碩士學位論文

단일영역에서 이미지를 다단계 처리할 수 있는 새로운 레이어구조 및 제어 기법

濟州大學校 大學院

電算統計學科

梁 요 한

2002年 10月

단일영역에서 이미지를 다단계 처리할 수 있는 새로운 레이어구조 및 제어 기법

指導敎授 李 鳳 奎

梁 요 한

이 論文을 理學 碩士學位 論文으로 提出함

2002年 10月

梁요한의 理學 碩士學位 論文을 認准함

審査委員長 _____ (印)

委 員 _____ (印)

委 員 _____ (印)

濟州大學校 大學院

2002年 10月

목 차

<국문초록>

현재 세계적으로 영상처리 응용분야에 대한 연구/개발이 지속적으로 이루어지고 있으며, 기술적인 진보도 급격히 이루어지고 있다. 특히 멀티미디어 정보의 활용이 증가하고 있어 광학과 펜 기반 문자 인식, 멀티미디어 내용 기반 정보 검색과 이해 등에 대한 흥미와 기대감이 증가하고 있는 추세이기 때문에 향후 영상처리 응용 기술 개발은 확대될 것이며, 이에 맞추어 영상처리 응용시스템 개발에 사용하는 SDK(System Development Kits)에 대한 요구도 증가하고 있다. 이런 이유로 국내외적으로 영상처리 전용 SDK에 대한 연구가 활발히 진행되고 있으며, 실제 몇 몇 개발 사례도 보고되고 있다.

그러나 이런 노력에도 불구하고 아직까지 영상처리를 위한 일반화된 SDK(Well Formed SDK)가 개발된 사례는 없는데, 그 이유는 일반화된 SDK에 적합한 내부 자료구조에 대한 연구가 없었기 때문이다. 기존의 영상 처리 툴(편집 기능 위주)과는 달리 영상처리 응용 S/W (SoftWare) 개발에 사용될 SDK의 경우는 하나의 S/W내에서 외부 영상의 획득, 복잡한 영상처리 기법의 반복적 수행 및 인식 실험이 가능해야 한다. 따라서 이전과는 다른 새로운 형태의 다단계 처리를 위한 영상 관리, 처리된 복수개의 결과영상의 제어 및 저장 기술을 필요로 하며, 이를 위해서는 먼저 영상처리의 특성을 잘 반영할 수 있는 특화된 자료구조와 이들 자료구조를 제어하는 방법이 개발되어야 한다. 그러나 편집도구에 단순히 영상처리 과정을 결합하는 방법이나 기존의 SDK로는 다단계 반복 처리에 따른 각각의 결과를 조정하고 제어하는데 한계를 가진다.

본 논문에서는 일반화된 SDK를 만드는데 장애가 되는 현재의 기술적인 문제를 해결하기 위한 연구의 일환으로 1) 영상처리에 관련된 입력/처리 결과를 효과적으로 관리할 수 있는 새로운 레이어 (Layer) 및 레이어 리스트 (Layer-List) 구조와 2)이들 구조를 관리/저장하는 방법을 제안한다. 또한 제안한 방법을 이용하여 실제 SDK를 구현하고 평가해본다.

I. 서론

현재 세계적으로 영상처리(Image Processing) 응용분야에 대한 연구/개발이 지속적으로 이루어지고 있으며, 기술적인 진보도 상당해 이루어지고 있다. 또한 멀티미디어 정보의 활용이 증가하고 있어 광학과 펜 기반 문자 인식, 멀티미디어 내용 기반 정보 검색과 이해 등에 대한 흥미와 기대감이 증가하고 있는 추세이기 때문에 향후 영상처리 응용 기술 개발은 확대될 것이다 (Landgrebe, 1997; 장동혁, 2001). 이런 다양한 응용분야 중에서도 특히 생체인식(Biometrics), 영상자동검색(Contents Based Retrieval) 등은 차세대 선도기술로 중요성이 날로 증가하고 있다(Tan, 1998). 따라서 영상처리 응용 분야 중 하나인 생체인식분야를 통해 영상처리 응용 개발에 적합한 SDK (System Development Kit, 본 논문에서의 SDK는 영상처리에 관련된 개발 툴만을 대상으로 함)의 중요성을 알아볼 수 있다.

생체정보인 지문, 홍채, 손등정맥, 얼굴 등을 이용하여 개인에 대한 보안/인증을 수행하는 생체인식에 관한 연구는 자체의 편리성, 정확성, 유용성으로 인하여 획기적인 기술로 평가받고 있다 (Koehler, 1998; Minte, 1998; Shen, 1997). 이런 추세를 반영하듯, 다양한 생체정보에 대한 인식방법 (Daugman, 1998; Daugman, 1999; Green 등 1996), 이를 활용한 제품 (Daugman, 1999; Wayman, 1999) 등이 국내외적으로 폭넓게 연구/개발이 진행되고 있으며, 많은 결과들이 소개되고 있다. 이런 연구/개발의 추세는 자연스럽게 영상처리를 위한 SDK에 대한 필요성을 증가시키고 이에 최근 국내외에서는 생체인식을 포함한 영상처리 응용 전반에 사용할 수 있는 일반화된 SDK (Well Formed SDK)를 개발하려는 노력을 활발히 전개하고 있다(문지현 등, 2001). 그러나 이런 노력에도 불구하고 아직까지는 일반화된 SDK를 개발한 사례는 없는데, 그 이유는 이런 종류의 SDK인 경우, 기존의 영상 처리 툴(편집 기능 위주)과는 근본적으로 다른 특성이 있기 때문이다.

생체인식을 포함한 모든 영상처리 응용 시스템들은 Figure 1과 같은 공통된 개발 흐름을 가지고 있다(Wayman, 2000). 개발의 예를 생체인식으로 든다면 우선 입력장치(카메라, 생체정보 측정장치 등)를 통해 인식에 사용할 생체영상을 컴퓨터에 저장하고(Figure 1-①) 2) 저장된 영상을 영상 프로세싱 기법을 활용하여 인식에

필요한 특징을 추출한 후(Figure 1-②) 3) 추출된 특징을 데이터베이스 (Data Base, DB)에 저장된 각 개인의 특징들과 비교하여 인식하는 과정(Figure 1-③)을 반복하는 흐름을 가진다. 이런 절차는 보통 인식 대상에 무관하게 공통적으로 적용되며 특히 2)번 단계의 경우 다양한 영상처리 기법으로 최적의 특징을 찾는 것이 생체인식과정에서 핵심적인 요소이다.

이런 2)번 단계에서 처리과정을 거쳐 만들어지는 중간 결과물들은 자신의 이전 단계에서 처리된 결과를 입력으로 받는 등 밀접한 관계를 가지고, 각 단계는 자신에게 적용된 영상 처리기법에 대한 세부정보를 유지해야 한다. 따라서 SDK의 경우는 하나의 S/W내에서 입력 장치로부터의 영상 입력, 단일 영역에서 입력 영상에 대해서 복잡한 영상 처리기법을 반복적으로 여러 번 수행이 가능하도록 다단계 처리를 위한 영상 관리와 처리된 복수개의 결과물에 대한 연관성을 제어/저장하는 기술이 필수이다. 기존의 단순 편집 기능을 위주로 한 영상 툴에서는 다단계의 복잡한 영상 프로세싱을 제공하고 있지 않으며, 다단계 반복 처리에 따른 각각의 결과를 조정하고 제어하는데 한계를 가진다.

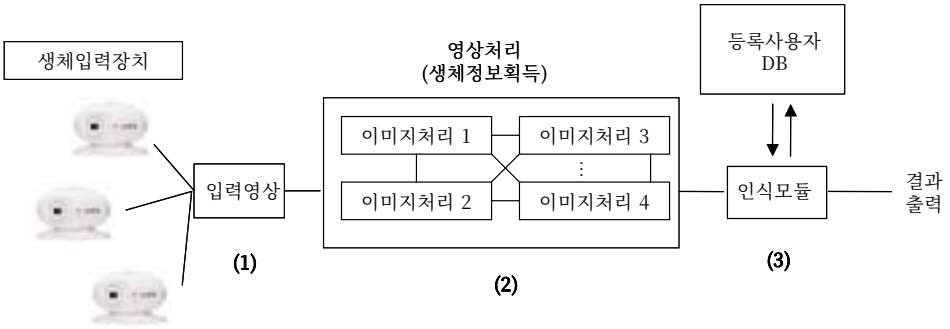


Figure 1. Processing Sequence of Biometrics

이렇게 기존의 SDK에서 문제점이 나타나는 근본적인 이유는 영상처리를 위한 SDK의 구현에 기존의 영상처리 S/W에서 사용했던 내부 자료 구조 및 제어 방식을 그대로 사용하거나 제한된 영역에서 유용한 구현 방법을 이용하기 때문이다. 이런 접근 방법으로는 일반화된 SDK를 구현하는데 한계가 있다. 기존의 SDK 개발 방법을 통해 이들 문제점을 지적해보면 다음과 같다.

현재까지 SDK 개발을 위해 사용되었던 방법은 2가지 정도로 분류해 볼 수 있다. 첫 번째 방법은 기존의 편집위주 S/W에서 사용되었던 내부의 구조나 방법을 사용하여 그 위에 단순히 영상처리 기법만 구현하여 추가하는 방법이다 (Mathwork Inc., 2002). 이 방법을 사용한 개발 사례로 대표적인 것이 Matlab, photoshop(Adobe, 2002)등을 이용한 것이다. 즉 Matlab에서 동작하는 프로그램을 작성하거나(M file 포맷), Photoshop에서 작동하는 플러그인 루틴이 대표적인 예이다. 이런 접근 방법은 범용성을 지닌 검증이 된 S/W를 그대로 사용할 수 있다는 장점이 있기는 하지만 영상처리 응용을 개발하는데 충분한 기능을 제공하는 것에 한계를 가지는 것은 물론 다단계의 영상 처리에는 적합하지 않다는 근본적인 문제를 지니고 있는 방법이다(Figure 2-①). 두 번째 방법으로는 독립적인 별도의 SDK를 개발하는 방법이다. 이 방법을 이용한 대표적인 사례가 HIPS(Mattew 등, 2002), Khoros (Konstantinides 등 1992; Konstantinides 등 1994), XITE (XITE, 2002) 등이다. 이들 구현 SDK들은 라이브러리형태이거나 독립된 플랫폼 등 2가지의 형식으로 나누어 볼 수 있다 (Figure 2).

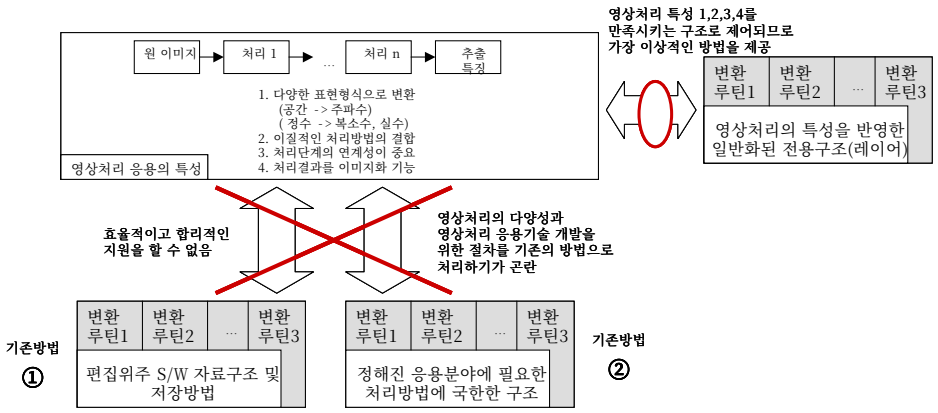


Figure 2. Three Approaches for SDK Implementation

이런 접근 방법의 경우, 첫 번째와는 달리 자체적인 전용 구조를 이용하여 영상을 처리하고 있으며, 목적에 적합한 다수의 복잡한 처리루틴을 제공하는 것이 특징이다. 그러나 이런 방법 역시 영상처리에 적합한 범용의 구조보다는 처리 과정에 중점을 두었기 때문에 영상처리를 위해 필요한 각 처리 요소간의 연관성 제어는 할 수 없으며, 결과영상의 저장방식도 기존의 영상 저장형식을 그대로 사용하기 때문에 처리 결과에 대한 관리가 어려운 것은 사실이다.

이렇듯 기존의 개발 방법으로는 최종목표인 일반화된 SDK를 구성하는데 많은 제약이 따르게 되는데, 그 근본적인 이유는 영상처리 시스템의 특성을 정확히 분석하여 영상처리에 적합한 SDK의 내부 자료구조를 명확히 제시하고 이를 바탕으로 필요한 세부기능들을 구현하는 방법을 사용하지 않았기 때문이다. 즉, 일반화된 SDK를 구현하기 위한 가장 기본적인 연구를 생략하였기 때문에 나타나는 문제점들이다.

본 논문에서는 지금까지 개발되었던 SDK에서 공통적으로 나타나는 문제점을 근본적으로 해결하여 일반화된 SDK를 구현하기 위해, 다단계 영상처리를 효과적으로 지원할 수 있는 내부구조로써 새로운 레이어(Layer) 구조와 이 구조에 대한 저장 및 연관성제어 방법 등을 제안한다. 제안하는 레이어 구조는 특정 영상을 처리한 결과에 대한 모든 정보를 가지는 독립구조로써 생성 순서에 따라서 상호 연결된 리스트(List)를 형성함으로써 다단계로 처리된 모든 처리 결과를 단일 화면에서 관리할 수 있어 자체가 처리 결과간의 연관성을 표현할 수 있다. 또한 현재 처리된 레이어의 결과들은 하나의 파일에 통합 저장하는 방법과 레이어 상호간의 연관성 자동 제어방법도 제안한다. 또한 제안된 방법들을 실제 구현하여 제안된 방법이 기존의 방법에 비해 효과적임을 직접적으로 보인다.

본 논문의 구성은 다음과 같다. II장에서는 영상처리의 기본적인 개념과 응용분야, 그리고 기존의 SDK 구현 사례와 문제점을 기술한다. III장에서는 제안하는 레이어 및 제어 방법을 설명하고 IV장에서는 실제 구현된 일반화된 SDK를 보인다. 그리고 V장에서 결론을 맺는다.

II. 배경과 관련 연구

이 장에서는 본 논문에서 제안하는 방법들을 이해하는데 필요한 영상처리의 기본적인 개념과 응용분야에 대해서 기술한다(1, 2절). 3절에서는 앞에서 기술한 영상처리를 활용하는 응용 시스템 개발에 사용되고 있는 기존의 SDK들의 개발사례와 문제점을 보인다.

1. 영상처리 개요

영상처리의 의미는 응용분야에 따라서 다를 수 있기 때문에 일반적인 정의를 내리기는 어렵다. 예를 들면 컴퓨터비젼(Computer Vision), 패턴인식 (Pattern Recognition)등에서는 영상처리의 의미를 디지털(Digital)화 된 입력 영상에 다양한 변환기법(Transform)을 적용하여 인식/해석 등에 필요한 특징(Feature)을 추출하는데 필요한 영상습득, 저장, 조작, 표현의 전 과정으로 정의한다 (Marr, 1982). 그러나 이런 정의가 유일한 것은 아니며, 컴퓨터 그래픽이나 영상압축/전송 등의 분야에는 “입력영상의 복구나 재구성 과정”으로 표현하는 것이 보다 적절하다 (Bow, 1992). 따라서 영상처리에 대한 정의는 해당 응용분야에 맞게 적합한 것을 선택할 수 있으며, 본 논문에서 제안하는 방법들은 비젼이나 인식분야에 사용되는 SDK에 관련된 것이므로 패턴인식에서의 정의를 따른다. Figure 3은 본 논문에서 정의하는 영상처리의 과정을 나타낸 것이다(Gonzalez, 1987).

처음 영상처리가 대두되게 된 것은 1920년 초반에 신문에 쓰일 사진을 케이블을 통해 전달하게 되면서부터이다. 이런 배경을 바탕으로 영상처리는 초기에 주로 입

력 영상의 향상(Enhancement) 과 복구(Restoration) 등의 제한된 영역에서 사용되었으나, 현재는 그 응용분야가 인식, 비젼 등 지능적 컴퓨터(Machine Intelligence)를 구축하는 모든 영역으로 넓어지고 있다 (Jain, 1989).

컴퓨터에서 대상으로 하는 디지털 영상은 실수 혹은 복소수 등의 숫자로 표현되어지는 값들의 2차원 배열이다. 이런 값들은 컴퓨터가 표현할 수 있는 타입의 제한된 크기의 범위를 가지게 되며 이를 레벨(Level)로 부른다 (Gonzalez, 1987). 이 레벨의 의미는 빛에 대한 감도 즉, 빛의 세기(Intensity)를 의미한다. 이는 간단히 2차원의 빛의 감도에 대한 함수 $f(x, y)$로 표현 할 수 있을 것이다. 여기서 x, y는 영상의 좌표이며, 함수의 출력은 빛의 세기가 될 것이다. 이렇게 디지털 영상은 빛의 세기값을 가지는 점인 픽셀(Pixel, Picture Element)들을 원소로 가지는 2차원 배열로 정의된다 (Figure 4).

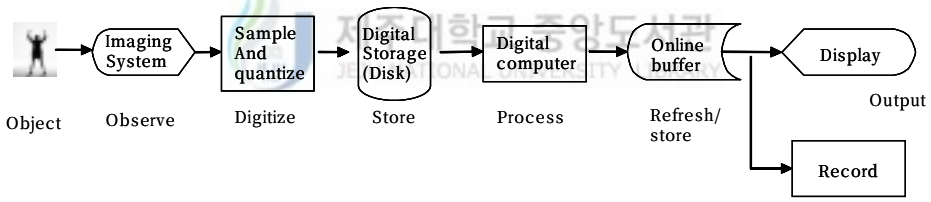


Figure 3. A Typical Image Processing Sequence

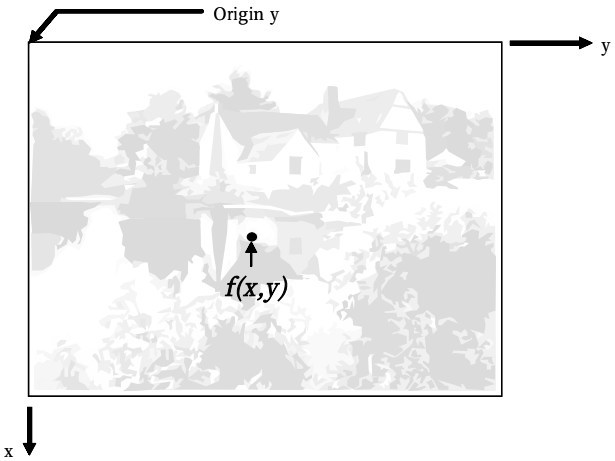


Figure 4. Axis Convention used for Image Representation

효과적인 영상처리를 위한 시스템은 Figure 5와 같이 H/W (HardWare)와 S/W로 구성된다 (Gonzalez, 1987). H/W는 입력장치, 저장장치, 표현장치 및 전용 H/W 모듈(Module)로 구성되는데 이것들의 역할은 주로 실세계의 특정 대상의 영상을 컴퓨터 (Computer) 에서 처리 가능한 디지털 형태로 변환하거나 디지털 영상을 아날로그(Analog)로 변환시키는 역할을 담당한다. S/W는 디지털 영상에 대한 실제적인 처리를 담당하는 부분으로 영상에 대한 변환, 특징추출 등의 역할을 수행한다. 각 구성요소의 세부적인 기능들은 다음과 같다.

1) 영상 처리 프로세서(Image Processor)

영상처리 프로세서는 영상의 획득, 저장, 저 수준의 영상처리 등의 기본적인 기능을 하는 하드웨어 모듈들로 이루어진다. 일반적으로 영상의 획득 모듈은 입력받은 신호를 디지털의 형태로 변환하는 샘플링(Sampling)과 양자화(Quantization)를 수행한다. 처리 모듈은 산술 연산이나 논리 연산과 같은 저 레벨의 함수들을 수행한다.

2) 디지타이저(Digitizer)

획득한 영상을 컴퓨터가 처리하기에 적당한 숫자로 변환한다.

3) 디지털 컴퓨터 (Digital Computer, S/W에 해당)

고수준의 영상처리를 수행하는 프로그램 (Program) 이다. 실제 영상처리의 핵심적인 부분으로 다양한 구성이 가능하다. 본 논문에서의 SDK가 이것에 해당된다.

4) 디스플레이 및 기록 장치(Display and Recording Devices)

모니터 (Monitor)와 같은 디스플레이 (Display) 장치는 컴퓨터에 의해서 처리된 영상을 표시하는 역할을 하며 기록장치의 내용은 슬라이드, 사진 등의 매체로 하드카피(Hard Copy) 될 수 있다.

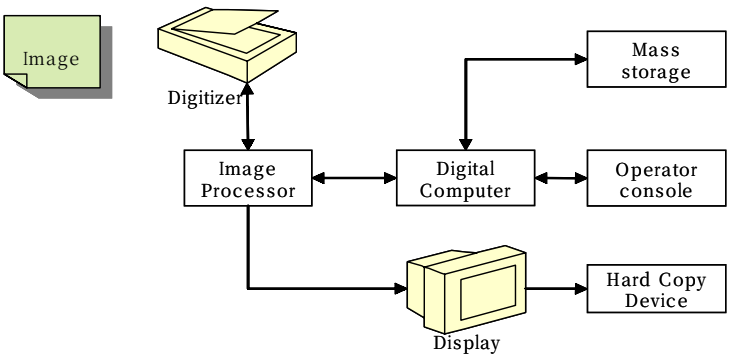


Figure 5. Elements of an Image Processing System

2. 응용분야

영상처리는 여러 영역으로 응용될 수 있는데 대표적인 것은 다음과 같다.

1) 영상인식(Bow, 1992; Dubios, 1981; Gonzalez, 1987; Jain, 1989; Koehler, 1998; Marr, 1982)

사람의 눈으로 영상의 특징을 발견하기란 쉽지가 않다. 영상인식이란 미세한 영상물의 차이점을 발견하고 다른 영상과 비교/분석하여 특징을 찾아 영상을 인식할 수 있도록 하는 영역이다. 예를 들어 사원들의 지문에 대한 데이터가 저장된 데이터베이스의 데이터들과 특정 영역을 출입하려는 사람의 지문을 획득해 이를 비교함으로써 출입의 허가를 결정하는 시스템 등이 이런 것이다.

2) 생물학 분야(Fickett, 1996; Halperin, 1999)

생물학에서는 미세한 세포의 세계를 다루어야 하고 관찰해야 한다. 이를 위해 현미경을 이용하고 세포의 영상 등을 획득하게 되는데 이러한 영상의 인식률을 높이고 영상의 특정 부분을 부각시키는 등의 영상처리가 활용되고 있으며 생물공학분야에서는 DNA (Deoxyribo Nucleic Acid) 및 RNA (Ribo Nucleic Acid)의 분석, 분류, 정합 등의 작업들을 영상처리를 응용해 해내고 있다.

3) 인공위성 분야(Hara, 1994; Munechika 등, 1993, Pan 등, 1992; Salu, 1993)

영상처리의 기술을 발전시키는데 중요한 역할을 하게 된 분야로써, 인공위성으로부터 획득된 영상을 처리하여 지구의 지리정보 혹은 기상정보 등을 얻어 낼 수 있으며, 군사적인 목적으로 적진의 군사 시설물이나 장비 등을 식별 해내고 공격 목표물을 포착하기 위한 작업등에 영상처리가 쓰이고 있다.

4) 문서처리(Sakar 등, 1998)

문서를 보관, 이용하는데 드는 시간과 인력을 줄이기 위해 문서를 디지털화 시키는 과정에서 영상처리가 이용된다. 문서를 컴퓨터가 저장하고 처리할 수 있는 디지

털 형태로 변화하는 과정과 특정 양식에 표기된 문자의 인식 등에 영상처리가 쓰이고 있다.

5) 의료 진단 시스템 (Prasad, 1997; Rao, 1996)

병원 등에서 촬영하는 X선 (X-ray) 사진에서 특정 기관을 부각시키거나 혹은 다른 X선 사진과 비교하거나 하는 등의 작업에 영상처리가 사용된다.

6) 영상의 내용기반 검색 (Bongiovanni, 1993)

데이터베이스 내의 데이터를 검색하는 방법은 문자를 기반으로 하는 방법이 주를 이뤄왔는데 이는 데이터베이스 내의 데이터가 대부분 문자 위주로 구성되기 때문이었으나 근래에 들어 데이터베이스의 저장 능력이 증가되면서 지리정보, 동식물 도감, 의료정보 등의 데이터베이스에 영상을 직접 저장하기에 이르렀다. 이러한 환경에서 데이터베이스내의 데이터를 검색하려면 저장된 영상을 설명하는 문자들을 기반으로 해서 검색을 해야 하는데 영상처리를 응용하게 되면 데이터베이스에 저장된 영상들과 검색하려는 영상들을 직접 비교함으로 해서 더욱 직관적이고 효용성 높은 검색을 할 수 있다.

3. 영상처리를 위한 SDK 개발 사례 및 문제점

2절에서 언급한 응용분야들에서 영상처리를 적절하게 사용하기 위해서는 해당 분야에 특화된 응용 S/W의 개발이 필수적이다. 그러나 응용마다 사용되는 영상처리 기법이 다를 수 있기 때문에 유사한 응용분야일지라도 각기 다른 개발방법을 사용해야 한다. 따라서 이런 번거로움을 해결하기 위해 영상처리 응용 시스템 개발을 위한 일반화된 SDK에 대한 필요성이 증가하고 있는 추세이다. 이런 SDK는 응용분야에 무관하게 모델링 (Modeling)에서부터 테스트(Test)에 이르는 개발의 전 과정을 지원해 줌으로써 개발에 따르는 시간과 노력을 줄일 수 있기 때문이다.

이 절에서는 일반화된 SDK를 지향하여 개발되었지만 미흡한 다른 SDK들에 대한 장/단점을 분석해 봄으로써 본 논문에서 제안하는 방법의 우수성을 보인다.

1) HIPS (Mattew 등, 2002)

HIPS 패키지는 C언어로 기술된 영상처리 전용 라이브러리로 구성되어져 있으며 스프레드시트(Spreadsheet) 형식의 사용자 인터페이스를 제공한다. HIPS가 사용자 인터페이스로 스프레드시트를 사용하는 이유는 우선 많은 사용자들이 스프레드시트 형태에는 이미 친숙해져 있어 쉽게 배울 수 있다는 데 있다. 또한 스프레드시트의 경우 한 화면에 대량의 정보를 표시할 수 있고 시각적인 응답이 빠르다는 것이다. 또 한가지의 중요한 HIPS의 특징은 자체적으로 고유한 영상 저장형식을 제공한다는 것이다. 기존에 있는 그래픽 파일의 저장방법과는 달리 HIPS에서는 영상데이터와 함께 영상처리에 필요한 정보들을 함께 저장하는 방식을 제공하기 때문에 영상처리에서 유용하게 사용할 수 있도록 하였다.

이러한 특성의 HIPS는 1) 일관된 형태를 갖춰 재사용이 가능한 라이브러리로 구성되어 있고 2) 스프레드시트를 통해 시각적인 정보의 화면 출력을 간소화시켰으며 3) 명령어 방식에 익숙한 사용자를 위해 스크립트 형식의 언어를 제공하고 있다는

장점을 보인다.

그러나 사용자 인터페이스인 스프레드시트의 경우 다른 형식으로 변환하는데 어려움이 있고 단계별로 나타나는 중간 결과 영상들의 확인과 수정이 용이하지 않다는 단점이 있다. 또한 스크립트를 이용할 경우 해당 언어를 익혀야 하기 때문에 개발에 장애 요인이 될 수 있다. 영상의 저장방식에 있어서도 영상 처리에 따른 다양한 파라미터 (Parameter)들은 저장하지 않기 때문에 현재의 처리들을 다시 재현하는데는 한계를 가진다.

2) XITE (XITE, 2002)

XITE (X based Image Processing Tools and Environment)는 UNIX의 X (Argiro 등, 1992; Mikes, 1992)를 기반으로 동작하는 C언어로 기술된 SDK(로써, 영상의 화면에 출력해주는 부분과 200여 개의 영상처리를 담당하는 부분으로 구성되어 있다.

XITE에서는 HIPS와 마찬가지로 고유의 영상저장 형식(BIFF형식)을 제공하고 있는데, 이 형식은 현재 처리가 되고 있는 모든 단계의 영상에 대한 정보(영상의 크기, 영상 데이터의 타입 등)를 모두 저장할 수 있는 특징이 있다. 또한 저장된 파일에 있는 영상들에 대해서 부분적인 입출력이 가능하며, 범용으로 사용되는 다양한 그래픽 저장 형식으로 변환도 가능하다.

XITE의 장점은 C로 작성된 소스코드(Source Code)를 제공하기 때문에 필요시 재 컴파일(Compile)을 통해 다양한 운영체제에서 사용 할 수 있다는 것이다. 또한 풍부한 영상처리 기법을 제공하기 때문에 대부분의 응용에서 사용이 될 수 있는 범용성이 있다.

그러나 사용자 인터페이스에 해당되는 영상출력은 X기반하에서만 작동하기 때문에 Windows 플랫폼에서는 사용자가 자체적으로 영상출력 부분을 제작해야 하는 번거로움이 있고, 다른 기존의 SDK와 마찬가지로 하나의 문서(Document)에 하나의 영상만 표시되는 기능만을 구현 할 수 있기 때문에 다단계 영상처리에 따른 수행과정을 쉽게 인식하기가 힘들다는 단점이 있다.

3) Khoros (Konstantinides 등 1992; Konstantinides 등 1994)

Khoros는 데이터흐름(Data-Flow) 모델 (Dyer, 1990; Upson, 1989; Williams, 1992)을 사용하여 영상처리를 위한 S/W개발 환경을 제공하는 새로운 형식의 SDK이다. 즉, 다른 SDK에서 사용되던 영상처리 SDK의 접근방식 (Figure 6-a)을 탈피하여 데이터 흐름에 기반한 시각적 프로그래밍이라는 새로운 접근방식(Figure 6-b)을 통해 개발되었다.

Khoros는 glyph라 불리는 블록(Block) 모양들로 구성된 다이어그램(Diagram)을 통하여 영상처리의 각 단계를 형상화하며 이 glyph들을 선으로 연결시킴으로써 처리 과정들을 연결시킨다. 각각의 glyph는 영상처리의 각 단계를 표현할 뿐 아니라, 각 단계에서 쓰이는 파라미터를 유지하고 있어서 일단 테스트된 일련의 영상처리 단계를 차후에 수정 할 수 있게 한다. 이 glyph는 영상처리의 과정에서 특정 모듈로도 볼 수 있다. Figure 7은 Khoros의 처리과정을 보여주고 있다.

Khoros에서의 중요한 특징 중 하나는 Figure 7에 나타난 처리결과들을 VIFF라는 자체 파일 저장형식을 사용하여 저장하는데, 이 형식은 HIPS의 파일 저장형식과 마찬가지로 모든 결과영상들에 대한 정보를 한꺼번에 저장할 수 있다.

이렇듯 장점이 많은 Khoros이지만, 각 모듈을 거치면서 생성된 결과영상을 실시간으로 보여주지 못한다는 점과 중간에 하나의 과정이 변화(삽입이나 삭제 등의 변경)하게 되면 모든 과정을 다시 처리한다는 것 때문에 다단계 영상처리를 통한 실제 개발과정에 사용하기에는 부족한 점이 있는 것은 사실이다.

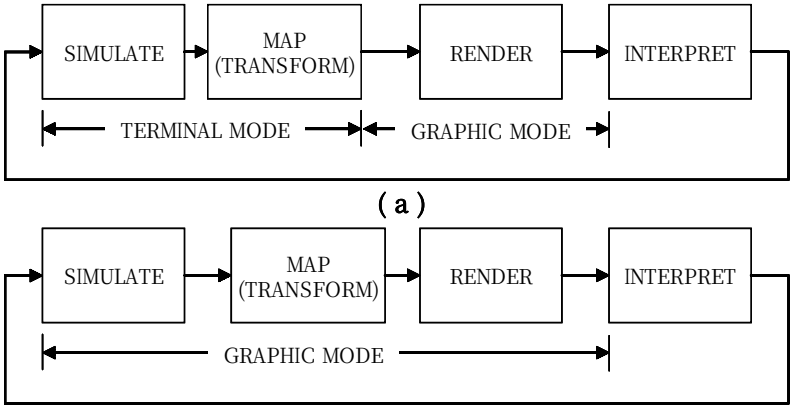


Figure. 6. Iterative Scientific Processing

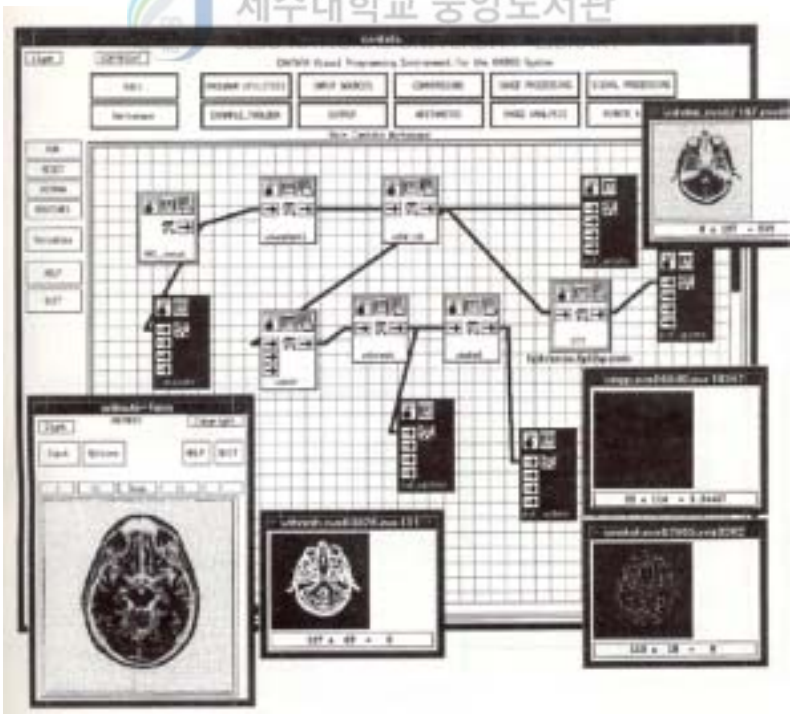


Figure 7. Block Diagram of an Image Processing Application in Khoros

III. 제안 레이어 구조 및 제어 방법

일반화된 SDK개발을 위해 필요한 새로운 내부 자료구조가 갖추어야 할 가장 중요한 특성은 단일 영역에서 영상을 다단계처리 하고 그 결과들은 저장/제어할 수 있어야 한다는 것이다. 지금까지는 한 영역에 있는 영상을 처리하면 다른 영역에 그 결과가 나타나는 방식이므로 다단계 처리를 할 경우 상호 연관성을 파악하기가 어려웠기 때문에 일반적인 SDK의 개발이 어려웠다 (Figure 8). 또한 영상처리의 경우 단순 영상 편집과는 달리 사용되었던 처리 방법이나 관련 파라미터에 대한 내용이 매우 중요한 기능을 가지는 것이 일반적이지만 기존에는 처리 방법에 대한 내용을 직접적으로 제어할 수 없었기 때문에 정확히 개발과정을 재현하는 것이 어렵다. 이에 우리는 이런 문제점을 근본적으로 해결할 수 있는 새로운 레이어 및 레이어 연결구조 제안하게 되었다. 제안하는 방법은 입력영상을 하나의 영역에서 처리가 가능토록 한다. 따라서 상호 연관성의 파악이 용이해지고 이를 바탕으로 기존에는 볼 수 없었던 일반화된 SDK를 구현하는 것이 가능해 지는 것이다.

이 장에서는 새로운 SDK 구현에 필요한 레이어 자료구조와 이들 자료구조의 저장방법 및 제어기술에 관련된 핵심내용을 자료구조 및 흐름도 등을 통해 제안한다.

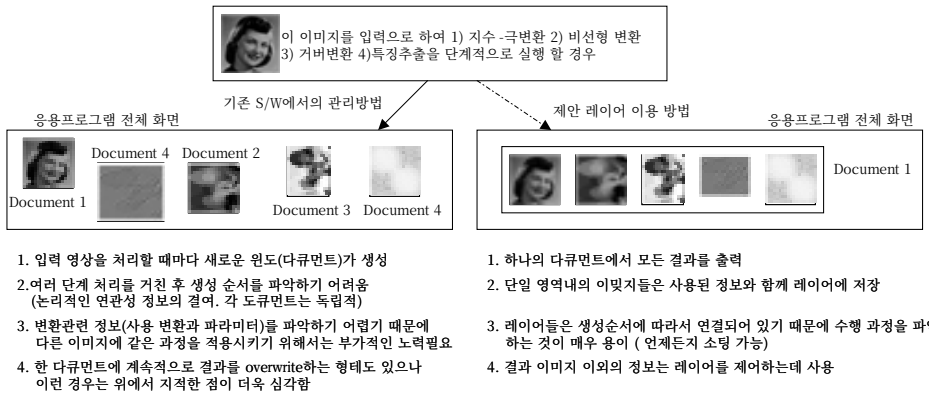


Figure 8. Image processing in a single document

1 레이어

본 논문에서 제안하는 단일 영역에서의 다단계 영상 처리를 위한 레이어는 Figure 9.와 같은 구조를 가진다. 이 구조의 가장 큰 특징은 일반성이다. 특정 응용 시스템개발을 위해 SDK내부에 입력되는 영상이 있을 경우 이 입력 영상은 제안한 레이어 자료구조에 저장된다. 만약 해당 입력 영상이 특정한 처리방법을 통해 처리될 경우 그 결과 역시 동적으로 생성되는 레이어에 저장된다. 이런 단일화된 구조를 통한 영상의 내부 관리기법을 기존의 방법으로 구현하기는 어렵다. 그 이유는 기존의 영상 편집 S/W나 영상처리 S/W에서는 레이어에 대해서 단순히 “한 화면에 겹쳐 싸여있는 아세테이트지” (Adobe, 2002) 라고 정의하는데 이는 한 화면에 배경->영상 1 -> 영상2를 겹쳐 배열하여 복합적인 영상을 만드는 기술로 마치 여러 장의 사진을 겹쳐놓은 것과 같은 형태를 띄도록 하는 것으로 다단계처리와는 무관한 기능이다. 또한 각 레이어는 실제 입력되는 영상의 형태가 제한되어 있어(n비트로 표현되는 그레이(Gray) 값), 입력영상을 다양한 영역(주파수 영역, 복소수 영역 등)으로 변환시켜야 하는 영상처리들에 그대로 적용시킬 경우 처리 결과의 표현이나 저장이 불가능할 수 있다.

이런 문제점을 해결하기 위해 제안하는 레이어 구조는 우선 입력된 영상을 처리할 경우 나타나는 모든 영역에 대해 처리가 가능하도록 일반화된 내부구조를 가지

고 있다. Figure 9에서 보듯이 전체 구조는 3가지의 세부구조(제어, 영상 저장, 인자전달)로 나뉜다. 제어요소는 현재 레이어가 가리키는 영상의 크기(가로, 세로), 채널의 수와 영상 저장형식을 담는 일반적인 필드(Field)와 현재 활성화되었는지(m_bactive) 여부, 화면에 나타낼지 여부(m_bvisible)와 레이어에 적용된 처리와 관련된(처리 후 원래 영상과 크기가 다를 수 있음) m_nX, m_nY, m_fZX, m_fZY 변수로 구성된다. 영상 저장부분은 실제 처리된 영상을 저장하는 영역인데 어떤 영상이 어떤 변환에 사용될지는 사전에 알 수 없기 때문에 현존하는 모든 형식의 영상을 처리한 결과를 모두 받을 수 있도록 실수영역 영상 (Real Part Image), 허수영역 영상 (Imaginary Part Image), 컬러 (Color) 영상, 그레이 영상 저장 영역을 모두 가지고 있다. 인자 전달 요소는 현재 레이어에 적용했던 변환기법의 실제 이름(strArgName)과 사용했던 파라미터의 실제 값(dArgValue)을 모두 가지고 있다. 이 값들은 실제로 이후에 나오는 레이어 자동 계산에 사용되는 중요변수들이다. 이러한 복합적인 구조를 가지고 있는 레이어이기 때문에 처리방법에 따라서 나타나는 결과가 다른 형식을 띄더라도 자료구조나 저장영역의 추가나 변경이 없이 단지 레이어에 모든 정보를 저장하고 관리할 수 있는 것이다.

제안된 레이어는 동적으로 생성되는데 그 이유는 다음과 같다. 입력받은 영상을 여러 가지 변환방법을 통해 처리할 경우 우선 입력 영상을 입력받아 레이어에 저장한 다음, 이 레이어에 존재하고 있는 현재의 영상에 원하는 처리방법을 적용하여 결과를 얻은 후, 이를 새로운 레이어에 적용하는 과정을 반복하는 것이다. 이때 몇 번의 처리를 할 것인지는 개발하고자하는 응용마다 다를 수 있기 때문에, 레이어는 필요할 때마다 동적으로 생성되어야 한다. 즉 새로운 처리결과가 발생하거나 새로운 입력 영상을 읽어 들이는 경우에 맞추어 레이어가 동적으로 생성되는 것이다.

구조명	요 소	각 요소들을 구성하는 필드명	각 필드의 설명
레이어 데이터구조 (Layer Data Structure)	제어요소	m_bactive	제어 변수
		m_bVisible	
		m_ID	레이어 id
		Transform_id	레이어에 담긴 이미지를 얻은 처리방법 id
		m_uBitPerPixel	이미지를 구성하는 픽셀의 비트 수
		m_uChannelCount	이미지의 채널 수
		m_uChannelMode	채널의 형식
		m_uEnableBit	제어 변수
		m_uWidth	이미지 가로 크기
		m_uHeight	이미지 세로 크기
		m_uX	제어 변수
		m_uY	
		m_fZX	
		m_fZY	
		m_pLink	다음에 생성될 레이어에 대한 연결 포인드
		m_strLayerName	현재 레이어의 이름
	이미지 저장요소	m_pData	실제 데이터의 저장 영역 (모든 이미지 변환 기법에 적합한 영역을 모두 포함)
		m_pDataR	
		m_pDataG	
		m_pDataB	
		m_pGrayData	
		m_pData16	
		m_pData16R	
		m_pData16G	
		m_pData16B	
		m_pDataRe	
		m_pDataIm	
		m_pDataReR	
		m_pDataReG	
		m_pDataReB	
		m_pDataImR	
		m_pDataImG	
		m_pDataImB	
		m_pData16Im	
		m_pBitmap	
		m_pBitmapR	
		m_pBitmapG	
		m_pBitmapB	
	인자전달 요소	strArgName;	이미지를 얻은 변환에서 사용한 파리미터 이름
		dArgValue;	해당 파라미 터의 실제 값

Figure 9. Layout of the Proposed Layer

2. 레이어간의 연관성(레이어 연결 리스트)

동적인 생성이외에 제안하는 레이어 구조가 가지는 특성은 생성되는 레이어가 상호 연결된 구조를 가진다는 것이다. 이런 레이어들 사이의 연결리스트 (리스트, Linked List)를 통해 특정 입력영상에 관련된 영상처리 결과물들을 다른 입력영상들에 대한 것과 구별하여 관리함으로써, 단일영역에서 입력 영상을 다단계 처리할 수 있도록 해 준다. 단일 영역에서 한 영상의 다단계 처리결과를 모두 나타내기 위해서는 현재 영역에 나타내어야 할 영상들과 다른 영역에 표시할 영상을 구별하여 관리해야 한다. 영상처리 SDK의 경우 상이한 여러 개의 입력 영상을 동시에 처리하는 경우가 빈번하기 때문에 이런 구별이 불가능할 경우 한 영역에 관련있는 여러 개의 처리결과를 한꺼번에 나타낼 수 없게 된다. 따라서 단일영역에서 영상을 다단계 처리하는 기능을 위해서는 특정 영상이 입력되면 이 입력을 저장한 레이어를 시작으로 처리가 진행 될 때마다 레이어가 동적으로 생성되어 결과를 저장한 후 입력으로 사용한 레이어와 결과를 받는 레이어 사이를 상호 연결시켜 하나의 리스트로 만들어 줌으로써 레이어들을 입력에 따라 그룹화하는 기능은 필수적이다.

Figure 10은 레이어가 상호 연관되어진 리스트의 구조를 보여주는 것으로 처음에 들어오는 입력을 저장한 레이어에 데이터를 이용하여 3단계 처리 후 메모리 내에 구성되는 레이어의 리스트를 보여주고 있다. 시작 영상(레이어 0)은 처음 읽어들여진 영상으로 우선 번호 0을 부여받는다. 이 0번 레이어를 단계적으로 처리하여 새로운 레이어들을 형성하게 되는 것이다. 만약 0번 레이어에 특정 변환을 적용하면 1번 레이어가 새로이 생성되고 이 레이어에 변환 결과를 저장하는 것이다. 다시 1번 레이어에 특정 변환을 적용하면 2번 레이어가 새로이 생성되는 과정을 반복한다. 그런데 이들 생성되는 레이어는 서로 독립적으로 존재하는 것이 아니라, 상호 연결을 하고 있다. 그 이유는 우리가 0번 레이어의 위치만을 기억하고 있으면 전체 레이어를 모두 살펴볼 수 있도록 리스트화 되어야 다단계 처리에 효과적이기 때문이다. 즉 0번 레이어의 영상을 처리하여 1번 레이어를 만든 후에는 0번 레이어의 링크 변수(m_plink)에는 새로 생성된 1번 레이어의 시작 주소가 부여되는 것이다.

이런 과정은 영상이 처리될 때마다 반복된다. 따라서 그림 .과 같이 리스트를 구성하게 되는 것이다. 마지막에 생성된 레이어(그림에서 4번 레이어)의 링크변수 값은 0으로 되어 리스트의 끝임을 알 수 있도록 한다.

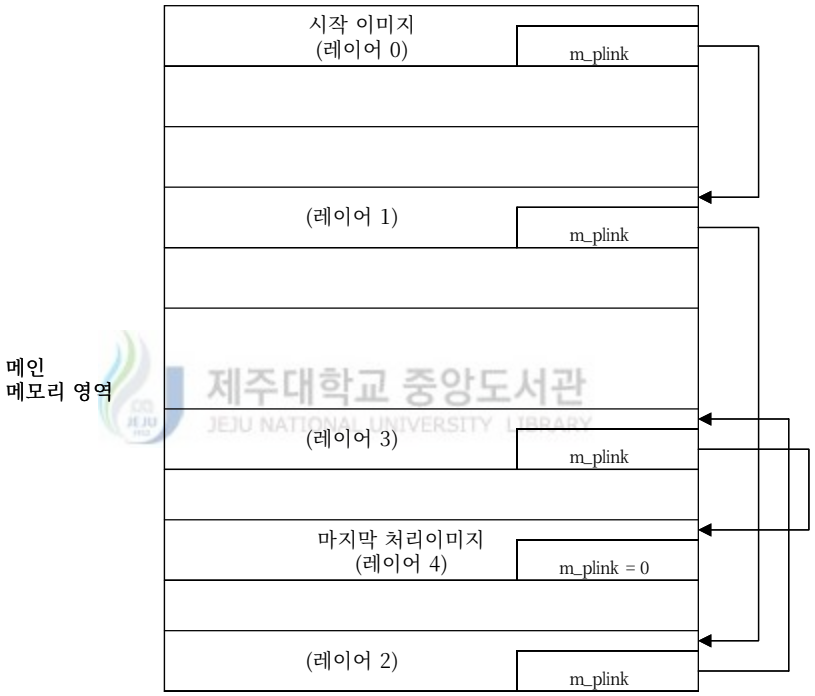


Figure 10. A Linked List with 5 Layers.

3. 레이어 연관성을 이용한 기능

1) 레이어 리스트의 단일 파일 저장

제안된 레이어 및 리스트 구조는 획득영상에서부터 최종 처리까지의 다단계 과정에 대한 처리결과, 처리정보 등을 모두 가지는데 이런 복수 개의 정보를 모두 파일에 기록하기 위해서는 새로운 파일 (File) 형식이 필요하다. 기존의 영상저장 방식인 JPEG, BMP 등의 방법으로는 영상과 텍스트 (Text)가 혼합된 복잡한 레이어 및 리스트를 저장할 수 없다. 제안 기술에서는 이런 문제점을 해결하기 위해 영상과 텍스트가 혼합된 레이어 및 리스트 구조를 단일 파일에 저장할 수 있는 방법을 개발하였다(Figure 11). Figure 12에서는 현재까지 작업한 결과(다단계 영상 처리 결과)를 가지고 있는 리스트의 모든 내용을 단일 파일에 통합 저장하는 프로시저 (Procedure) 인 “save_layer_list”에 대한 흐름도를 보여준다.

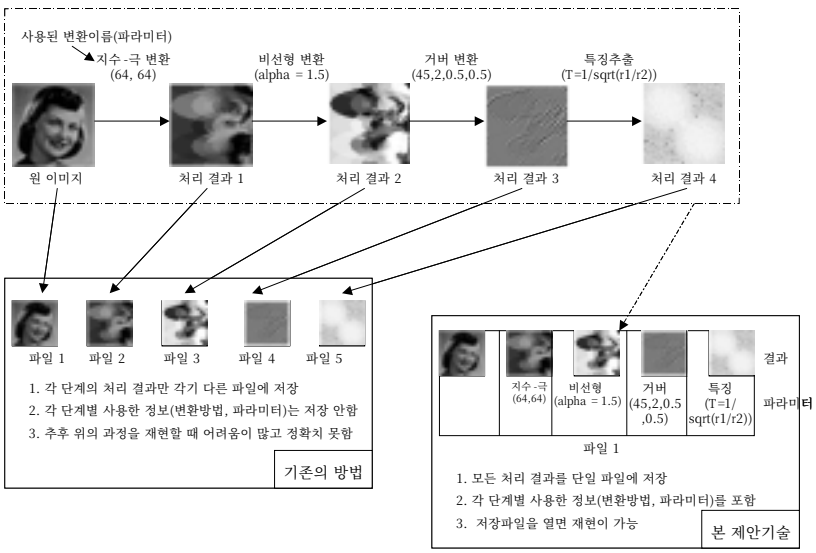


Figure 11. Save Results into a single file

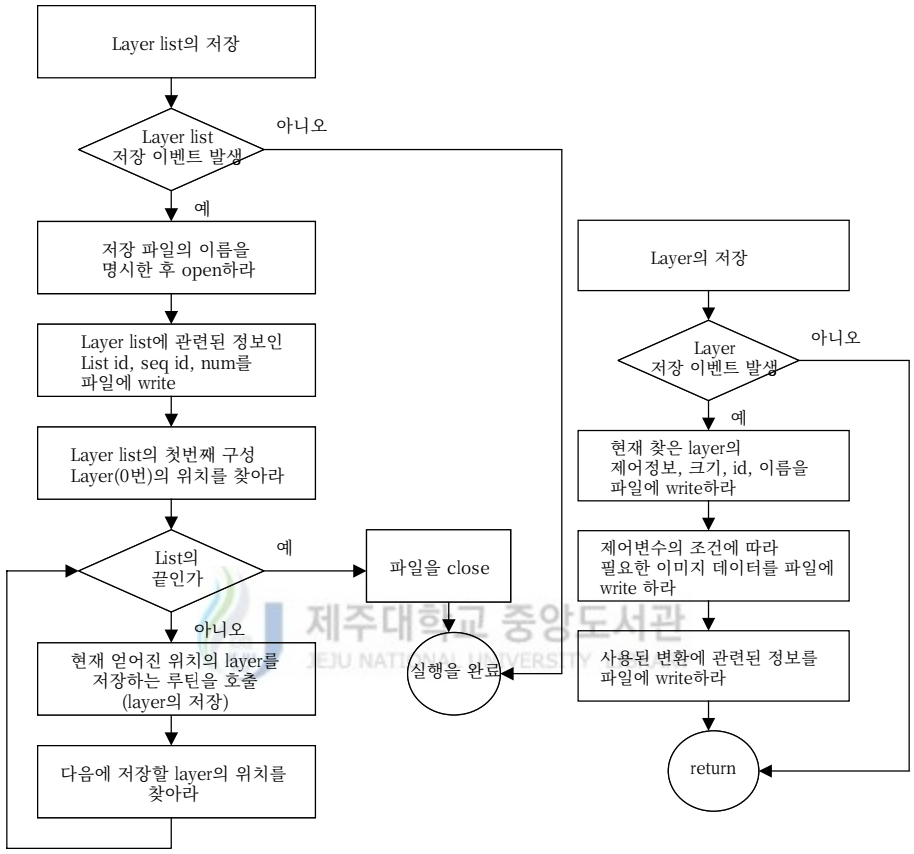


Figure 12. Flow Diagram of the "save_layer_list" Procedure

2) 리스트의 연관성 자동 제어 방법

Figure 13은 현재 작업이 진행되고 있는 리스트에 새로운 레이어가 삽입되거나 현재 있는 레이어 중 하나를 재처리 할 경우 삽입/변경된 레이어에 영향을 받는 다른 레이어들을 자동적으로 재계산하여 올바른 연관성을 유지할 수 있게 하는 방법에 대한 프로시저 (Procedure) 인 “layer_control”에 대한 흐름도를 보여준다.

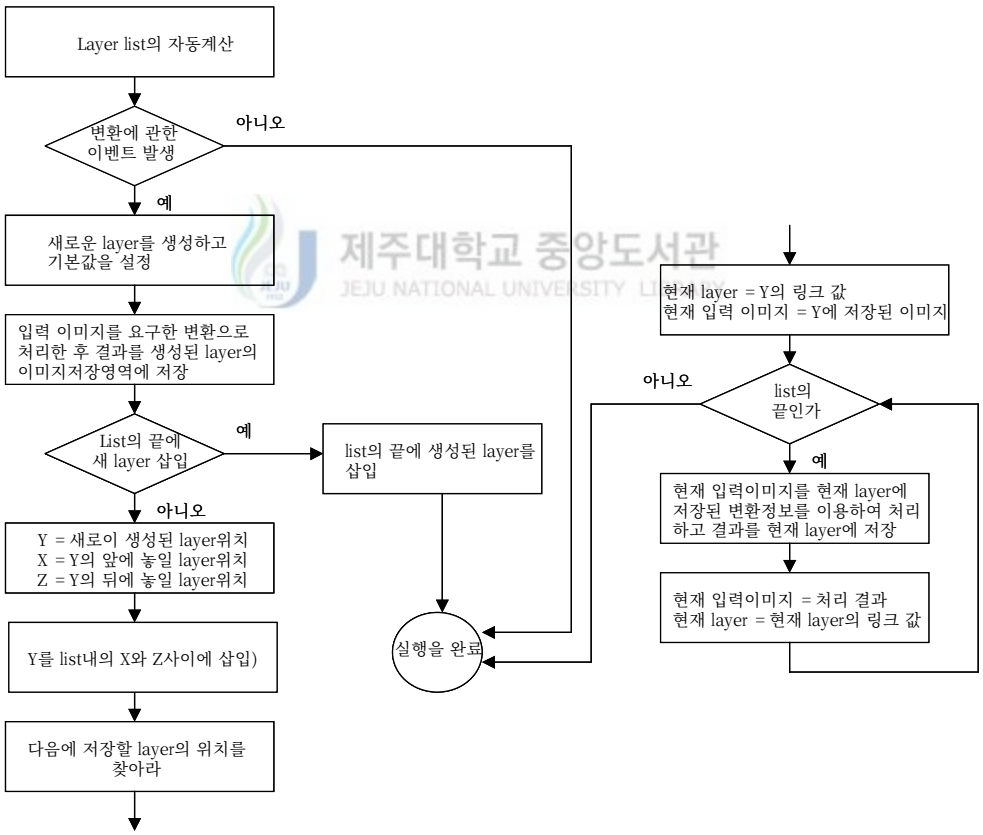


Figure 13. Flow Diagram of the "layer_control" Procedure

3) 영상처리에 대한 매크로 기능 구현

Figure 14는 현재 작업하고 있는 레이어 리스트에 대해서 영상 변환정보만을 추출하여 파일(tsq 확장자)로 저장한 후, 새 영상에 일괄적으로 적용하는 매크로 기능에 대한 프로시저 (Procedure) 인 “image_macro″에 대한 흐름도를 보여준다.

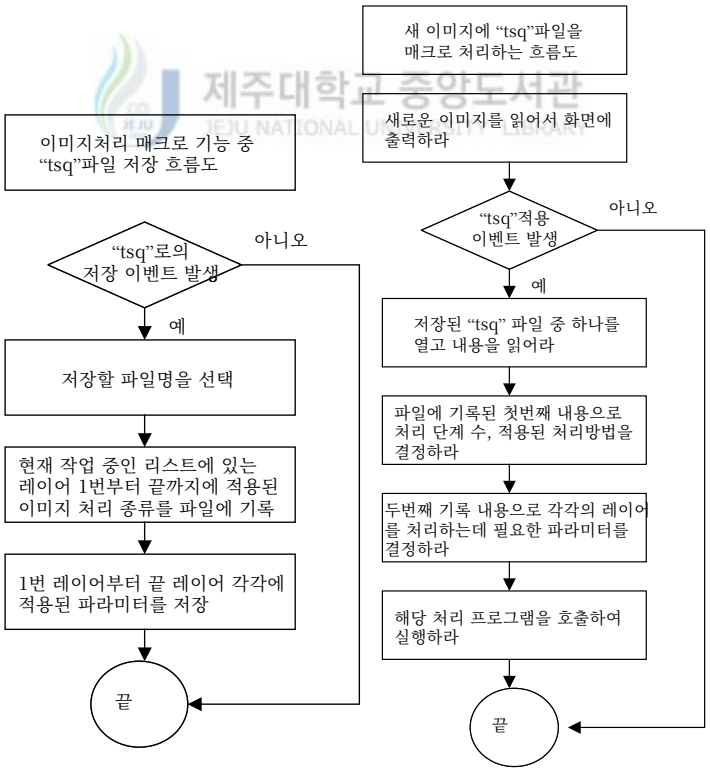


Figure 14. Flow Diagram of the ″image_macro″ Procedure

IV. 구현

본 장에서는 III장에서 제안한 방법들을 실제 구현하고, 이를 바탕으로 제안된 방법이 기존의 문제점을 해결할 수 있음을 보인다. 구현하는 대상은 기존의 단순 편집기능 위주의 영상처리 S/W가 아닌 생체인식, 컴퓨터비젼, 패턴인식 등 응용에 무관하게 시스템(System) 개발에 사용할 수 있는 일반화된 SDK이며 그 내부구조에 대한 개략적인 블록도 (Block Diagram)는 Figure 15에 나타나 있다. 그림에서 보듯이 구현된 프로그램은 제안된 자료구조인 레이어 및 리스트를 기반으로 한다. 이런 기본 자료구조 위에 자동제어 및 입출력 기능을 구현함으로써 전체적인 틀을 구현하였다. 구현된 틀 위에 실제 영상처리를 담당할 영상처리 기법 알고리즘(Algorithm)을 구현하여 실제 영상을 처리 할 수 있도록 하였다. 따라서 구현된 SDK는 영상진단, 생체인식, 로봇공학, 내용기반검색 등 영상처리 응용 시스템 개발에 실제 사용이 가능한 독립된 S/W로 구동이 가능하다.

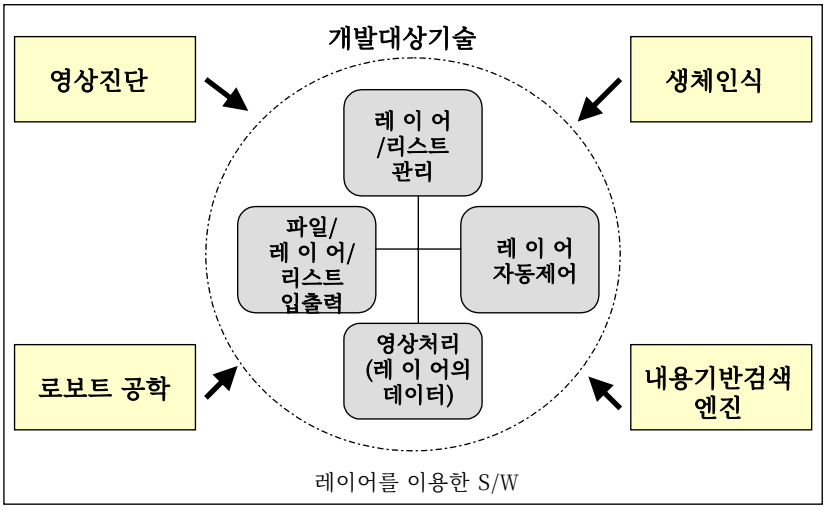


Figure 15. Block Diagram of the Implemented SDK

1. 구현 플랫폼 및 평가 방법

제안한 방법에 대한 구현은 윈도 (Windows) 기반의 PC(Personal Computer)상에서 비주얼 C++ (Visual C++) 6.0을 사용하였다. 이처럼 실제 구현을 위한 플랫폼(Platform)을 윈도로 함으로써, UNIX기반의 W/S를 (Workstation) 기반으로 구현된 대부분의 영상처리 SDK구현에 비하여 여러 가지 장점을 가질 수 있다. 우선 하드웨어적인 측면에서 본다면 PC는 W/S에 비해서 성능향상(계산 속도의 증가) 속도가 빠르기 때문에 대량의 계산을 요구하는 영상처리에 보다 적합하다. 또한 PC의 운영체제인 윈도는 다양하고 강력한 개발도구를 (비주얼 C++ 등) 다수 제공하고 있기 때문에 S/W 개발 측면에서도 PC가 보다 나은 개발조건을 가졌다고 볼 수 있어 윈도기반의 PC를 플랫폼으로 사용하게 되었다. 더욱이 현재 영상처리 응용 시스템을 개발하는 개발자 대다수가 윈도 환경에 익숙해져 있기 때문에 운용적인 면에서도 타당하다. Table 1 은 구현에 대한 세부적인 내용을 정리한 것이다.

구현된 SDK에 대한 성능평가는 구현된 SDK가 관련하여 본 논문에서 제시한 목표를 달성했는지를 알아보는 방법을 이용한다. 구현된 SDK의 경우 현재까지 시도된 적이 없는 새로운 개념을 바탕으로 하고 있고, 기존의 SDK는 대부분 현재 관련 분야의 연구 종사자들이 그 한계점 등을 잘 이해하고 있기 때문에 실제 기존의 방법과의 비교는 큰 의미를 가지지 않는다. 따라서 기존 SDK에서 볼 수 없는 기능을 구현하고 이를 확인하는 상대적인 비교방법을 이용한다. Table 2는 실제 구현된 SDK의 성능평가에 사용된 항목 및 평가방법을 보여준다.

Table 1. Implementation Details

구분	내용
구현언어	비주얼I C++ 6.0, 개발 라이브러리, 기타 윈도상의 개발 툴
개발목표	제안한 레이어 구조 및 제어방법을 윈도환경에서 실현한 응용 S/W
개발용 PC 사양	팬티엄 II 200MHz 이상, 128M 바이트 이상의 기억장치, 스캐너/디지털 카메라를 위한 USB port, 윈도 계열 운영체제

Table 2. Evaluation Methods

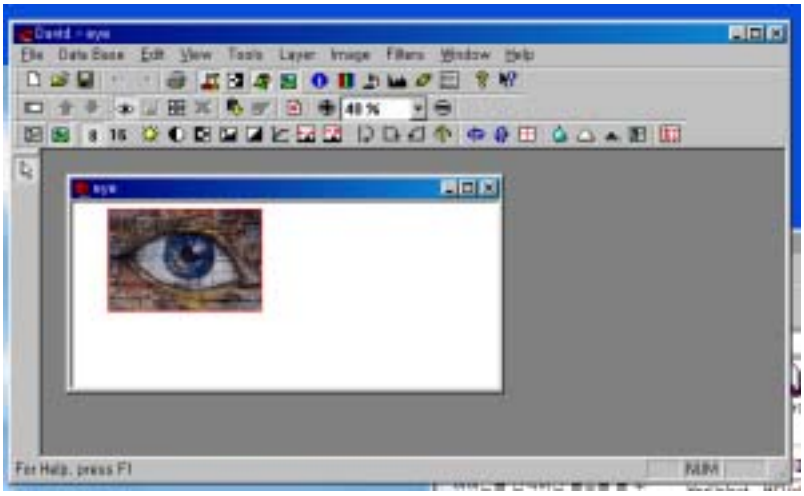
평가항목	평가방법
1. 레이어 및 리스트 구조의 구현	제안하는 레이어 및 리스트를 실제 구현하였는가
2. 단일 영역에서 영상 다단계 처리	한 영역(다큐먼트)에서 영상을 다단계 처리 가능한가
3. 처리결과의 단일파일 저장	다단계 처리결과를 한 파일에 저장하고 제어 가능한가
4. 동적인 레이어의 삽입과 그에 따른 연관성 제어	제안하는 연관성 처리가 동적으로 이루어지는가
5. 영상에 대한 매크로 기능	한 영상에 적용된 처리절차를 그대로 다른 영상에 적용시킬 수 있는가
6. 윈도 환경 변화에 따른 동작 여부	윈도의 종류에 관계없이 동작이 가능한가

2. 구현 및 평가

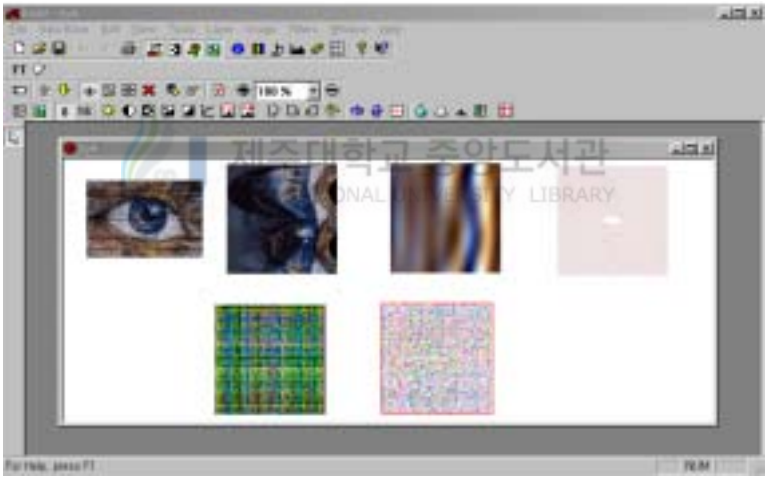
이 장에서는 실제 구현된 SDK에 대해서 Table 2에서의 평가항목을 적용시켜보는 방법을 통해 제안한 방법들의 타당성과 유용성을 보인다. 평가순서는 먼저 구현된 SDK의 인터페이스에 해당되는 전체적인 모양과 다단계 처리의 한 예를 제시함으로써 평가 항목 1,2와 6에 해당되는 내용이 완료되었음을 보인다. 평가항목 3, 4, 5에 대해서는 SDK를 실제 활용한 예를 이용하여 성능을 증명한다.

1) 평가항목 1, 2

Figure 16은 실제 구현된 레이어 및 리스트를 기반으로 한 영상처리 SDK의 초기 화면과 실제 한 영역에서 영상을 단단계 처리하는 과정을 보여준다. 구현된 SDK는 다양한 버전 (Version)의 윈도 (98/ME/2000/XP)에서 모두 동작할 수 있으며, GUI(Graphical User Interface)를 이용하였다. 그림의 (a)는 구현된 SDK를 구동할 경우 나타나는 초기 화면이다. 그림에서 보듯이 입력 영상을 읽어들이게 되면 내부적으로는 새로운 레이어가 자동 생성되어 입력 영상을 저장하고 화면에 출력하게 된다. (b)는 입력영상을 이용한 다단계 영상처리 결과를 보여준다. 입력영상에 대해서 처리할 경우, 각 단계마다 레이어들이 자동으로 생성되면서 현재 단계에서의 영상처리 결과를 저장하게된다. 이런 일련의 과정을 거치면 (b)에서 보이는 것과 같이 다단계 처리된 모든 결과를 화면에서 확인할 수 있게 된다. 여기서 주목할 것은 바로 (b)에서 보듯이 모든 처리 결과 및 입력 영상이 단일 영역에 표시된다는 것이다. 기존의 SDK의 경우 입력 영상이 처리될 경우 새로운 영역(다큐먼트)이 생성되어 그 결과를 저장하고 화면에 표시하는 반면, 제안된 레이어 및 리스트를 이용한 구현 SDK의 경우는 특정 입력 영상에 관련된 모든 처리결과는 한 영역에 표시되기 때문에 개발자의 경우 처리과정의 흐름을 파악하기가 용이하기 때문에 응용 시스템 개발 공정이 단순화되어 개발의 효율성을 높일 수 있는 것이다.



(a)

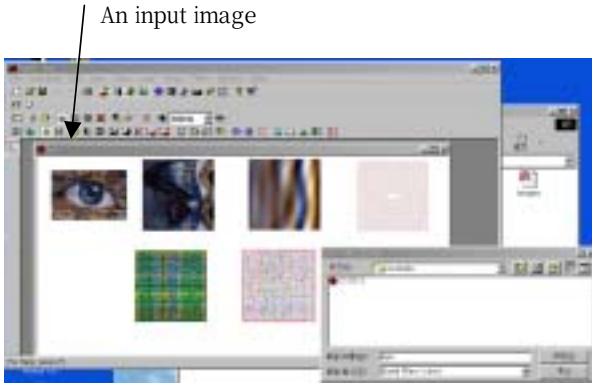


(b)

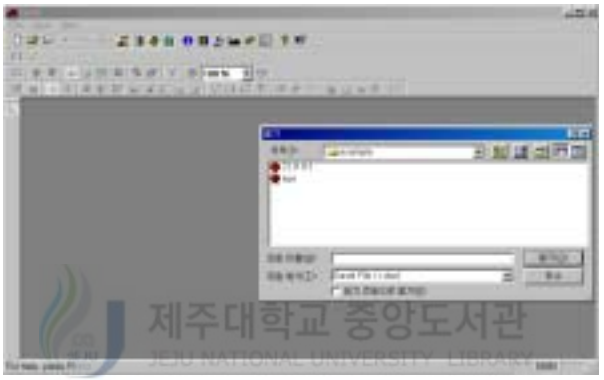
그림 16. Workspace and User Interface of the Implemented SDK

2) 평가 항목 3

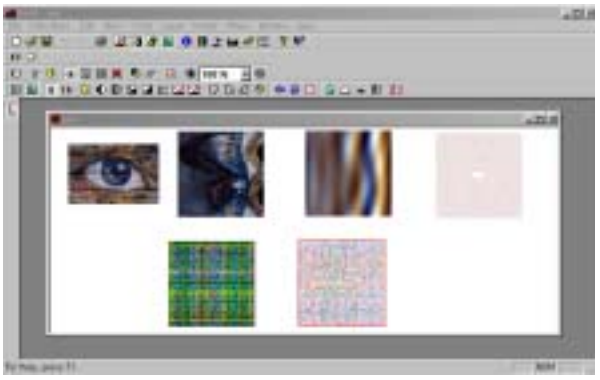
평가항목 3은 단일 영역에서 처리된 다단계 처리결과를 하나의 파일로 저장하는 기능이 가능한지를 알아보는 것이다. 본 논문에서 제안한 방법을 통해 구현된 SDK에서는 Xite (XITE, 2002), Khoros (Konstantinides, 1992; Konstantinides, 1994) 등과 마찬가지로 자체의 영상저장 형식인 “Dav″를 구현함으로써 항목 3의 목표를 달성하였다. 구현된 ”Dav″ 형식은 기존의 영상저장 방법과는 달리 한 영역에 있는 모든 처리 결과들(레이어 리스트 전체)에 대해서 데이터, 영상처리방법, 사용 파리미터, 영상의 특성을 모두 저장한다. 이런 방법이 가능한 것은 구현 SDK의 경우 특정 영역에 있는 모든 결과를 에이어 리스트로 관리하기 때문에 개개 레이어의 연관성을 알 수 있기 때문이다. 입력영상을 시작으로 리스트내의 모든 레이어를 추적하면서 현재 도달한 레이어가 가지는 모든 내용을 순차적으로 파일에 기록하기 때문에 작업한 모든 내용을 한 파일에 기록할 수 있는 것이다. Figure 17은 단일 파일 저장을 위한 프로시저 “save_layer_list″의 동작과 관련된 예를 보여준다. 그림의 (a),(b)에서는 입력 영상을 5단계로 처리한 후의 화면 결과를 ″eye.dav″라는 단일 파일에 저장하는 것을 보여준다. (c)에서는 저장결과를 다시 읽어봄으로써 저장된 파일이 현재 영역의 모든 결과를 올바르게 유지하고 있음을 증명하고 있다.



(a)



(b)

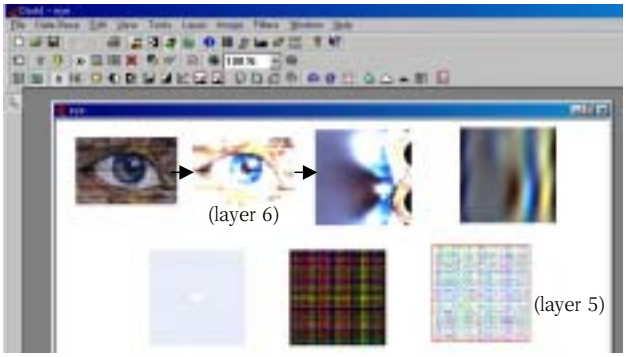


(c)

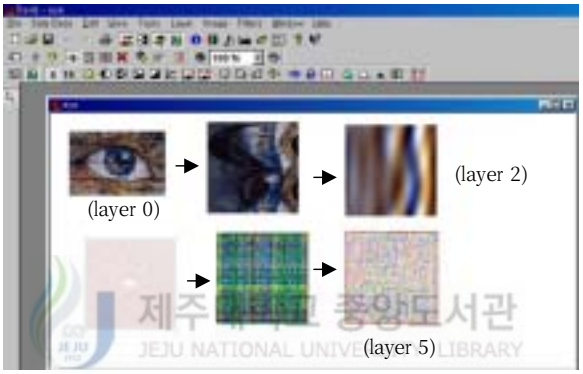
Figure 17. An Example of Workspaces of "save_layer_list" Procedure

3) 평가항목 4

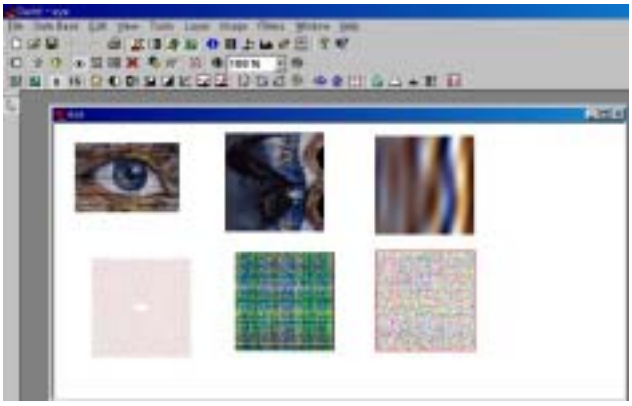
평가항목 4는 구현된 SDK에서만 볼 수 있는 가장 중요한 기능 중 하나인 처리 결과의 연관성 자동제어 (프로시저 “image_control”) 이다. 이 기능의 경우 기존의 SDK에서는 없는 새로운 기능이며 (Khoros의 경우는 제한적인 구현), 제안된 레이어 및 리스트의 유용성을 직접적으로 볼 수 있는 기능이다 (Figure 18). 우선 그림의 (a)에서는 입력된 영상을 5단계로 처리 한 후의 결과를 보여준다. 5 단계에 적용된 처리방법은 모두 다르지만 모두 같이 레이어 구조에 저장되고 상호 리스트를 유지하고 있다. 각 레이어들은 입력영상을 선두로 생성된 순서에 따라서 번호(0, 1, 2, 3, 4, 5)로 번호가 부여되어 있다. 이런 번호를 통해 특정 레이어에 대한 선행자와 후속자를 알 수 있는데 예를 들면 2번 레이어의 경우는 1번 레이어의 결과 영상을 입력으로 받아 처리한 후 자신의 결과로 보관하면서 동시에 3번 레이에에 입력으로 전달한다. 이런 상황에서 만약 입력 영상(0번)에 새로운 처리방법을 적용(1과 1 사이에 새로운 6번 레이어가 삽입)한 경우 기존의 1번에서 5번까지의 레이어는 자신이 받아들이는 입력과 출력이 바뀌게 되며 따라서 처리 결과도 달라지게 된다. 즉 현재 표시된 내용을 변경해야 하는 것이다. 기존의 SDK에서는 이럴 경우 사용자가 일일이 수작업을 통해서 만들어 주어야하기 때문에 어려운 점이 많았다. 그러나 구현된 SDK는 레이어 리스트의 연관성을 이용하여 이 작업을 자동으로 진행한다. 즉 특정 레이어가 새로 삽입되면 삽입된 레이어 이후에 있는 리스트 요소들에 대해서 자동으로 계산과정을 다시 수행함으로써 이 문제를 쉽게 해결하는 것이다. (b)와(c)가 이런 특성을 보여준다. (b)의 경우 새로운 레이어 6이 삽입되면 1에서 5번까지의 레이어가 자동으로 다시 처리되는 결과를 보여준다. (c)에서는 새로 삽입된 6번이 삭제되면 이전으로 돌아감을 보여준다.



(a)



(b)

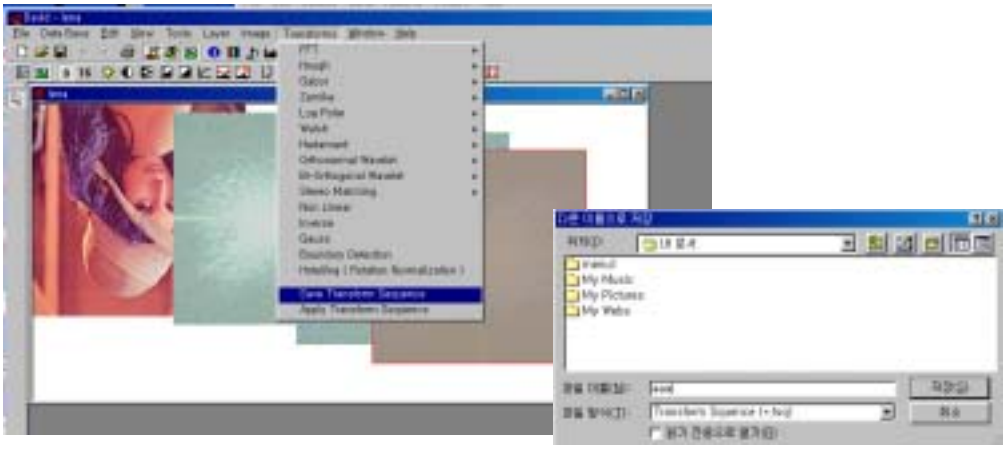


(c)

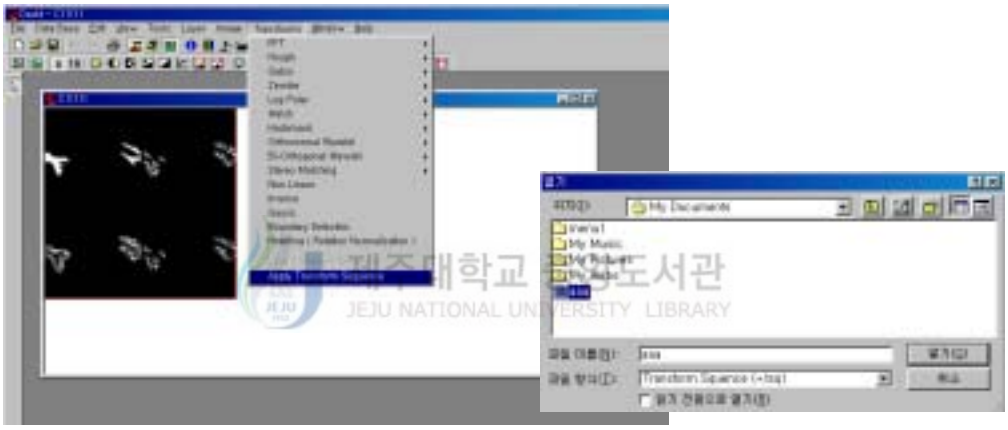
Figure 18. An Example of Workspaces of "image_control" Procedure

4) 평가항목 5

입력 영상을 다단계 처리하여 만든 리스트의 모든 내용은 2.2절에서 설명한 대로 단일 파일에 저장이 가능하다. 이런 현재 리스트에서 영상 처리 정보만(변환종류, 파라미터들)을 따로 뽑아 파일에 저장한 다음 새로운 영상에 일괄적으로 적용할 수 있다면 과거의 처리 절차를 손쉽게 재현할 수 있을 것이다(Figure 19). 이런 영상 처리에 대한 매크로 기능은 기존의 SDK에는 없는 기능으로 제안된 레이어 및 리스트를 활용하여 구현된 기능이다. 그림의 (a)에서는 현재 입력된 영상에 대한 4단계로 처리한 결과와 이런 결과를 얻는데 사용되었던 처리에 대해서 결과 영상을 제외하고 (변환방법, 사용 파라미터) 등에 대한 정보만을 현재 리스트에서 추출하여 파일의 형식으로 저장하는 과정을 보여준다. 이 과정을 거치게 되면 현재 리스트를 만들 때 사용되었던 처리 정보가 파일로 저장된다. 그림의 (b),(c)는 새로운 입력 영상에 대해서 저장했던 매크로 처리 파일을 적용한 결과를 보여준다. 새로운 입력 영상에 대해서 이전의 4단계 처리를 그대로 적용하기 위해서는 사용자는 단순히 이전에 저장했던 매크로 파일을 읽어서 실행만 시키면(Figure 19-b), 이전의 처리과정을 다른 영상에 적용한 결과를 바로 얻을 수 있는 것이다 (Figure 19-c).



(a)



(b)

(c)

Figure 19. An Example of Workspaces of "image_macro" Procedure

V. 결론

본 논문에서는 영상처리 응용 시스템 개발을 지원하는 SDK를 구성하는데 필요한 자료구조(레이어 및 리스트)와 이들 자료 구조의 저장/제어 방법에 대해서 제안하였다. 또한 제안된 방법을 바탕으로 실제 영상 획득, 영상의 다단계처리, 인식을 종합적으로 지원하는 SDK를 실제로 구현하고 실험하였다.

레이어 및 리스트 구조의 개발은 영상처리에서 사용되는 여러 영상들의 관리를 용이하게 해주기 때문에 효율적인 영상처리 SDK의 구성을 가능케 하였다. 또한 현재 처리되고 있는 영상들에 대한 모든 처리과정을 단일 파일에 저장시킴으로써 개발과정자체를 기록할 수 있도록 하였다. 이 방법의 개발로 인하여 영상처리 응용기술 개발에 대한 효율성을 증가시키는 길이 제공되었다.

지금까지 구현된 적이 없었으나 영상처리 SDK에 필수적인 기술이었던 입력영상과 처리된 영상간의 관련성 문제를 제안된 레이어 및 리스트를 통해 해결함으로써 부가적으로 구현된 기능이 자동 연관성 제어이다. 획득된 영상을 여러 단계로 처리한 결과들 중간에 새로운 단계를 삽입하거나 임의의 단계를 수정할 경우, 삽입되거나 수정된 단계 이후의 내용을 자동적으로 다시 계산해주는 이런 기능은 개발자로 하여금 자유롭게 처리 단계를 변화시키거나 삽입/삭제시킬 수 있도록 한다.

이와같이 본 논문에서는 지금까지 연구/개발된 적이 없는 독창적이고 새로운 형태의 영상처리 SDK를 만드는데 필요한 핵심기술을 제안하고 실제 구현을 통해 타당성을 검증해 보았다. 이러한 본 논문에서의 연구분야는 영상처리 SDK에 대한 필요성이 증가하는 현재의 추세에 부합되는 적절한 것이며, 아울러 새롭고 획기적인 개념을 바탕으로 한 영상처리 응용시스템 개발 SDK 구성에 필요한 독창적인 기반기술을 확보할 수 있었다.

VI. 참고문헌

Adobe. 2002. Adobe PhotoShop 7.0 Manual. Adobe Inc.

Andrews, H. C., Tescher, A. G. and Kruger, R. P. 1972. Image Processing by Digital Computer. IEEE Spectrum, 9(7). 20-32.

Argiro, D. and Rasure, J. An X windows based application programming system. in Proc. Xhibition 92: A Window on Distributed Computing(San Jose).

Bongiovanni, G., Cinque, L., Levialdi, S. and Rosenfeld, A. 1993. Image segmentation by multiresolution approach. Pattern Recognition, 26(12). 1845-1854.

Bow, S. T. 1992. Pattern Recognition and Image Preprocessing, Marcel Dekker.

Daugman, J. 1998. Phenotypic versus genotypic approaches to face recognition. Springer-Verlag. 108-123.

Daugman, J. 1999. Biometric decision landscapes. Univ. of Cambridge Computer Lab.

Dubois, E., Prasada, B. and Sabri, M. S. 1981. Image Sequence Coding - In Image Sequence Analysis. New York : Springer-Verlag. 229-288.

Dyer, D. S. 1990. A dataflow toolkit for visualization. IEEEE Comp. Graphics Applications, 10(4). 60-69.

Fickett, J. 1996. Finding Genes by Computer: the State of the Art. Trends

Genet., 12(8). 316-320.

Gonzalez, R. C. and Paul W. 1987. Digital Image Processing. Addison-wesley.

Green, D. M. and Swets, J. A. Signal detection theory and psychophysics. Wiley.

Halperin, E., Faigier, S. and Gill-More, R. 1999. FramePlus - Aligning DNA to Protein Sequences. Bioinfomatics, 15(11). 867-873.

Hara, Y., Atkins, R. G., Yueh, S., Shin, R. T. and Kong. 1994. Application of Neural Networks to Radar Image Classification. IEEE Trans. on Geosci & Remote Seeing, 32(1). 100-109.

Jain, A. K. 1989. Fundamentals of Digital Image Processing. Prentice-Hall. 1-10.

Koehler, G. 1998. Biometrics: A case study- Using Finger Image access in an Automated branch. Proc. of CTST'98, 1. 535-541.

Konstantinides K. and Rasure, J. R. 1992. Khoros Programmer's Manual. Univ. of New Mexico.

Konstantinides K. and Rasure, J. R. 1994. The Khoros Software Development Environment for Image and Signal Processing. *IEEE Trans. Image Processing*, 3(3). 243-252.

Landgrebe, D. A. 1997. On Progress Toward Information Extraction Methods for Hyperspectral Data. SPIE 42nd Annual Meeting, San Diego CA.

Marr, D. 1982. Vision. W. H. Freemen and Company.

Mathwork Inc. 2002, Mathtool.net, http://www.mathtool.net.

Matthew C. 2002. The HIPS package, http://www.permutationcity.co.uk

Mikes, S. 1992. Visual Programming tools for X.. X. J, 1(5). 50-62.

Mintie, D. 1998. Biometrics for state identification applications-operational experiments. Proc. of CTST'98, 1. 299-312.

Munechika, C. K., Warnick, J. S., Salvaggio, C. and Schoot, J. R. 1993. Resolution Enhancement of Multispectral Image Data to Improve Classification Accuracy. PE & RS, 59(1). 67-72.

Pan, J. J. and Chang, C. I. 1992. Destriping of Landsat MSS Images by Filtering Techniques. PE & RS, 58(1). 1417-1423.

Prasad, L. and Iyenger, S. S. 1997. Wavelet analysis with applications to image processing. CRC Press.

Rao, K. R. and Hwang, J. J. 1996. Techniques & standards for image, video & audio coding. Prentice-Hall.

Saker, P., Nagy, G. Z. and Lopresti, D. 1998. Spatial Sampling of Printed Patterns. PAMI, 20(3). 344-350.

Salu, Y. and Tilton, J. 1993. Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods.

IEEE. Trans. on Geosci. & Remote Seeing, 31(3). 600-617.

Shen, W. 1997. Evaluation of automated biometrics-baed identification and verification system. Proc. IEEE, 85. 1464-1479

Tan, T. N. 1998. Rotation Invariant Texture Features and Their Use in Automatic Script Identification. PAMI, 20(7). 751-756.

Upson. C. 1989. The application visualization system : a computational environment for scientific visualization, IEEE Comp. Graphics Applications. 30-42.

Wayman, J. L. 1999. Error rate equations for the general biometric system, IEEE Automation and Robotics Magazine.

Wayman, J. L. 2000. National Biometric test center collected works. San Jose state Univ.

Williams, C., Rasure, J. and Hansen, C. 1992. The state of the art of visual languages for visualization. Proc. Visualization(92).

XITE. 2002. http://www.ifi.uio.no/forskning/grupper/dsb/Software/Xite/

문지현, 안도성, 김학일, 2001. 지문 인식 시스템 성능 평가를 위한 플랫폼 구현. 2001년 정보과학회 춘계 학술대회 논문집, 28(1). 601-603.

장동혁. 2001. 디지털 영상처리의 구현. 정보게이트.

Abstract

The repetition of image transforms, known as iterative processing, is a key element in image processing, pattern recognition, biometrics and other related fields. In this process, an image is repeatedly transformed with different parameters until a satisfactory solution is found. A well formed image processing SDK(System Development Kit) allows users to create a processing results of their applications and interactively control input, output and various parameters. Although there are many researchers working on the implementation of SDKs, little success has been achieved because of the lack of researches in defining appropriate data structures for a well formed SDK.

In this thesis, for the implementation of a well formed SDK, we proposed two internal structures called layer and layer-list. The layer is an unified structure for image transforms and contains images (multiband arrays of data) and informations of an applied transform. Layers are connected together into a list called layer-list by the data dependencies. The starting point of a layer-list is a layer with no data dependencies (source layer). Using the proposed internal structures, we implemented a well formed SDK. A major advantage of the implemented SDK from other SDKs is that a single document (workspace) is used for both image processing and layer control. This advantage is obtained due to the proposed internal structures.

List of Tables

List of Figures