

# 비트맵과 벡터방식을 혼합한 이미지 편집도구 구현에 관한 연구

김진호\*, 이규남\*, 나인호\*

\*군산대학교

## A Study on Implementation of Image Editing Tool based on Combining of Bitmap and Vector Image

Jin-ho Kim\*, Kyu-nam Lee\*, In-ho Ra\*

\*Kunsan National University

E-mail : golbaeng2@kunsan.ac.kr

### 요 약

이미지 데이터는 표현 방식에 따라서 비트맵과 벡터 방식으로 나누어진다. 비트맵 방식은 이미지 데이터를 픽셀이라는 2차원 맵으로 구성한 것이며, 벡터방식은 주어진 2차원 또는 3차원 공간에 선이나 형상을 그리기 위해 일련의 명령어나 수학적 표현으로 나타낸 것이다. 따라서 서로 다른 방식의 이미지를 제작 및 편집하고자 하는 경우에는 각기 다른 응용프로그램을 사용해야 문제점이 있다. 본 논문에서는 비트맵과 벡터 방식의 혼합을 기반으로 하여 하나의 응용프로그램에서 이미지의 제작 및 편집이 가능한 이미지 편집 도구의 설계 및 구현 방법에 대하여 기술하였다.

### ABSTRACT

It is possible to classify image data into two types according to the internal representation: one is bitmap, the other is vector. A bitmap image is represented by the two dimensional pixels whereas a vector image is represented by mathematical functions to draw vector objects such as line, rectangle and circle on the two or three dimensional space. So it is necessary for users to use a individual application program for each different image. In this paper, we present a method for designing and implementation of image editing tool based on combining of bitmap and vector image.

### I. 서론

정보를 표현하기 위해서 텍스트, 동영상, 이미지, 사운드 등 멀티미디어 매체를 이용하고 있다. 그 중에서 시각적이 효과를 내기 위해 많이 사용되는 것이 이미지 매체이다.

이미지 매체는 그림, 차트, 그래프 같은 영상 매체들이 많은 정보를 함축적으로 표현할 수 있다. 사진이나 그래픽 영상들을 디지털형태로 컴퓨터에 저장하기 위해서는 많은 저장 공간을 필요로 하는 단점이 있으나 자료의 시각적 효과가 뛰어나므로 멀티미디어 환경에서 지금까지 많이 사용하였던 텍스트정보보다는 그래픽과 이미지로 구성된 정보의 사용이 늘고 있다.

일반적으로 그래픽과 이미지라는 용어는 구별 없이 사용되고 있으나 좀더 명확히 구분한다면, 그래픽 편집기 등을 이용하여 제작한 데이터를 그래픽(graphic) 데이터라 하고, 스캐너나 비디오 보드를 이용해 얻어진 데이터를 이미지(image) 데이터라고 한다. 다시 말해, 그래픽은 선, 원, 사각형 등을 이용해 그려진 것으로 벡터 그래픽(vector

graphics), 벡터 드로잉(vector drawing)이라고 한다. 그래픽을 생성하는 소프트웨어를 드로잉 도구(drawing tool), 그래픽 편집기(graphic editor)라고 한다. 이미지는 픽셀을 2차원 맵으로 구성한 것이며, 비트맵 이미지(bitmap image), 래스터 이미지(raster image)라고 한다. 이미지를 생성하는 소프트웨어를 페인팅도구(painting tool), 이미지 편집기(image editor)라고 한다[1][2]. 과거에는 주로 스캐너를 이용하여 이미지 파일을 생성하였으나 요즘에는 디지털카메라를 이용하여 영상을 바로 이미지 파일형태로 저장하고 있다. 본 논문에서는 그래픽과 이미지를 같은 시각적 요소로 보고, 이미지나 그래픽 데이터를 소프트웨어를 이용하여 생성하는 방법에 대해 기술하였고, 특히 벡터방식과 비트맵 방식을 혼합하여 동시에 편집할 수 있는 이미지 에디터 구현 방법에 대하여 기술한다.

### II. 비트맵과 벡터

벡터 그래픽은 주어진 2차원 또는 3차원 공간

에 선이나 형상을 그리기 위해 일련의 명령어나 수학적 표현을 디지털 이미지를 만든다. 예를 들면, 선을 그리기 위해 벡터그래픽 파일에는 각 비트들이 저장되어 있는 대신에, 연결될 시작점과 끝점의 위치가 들어있다. 그로 인해 파일크기가 작아지는 결과를 가져온다. 그림 1은 사각형과 선을 표현하기 위한 벡터 명령문이다.

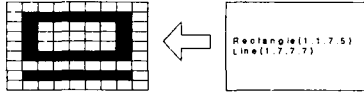


그림 1. 벡터방식의 이미지 표현

비트맵 이미지는 드로잉 유형의 이미지와 달리 픽셀 또는 도트를 이용하여 표현된다. 각 픽셀은 1비트의 색상 정보를 포함한다. 즉, 비트맵 이미지는 정사각형의 픽셀들이 모여서 이미지를 구성하는 방식이다. 픽셀들은 각각 하나의 독립된 형식인 색상을 가지고 있으며 비트맵 이미지를 구성하기 위해 많은 수의 픽셀을 사용하면 이미지를 더욱 섬세하게 표현할 수 있기 때문에 사진이나 회화 이미지에 적합하다. 그러나 이미지의 크기, 해상도, 색상 깊이에 따라 비트맵 파일은 드로잉 이미지보다 훨씬 더 많은 기억장치 공간을 요구하여 용량이 커질 수 있다. 그림 2의 예는 2차원 형태의 배열에서 글이 써진 부분만 '1' 그렇지 않은 부분은 '0' 으로 표현하였다.

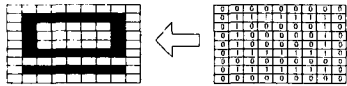


그림 2. 비트맵방식의 이미지표현

벡터 형식의 이미지 표현은 다음과 같은 특징을 가진다.

- 객체들이 다양하고 많을수록 그 영상을 저장, 상영하기 위한 처리 시간이 많이 소요
  - 처리시간과 저장공간은 영상마다 가변적
  - 좌표값을 이용한 각 픽셀을 개별적으로 조작
  - 영상을 축소하거나 확장할 때 영상의 모든 영역의 재계산에 의한 해상도 유지
- 이에 반해 비트맵 형식의 이미지 표현은 다음과 같은 특징이 있다.
- 깊이 있는 색조와 부드러운 질감, 그리고 자연스러운 분위기
  - 사진, 회화 이미지에 적합
  - 축소, 확대하는 경우 이미지 품질이 저하
  - 다양한 질감과 사실적인 효과의 연출 가능
  - 해상도에 따라 이미지 품질이 결정된다
  - 비트맵 이미지를 구성하는 요소는 픽셀

기존 도구들의 주요기능을 알아보면, 비트맵 프로그램의 기능은 이미지 파일(JPEG, GIF, BMP 등)파일을 열어서 각 픽셀단위를 대상으로 편집하는 기능이다. 색상 모드 조정, 색상 정보 변경 및 보정, 효과 주기 등 픽셀의 색상값을 이용하여 편

집하는 방식이다. 벡터프로그램은 주로 다각형이나 텍스트 등 차트나 도형을 그리는 도구로 이루어져 있다. 각 객체마다의 위치, 회전, 정렬등 각 객체의 좌표를 계산함으로써 이미지를 편집하며, 객체의 내부를 채우는 방법과 패턴으로서 색을 지정한다. 즉, 산술식으로 표현할 수 있어 수정 및 편집이 용이하다. 그러나 각각 프로그램은 비트맵과 벡터의 특징을 살려 편집하도록 되어 있어 이미지 데이터를 비트맵과 벡터 방식을 같이 사용하여 편집하려 할 때는 각각의 프로그램을 따로 이용해야 한다는 불편한 점이 있다. 본 논문에서는 이를 착안하여 비트맵과 벡터를 생성, 편집, 저장 이 가능한 이미지 편집 도구 구현에 관하여 설계 방법과 구현방법에 대해 기술하였다.

### III. 편집도구 설계 및 구현

벡터이미지와 비트맵 이미지를 혼합하여 동시에 편집할 수 있도록 하기 위해 본 시스템에서는 비트맵과 벡터 요소를 가지는 객체로 구성하였다. 객체는 크게 도형관련 객체, 이미지 관련 객체, 도형 관련 객체, 그룹관련 객체, 텍스트 관련 객체, 브러시 관련 객체로 나뉘었다. 각 객체는 생성 기능, 출력 기능, 편집기능, 효과기능을 묶어 객체화하였다.

#### 1. 시스템 구조

비트맵과 벡터의 요소를 가지는 객체는 크게 벡터 객체로는 사각형, 다각형, 원형, 둥근 사각형, 선, 텍스트, 그룹 객체로 이루어져 있고 비트맵 객체로는 이미지, 브러시로 구분할 수 있다. 각 객체는 생성 과정을 통하여 생성되면 객체리스트에 등록하여 하나의 작업 영역에서 동시에 편집함으로써 비트맵과 벡터 이미지를 혼합화하였다.

전체적인 시스템 흐름도는 그림 3과 같이 구성되었다. 먼저 기존 데이터 파일이나 새 파일을 열거나 생성하여 편집 과정을 거친다. 편집과정은 데이터 객체 생성, 객체 그리기, 객체 이동, 객체 편집, 객체 출력, 객체 소멸, 데이터 파일 저장과정으로 이루어진다. 객체 편집에는 속성 지정, 크기수정, 회전, 효과, 그리고 이미지 및 텍스트 삽입 과정이 있다.

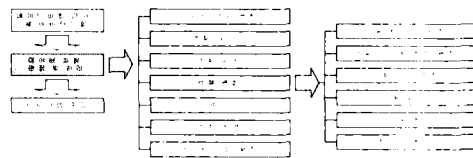


그림 3. 전체적인 시스템 흐름도

#### 2. 객체 정보

각 객체의 특성은 객체의 용도와 필요한 모든 속성들을 객체별로 구분하여 표 1과 같이 구성하

였다[3][4]. 속성은 객체 종류와 객체를 표현하는데 필요한 정보로 구성되어 있다[5]. 각 객체는 속성을 값을 통해 객체를 화면에 출력하거나, 속성값을 변경함으로써 편집을 할 수 있다.

표 1. 객체의 종류와 속성

객체	속성	
벡터객체	선	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 화살표 정보, 선 굵기, 선 색상
	원	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 채우기 유부/색상/패턴
	사각형	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 채우기 유부/색상/패턴
	둥근 사각형	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 채우기 유부/색상/패턴, 모서리 폭/높이
	다각형	객체 종류, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 채우기 유부/색상/패턴, 꼭지점의 수, 꼭지점 좌표
	그림	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 그림내 객체 포인터
비트맵객체	이미지	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 투명영역
	텍스트	객체 종류, 시작점 좌표, 끝점 좌표, 선 그리기 유/무, 선 굵기, 선 색상, 이미지, 텍스트, 낱말, 낱말정보, 채우기 유부/색상/패턴
	브러시	객체 종류, 메모리DC, 브러시 종류, 색상, 패턴

각 객체에 대한 정보를 가지고 시스템을 구현하기 위하여 클래스를 설계하여 보면, 우선 모든 객체의 공통된 내용인, 객체 종류, 선에 관한 정보, 브러시에 관한 정보를 가진 객체의 베이스 클래스인 CAreaObject Class를 두었고, 선, 원, 사각형, 둥근 사각형의 객체는 그림 4와 같이 사각좌표를 기준으로 사각좌표의 왼쪽 위와 오른쪽 아래의 값을 활용하여 하나의 데이터 클래스 CRectAreaObject Class로 표현이 가능하며, 그 외 다각형, 이미지, 텍스트, 브러시는 따로 두었다. 물론 이미지, 텍스트, 브러시는 기존 원, 사각형, 둥근 사각형객체에 포함 가능하다.



그림 4. CRectAreaObject Class

그림 5는 시스템의 객체 클래스 계층도이다.

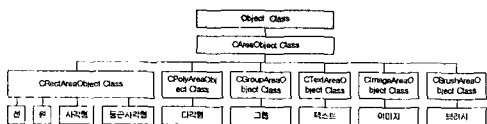


그림 5. 시스템 객체 클래스 계층도

설계과정에서 객체 클래스의 계층도와 각 클래스의 기능을 보면 그림 6과 같다. 각 클래스는 표 1에서 열거한 속성값을 이용한 객체함수를 정의하였다.

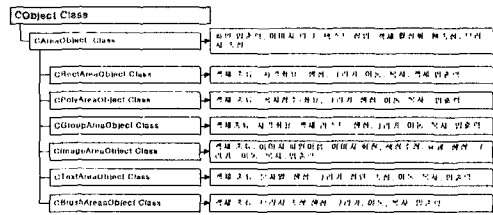


그림 6. 객체의 클래스와 기능

### 3. 파일정보

시스템에서 작업할 수 있는 파일 형태를 보면, 비트맵 이미지인 BMP, GIF, JPEG, TIFF등의 포맷의 문서를 열어 편집하거나 상호 포맷을 변환하여 저장 가능하도록 하였으며, 벡터 형식으로 저장 가능하다. 이를 위해 각 객체를 리스트로 구현하는데, 여러 개의 리스트를 가짐으로써 레이어(Layer)기능도 구현 가능하게 하였다. 또한 리스트 정보를 가지는 새로운 데이터 파일 포맷을 생성하였다. 각 객체의 정보를 리스트로 구성하는 방법은 처음 새 문서를 생성하면 아무 내용이 없는 빈 리스트를 생성한 다음, 도구를 이용하여 각 객체를 생성하면 순서대로 리스트에 추가하게 된다. 이 리스트의 정보를 이용하여, 화면에 보여주는 순서를 지정할 수 있다. 그리고 작업영역을 비트맵 형식의 이미지 파일로도 저장이 가능하도록 하였다.

### 4. 객체 편집

각 객체의 편집은 크게 생성, 찾기, 이동, 복제, 출력, 소멸, 그룹화, 그룹해제, 효과 등으로 구분하였다.

객체 생성은 객체의 종류에 따라 조금씩 다르다. 일반적인 과정은 객체종류설정→표시영역설정→크기조정→화면출력이다. 여기서 다각형 객체는 각 꼭지점 좌표를 추가하게 되고, 이미지 객체의 경우는 삽입할 이미지를 선택하며, 텍스트 객체는 내용을 입력하게 된다.

객체 찾기는 리스트에 하나의 객체를 선택하는 것으로 객체의 속성을 변경하거나, 커서 변경을 목적으로 한다. 클릭된 마우스 좌표를 얻어 객체 리스트의 마지막 객체부터 좌표검사를 하여 처음으로 객체 내부에 위치한 객체를 선택하게 된다. 또한 크기 조절점인 트래커에 위치하게 되면 커서 모양의 그 위치에 맞게 변경하여 크기조정 및 이동을 할 수 있도록 한다.

객체 이동은 선택된 객체에 대하여 표시영역 좌표를 변경하는 경우와 크기 조정하는 경우로 나눌 수 있다. 키보드를 이용하여 이동이 가능하도록 설계하였다. 또한 좌표의 변환을 이용하여 회전 및 대칭 이동을 할 수 있다[6].

객체 복제는 똑같은 종류의 객체와 같은 속성을 지닌 복사본의 객체를 만드는 작업이다. 복제하여 삽입된 객체는 리스트의 맨 뒤에 추가된다.

객체 출력은 생성된 객체를 출력하는 것으로서 이동중이거나 그룹화 등 다양한 환경에 맞게 달리 표현하도록 하였다. 출력 역시 화면이 아닌 프린터와 파일로도 가능하다.

객체 소멸은 선택된 객체를 삭제한다. 이때 리스트에서 삭제하고 객체 리스트를 다시 연결해주면 인접한 객체를 활성화 해준다.

객체 그룹화는 개개의 객체를 하나로 묶어 편집을 쉽게 하도록 한다. 그룹화의 과정은 그림 7 처럼 객체를 다중 선택한 후 선택된 그룹의 새로운 좌표를 구하여 이를 표시해 준다[2].

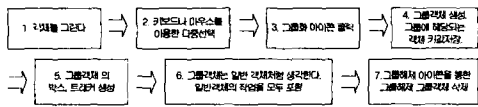


그림 7. 객체 그룹화 과정

객체 편집 과정은 주로 마우스 이벤트에 따라서 발생하게 된다. 처리 흐름도는 다음과 같다.

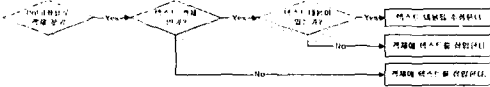


그림 8. 마우스 더블클릭시 처리 흐름도

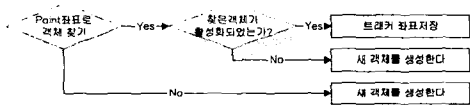


그림 9. 마우스 버튼을 눌렀을 때의 처리 흐름도

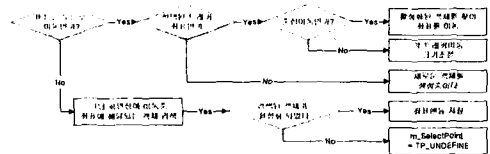


그림 10. 마우스 이동시 처리 흐름도

5. 객체 효과

이미지 객체에 대해서는 비트맵 이미지 효과를

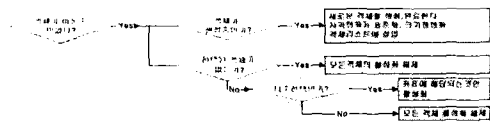


그림 11. 마우스 버튼을 떼었을 때의 처리 흐름도

적용할 수 있다. 픽셀정보를 이용하여 계산하여, 명암조정, 채도 조정 등 색상 조정과 외곽선 추출, 색상 반전, 엠보싱효과 등으로 이루어진다.

6. 인터페이스

그래픽 편집기는 시각적 요소가 강하므로 화면 구성 및 도구의 구성을 중요시하며, 크게 메뉴, 도구바, 작업영역, 상황표시줄로 나누어진다. 도구바는 기본도구, 그리기 도구, 회전도구, 이미지효과 도구, 텍스트 도구, 정렬도구로, 작업 영역은 일정한 문서 크기영역에 작업하며 확대/축소, 안내선, 격자, 눈금자 등으로, 상황 표시줄에 객체 및 좌표에 대한 정보를 표시로 구성한다. 위의 설계를 바탕으로 구현한 시스템 실행 화면이다.

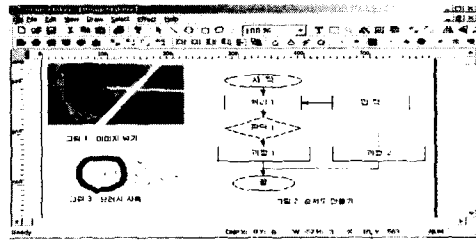


그림 12. 시스템 실행화면

IV. 결론

본 논문에서는 비트맵과 벡터형식의 이미지를 혼합하여 편집할 수 있는 편집 도구의 구현을 위한 시스템 설계방법과 과정에 대하여 기술하였다. 제안된 이미지 편집기는 윈도우환경에서 Visual C++ 6.0을 사용하여 개발되었으며, 향후 개발된 이미지 편집기의 성능을 향상시키기 위해서는 지원 포맷의 확대, 비트맵 이미지에 대한 편집 도구의 추가 기능이 필요하다

참고문헌

[1] Gregory A. Baxes, "Digital Image Processing : Principles and Applications", 2000  
 [2] Narendra Ahua, B. J. SCHACHTER "Image Models", Computing Surveys, Vol. 13, No. 4, December 1981  
 [3] 정정복, "애플리케이션 개발", 열린기술, 1998  
 [4] Erik Wlstrand "Creating Image Context using ImageTrees, CHI 96 APRIL, 13-18, 1996  
 [5] Manuel M. Oliveira and Gary Bishop, "Image Based Objects", 1993  
 [6] Christopher Dean, "Bitmap Image Transformations", ACM journals, Dec, 1993