

움직임 궤적 분석 기반의 원거리 판서 기술

임승민, 정현석, 김성영*

Remote Drawing TechnologyBased on Motion Trajectories Analysis

Seung-min Leem, Hyeon-seok Jeong, Sung-young Kim*

요약 본 논문에서는 3차원 공간에서 손 위치를 추적하고 움직임 궤적을 분석하여 원거리에서 판서가 가능한 기술을 제안한다. 3차원 공간에서 손으로 입력하는 한글 음절은 글자 획과 이동 획이 구분되지 않아 음절의 종류를 구분하기 힘들다. 이에 본 논문에서는 한글 음절을 구성하는 획을 글자 획과 이동 획으로 구분한 후 이동 획은 제거하고 글자 획만을 출력하는 방법을 제안한다. 우선, 필기체 음절의 궤적에서 획의 끝 점을 검출하고, 검출한 끝 점 정보를 이용하여 입력 음절을 획 단위로 분리한다. 음절 집합으로부터 8가지의 획 패턴을 정의한 후 분리한 획에 대해서는 방향 코드를 기반으로 획 패턴을 분류한다. 그리고 이를 기반으로 최종적으로 획의 유형을 글자 획과 이동 획으로 분류한다. 분류된 획의 유형을 기반으로 입력된 음절에서 이동 획은 제거하고 글자 획만을 출력하여 가독성이 있는 음절 표시가 가능하도록 한다. 360개의 음절 집합에 대해 정확도를 측정하여 획의 패턴은 88.3%, 획의 유형 구분은 91.1%의 정확도를 얻었다.

Abstract In this paper, we suggest new technology that can draw characters at a long distance by tracking a hand and analysing the trajectories of hand positions. It's difficult to recognize the shape of a character without discriminating effective strokes from all drawing strokes. We detect end points from input trajectories of a syllable with camera system and localize strokes by using detected end points. Then we classify the patterns of the extracted strokes into eight classes and finally into two categories of stroke that is part of syllable and not. We only draw the strokes that are parts of syllable and can display a character. We can get 88.3% in classification accuracy of stroke patterns and 91.1% in stroke type classification.

Key Words : Electric White-Board, End Points Detection, Stroke Extraction, Stroke Type Classification, Virtual Drawing, Trajectories of hand positions

1. 서론

움직임 검출 및 인식 기술은 스마트TV, 게임, CCTV 등 다양한 분야에서 사용되는 유용한 기술이다. 이러한 기술의 발전과 함께 교육환경도 많이 바뀌고 있다[1]. 컴퓨터 기반의 터치스크린을 예로 들 수가 있다. 우리나라의 경우 전자칠판은 40인치가 80~120만원, 60인치는 200만원 내외, 80

인치는 400~500만원 내외 가격으로 판매되고 있다. 터치스크린은 전자칠판가격의 50% 이상을 차지할 정도로 가격 비중이 높다. 그 이유는 기존 TV에 강화유리와 철펠프레임을 사용하여 터치 패널을 부착하기 때문이다. 이로 인해 설치 및 유지 보수비용이 비싸고 크기가 커질수록 제작에 어려움이 발생한다. 따라서 기존의 터치스크린을 가상 터치 방식으로 변경하면 제품 가격을 획기적으로

This work was supported by research fund of Kumoh National Institute of Technology

*Corresponding Author : Dept. of Computer Engineering, Kumoh National Institute of Technology(sykim@kumoh.ac.kr)

Received March 17, 2016

Revised March 27, 2016

Accepted April 05, 2016

로 감소시킬 수 있다. 이를 통해 전자칠판의 무게 증가로 인해 교실이나 강의실 내부에서 안전사고를 미연에 방지할 수 있다. 그리고 일반 터치 시스템을 부착한 전자칠판에서는 강사가 스크린 앞에 위치한 경우에만 입력이 가능하지만 가상터치 방식을 도입하면 원거리 위치에서도 입력이 가능할 수 있다.

가상터치입력 방식은 터치스크린 대신에 카메라를 사용하여 입력을 처리하는 방식을 의미한다. 카메라 앞에서 손을 움직이면 터치 패널은 누른 것과 같이 입력이 이루어지도록 처리한다. 일반적인 터치스크린이나 터치 패널에서는 사용자가 스크린의 표면에 터치하여 입력을 수행하여 터치하는 순간 입력이 발생하고 스크린 표면에서 손을 떼면 입력이 이루어지지 않는다. 그러나 가상터치 입력방식에서는 입력의 시작과 끝을 구분하기가 쉽지 않다. [그림 1]은 가상터치 방식으로 입력한 일부 음절에 대한 예시이다. 일반 터치방식에서는 음절을 구성하는 획 성분만 표시되어 글자모양에 대한 확인이 가능하지만 가상터치 방식에서는 글자 획과 이동 획이 구분되지 않아 글자모양의 구분이 어렵다. 따라서 원거리 판서가 가능하기 위해서는 글자 획과 이동 획을 구분하여 이동 획은 글자 입력 단계에서 제거해야만 한다.

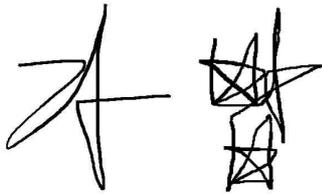


그림 1. 가상입력예시
Fig.1. Examples of two syllable wrote on virtual touch system

원거리 판서를 위해 음절 또는 자모 단위로 인식 처리를 수행하여 글자 획만 입력하는 처리 방법을 고려할 수 있다. 한글 필기체 인식 기술은 불과 얼마 전까지 활발하게 연구가 진행되었다. 온라인 한글 필기체 인식은 음절 단위의 인식 기술[2,3]과 자모 단위의 인식 기술[4,5]로 구분할 수

있다. 음절 단위의 인식은 현대 한글에서 자모의 조합 가능한 12768개 음절에 대해 다양한 사용자 입력 패턴을 고려한 방대한 훈련 집합을 생성해야 하는 어려움이 있다. 또한 음절 또는 자모 단위의 인식 기술은 글자 획의 입력이 있는 경우에만 적용할 수 있다. 중국 인쇄체 문자 인식을 위해 획 단위의 분리 처리를 위한 연구가 이루어졌다[6,7]. 본 논문에서는 이동 획과 글자 획을 구분하기 위해 이전 연구[8]에서는 획의 끝점을 검출하여 획을 분리하고 획의 필기 속도를 고려하여 이동 획과 글자 획을 구분하기 위해 시도하였다. 획 단위의 처리는 획 분리 과정이 어렵지만 훈련 데이터 생성과 인식 과정이 용이하다[9,10].

본 논문에서는 이전 연구[8]보다 끝점 검출과 획 유형의 구분 정확도에 대한 향상이 가능하도록 입력된 음절 데이터로부터 곡률을 기반으로 획의 끝 점을 검출하여 획 단위로 분리한 후 분리된 각 획에 대해 방향성을 나타내는 특징을 추출한다. 방향성 특징을 체인코드로 변환을 한 다음, 체인코드를 분석하여 이동 획과 글자 획을 구별한 뒤 이동 획만 제거하여 최종적으로 글자 획들만 추출한다.

본 논문의 구성은 2장에서 문자인식 시스템을 위한 훈련데이터베이스를 구축한 방법에 대해 설명을 하고, 3장에서는 획 단위로 나누기 위해 끝점을 찾아내는 알고리즘에 대해 기술한다. 4장에서는 분리된 획을 처리하는 방법, 5장에서는 실험 결과 및 분석을 하여 마지막 6장에서 결론을 맺는다.

2. 훈련 데이터베이스 구축

우선 분류 및 정확도 측정을 위한 문자를 정의한다. 한글은 자음, 모음에 해당하는 글자 획과 이동 획에 대해 다양한 패턴들이 존재하기 때문에 이를 모두 수용할 수 있는 음절 집합을 정의할 필요가 있다.

2.1 획의 기본패턴 정의

본 논문에서는 사용하는 훈련 집합의 크기를 최소화할 수 있도록 현대 한글의 자음, 모음에 대해 획 단위로 분리를 하여 훈련 데이터를 수집한다. 음절을 획 단위로 분리하여 처리하면 인식 과정이 쉽고 적은 훈련 집합만으로도 훈련 및 인식이 가능하다. 획 단위로 인식을 수행하기 위해서는 음절을 획 단위로 분리하는 처리 과정이 필요하다.

[그림 2]에서 입력 음절의 글자 획과 이동 획을 구성하는 획들의 패턴을 확인할 수 있다. 초성 ‘ㄱ’의 두 가지 패턴인 →, ↘, 중성 ‘나’는 패턴 ↓, → 그리고 종성 ‘ㄱ’에는 세 가지 패턴 →, ↓, →이 존재한다. 그리고 초성에서 중성으로 이동할 때 발생하는 이동 획에 대한 패턴 ↗, 중성의 입력 과정에서 발생한 이동 획 패턴 ↑, 중성에서 종성으로 이동할 때 발생하는 이동 획의 패턴 ↙, 마지막으로 종성에서 발생하는 이동 획 패턴 ↖을 확인할 수 있다. 결과적으로 글자 획 7개, 이동 획 4개를 포함하여 음절 ‘각’에는 총 11개의 획이 존재한다. 획은 곡선으로 구성될 수도 있지만 본 논문에서는 획의 형태를 단순화시켜 모두 직선으로 표현한다.

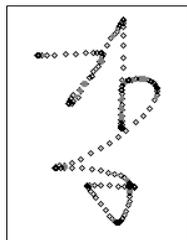


그림 2. 가상터치 방식의 음절 입력
Fig.2. Syllable written by virtual touch

[표 1]은 음절에서 발생할 수 있는 모든 글자 획과 이동 획의 기본 패턴을 정의한 것이다. 그런데 ↙ 패턴은 글자 획과 이동 획에서 모두 발생할 수 있다. ↙ 패턴을 제외한 나머지 모든 패턴은 패턴 유형에 따라 자동적으로 글자 획과 이동 획으로 분류가 가능하지만 ↙ 패턴의 경우 진후에

발생하는 패턴의 맥락을 이용하여 추가적인 분류 과정이 필요하다.

표 1. 획 패턴
Table 1. Stroke pattern

구분	↑	↗	→	↘	↓	↙	←	↖
글자 획	○	○	○	○	○	○	○	○
이동 획	○	○	○	○	○	○	○	○

2.2 음절 집합 정의

[표 1]의 획 유형을 기반으로 하여 [그림 3]과 같이 총 40개의 음절을 정의하여 정확도 측정과 인식 정확도 계산을 위한 훈련 집합으로 정의하였다. [그림 3]에 정의된 음절에 대한 패턴 유형별 발생 빈도는 [표 2]와 같다. 모든 획 패턴의 빈도가 균등하도록 음절 집합을 정의하려고 했지만 ‘→’, ‘↓’와 같은 패턴들이 대부분 음절에 높은 빈도로 발생하고 있어 다른 패턴들보다 상대적으로 빈도수가 높게 나타나고 있다.

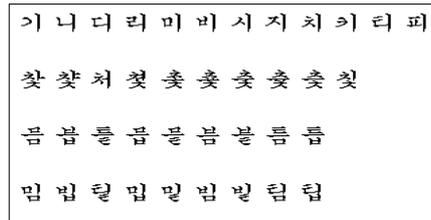


그림 3. 훈련 및 정확도 분석을 위한 음절 집합
Fig.3. Set of syllable for training and calculation on accuracy

표 2. 음절 집합에 대한 패턴 발생 빈도
Table 2. Pattern occurrence frequencies on the syllable set

유형	↑	↗	→	↘	↓	↙	←	↖
발생 빈도	18	62	145	45	110	89	33	30

3. 획 끝점 추출

입력 음절로부터 획을 분리시키기 위해 획의 끝점에 대한 검출 처리가 필요하다. 판서를 할 때 한 획에서 다음 획으로 바뀌는 과정에서 방향이

바뀌는 특징이 있다. 이러한 특징을 이용하여 순차적으로 입력되는 3개의 점을 이용하여 곡률을 구한 다음 문턱치 연산을 통해 획의 끝점 여부를 판단한다. 이때 한 가지 문제점은 입력 디바이스에서의 입력 신호는 판서를 할 때 입력 속도에 따라 점들의 간격이 달라진다. 예를 들면, 일반적으로 판서를 할 때 획의 끝점으로 갈수록 속도가 느려진다. 따라서 획의 끝점 부근에는 상대적으로 많은 수의 입력 점들이 존재한다. 이런 환경에서 3점의 곡률을 구하게 되면 문턱치 연산을 하는데 있어서 원하는 방향의 값을 얻지 못하는 경우들이 있다. 우리는 이런 문제점을 해결하기 위해 점들을 먼저 등 간격으로 배치한 후 곡률을 구한다.

3.1 입력 점에 대한 동일 간격 배치 처리

입력 점의 동일 간격 처리를 위해 순차적으로 입력되는 두 점의 유클리드 거리를 구한다. 그리고 [그림 4]처럼 두 점 사이의 간격에 따라 3가지 경우로 구분하여 처리한다.

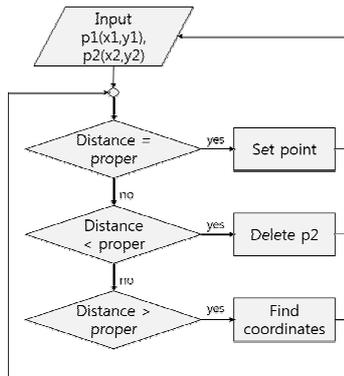


그림 4. 동일 간격 배치를 위한 처리 알고리즘
Fig.4. Flowchart on equal : arrangement of input sequences

첫 번째 분기는 두 점의 거리가 적정범위인 8~12 범위인 경우이며 입력 점을 사용한다. 두 번째 분기는 두 점 사이의 거리 값이 8미만일 때 두 점 중에서 끝점을 삭제하고 다음 입력 점과 이전 시작점 사이의 거리 값을 구하는 과정을 다시 시작한다. 세 번째 분기는 두 점 사이의 거리

가 적정범위보다 큰 경우이며 두 점 사이의 직선 방정식을 구한 후에 적정 범위 내부에 새로운 점을 추가한다.

3.2 곡률에 의한 끝점 추출

균일하게 배치한 입력 점들을 대상으로 3점의 곡률을 구한다. 3점 (x1,y1), (x2,y2), (x3,y3)에 식 (1)을 사용하여 곡률을 계산한다.

$$K = \frac{2*((p)*(t) - (s)*(q))}{\sqrt{((p)^2 + (s)^2)*(q)^2 + ((t)^2 + (h)^2)}} \quad (1)$$

단, $p = (x2 - x1), q = (x3 - x2),$
 $s = (y2 - y1), t = (y3 - y2),$
 $g = (x1 - x3), h = (y1 - y3)$

계산된 곡률을 기준으로 임계치 결정하여 끝점 여부를 판단한다. 임계치는 실험적으로 결정한다. 본 논문에서는 0.1로 결정하였다. [그림 5(a)]는 끝점 검출 결과이다.

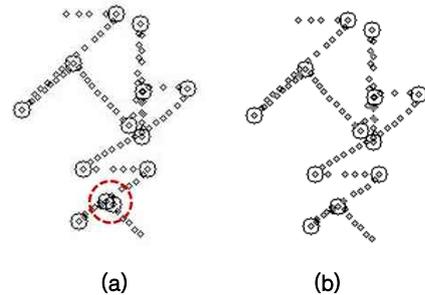


그림 5. 잘못 선택된 끝점을 포함하는 끝점 추출 결과
Fig.5. A result on end-points extraction of strokes with a false end-point

그런데 개인의 판서 성향에 따라 유사지점에서 연속으로 끝점이 검출되는 경우가 발생한다. [그림 5]에서는 점선 원형으로 표시한 부분에서 2개의 끝점이 추출된 것을 확인할 수 있다. 다음 획으로 이동할 때 곡선의 궤적으로 나타나며 이런 현상들이 종종 발생한다. 이런 경우 검출된 끝점 중에서 곡률이 최대인 것만을 남기고 나머지는 제

거하였다.

이런 문제를 해결하기 위해 연속으로 끝점이 검출되면 연달아 발견되면 그 중 곡률이 제일 높은 것만 추출하고 그렇지 않은 것은 끝점에서 제외시키는 방법으로 해결하였다. [그림 5(b)]는 불필요한 끝점을 제거한 결과이다.

4 획 유형 구분

검출된 끝점 정보를 바탕으로 획 단위로 분류를 수행한다. 예를 들어 ‘가’의 경우 끝점은 5개가 존재한다. 추출한 끝점 정보를 통해 획을 분리한 후, 획 단위로 분류를 수행한다. 최종적으로 분류한 획을 토대로 한글의 구조적 특징과 필기 순서 정보를 이용해 가상터치 방식의 가장 큰 문제점인 이동 획을 제거하여 사용자가 원하는 글자만 추출한다.

4.1 체인코드를 이용한 분류

입력된 한글 음절에서 끝 점 정보를 이용해 획들을 분리하고 분리된 획에 대해서는 주방향을 계산하여 정의된 획 패턴중의 한 가지로 분류한다. 획의 주방향을 계산하기 위해서는 획의 시작점과 마지막 점의 각도를 계산하여 미리 설정한 각도 범위 내에 속하는 부류에 매핑을 시킨다.

끝 점 정보를 이용하여 음절을 획 단위로 분리하고 각각의 획을 체인코드로 변환시킨 결과와 수작업으로 설정한 해당 음에 대한 체인코드를 비교하여 정확도를 측정하였다. 한글 ‘피’의 경우 글자 획, 이동 획을 코드로 표현하면 ‘264146214’이다. 실제로 프로그램을 구동시켜 ‘피’를 판서한 뒤 체인코드를 추출해 보니 체인코드 ‘264146204’이 추출되었다. 7번째 체인코드가 ‘1’이 아니라 ‘0’으로 인식하였다. 이유는 자음‘교’에서 모음‘ㅣ’으로 이동하는 획의 각도가 체인코드 0의 범위 내에 들어왔기 때문이다. 이런 문제점은 충분히 발생할 수 있다. 직관적으로 봤을 때 이 경우, 자음에서 모음으로 이동하는 획은 ‘↘’ 패턴으로 움직인다. 하지만 자음과 모음의 간격

이 많이 붙어 있는 경우라면 수치상으로 볼 때 ‘↑’패턴으로 인식할 가능성이 더 높기 때문이다.

4.2 이동 획 제거

획 단위로 인식을 하고 한글의 구조적 특징과 순서 정보를 이용한다면 이동 획은 쉽게 제거할 수 있다. [표 1]에서 확인할 수 있었듯이 글자 획과 이동 획의 패턴은 이미 구분되어 있다. 따라서 ‘→’, ‘↘’, ‘↓’ 패턴이 발생하면 이동 획이라 간주하고 제거를 한다. 그런데 ‘↙’의 경우는 이동 획과 글자 획에서 모두 존재한다. 이동 획이 발생하는 패턴을 살펴보면 전과 후로 반드시 글자 획이 존재한다. 따라서 ‘↙’ 패턴이 발생하면 전과 후에 모두 글자 획 패턴이 존재하는지 확인한 다음 모두 존재한다면 이동 획, 그렇지 않다면 글자 획으로 분류한다.

5. 실험 결과

5.1 끝 점 추출에 대한 정확도 분석

360개의 데이터를 수동으로 잡은 끝점과 곡률에 의해 자동으로 잡은 끝점의 일치여부를 가지고 정확도 측정을 하였다.

끝점 검출에 대한 정확도는 [표 3]에서 확인할 수 있다. [표 3]의 ‘case 1’은 수동으로 선택한 끝점과 자동으로 검출한 끝점이 일치하는 경우이다. ‘case 2’는 자동 검출 끝점이 수동 선택 끝점과 인접하는 경우이다. 자동 검출 끝점과 수동 선택 끝점이 허용 범위 내에 위치하는 경우이다. ‘case 3’은 실제 끝점이지만 자동으로 잡은 끝점이 존재하지 않는 경우이고 마지막으로 ‘case 4’는 실제 끝점이 아닌데 자동으로 잡은 끝점이 존재하는 경우이다. 총 360개의 글자 데이터 중 실제 끝점이 4293개 존재하고 자동으로 잡은 끝점이 일치하는 점은 case 2를 포함하여 4245개를 정확하게 찾았고 48개의 끝점은 찾지 못했다. ‘case 4’를 제외한다면 이는 98.8%의 정확성을 보여준다.

표 3. 끝점 정확도 분석

Table 3. Accuracy analysis of the endpoint

구분	count
case 1	4224
case 2	21
case 3	48
case 4	160

실제 끝점이지만 자동으로 끝점을 검출하지 못한 경우(case 3)는 [그림 6(a)]에서 확인 할 수 있다. 점선의 원으로 표시 된 것처럼 ‘口’을 쓸 때 오른쪽 위의 모서리부분을 흘려 쓰게 되어 곡선 형태의 획이 생성되었다. 따라서 이 부분은 곡률이 낮게 나오게 되어 문턱치 연산을 통해 자동으로 끝점을 검출하기가 어렵다. ‘case 3’의 경우를 분석한 결과 대부분이 유사한 이유로 인해 발생하는 것을 확인할 수 있었다. [그림 6(b)]는 끝점이 오검출된 경우(case 4)이다. 획의 굵기는 과정에서 순간적으로 어긋나게 입력하여 곡률이 높게 나타나고 있다. ‘case 4’의 대부분의 경우가 이런 이유에서 발생하였다.

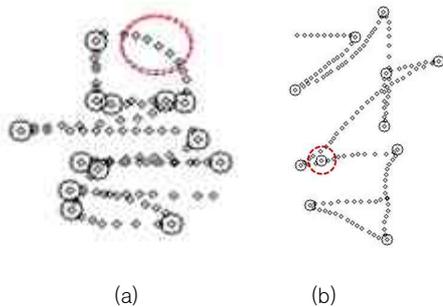


그림 6. ‘case 3, case 4’ 음절의 예
Fig.6. Syllable example on ‘case 3, case 4’

5.2 획 패턴 인식에 대한 정확도 분석

[그림 3]에 정의한 40개의 음절을 대상으로 수작업으로 획을 분리하여 획의 패턴을 정의하였다. 그리고 360개의 입력 데이터로부터 자동 추출한 방향 코드를 사용하여 정확도를 평가하였다. 분류 정확도는 [표 4]와 같다. 입력 데이터의 획 단위 패턴 구분의 정확도는 88.3%이고 이동 획과 글자 획에 대한 구분 정확도는 91.1%이다.

표 5. 획 패턴 및 획 유형 분류의 정확도

Table.5. Classification accuracy of stroke pattern and type

구분	정확도
획 패턴	88.3%
획 유형	91.1%

[그림 7]은 가상터치패널 환경에서 입력한 데이터에 대해 이동 획을 구분하여 제거한 후 글자 획에 해당하는 점들을 출력한 결과이다. 이동 획 제거 부분에서는 끝 점 추출이 얼마나 정확하게 선택되는지에 따라 입력 성능이 판가름된다.

음절	입력 궤적	결과 궤적
틀		
봄		
첫		
밑		
훑		

그림 7. 입력 궤적과 이동 획을 제거한 결과 궤적
Fig..] Some examples of input trajectories and result trajectories after removing move-stroke

6. 결론

카메라를 탑재한 스마트 전자칠판 시스템에서는 손 움직임을 통해 원거리에서 입력되는 궤적으로부터 판서가 가능하다. 본 논문에서는 3차원 공간에서 카메라를 통해 입력된 음절 궤적으로부터 이동 획은 제거하고 음절을 구성하는 글자 획만을 추출하여 글자 모양의 입력이 가능할 수 있는 기술을 제안하였다.

이를 위해 가상터치방식으로 한글을 판서하여 입력되는 음절에 대해 획의 끝 점을 찾아내고 이를 기반으로 음절을 획 단위 분리한 후 획의 방향 코드를 사용하여 이동 획과 글자 획을 구분할 수 있도록 하였다.

제안한 방법은 지능형 전자칠판 시스템을 구축하는 핵심 기술로 사용될 수 있을 뿐만 아니라 카메라 사용이 손쉬운 최근 환경에서 카메라를 부착한 데스크톱 컴퓨터를 비치한 공간이면 강의나 교육의 목적으로 활용할 수 있을 것으로 기대한다.

REFERENCES

- [1] Bai, Y., Park, S. Y., Kim, Y. S., Jeong, I. G., Ok, S. Y., & Lee, E. J. (2011). Hand Tracking and Hand Gesture Recognition for Human Computer Interaction. *Journal of Korea Multimedia Society*, 14(2), 182-193..
- [2] Heng-Kil Lee, "A study on the efficient syllable segmentation method for the korean language recognition", 한국과학기술원 석사학위논문, 1986.
- [3] Soohoo Kim, Sangberm Koh, Siyoung Hur, "A Study on the Syllable Recognition Using Neural Network Predictive HMM", *Journal of the Acoustical Society of Korea*, Vol.17 No.2, 1998.
- [4] Mingki Piao, Sekin Kim, Rongyi Cui, "Structure Based Modern Korean Character Set Partitioning and Pre-Classification Method of Korean Character Recognition", *Int'l Con. on Computer Science and Information Processing*, 2012.
- [5] Jae-Ook Kwon, Bong-Kee Sin, Jin-H. Kim, "Recognition of On-line Cursive Korean Characters Combining Statistical and Structural Methods", *Pattern recognition*, Vol.30 No.8, 1997.
- [6] Cheng-Lin Liu, In-Jung Kim, Jin H. Kim, "Model-based stroke extraction and matching for handwritten Chinese character recognition", *Pattern Recognition*, Vol.34, pp.2339-2352, 2001.
- [7] Yih-Ming Su, Jhing-Fa Wang, "A novel stroke extraction method for Chinese characters using Gabor Filters", *Pattern Recognition*, Vol.36, pp.635-647, 2003.
- [8] Sung-Young Kim, Jeong-Jin Cheon, "Strokes Separation from Hand Motion Trajectories", *4th Int'l Conf. on Circuits, Control, Communication, Electricity, Electronics, Energy, System, Signal and Simulation*, Vol.51, pp.64-67, 2014.
- [9] Pyeoung Kee Kim, Jong Kook Lee, Hang Joon Kim, "Handwritten Korean Character Recognition by Stroke Extraction and Representation", *IEEE TENCON*, 1993.
- [10] Ming Ma, Dong-Won Park, Soo-Kyung Kim, Syung-Og An, "Online Recognition of Handwritten Korean and English Characters", *Journal of Information Processing Systems*, Vol.8 No.4, pp.653-668, 2012.

저자약력

임 승 민 (Seung-Min Leem) [학생회원]



- 2015년 2월 : 금오공과대학교 컴퓨터학과 (공학사)
- 2015년 3월 ~ 현재 : 금오공과대학교 대학원 컴퓨터공학과 석사과정

<관심분야> 영상처리, 패턴인식

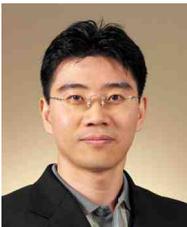
정 현 석 (Hyeon-Seok Jeong) [학생회원]



- 2011년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 학부과정

<관심분야> 영상처리, 정보통신

김 성 영 (Sung-Young Kim) [정회원]



- 2003년 8월 : 부산대학교 컴퓨터공학과 (공학박사)
- 2004년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 (교수)

<관심분야> 멀티미디어 정보검색, 패턴인식