않고 동일한 알고리즘을 사용하여 처리할 수 있다는 장점을 가진다

3.1 프리미티브의 구조

그래픽스의 특성상 여러 개의 프리미티브들이 연결되어 있을 경우 디스플레이의 속도를 빠르게 해야하기 때문에 그림 2에서처럼 프리미티브 클래스를 크게 두 가지의 형태로 분리하였다. 하나는 단순히 프리미티브들의 형태(모양)를 표현하기 위한 특성값을 가지는 형태(Shape) 프리미티브 클래스이고 다른 하나는 프리미티브들의 색이나 두께와 같이 속성들을 표현하기 위한 속성(Attribute) 프리미티브 클래스이다



[그림 2. 프리미티브의 구조]

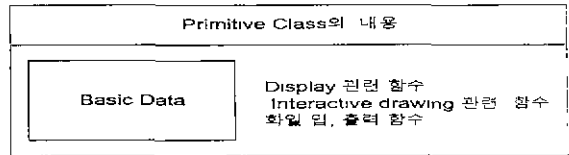
- 형태 프리미티브 (Shape Primitive)

형태를 나타내는 프리미티브에서 그래픽스의 프리미티브와 영상처리 결과를 표현하는데 이것은 두 가지 형태로 나뉜다. 첫번째 것은 Basic Graphics Primitive Class이다 이것은 영상처리의 결과와 기존의 그래픽스의 프리미티브가 일치하는 것으로 line, ellipse, curve 등과 같은 것이 포함된다. 이것은 클래스내에 그래픽스에서 사용되는 정보를 유지하고 영상처리에서 필요한 특성값들을 추가한 것이다. 두번째 것은 Image Feature Primitive Class로 영상처리에서 사용되는 값(edge, region)들을 표현하기 위한 클래스이다

• 속성 프리미티브 (Attribute Primitive)

그래픽스에서는 일괄적인 디스플레이를 하기 위해서 프리미티브들을 리스트 구조로 연결한다. 기존의 것들은 리스트에 연결된 하나의 프리미티브를 디스플레이하기 위해 항상 자신의 속성을 설정한 후 그려야하는 문제가 있었다. 본 연구에서는 이런 오버헤드를 줄이기 위해 대략 6가지 정도로 필요한 속성값들을 구분하여 각각 프리미티브로 표현한다. 이러한 속성값들은 자신에 해당하는 속성이 설정되면 설정된 속성을 나타내는 클래스가 리스트에 연결된다. 이러한 구조는 하나의 프리미티브가 선행하는 프리미티브와 동일한 속성을 가질 경우에는 다시 속성을 설정할 필요 없이 비로 디스플레이함으로써 속도를 향상시킬 수 있다.

기존의 그래픽스 제품들은 시스템이나 운영체제에 의존적이었는데 본 연구에서는 그림 3처럼 Display나 파일 입, 출력 부분과 같은 시스템 의존적인 부분을 분리하였다. Basic Data라는 것은 영상처리나 그래픽스에서 사용되는 실제 데이터(중심점, 시작점 끝점 등)와 그것을 설정하는 함수들(Get(), Set() 등)로 구성된다.



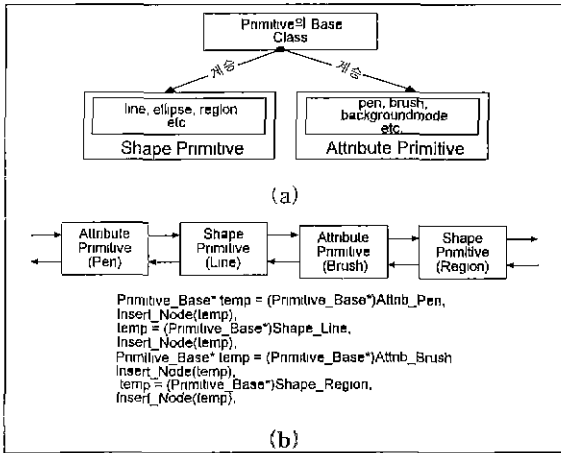
[그림 3. 형태 프리미티브 클래스의 내용]

영상처리에서는 디스플레이보다 실제 데이터(영상 특성값)를 사용하여 새로운 영상특성 추출을 목표로 하기 때문에 효율적으로 자료를 전달할 수 있어야 한다. 이를 위해 Basic Data 부분만을 때어내어 영상처리의 알고리즘간 자료 전달이나 그래픽스와의 자료 전달에 최적화된 모듈을 사용함으로써 속도 및 메모리의 효율을 높일 수 있다. 또한 Basic Data 부분과 해당 시스템 및 운영체제에 알맞는 디스크객이 루틴을 사용하면 언제든지 원하는 시스템에 적합한 프로그램들 개발할 수 있기 때문에 확장성이 높고 표준화된 자료구조의 제공이 가능하다.

3.2 프리미티브의 리스트 설계

클래스로 정의된 프리미티브들을 연결하기 위해 리스트(더블 링크드 리스트)구조를 사용한다. 하지만 위에 제시된 것과 같이 각 프리미티브가 서로 다른 타입의 클래스로 구성되면 서로 연결하기가 어렵다. 이를 위해서는 클래스에 구애 받지 않고 모든 클래스의 pointer를 받는 일반화된 클래스가 필요하다. 이에 본 연구에서는 모든 프리미티브들이 그림 4의 (a)처럼 베이스 클래스를 상속받아 생성되도록 하

었다. 이것은 형태 프리미티브와 속성 프리미티브를 구분하지 않고 단순히 베이스 클래스의 포인터형을 사용함으로써 모든 프리미티브들을 쉽게 연결할 수 있도록 해준다.

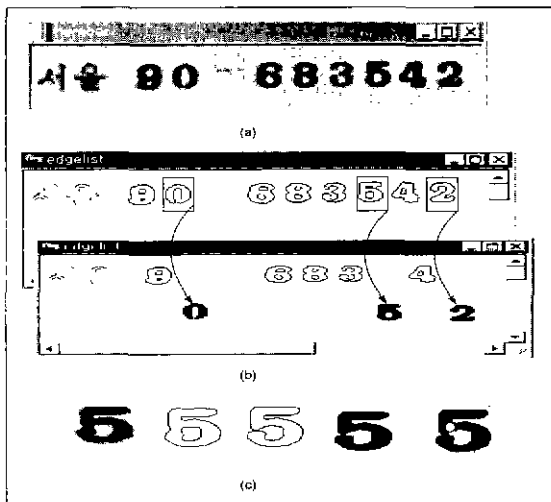


[그림 4. 프리미티브의 연결]

그림 4의 (b)에 서로 다른 형태인 속성 프리미티브와 형태 프리미티브가 베이스 클래스의 포인터 형태를 사용하여 리스트로 연결되는 구조와 이를 위한 실제 코드가 나타나 있다. 그림 4에서 처럼 프리미티브들은 베이스 클래스의 멤버 함수를 상속받았기 때문에 리스트에 연결된 프리미티브들은 동일한 멤버 함수를 포함하게 된다. 리스트를 순차적으로 따라 가면서 이 멤버 함수를 수행하게 되면 빠르게 액세스할 수 있고 리스트에 연결된 프리미티브들을 일괄적으로도 처리할 수 있다

4. 구현 및 결과

본 연구에서 구현된 자료 구조의 적합성 여부를 테스트하기 위해 테스트 환경이 필요할때 본 연구실에서 개발된 영상처리 알고리즘 개발 도구인 Hello-Vision에 이식하여 구현하였다



[그림 5. 결과]

자료 구조의 적합성 실험은 영상으로부터 feature를 추출하여 Basic Data Class에 값을 저장하고 그래픽 도메인으로 넘겨 Display 및 Interactive Drawing 함수들을 사용하여 수정과 수정된 결과를 다시 영상화하는 과정을 통해 이상적인 모델을 생성할 수 있는지 실험해 보았다. 그림 5의 (a)는 카메라로 입력받은 영상이고 (b)는 edge detection을 하여 Basic Data Class에 값을 넣고 그래픽 도메인으로 전달하여 디스플레이한 다음 임의로 이동시키고 내부를 여러가지 색으로 채운 결과이다. 이 과정을 통해 구현된 자료 구조가 영상처리의 그래픽 도메인에서 모두 제대로 동작함을 일 수 있었다 그리고 그림 5의 (c)는 (b)의 과정과 유사하게 원본에서 숫자 5를 확대하여 canny edge detection을 적용한 후 정확한 형태를 얻기 위해 그래픽 도메인에서 interactive한 수정을 하고 재영상화한 것이다 이 과정에서는 구현된 자료구조를 사용하여 영상에 있는 숫자 5를 완전하게 복원함으로써 이상적 모델 생성이 가능함을 확인하였다.

5. 결 론

본 연구에서 영상처리와 그래픽스 영역을 모두 지원하는 자료 구조를 설계하기 위해 그래픽스의 프리미티브 계층에 영상처리 결과를 저장할 수 있는 자료 구조를 추가하였다 구현된 자료구조를 영상처리와 그래픽스 모두에 적용하여 이상적 모델을 생성하는 실험을 해보았는데 제안된 자료구조가 영상처리와 그래픽 영역에서 모두 사용 가능하고 효율적인 데이터 전달과 공유가 가능하다는 것을 확인할 수 있었다 또한 구현된 자료 구조를 사용하여 이상적 모델을 생성하거나 재모델링에도 적합하다는 것을 알 수 있었다 향후 본 연구에서 제안된 자료구조를 표준화하여 새로운 시스템이 개발될 수 있음을 보여 주었다.

6. 참고 문헌

- [1] J.L. Mundy and IUE Committee, Spatial objects in the Image Understanding Environment. Proceedings : IMAGE UNDERSTANDING I, pp 317-331, 1993
- [2] William K. Pratt, PIKS Foundtion C Programmers Guide, Prentice Hall, pp 18-26, 1995
- [3] Khoros Program Service, Khoral Research INC. <http://www.khoral.com/>
- [4] Dana H. Ballard, Christopher M. Brown, Computer Vision, Prentice Hall, pp. 119-165, 1982
- [5] Ranesh jam, Rangachar Kasturi, Brian G Schunck, MACHINE VISION,McGRAW-HILL INTERNATIONAL EDITIONS, pp 73 105, 140-180, 186-223, 1995
- [6] Joseph E. Kasper David Arns, Graphics Programming with PHIGS and PHIGS PLUS. Addison Wesley, pp 101-195, 258-338, 1993
- [7] 지용화, 박래홍, 여러 스케일을 이용한 디각형 근사화 알고리즘, 한국정보과학회 논문지 93 3 Vol 20, No 3, pp. 293-304