

---

# 비트맵과 벡터방식을 혼합한 새로운 이미지 편집기

김진호\* · 이규남\* · 나인호\*

New Image Editor based on Combination of Bitmap and Vector Method

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

## 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 하는 문제점이 있다. 본 논문에서는 비트맵과 벡터 방식을 혼합하여 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 design and implementation of image editing tool based on combining of bitmap and vector image.

## 키워드

비트맵 방식, 벡터 방식, 이미지프로세싱, 이미지 편집도구

## 1. 서론

최근 인터넷과 멀티미디어 저작도구의 보급의 보편화 및 이미지 저작 툴, 디지털 카메라와 스캐너 보급으로 이미지데이터를 쉽게 얻을 수 있다. 이러한 이미지는 멀티미디어 매체 중에서 텍스트와 마찬가지로 시각

적인 요소를 표현하는데, 이것은 텍스트를 이용하여 어떠한 내용을 서술하는 것보다 이미지나 사진, 그래프, 차트로 나타내면 사용자에게 그 뜻을 전달할 수 있기 때문이다. 이러한 이점 때문에 이미지의 사용이 증

가하고 있으며, 이미지 데이터를 원본 그대로 사용하기보다는 사용자가 목적이나 사용할 곳에 맞게 가공을 하게 된다. 이때 사용하는 가공 방법으로는 색상의 명암이나 대비 같은 색상 값을 변화시키는 색상 보정, 이미지 파일 크기나 가로 세로 폭을 조절하는 크기조정과 회전, 대칭 이동과 같은 변환과 이미지 필터와 같은 효과 등의 이미지 조작으로 나누어지며, 이러한 조작을 위해 이미지 편집 도구를 이용한다<sup>[1][2]</sup>. 현재 많은 사용이 활용하고 있는 이미지 편집기에는 포토샵, 일러스트레이터, 비저오와 같은 프로그램이 있다.

현재 나와 있는 이미지 편집도구는 표현 방식에 따라서 비트맵 방식과 벡터 방식의 이미지 데이터를 생성, 편집, 저장하고자 할 때 각각의 표현 형식을 중심으로 작업이 이루어지고 있다. 또한 다른 표현 형식을 지원한다 하더라도 그 편집 지원이 극히 일부분이다. 따라서 기존 편집 도구는 하나의 벡터와 비트맵의 두 방식으로 이루어진 이미지 데이터를 제작 편집하고자 하는 경우 서로 다른 응용프로그램을 사용해야만 한다.

이에 본 논문은 비트맵과 벡터 방식의 혼합을 기반으로 하여 하나의 편집 도구를 이용하여 두 방식을 동시에 생성, 편집할 수 있는 이미지 편집 도구에 대한 설계와 구현을 목적으로 하였다.

## II. 관련 연구

### 2.1 영상의 분류

그림, 그래프와 같은 정지영상뿐만 아니라 애니메이션, 비디오와 같은 동영상까지 영상 데이터라고 한다.

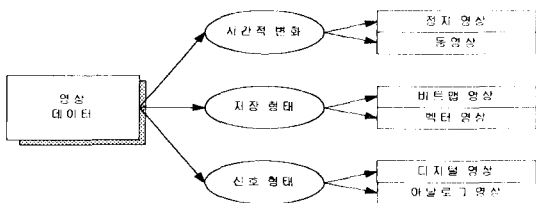


그림 1. 이미지 데이터의 분류  
Fig. 1 Classification of image

이러한 영상 데이터들은 여러 기준에 의하여 구분되어지는데 첫째, 시간상으로 정지영상과 동영상으로

구분되어지며, 둘째, 신호 형태에 따라서 디지털 영상과 아날로그 영상으로 나누어지며, 마지막으로, 표현 방식에 따라서 비트맵 방식과 벡터 방식으로 나누어진다. 다음의 그림 1과 같이 분류 할 수 있다.

### 2.2 비트맵과 벡터이미지

벡터 그래픽은 주어진 2차원 또는 3차원 공간에 선이나 형상을 그리기 위해 일련의 명령어나 수학적 표현으로 디지털 이미지를 만든다. 예를 들면, 선을 그리기 위해 벡터그래픽 파일에서는 연결될 시작점과 끝점의 위치가 들어있다. 그림 2는 사각형과 선을 표현하기 위한 벡터 명령문이다.

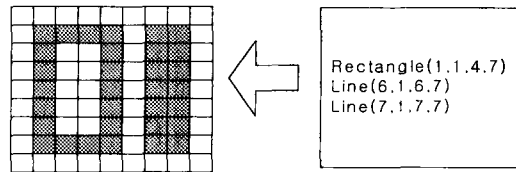


그림 2. 벡터방식의 이미지 표현  
Fig. 2 Vector image

비트맵 이미지는 벡터 유형의 이미지와 달리 픽셀 또는 도트를 이용하여 표현된다. 각 픽셀은 1비트 이상의 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 3의 예는 픽셀의 비트수가 1비트의 경우 2차원 형태의 배열에서 글이 써진 부분만 '1' 그렇지 않은 부분은 '0'으로 표현하였다.

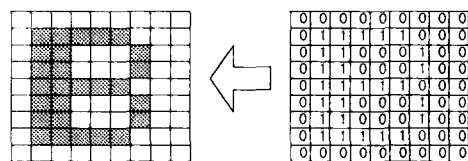


그림 3. 비트맵방식의 이미지표현  
Fig. 3 Bitmap image

### 3.3 이미지프로세싱

영상처리(Image Processing)라는 표현은 일반 대중들도 친숙하게 느끼는 일상용어로 널리 사용되고 있다. 영상처리는 산업현장, 연구실 등에서 수행하는 각종의 작업에서는 물론 개인용 컴퓨터의 그래픽 프로그램이나 워드프로세서 등을 통해서도 자주 접해온 용어이다. 일반적인 의미에서 영상처리는 2차원 시각영상을 의미하는 그림정보를 조작, 해석하는 것으로 언급되고 있으나, 엄밀하게 말하면 영상처리는 영상을 향상시키고 보정하며 해석하고 변환하는 처리라고 할 수 있다<sup>[1]</sup>.

영상처리의 근간이 되는 네 가지의 종류는 영상 개선 및 복원, 해석, 압축, 합성이 있는데 본 논문에서는 이미지의 색상을 이용한 영상개선, 복원 그리고 해석 등 영상처리 관련 기술을 적용하였다.

### 3.4 기존 편집도구

기존 도구들의 주요기능을 알아보면, 비트맵형식의 프로그램은 이미지 파일(JPEG, GIF, BMP 등)을 열어서 각 픽셀단위를 대상으로 편집하는 기능이다. 색상 모드 조정, 색상 변경 및 보정, 효과 등 픽셀의 색상을 이용하여 편집하는 방식이다. 벡터프로그램은 주로 다각형, 텍스트, 차트, 도형을 그리는 도구로 이루어져 있다. 객체마다의 위치, 회전, 정렬 등 각 객체의 좌표를 계산함으로써 이미지를 편집하며, 객체의 내부를 채우는 방법과 패턴으로서 색을 지정한다. 즉, 산술식으로 표현할 수 있어 수정 및 편집이 용이하다.

트, 그룹 객체와 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 4와 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.

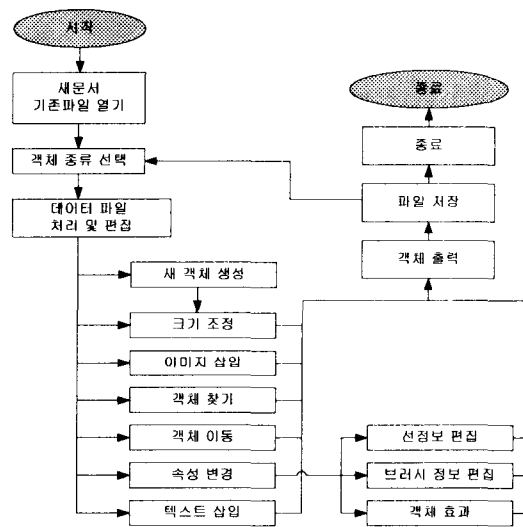


그림 4. 시스템 흐름도  
Fig. 4 System flowchart

## II. 시스템 설계

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

### 3.1 시스템 구조

비트맵과 벡터의 요소를 가지는 객체는 먼저 벡터 객체로는 사각형, 다각형, 원형, 둥근사각형, 선, 텍스

### 2.2 객체 정보

각 객체의 특성은 객체의 용도와 필요한 모든 속성들을 객체별로 구분하여 표 1과 같이 구성하였다<sup>[3][4]</sup>. 속성은 객체 종류와 객체를 표현하는데 필요한 정보로 구성되어 있다<sup>[5]</sup>. 각 객체는 속성값을 통해 객체를 화면에 출력하거나, 변경함으로써 편집을 할 수 있다. 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 CBase Class를 두었고, 선, 원, 사각형, 둥근사각형의 객체는 사각좌표를 기준으로 사각좌표의 원

쪽 위와 오른쪽 아래의 값을 활용하여 하나의 데이터 클래스 CRect Class로 표현이 가능하며, 그 외 다각형, 이미지, 텍스트, 브러시는 따로 두었다. 물론 이미지, 텍스트, 브러시는 기존 원, 사각형, 둥근 사각형객체에 포함 가능하다. 그림 5는 시스템의 객체 클래스 계층도이다.

표 1. 객체 종류와 속성  
Table. 1 Object type and attribute

객체	속성	
벡터 요소	선	객체종류, 시작점/끝점좌표, 선 색상, 화살표 경향
	사각형	객체종류, 사각형의좌표, 선 색상, 채우기 색상, 텍스트, 이미지
	둥근사각형	객체종류, 사각형의좌표, 선 색상, 채우기 색상, 텍스트, 이미지, 모서리 크기
	원	객체종류, 사각형의좌표, 선 색상, 채우기 색상, 텍스트, 이미지
	다각형	객체종류, 선 색상, 채우기 색상, 텍스트, 이미지, 꼭지점 개수, 꼭지점 좌표
	그림	객체 종류, 사각형의좌표, 그림내 객체 모인단.
비트맵 요소	이미지	종류, 사각형의좌표, 선 색상, 이미지 텍스트, 투명 영역
	브러시	객체 종류, 메모리 DC, 브러시 종류, 색상, 패턴

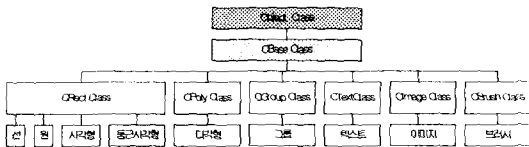


그림 5. 시스템 객체 클래스  
Fig. 5 Object Class

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

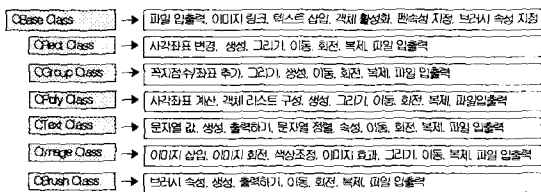


그림 6. 객체 클래스와 함수  
Fig. 6 Object Class and Function

### 2.3 파일정보

시스템에서 작업할 수 있는 파일 형태를 보면, 비트

맵 이미지인 BMP, GIF, JPEG, TIFF등의 포맷의 문서를 열어 편집하거나 상호 포맷을 변환하여 저장 가능하도록 하였으며, 벡터 형식으로도 저장이 가능하다. 이를 위해 작업한 각 객체를 리스트로 구현하는데, 여러 개의 리스트를 가짐으로써 레이어(layer)기능도 구현 가능하게 하였다. 또한 리스트 정보를 가지는 새로운 데이터 파일 포맷을 생성하였다.

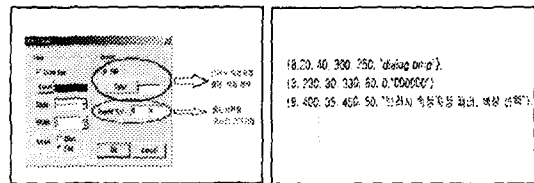
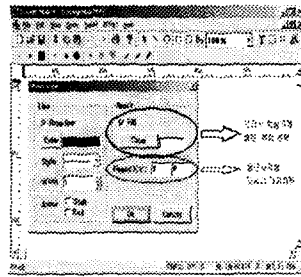


그림 7 작업 내용의 출력  
Fig. 7 Output of image data

각 객체의 정보를 리스트로 구성하는 방법은 처음 새 문서를 생성하면 아무 내용이 없는 빈 리스트를 생성한 다음, 도구를 이용하여 각 객체를 생성하는 순서대로 리스트에 추가하게 된다. 그림 7은 작업한 데이터를 비트맵과 벡터 형식으로 저장한 예이다.

### 2.4 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 리스트에서 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.

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<sup>[6]</sup>.

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 객체를 다중 선택한 후 선택된 그룹의 새로운 좌표를 구하여 이를 표시해 준다<sup>[2]</sup>.

객체 편집 과정은 주로 마우스 이벤트에 따라서 발생하게 된다. 마우스 이벤트는 마우스 오른쪽 버튼을 클릭 했을 때, 왼쪽 버튼을 더블클릭 했을 때, 또한 왼쪽 버튼을 눌렀을 때, 이동할 때 그리고 떼었을 때로 구분할 수 있다.

### 2.5 객체 효과

이미지 객체는 비트맵 이미지 효과를 적용할 수 있다. 픽셀정보를 계산하여 명암조정, 채도 조정 등 색상 조정과 외곽선 추출, 색상 반전, 엠보싱 효과 등으로 이루어진다<sup>[7][8]</sup>.

### 2.6 인터페이스

그래픽 편집기는 크게 메뉴, 도구바, 작업영역, 상황 표시줄로 나누어진다. 도구바는 기본도구, 그리기도구, 회전도구, 이미지효과도구, 텍스트도구, 정렬도구로 나누어진다. 작업 영역은 일정한 문서 크기영역에서 작업하며 확대/축소, 안내선, 격자, 눈금자 등으로 구성

된다. 상황 표시줄에 객체 및 좌표에 대한 정보를 표시로 구성한다.

## N. 구현결과

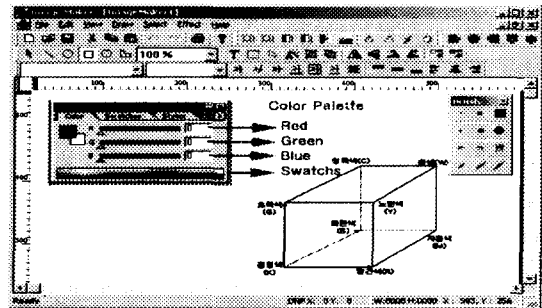


그림 8. 프로그램 실행화면 예  
Fig. 8 A view of the program execution window

본 논문에서 제안한 설계방법에 의해 구현된 편집 도구의 실행화면이다. 비트맵 이미지 및 벡터 객체를 조합하여 작업을 하였다.

## V. 결론

본 논문에서 제안된 이미지 편집기는 윈도우환경에서 Visual C++ 6.0을 사용하여 개발되었으며, 비트맵과 벡터 방식의 이미지를 혼합하여 편집할 수 있는 도구의 설계 및 구현과 다양한 이미지 프로세싱 기능과 객체 단위 편집기능의 효율적 조합을 다루었다. 또한 이미지 처리에 관련된 멀티미디어 콘텐츠 저작분야에서 다양한 활용이 가능할 것으로 기대된다. 향후 개발된 이미지 편집기의 성능을 향상시키기 위해서는 지원 포맷의 다양화, 비트맵 이미지에 대한 편집 도구의 추가 기능이 필요하다.

## 참고 문헌

- [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 1991

[3] 정정복, "에플리케이션 개발," 열린기술, 1998

[4] Erik Wlstrand "Creating Image Context using ImageTrees, CHI 96. 4 L 13-18, 1996

[5] Manuel M. Oliveira and Gary Bishop, "Image Based Objects", 1993

[6] Christopher Dean, "Bitmap Image Transformations", ACM journals, Dec, 1993

[7] Crane, Randy. "A Simplified approach to image processing", Prentice Hall PTR, 1997

[8] Scott E Umbaugh, "Computer Vision and Image Processing", Prentice Hall PTR, 1998

1995년 8월 중앙대학교 대학원 전자계산학과(공학박사)

1995년 9월~현재 군산대학교 전자정보공학부 부교수

1997년 7월~현재 한국 해양정보통신학회 편집 위원

1997년 12월~1998년3월 전자통신연구원 초빙 연구원

1999년 10월~2000년10월 전주국제컴퓨터게임 축제조직위원

2001년 9월~현재 전라북도 과학기술정책 자문위원

2001년 5월~현재 행정자치부 정보화시범마을 기획단 위원

\*관심분야 : 멀티미디어 통신시스템, 분산시스템, 병렬 처리

저 자 소 개



김진호(Jin-Ho Kim)  
 2000년 군산대학교 정보통신공학과 졸업(공학사)  
 2002년 군산대학교 정보통신공학과 (공학석사)

※관심분야 : 영상처리, 멀티미디어 시스템



이규남(Kyu-Nam Lee)  
 1999년 군산대학교 정보통신공학과 졸업(공학사)  
 2001년 군산대학교 정보통신공학과(공학석사)  
 2001. 3~현재 군산대학교 전자정보공학부 박사과정

※관심분야 : 멀티미디어 통신시스템, 멀티미디어 응용 프로그램, 시스템 프로그래밍



나인호(In-Ho Ra)  
 1998년 2월 울산대학교 전자계산학과(공학사)  
 1991년 2월 중앙대학교 대학원 전자계산학과(공학석사)