

## 향상된 트래킹 시스템과 실시간 수성 사인펜을 위한 사실적 드로잉

허혜정\*, 이주영\*\*

# Improved Tracking System and Realistic Drawing for Real-Time Water-Based Sign Pen

Hyejung Hur\*, Ju-Young Lee\*\*

### 요약

본 논문에서는 저가의 웹 카메라를 사용하여 마커 없이 손끝과 붓을 트래킹 하는 시스템을 제시한다. 트래킹 시스템은 CUDA를 사용하여 병렬처리를 적용했다. 이 트래킹 시스템은 노트북이나 데스크탑과 같은 환경에서 수행이 가능하고, 실시간 애플리케이션에 사용 가능한 성능을 가진다. 또한 본 논문에서는 사적인 수성 사인펜 드로잉 모델을 제시하고 구현된 결과를 보여준다. 제안하는 시스템은 손끝과 붓을 트래킹 하는 저가의 실시간 트래킹 시스템으로 사실적 드로잉 애플리케이션과 연동하여 미래 최첨단 교육 환경 구축의 테스트베드로의 활용을 기대한다.

▶ Keywords : 사실적 드로잉, 비전 기반 트래킹, 병렬처리

### Abstract

In this paper, we present marker-less fingertip and brush tracking system with inexpensive web camera. Parallel computation using CUDA is applied to the tracking system. This tracking system can run on inexpensive environment such as a laptop or a desktop and support for real-time application. We also present realistic water-based sign pen drawing model and implementation. The realistic drawing application with our inexpensive real-time fingertip and brush tracking system shows us the art class of the future. The realistic drawing application, along with our inexpensive real-time fingertip and brush tracking system, would be utilized in test-bed for the future high-technology education environment.

▶ Keywords : Realistic Drawing, Vision-Based Tracking, Parallel Processing

•제1저자 : 허혜정 •교신저자 : 이주영

•투고일 : 2014. 2. 10, 심사일 : 2014. 2. 20, 게재확정일 : 2014. 2. 25

\* University of Illinois at Chicago 전산학과 박사과정

\*\* 덕성여자대학교 컴퓨터학과(School of Information and Media, Duksung Women's University)

※본 연구는 덕성여자대학교 2012년 교내연구비 지원으로 이루어졌음

## I. 서론

우리는 일상생활에서 스마트 폰의 터치스크린을 누르며 책을 읽거나, 음악을 듣거나, 웹 서핑을 하거나, 게임 등을 하고 있는 사람들을 쉽게 볼 수 있다. 또한, 쇼핑몰, 호텔, 공항 등과 같은 공공장소에서 현재 위치와 상점의 위치를 조회할 수 있는 터치스크린을 종종 볼 수 있다. 우리 생활에 터치스크린은 보편화 되어있다. 버튼 선택, 화면 이동, 확대와 같은 단순 내비게이션 및 조작 용도로 많이 사용되고 있으며, 입력 장치인 마우스를 대체하는 용도로 성공적이다.

터치스크린이 좀 더 정밀화 된다면 종이와 펜 등을 사용하는 모든 활동들을 수행할 수 있을 것이다. 인간의 삶의 패턴을 바꾼 스마트 폰의 영향력보다 더 크게 변화된 또 다른 시대를 열 것이다. 터치스크린의 용도를 확장하여 교육에 활용할 수 있는 예를 살펴보자. 교실의 책상이 터치가 가능한 스크린으로 바뀌게 된다면, 영화에서 나올법한 가능성들을 쉽게 상상해볼 수 있을 것이다. 평소에는 책상의 표면 같다가, 전원을 켜면 모니터 화면이 나오고, 학생들은 교과서와 노트를 가지고 다닐 필요가 없이, 스크린에서 자신의 교과서를 볼 수 있고, 노트 필기를 스크린에 직접 할 수 있게 되고, 필기된 내용은 클라우드 서비스를 제공하는 리모트 하드 디스크에 저장된다. 미술 시간에는 그림 도구 필요 없이 그림을 그릴 수 있게 되고, 재질을 조절해서 같은 그림을 여러 다른 재질의 캔버스에 그린 것으로 변경할 수 있다. 터치스크린을 교육적이고 생산적인 용도로 활용하기 위해서는 트래킹의 정밀함이 필수적이다.

세종시 교육청은 스마트한 교육 시스템을 도입하기 위해서 최첨단 교육시설을 갖추도록 스마트 교육 기반 시스템 구축과 더불어 기반 조성에 공을 들이고 있다. 다양한 콘텐츠 개발과 유비쿼터스 클라우드 환경 조성 등과 같이 스마트한 학업 환경을 구축하려 한다. 이는 앞에서 언급한 교육 환경과 목표를 같이 한다고 볼 수 있다. 단순히 장비를 최첨단으로 변경하는 것이 스마트한 교육 환경을 보장하지는 않는다. 스마트한 교육 시스템 도입은 최첨단 교육시설과 더불어 콘텐츠 구축도 중요하며, 학생들에게 혼동을 주지 않기 위해서, 교육 활용 가능성을 유저 스테디(user study)를 통해 확인할 필요가 있다. 유저 스테디를 통해 검증된 교육 환경을 모델 학교에서 적용하고, 수정 보완하여, 점차적으로 확장되어야 한다.

본 논문에서는 맨손 트래킹 부분을 병렬처리를 이용하여 속도를 향상시키고, 3차원이 아닌 손 인덱싱과 붓의 2차원 위치를 찾아내는 문제로 축소하여 웹캠을 사용하는 저가의 비전

(vision) 기반 트래킹 시스템을 제안한다. 또한 사실적인 수성 사인펜 드로잉 효과가 적용된 그림 툴과 결합하여 트래킹 시스템의 교육 활용 가능성을 보여준다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로서 비전기반 트래킹과 사실적 드로잉 기법을 살펴본다. 3장에서는 성능을 향상시키기 위해 CUDA를 사용하여 병렬처리를 적용한 이미지 프로세싱 단계를 제시하고, 사실적 드로잉 효과를 보여주기 위한 기법을 제시한다. 4장에서는 구축된 트래킹 시스템에 사실적 드로잉 기법을 적용한 실험 결과들을 보인다. 마지막으로, 5장에서 결론을 맺는다.

## II. 관련연구

### 1. 비전 기반 트래킹

비전 기반 트래킹 시스템에는, 우선, 전통적인 마커 기반 트래킹(marker-based tracking)이 있다. 영화나 애니메이션 제작에 많이 사용되어 일반인에게 많이 알려져 있다. ART(Advanced Realtime Tracking) 사의 ARTTRACK[1]과 같은 고가의 카메라 장비를 사용하며, 역반사 마커(retro-reflective marker)나 LED를 인식한다. 손 트래킹에 쓰일 경우 역반사 마커가 부착된 장갑이나 손에 부착하는 장비를 사용하여야 한다. Natural Point사의 OptiTrack[2]과 같이 비교적 저렴한 가격으로도 시스템 구축은 가능하나, 손 트래킹이 가능한 제품과 일반 카메라와의 가격은 최소 9배에서 최대 50배 비싸다. 고가의 카메라 장비와는 성능 면에서 차이가 크다. 일반 카메라 장비를 사용할 경우 여러 가지 색상으로 만들어진 장갑을 이용하여 손을 트래킹하기도 한다[3].

두 번째로 맨손 트래킹(bare-hand tracking)이 있다. 맨손 트래킹은 계속적으로 연구되는 분야 중에 하나로, 역반사 마커가 부착된 장갑이나 손에 부착하는 추가적인 장비 없이 카메라로 맨손을 트래킹 한다. 대부분의 맨손 트래킹 방법은 피부색을 기반으로 하여 손의 영역을 찾아내고, 손가락 마디와 뼈의 구성을 중심으로 형태학상의 분석을 통해 이루어진다[4,5,6,7,8,9]. 곡률(curvature) 분석을 통해 손 끝(fingertip)을 찾아내는 방법[5], 피부색과 손톱(fingernail)의 색상 패턴을 분석하여 손끝을 찾아내는 방법[8] 등 다양한 분석 방법이 소개 되었다. 빛에 따른 주변 환경 조건에 영향을 많이 받는 마커 기반 트래킹과 달리, 맨손 트래킹은 빛의 영향을 많이 받지 않아서 안정적인 방법이다.

그러나 손의 다양한 포즈들을 분석하는 알고리즘의 계산이 복잡해서 순수 맨손 트래킹은 실시간 애플리케이션에 적용하는 것은 불가능하다고 분석되고 있다[4].

마지막으로 데이터 기반 포즈 트래킹(data-driven pose tracking)이다. 다양한 포즈들로 구축된 합성 데이터베이스를 기반으로 손의 포즈를 데이터베이스 내의 콘텐츠와 비교하여 빠르게 가장 가까운 포즈를 찾아내는 방법이다. 데이터베이스 내의 특성(feature) 매칭에 기반을 하여 손끝을 찾아내는 알고리즘을 적용하기 때문에 높은 정확도로 손가락을 트래킹 하여 안정적으로 글자를 인식하는 시스템을 제안하였다[10]. 정확도와 속도를 향상 시키는 방법이지만, 반대로 데이터베이스내의 콘텐츠의 질에 따라 정확도가 달라질 수 있고, 데이터베이스 사이즈에 따라 비교 예측하는 속도를 높이는 기법이 요구된다[11].

본 논문에서는 비전 기반 맨손 트래킹 시스템을 개발하였다. 비전 기반 맨손 트래킹 시스템이 계산의 복잡성으로 인한 성능 저하로 실시간 애플리케이션에 실제로 적용하는 것이 불가능하다는 문제점을 병렬처리로 해결하고 3차원이 아닌 손 인덱싱과 붓의 2차원 위치를 찾아내는 문제로 축소하여 실시간으로 손을 트래킹 할 수 있도록 하였다.

## 2. 사실적 드로잉 기법

Sousa[12,13]는 3D 모델을 입력 받아서, 펜슬 드로잉 효과가 적용된 그림으로 변형시켜주는 프레임을 제안하였다. 흑연의 단단한 정도와, 연필의 팁(tip)의 모양(shape), 팁의 모양을 다각형으로 구성하였을 때 종이에 닿는 면적의 모양과 압력에 따른 흑연의 분배량을 계산하여 펜슬 드로잉 효과를 생성한다. 종이의 높이 맵(height map)을 구성하여 흑연 분배 계산 시 적용하였으나 실시간 변환을 지원하지 않는다. 또 다른 사실적 펜슬 드로잉 효과는 이미지를 입력 받아, 제안된 펜슬 드로잉 알고리즘을 적용하여 결과 이미지를 생성한다[14,15]. 일반적인 LIC (Line Integral Convolution) 기반 펜슬 드로잉 효과 생성 프로세스는 이미지를 입력 받아서, 그레이 스케일(gray scale)로 변환시킨 후, 3개의 중간 결과 파일을 생성한다. 에지(edge)를 추출한 이미지, 노이즈(noise)를 추출한 이미지, 벡터 필드(vector field)를 추출한 중간 결과 이미지를 생성한다. 노이즈 이미지와 벡터 필드 이미지를 통해 LIC 이미지를 생성하고, 에지 이미지와 LIC 이미지를 통해 스트로크(stroke) 이미지를 생성한다. 스트로크 이미지와 종이 샘플을 결합하여 완성된 펜슬 드로잉 효과를 생성한다[15].

Curtis[16]는 다양한 워터 칼라 효과를 만들 수 있는 방

법을 제안하였다. Curtis의 워터 칼라 모델은 얇은 물의 유체 시뮬레이션(shallow-water fluid simulation)을 통해 생성된 반투명한 글레이즈의 집합에 기반을 두어 생성한다. Way[17]는 서양화와 다른 동양화만의 칼라 잉크 효과를 만들어 낼 수 있는 기법들을 제안하였다. Way는 종이에 즉각적으로 흡수되는 효과와 종이의 조직을 따라 흐르는 특성을 표현하기 위한 퍼지를 생성하는 확산(diffusion) 모델을 제안하였다. Sun[18]은 입력 받은 이미지로부터 영역을 분할하고, 영역의 특성에 따라 모양과 색의 분산 효과에 대한 속성 값을 다르게 적용하여 결과 이미지를 생성하는 모델을 소개했다.

본 논문에서는 Sousa와 Curtis의 펜슬 드로잉 모델과 워터 칼라 모델을 기반으로 하여 수성 사인펜 드로잉 모델을 제안한다. 3D 모델이나 이미지를 입력 받아서 알고리즘을 적용하는 후처리 방식이 아닌, 트래킹 시스템과 연동하여 실시간으로 사용자의 입력을 받아 수성 사인펜 드로잉 모델을 통해 생성된 결과를 곧바로 볼 수 있도록 하는 실시간 시스템을 구축하였다.

## III. 제안하는 수성 사인펜 드로잉 기법

### 1. 병렬 이미지 프로세싱

그래픽 카드를 이용해서 범용 컴퓨터에서도 병렬 처리가 가능해졌다. CPU(Central Processing Unit)와는 달리 그래픽 카드의 GPU(Graphics Processing Unit)는 병렬의 멀티 코어 구조를 가지고 있고, 각 코어는 수천 스레드(thread)를 동시에 실행시킬 수 있다. 응용 프로그램이 수행하는 계산 작업이 이러한 병렬처리 연산에 적합할 경우 상당한 성능 향상을 기대할 수 있다. 트래킹 시스템의 이미지 프로세싱 작업은 병렬 처리에 적합한 구조를 가지고 있다. 따라서 본 논문에서는 이미지 프로세싱 단계에서 CUDA를 사용하여 병렬처리가 가능한 부분에 적용하였다. 손 트래킹을 위한 이미지 프로세싱 단계는 그림 1과 같다. 이미지 프로세싱 단계 중 왜곡 보정, 노이즈 필터링, 배경 필터링, 피부색에 따른 손 영역 분석, 원뿔 곡선 추출 단계는 GPU에서 병렬로 처리하여 속도 성능을 향상시켰다.

- 실시간 이미지 캡처: 웹캠과 연결하여 실시간으로 이미지를 가져온다.
- 왜곡 보정: 굴곡 현상 보정, 조명에 따른 색상 보정한다.
- 노이즈 필터링: 보정된 이미지에 가우시안 필터를 적용하여 노이즈를 필터링한다.

- 배경 필터링: 필터링된 이미지로부터 손 영역을 추출하기 위해서 배경을 필터링한다.
- 피부색에 따른 손 영역분석: 사람의 피부 톤은 다양하기 때문에, 피부 색상 히스토그램(skin color histogram)을 바탕으로 손 영역을 분석한다. 피부 색상 히스토그램 영역 밖의 오브젝트들을 제거한다.
- 윤곽(connected contour) 추출: 연결된 윤곽을 찾아서 영역 그룹들로 분류한다.
- 영역 사이즈별 필터링: 하나로 연결된 윤곽의 영역 사이즈가 기준치보다 작을 경우 제거한다.
- 원뿔 곡선(conic section) 추출: 윤곽들로부터 원뿔 곡선 포인트들을 찾고, 곡률(curvature)에 따라 피크(peak) 포인트와 밸리(valley) 포인트를 구별한다.
- 손등/손바닥 영역 추출: 분석된 피크 포인트와 밸리 포인트 정보를 통해 손등/손바닥 영역을 찾아낸다.
- 손가락 인텍싱 도출: 예상되는 손가락 인텍싱을 도출한다.
- 붓 트래킹: 엄지와 검지 사이에 붓으로 예상되는 일자 형태의 붓을 추출한다.
- 좌표 계산: 가상 2D 좌표계에서의 각 포인트 좌표를 계산한다.

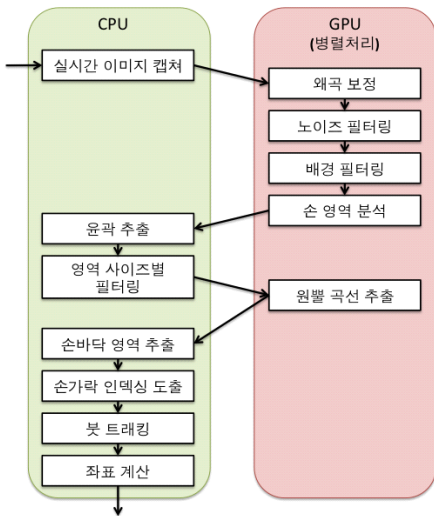


그림 1. CPU와 GPU에서의 이미지 프로세싱 단계도  
Fig.1. Steps of image processing on CPU and GPU

2. 사실적 드로잉 효과

본 논문에서는 앞 절에서 설명한 트래킹 시스템을 구축하여, 손끝이나 붓을 사용해서 그림을 그릴 수 있는 그림 툴을 사실적 드로잉 기법을 적용하여 개발한다. Sousa는 3D 모델

을 입력 받아서 연필, 종이, 블렌더 그리고 지우개의 모양과 물질의 특성을 분석하여 2D 펜슬 드로잉 이미지를 생성하는 모델을 제시하였다(12,13). Curtis의 워터 칼라 모델(16)에서는 번짐 효과를 제안하였다. 본 논문에서는 Sousa의 펜슬 드로잉 모델과 Curtis의 번짐 효과를 이용하여 수정된 실시간 수정 사인펜 타입의 드로잉 모델을 제안한다. 수정 사인펜 타입의 드로잉 모델은 사용자의 손끝이나 붓을 트래킹 한 결과 값을 입력 받아서, 실시간으로 사인펜 효과를 그려준다. 그림 2는 제안하는 드로잉 모델의 절차를 보여준다. 픽셀 맵을 사용하여 각 픽셀에 해당되는 값을 계산하여 화면에 그려주며, 각 절차에 따른 효과는 다음과 같이 계산된다.

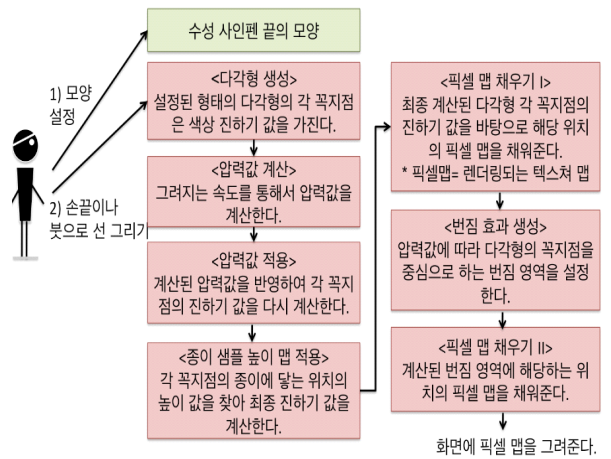


그림 2. 사실적 드로잉 절차  
Fig.2. Procedure of realistic drawing

- 그리는 속도에 따른 색상 진하기 정도 조절: 압력 값  $p$ 를
- 모양: 수정 사인펜 끝의 모양은 다각형( $n$ -각형)으로 나타내고 다각형의 각 꼭짓점의 색상 진하기를 다르게 설정한다. 펜 끝의 모양  $n$ -각형의 중심을  $g_0$ 라 두고, 각 꼭짓점을  $g_i$  ( $i=1, 2, \dots, n$ )으로 둔다. 펜의 중심은 가장 진하고 가장자리인 각 꼭짓점으로 갈수록 멀어지도록 진하기 효과를 낸다. 각 꼭짓점  $g_i$ 의 진하기 정도 값은  $f_i$ 로 표기한다. 그림이 그려진 위치에 덧칠을 하게 되면, 해당 위치에서 기존 색상에 더해져서 덧칠 효과를 표현한다. 본 논문에서는 다양한 다각형 모양들을 드로잉 툴에 적용해 보았다. 그림 3은  $n=6$ 인 육각형의 예이다.

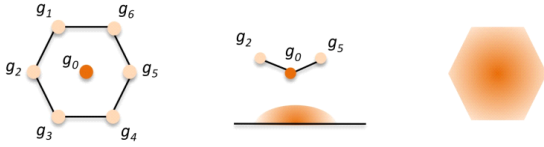


그림 3. 펜 끝의 모양 (육각형의 예제)  
Fig.3. Pen-tip shape (Example for Hexagon)

- 그리는 속도에 따른 색상 진하기 정도 조절: 압력 값  $p$  를 적용하여 사용자가 선을 그릴 때 소요되는 시간차를 계산해서 색상의 진하기 정도를 조절한다.  $p$  값의 범위는  $0 \leq p \leq 1$  이며, 다각형의 각 꼭짓점  $g_i$ 의 진하기  $f_i$ 는 압력 값을 반영한 진하기 값  $f'_i = f_i \times p$ 로 재계산된다.
- 종이: 의사-무작위 프로세스(pseudo-random process) 들 중에 하나를 선택하여 무작위하게 종이의 높이, 0보다 크며 1 이하인 값을 생성해서 종이의 각 셀(cell)의 높이 맵을 만든다[19]. 각 셀은 4개의 점들로 이루어지며 그 점들의 높이들로 높이 맵이 만들어진다. 그림 4는 생성된 종이 맵의 예를 보여준다.



그림 4. 종이 맵 예제  
Fig.4. Example of paper height map

그림을 그릴 때, 펜 끝 다각형의 꼭짓점이 종이에 닿는 위치에 해당하는 종이 셀의 높이는 그 셀을 이루는 4개 점들의 높이를 기반으로 계산한 선형보간 값(linear interpolated value)으로 나타낸다. 그림 5에서 펜 끝 다각형의 어느 한 꼭짓점이 종이 셀  $a_k$ 에 닿는다고 하면, 셀  $a_k$ 의 4개 점들의 높이(각 각  $h_1, h_2, h_3, h_4$ )들로 선형보간 값을 계산하여 그 값이 종이 셀  $a_k$ 의 높이(그림에서는  $h$ )가 된다. 펜의  $n$ -각형 꼭짓점  $i$ 가 종이 셀  $a_k$ 에 닿을 때, 종이 셀의 높이 선형보간 값을  $h(a_k)$ 라 두면 꼭짓점  $i$ 에서의 최종 진하기 값은 다음과 같이 최종적으로 계산된다.

$$p_i = f'_i \times h(a_k) \quad (\text{여기서, } i = 1, 2, \dots, n)$$

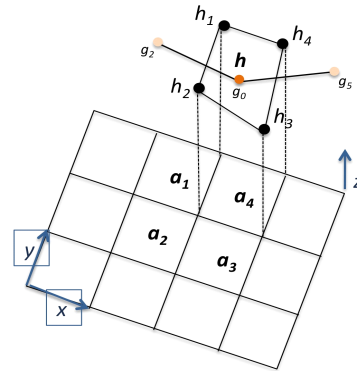


그림 5. 종이 맵과 펜 끝 다각형의 꼭짓점과의 접촉  
Fig.5. Pen-tip points and paper height map

- 번짐 효과: 수성 사인펜 끝의 다각형의 중심을 제외한 각 꼭짓점 진하기를 기준으로 번짐 효과를 생성한다. 일정하게 주어진 기본 번짐 영역에 각 꼭짓점에서의 압력 값  $p$ 를 적용하여 번짐 영역을 계산한다. 그림 6은 각 꼭짓점들에서의 압력 값을 반영한 번짐 영역의 예를 보여준다. 번짐 영역은 압력 값에 따라 다르게 나타나며 그림 6의 예에서는  $d_0$ 가 가장 크다. 각 번짐 영역의 중앙에서 외곽으로 갈수록 진하기가 열어지며, 외곽의 진하기는 0의 값을 갖는다. 주어진 기본 번짐 영역은 그림 툴 상에서 그 값을 조정 가능하도록 한다.

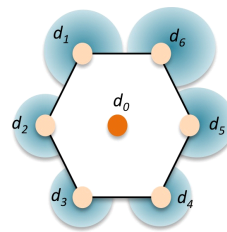


그림 6. 다각형의 각 꼭짓점에서의 번짐 효과  
Fig.6. Spread effect on each point of polygon

- 지우개: 종이가 가지고 있는 색상의 값을 빼주는 과정으로 한 번에 그 영역의 색상이 전체적으로 지워지는 것이 아니라 서서히 지워지도록 한다. 지우개도  $n$ -각형 모양을 가지며, 각 꼭짓점에 따라서 제거되는 색상의 진하기 값이 다르다. 지우개의 중심  $e_0$ 는 많이 제거되고, 가장자리  $e_1 \sim e_n$ 로 가면 중심보다는 적게 지워지는 효과를 표현한다. 진하기 값 계산의 경우와 마찬가지로 지우개로

지우는 속도에 따른 압력 값  $p$ 를 반영한 색상의 진하기 제거 값을  $p$ 로 나타내고, 종이의 해당 위치의 높이 값도 적용되어 실제 제거되는 색상의 진하기 값은 다각형의 각 꼭짓점의 위치에 따라 계산된다.

#### IV. 구현 결과

본 논문에서 제안하는 비전 기반 트래킹 시스템은 C++로 개발되고 웹캠으로부터 이미지를 가져오는 부분은 QuickTime 라이브러리를 사용했으며, 병렬처리 부분은 NVIDIA CUDA로 구현되었다. 트래킹 속도는 최소 24fps 유지하며 트래킹 시스템과 함께 구축된 그림 툴은 OpenGL로 개발했다. 그림 툴은 일반 그림 툴과 수성 사인펜의 사실적 드로잉이 적용된 그림 툴 두 가지 버전으로 구축되었다. 사용한 테스트 시스템은 애플사의 MacBook Pro 노트북으로 2 GHz Intel Core 2 Duo 프로세서, 2GB 메모리, NVIDIA GeForce 9400M 256 MB VRAM 그래픽 카드이고, 사용된 카메라는 Logitech HD Pro Webcam C910 1080p 30 fps 웹캠이다. 이는 저가로서 구축 가능한 것으로, 트래킹 시스템과 그림 툴은 같은 컴퓨터에서 사용된다.

##### 1. 손끝과 붓 트래킹

구축된 그림 툴은 트래킹 시스템을 통해 계산된 사용자의 손가락이나 붓의 위치를 받아 실시간 비디오 영상 스트림에 가상의 오브젝트들을 그려서 화면에 보여준다.

그림 7과 그림 8에서 그림 상단의 작은 사각형들은 색상 선택 버튼으로 손이나 붓을 해당 위치에 가져가면 색상이 변경된다. 붓으로 그렸을 때 조금 더 부드럽게 그려진다.

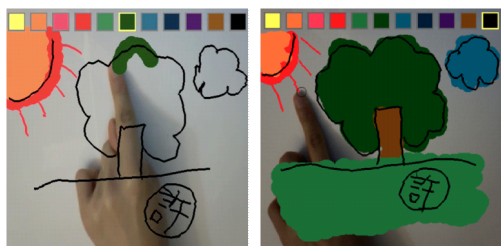


그림 7. 손으로 그림 그리기  
Fig. 7. Hand drawing



그림 8. 붓으로 그림 그리기  
Fig. 8. Pen drawing

##### 2. 수성 사인펜 드로잉 효과

그림 툴의 기능에 다양한 드로잉 효과를 적용한 사실적 드로잉 그림 툴을 구현하였다. 그림 9는 다양한 드로잉 효과가 적용된 선들을 보여준다. 펜의 모양, 수성 사인펜의 액체 물질의 성질, 선을 그리는 속도에 따른 다양한 효과를 보여준다.

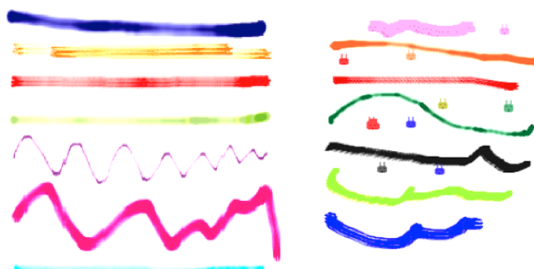


그림 9. 사실적 드로잉 효과가 적용된 라인들  
Fig. 9. Lines with water-based sign pen effect

그림 10은 선을 그리는 속도에 따른 압력을 반영한 색상 진하기의 효과를 보여주고 있으며, 그림 11은 이미 그려진 선 위에 다른 색으로 선을 겹쳐 그렸을 때의 색상의 섞임 효과를 보여준다. 그림 12는 지우개 효과를 보여주는데, 기존의 지우개와는 달리 한 번에 해당 영역의 색상을 흰 바탕으로 바꿔 버리는 것이 아니라, 해당 위치의 색상 진하기를 열게 하는 방식으로 자연스러운 효과를 생성한다.

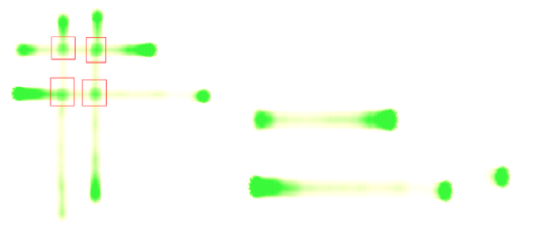


그림 10. 압력에 따른 색상의 진하기 효과  
Fig. 10. Pressure effect

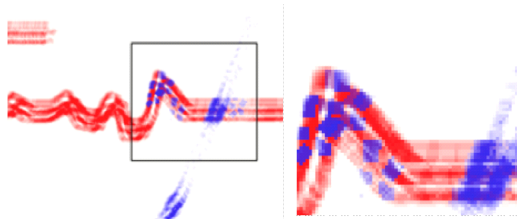


그림 11. 색 덧칠 효과  
Fig. 11. Overwrite effect



(a) 지우개 사용 전 (b) 지우개 사용 후  
그림 12. 지우개 효과  
Fig. 12. Eraser effect

## V. 결론 및 향후 계획

저가의 비전 기반 트래킹 시스템을 구축하고 병렬처리를 통해 실시간 애플리케이션에 적용 가능하도록 하였다. 사용자가 손이나 붓을 이용하여 그림을 그릴 수 있는 그림 툴을 개발하여, 단순한 트래킹 기반의 환경에서 사용하는 툴로서 국한시키지 않고, 제한한 실시간 수정 펜의 사실적인 드로잉 모델을 적용하였다. 구축된 환경은 미래 스마트 교육 환경 구축을 위한 테스트베드로 활용될 수 있을 것이다.

본 논문 결과물을 확장시켜 테이블 표면이 모니터의 표면이 되는 개인용 테이블을 제작해서 모니터 위에 직접 그림을 그릴 수 있고, 고해상도 카메라를 2개 이상 설치하여 트래킹의 정밀도를 높일 수 있다. 또한 한국화 수업을 위한 사실적인 한국화 그림 툴을 개발할 수 있을 것이다. 유저 스테디를 통해서 트래킹 시스템과 교육적인 애플리케이션의 발전 가능성, 추가 고려사항들을 도출하여 미래 스마트 교육 환경 구축을 위한 디자인에 도움이 될 것으로 기대한다.

## 참고문헌

- [1] <http://www.ar-tracking.com/products/interaction-devices/fingertracking/>
- [2] <http://www.naturalpoint.com/optitrack/>
- [3] R. Wang and J. Popovic, "Real-Time Hand-Tracking with a Color Glove", Proceedings of ACM SIGGRAPH, Vol.28, Issue 3, No. 63, 2009.
- [4] A. Erol, G. Bebis, et al., "Vision-based Hand Pose Estimation: A Review", Computer Vision and Image Understanding, Vol.108, Issue 1-2, pp.52-73, 2007.
- [5] D. Lee and S. Lee, "Vision-Based Finger Action Recognition by Angle Detection and Contour Analysis", ETRI Journal, Vol. 33 Issue 3, pp. 415-422, June 2011.
- [6] M. Bhuyan, D. Neog, and M. Kar, "Fingertip Detection for Hand Pose Recognition", International Journal of Computer Science and Engineering(IJCSE), Vol. 4, No. 3, pp. 501-511, 2012.
- [7] S. Nagarajan, T. Subashini, and V. Ramalingam, "Vision Based Real Time Finger Counter for Hand Gesture Recognition", International Journal of Technology, Vol. 2, No. 2, 2012.
- [8] J. Barrho, M. Adam, and U. Kiencke, "Finger Localization and Classification in Images Based on Generalized Hough Transform and Probabilistic Models", Proceeding of IEEE int'l Conf. Control, Automation, Robotics and Vision, pp.1-6, 2006.
- [9] S. Kang, M. Nam, and P. Rhee, "Color Based Hand and Finger Detection Technology for User Interaction", Proceeding of IEEE int'l Conf. Convergence and Hybrid Information Technology, pp. 229-236, 2008.
- [10] L. Jin, et al., "A Novel Vision-Based Finger-Writing Character Recognition System", Journal of Circuits, Systems, and Computers, Vol. 16, No. 3, pp. 421-436, Jan. 2007.
- [11] L. Ren, G. Shakhnarovich, et al., "Learning Silhouette Gestures for Control of Human Motion", ACM Transactions on Graphics, Vol. 24, Issue 4, pp.1303-1331, Oct. 2005.

- [12] M. Sousa and J. Buchanan, "Computer-Generated Graphite Pencil Rendering of 3D Polygonal Models", Computer Graphics Forum, Vol. 18, Issue 3, pp.195-208, September 1999.
- [13] M. Sousa and J. Buchanan, "Observational Models of Graphite Pencil Materials", Computer Graphics Forum, Vol.19, Issue 1, pp.27-49, March 2000.
- [14] H. Yang, Y. Kwon, and K. Min, "A Stylized Approach for Pencil Drawing from Photographs", Computer Graphics Forum, Vol. 31, No. 4, pp.1471-1480, June 2012.
- [15] M. Hata, M. Toyoura, and X. Mao, "Automatic Generation of Accentuated Pencil Drawing with Saliency Map and LIC", The Visual Computer, Vol. 28, pp. 657-668, June 2012.
- [16] C. Curtis, et al., "Computer-Generated Watercolor", Proceedings of ACM SIGGRAPH, pp.421-430, 1997.
- [17] D. Way, W. Lin, and Z. Shih, "Computer-Generated Chinese Color Ink Paintings", Journal of the Chinese Institute of Engineers, Vol. 29, No. 6, pp.1041-1050, 2006.
- [18] Y. Sun and J. Wang, "Representation of Watercolor Based on Regions", Proceeding of IEEE int'l Conf. Image Analysis and Signal Processing, pp. 496-500, 2011.
- [19] S. Worley, "A Cellular Texturing Basis Function", Proceeding of ACM SIGGRAPH, pp.291-294, 1996.

## 저 자 소 개



**허 혜 정 (Hur, Hyejung)**  
 1999년 덕성여자대학교 전산학과 학사  
 2001년 덕성여자대학교  
 전산정보통신학과 석사  
 2007~ Univ. of Illinois at  
 Chicago 박사과정  
 관심분야 : 컴퓨터그래픽스, HCI,  
 데이터가시화, 가상현실  
 Email : beinglikeanne@gmail.com



**이 주 영 (Lee, Ju-Young)**  
 1984년 이화여자대학교 수학과 학사.  
 1991년 The George Washington  
 Univ. 컴퓨터학과 석사.  
 1996년 The George Washington  
 Univ. 컴퓨터학과 박사.  
 1996~ 현재 덕성여자대학교  
 컴퓨터학과 교수.  
 관심분야 : 알고리즘, 분산/병렬처리,  
 그래프이론, HCI,  
 무선통신  
 Email : jylee@duksung.ac.kr