

# 테트리스 게임을 위한 비전 기반의 인터페이스

김상호\*, 장재식, 김항준  
경북대학교 컴퓨터공학과 인공지능연구실  
(shkim\*, jschang, kimhj)@ailab.knu.ac.kr

## Vision-based Interface for Tetris Game

Sang Ho Kim\*, Jae Sik Chang, Hang Joon Kim  
Dept. of Computer Engineering, Kyungpook National University

### 요 약

본 논문에서는 테트리스 게임을 위한 비전 기반의 인터페이스를 제안하고 있다. 제안된 인터페이스는 카메라로부터 실시간으로 입력되는 연속 영상에서 손의 제스처를 인식하고, 인식된 제스처를 게임의 명령으로 사용한다. 테트리스 게임에 필요한 6개의 명령은 손의 자세로 정의되는 세 종류의 정적 제스처와 손의 자세와 움직임으로 정의되는 세 종류의 동적 제스처로 정의된다. 손의 자세는 손 영역의 불변 모멘트로 표현하였고, 입력된 손 영역의 자세는 미리 학습된 불변 모멘트 값들과의 거리차이를 비교하여 분류한다. 실험 결과에서 제안된 시스템이 실시간 테트리스 게임의 인터페이스로 적용가능함을 보였다.

### 1. 서론

컴퓨터를 기반으로 하는 많은 기술들이 발전함에 따라 사람들은 점차 키보드나 마우스 또는 조이스틱과 같은 장치들을 직접 다루는 데에서 벗어나 좀 더 자유롭고 편리한 HCI를 요구하게 되었다. 또한 사용자가 게임의 가상현실 속에서 자연스럽게 게임으로 게임을 진행할 수 있는 새로운 게임 인터페이스의 개발이 요구되고 있다. 이를 동기로 비전 정보에 기반한 손 또는 팔의 제스처(gesture)에 대한 관심이 증가하고 있다[1-4].

제스처는 물체를 조작하거나 지시하는데 있어서 가장 직관적인 방법으로 사용된다. 이미 2D/3D 마우스, TV 제어, 그리고 컴퓨터 게임과 같은 응용을 통해 제스처는 실제 세계에서 컴퓨터와 인간의 상호작용(interaction)을 위한 직관적이며 효율적인 수단으로 사용될 수 있는 가능성을 보여주고 있다[1-4].

손 제스처는 보통 손의 자세(posture) 즉, 공간적인 정보만을 사용하는 정적(static) 제스처와 움직임 즉, 시간적인 정보를 사용하는 동적(dynamic) 제스처로 나눌 수 있다. 또한 일반적으로 처리하는 정보의 입력 방식에 따라 글러브 기반의 방법과 컴퓨터 비전을 기반으로 하는 방법으로 나눌 수 있다[1-2].

본 논문에서는 손의 제스처를 이용하여 테트리스(Tetris) 게임을 위한 비전 기반의 인터페이스를 제안하고 있다. 제안하는 방법은 정규화된 CrCg 색상 공간에 정의한 피부색 모델을 이용하여 카메라 영상에서 손 영역을 추출한다. 추출된 손 영역에서 손의 자세는 불변 모멘트로 표현하였고, 미리 학습된 불변 모멘트 값들과의 거리차이를 비교하여 분류된다. 분류된 손의 자세와 움직임 정보를 보고 제스처를 인식하였으며, 이 제스처를 테트리스 게임의 명령으로 사용하였다.

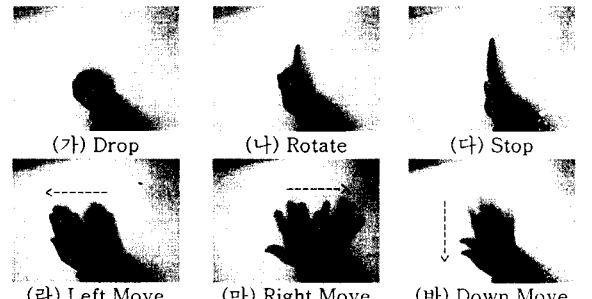
### 2. 시스템 개요

제안된 비전 기반의 인터페이스는 그림 1에서 보듯이 기본적으로 다섯 종류의 손의 자세를 정의하였다. 분류된 손의 자세와 움직임 정보를 보고 제스처를 인식하였으며, 이 제스처로

테트리스 게임을 조절하였다. 이를 위해 그림 2에서 처럼 각각 세 종류의 정적 제스처와 동적 제스처를 정의하여 사용하였다.



그림 1. 정의한 다섯 종류의 손의 자세 :  
Drop, Move1, Move2, Rotate, Stop.



(가) Drop (나) Rotate (다) Stop  
(라) Left Move (마) Right Move (바) Down Move  
그림 2. 사용된 세 종류의 정적 제스처((가),(나),(다))와 동적 제스처((라),(마),(바)).

테트리스 게임은 각기 다른 모양으로 붙어 있는 4 개의 블록을 차곡차곡 쌓아서 한 줄 이상을 채우면 채워진 줄이 없어지는 퍼즐형 게임이다. 1985 년 모스크바의 알렉시 파지노프(Alexey Pajitnov)에 의해 처음으로 개발되었고, 누구나 쉽게 할 수 있어 오랜 역사에도 불구하고 많은 사람들이 즐기는 게임 중의 하나이다.

이 게임을 진행하기 위해서는 총 여섯 종류의 명령이 필요하다; 블록의 움직임에 관련된 네 종류의 명령, 블록의 회전에 관련된 한 종류의 명령, 게임의 진행을 잠시 멈추는 한 종류의 명령. 각 명령에 관련된 조작키 및 제스처는 표 1 과 같다.

표 1. 데트리스 게임의 명령, 조작키 및 제스처

분류	명령	조작키	제스처
블록의 움직임 관련	왼쪽으로 이동	←	Left Move
	오른쪽으로 이동	→	Right Move
	아래로 이동	↓	Down Move
	한 번에 가장 아래로 이동	Space bar	Drop
블록의 회전 관련	반시계 방향으로 90도 회전	↑	Rotate
게임진행 관련	잠시 멈춤	Enter	Stop

제안된 인터페이스는 그림 3과 같이 책상 위에서 움직이는 손을 카메라가 위에서 아래로 보고 있는 환경에서 구현되었다.



그림 3. 시스템 환경

3. 손 영역 검출

본 장에서는 컬러 영상에서 손 영역을 검출하는 방법에 대해 설명한다. 제안하는 방법은 카메라로부터 입력되는 컬러 영상을 CrCb 색상 공간에서 가우시안 분포로 정의된 피부색 모델을 이용하여 피부색 영역과 배경 영역으로 이진화하였다. 이 이진 영상에서 모폴로지(morphology) 연산인 클로징 연산(closing operation)을 사용하여 잡음을 제거하고 연결성분(connected component)을 분석하여 박스리스트(boxlist)데이터를 만든다. 박스리스트데이터 중 일정 크기 이상의 가장 큰 박스를 손이라 생각하고 이진 영상에서 박스 친 손 영역을 검출해 낸다. 손 영역을 추출하는 과정의 예는 그림 4에서 보여 주고 있다.

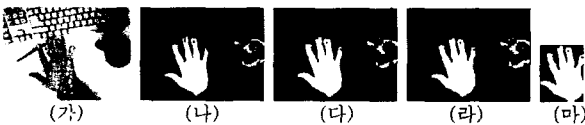


그림 4. (가) 컬러 영상 (나)살색 잦은 이진영상 (다) 잡음 제거한 영상 (라) 손 영역을 박스친 영상 (마) 손 영역 검출 영상

4. 특징 추출 및 손의 자세 분류

4.1 특징 추출

본 논문에서는 손의 자세의 특징으로 불변 모멘트를 사용하였다. 1962년 Hu[5]가 최초로 불변 모멘트를 만들었으며, 객체의 모양을 기술하기 위해 객체의 모멘트로부터 구하여진 7개의 변수를 사용하였다. 불변 모멘트는 천이, 회전, 크기 변화에 불변한 성질을 가지기 때문에 제안된 인터페이스에서 손의 자세를 표현하는데 적절하다.

2차원 연속 함수  $f(x,y)$ 에 대해, 차수  $(p+q)$ 를 가지는 모멘트는 다음과 같이 정의된다.

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad (1)$$

여기서  $p, q = 0, 1, 2, \dots$  이다. 디지털 영상에서, 식 (1)은

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad (2)$$

로 된다. 객체의 무게 중심은

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{와} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

이다. 중앙 모멘트는 다음 식 (3)으로 표현된다.

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \quad (3)$$

정규화된 중앙 모멘트는  $\eta_{pq}$ 로 표기하며,

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma} \quad (4)$$

로 정의된다. 여기서  $\gamma = (p+q)/2 + 1$ 이고,  $p+q = 2, 3, \dots$  이다.

7개의 불변 모멘트의 집합은 다음 식 (5)로 구할 수 있다.

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02}, & \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} + \eta_{21})^2, & \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \\ \phi_5 &= (3\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{21} - \eta_{03})[3(\eta_{03} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (5)$$

이 7개의 불변 모멘트는 손의 자세를 기술해주며, 천이, 회전, 크기 변화에 불변이다.

4.2 손의 자세 분류

현재의 영상에서 추출한 손의 자세에 대한 불변 모멘트와 미리 정의된 다섯 종류의 손의 자세에 대한 불변 모멘트의 거리 차이를 보고 손의 자세를 분류한다. 제안된 방법으로는 유클리디언 디스턴스(Euclidean Distance)를 측정하여 최소 거리차를 출력해 주는 손의 자세를 분류된 손의 자세로 결정하였다. 미리 정의된 각각의 손의 자세에 대한 불변 모멘트는 각각 손의 자세를 취한 100장의 영상에서 추출해낸 불변 모멘트들의 평균값이다. 사용된 영상들의 예는 그림 5와 같다. 이 영상들에서 추출해낸 불변 모멘트들의 평균값은 그림 6의 (가)와 같다.



그림 5. 각 손의 자세의 학습에 사용된 영상들

이 논문에서는 손의 자세를 보다 정확하게 분류하기 위해서 식 6과 같이 가중치가 부여된 불변 모멘트( $\phi_n$ )를 사용하였다.

$$\phi_n = \omega_n \times \phi_n \quad (6)$$

$n$ 은 1,2,...,7이고,  $\phi_n$ 은 각 손의 자세의 학습에 사용된 영상에서 추출한 불변 모멘트의 평균값이다.  $\omega_n$ 은 가중치이며 각각의 가중치 값은 1, 10, 100, 100, 1, 1, 1 이다. 그림 6의 (나)는

가중치가 부여된 평균 불변 모멘트를 보여주고 있다. 가중치는 실험적으로 구했으며, 가중치가 부여된 7개의 불변 모멘트 중 앞의 네 개만 사용하였다.

제스처 인식 능력을 측정하기 위한 실험의 결과를 보여주고 있다. 이 실험에서 제스처 인식률은 가중치를 부여한 불변 모멘트를 사용한 손의 자세 분류율과 비슷함을 알 수 있다.

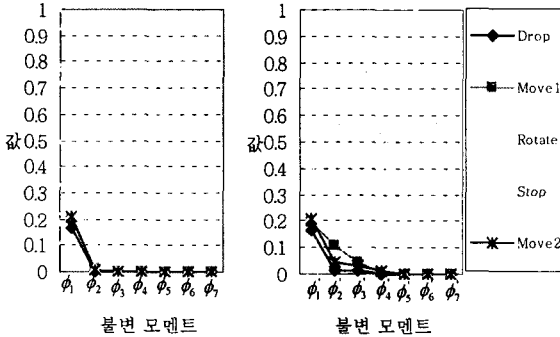


그림 6. (가) 각각의 손의 자세에 대한 평균 불변 모멘트  
(나) 가중치가 부여된 평균 불변 모멘트

5. 제스처 인식

제스처 인식은 카메라로부터 획득된 현재의 컬러 영상을 6개의 제스처 중 하나로 결정하기 위해 분류된 손의 자세와 손 중심의 움직임 정보를 사용한다. 분류된 손의 자세가 'Drop', 'Rotate', 'Stop'이면 현재의 영상을 각각 정적 제스처인 [Drop], [Rotate], [Stop]로 인식한다. 분류된 손의 자세가 'Move1'이나 'Move2'면 현재의 영상에서의 손의 중심과 그 전 영상에서의 손의 중심의 거리차와 이동 방향을 보고 동적 제스처로 인식한다. 손이 두 픽셀 이상의 거리로 왼쪽으로 움직였다면 [Left Move]로, 오른쪽으로 움직였다면 [Right Move]로, 아래로 움직였다면 [Down Move]로 각각 인식한다.

6. 실험 및 결과

제안된 인터페이스는 Pentium 4 1.8 GHz PC, Window XP 시스템에서 Visual C++ 6.0과 Open CV를 사용하여 구현되었다. 저가형 캠(cam)을 사용하여 초당 10프레임으로 320x240 크기의 컬러 영상을 처리하였다. 제안된 방법으로 인식된 제스처는 그림 7과 같이 테트리스 게임을 조절한다.

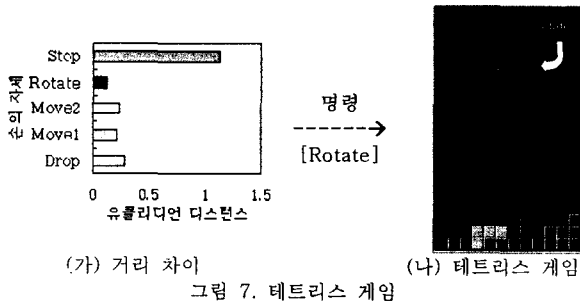


그림 7. (가) 거리 차이 (나) 테트리스 게임

표 2는 손의 자세 분류 능력을 측정하기 위한 실험의 결과를 보여주고 있다. 이 실험에서 가중치를 부여한 불변 모멘트를 사용하였을 경우 보다 잘 분류한다는 것을 알 수 있다. 표 3은

표 2. 손의 자세 분류율

손의 자세	영상수	가중치를 부여하지 않은 불변 모멘트 사용시	가중치를 부여한 불변 모멘트 사용시
		분류율(%)	분류율(%)
Drop	54	83.33	98.15
Move1	45	86.67	97.78
Move2	48	83.33	98.92
Rotate	52	76.92	94.23
Stop	46	84.78	93.48
Total	245	82.86	96.33

표 3. 제스처 인식률

제스처 종류	영상수	인식률(%)
Drop	53	98.11
Rotate	49	95.92
Stop	52	94.23
Left Move	52	98.08
Right Move	54	96.30
Down Move	55	98.18
Total	315	96.83

7. 결론

본 논문에서는 카메라로부터 실시간으로 입력되는 영상에서 손 제스처를 인식하여 테트리스 게임을 진행하는 인터페이스를 제안하였다. 테트리스 게임을 조절하기 위해서 여섯 종류의 제스처를 정의하였으며, 실험을 통해 빠르고 정확하게 게임을 할 수 있음을 보였다. 또한 제안된 방법으로 손 영역을 추출하고 손의 자세를 가중치가 부여된 불변 모멘트로 기술함으로써 보다 효과적으로 손의 자세를 분류할 수 있었다.

8. 참고 문헌

- [1] Oh-Young Cho, Hyoung-Gon Kim, Sung-Jea Ko, Sang Chu, Ahn, "A Hand Gesture Recognition System for Interactive Virtual Environment", *Journal of the Institute of Electronics Engineers of Korea*, Vol 36, No 4, pp 70-82, 1999.
- [2] William T. Freeman and Michal Roth, "Orientation Histograms for Hand Gesture Recognition", *IEEE Int'l Workshop on Automatic Face and Gesture Recognition*, pp 296-301, 1995.
- [3] William T. Freeman, David B. Anderson, Paul A. Beardsley, Chris N. Dodge, Michal Roth, Craig D. Weissman, William S. Yerazunis, Hiroshi Kage, Kazuo Kyuma, Yasunari Miyake, Ken-ichi Tanaka, "Computer Vision for Interactive Computer Graphics", *IEEE Computer Graphics I/O Devices*, pp 42-51, 1998
- [4] Jakub Segen, Senthil Kumar, "Human-Computer Interaction using Gesture Recognition and 3D Hand Tracking", *IEEE International Conference on Image Processing*, pp 188-192, 1998
- [5] Hu, M.K., "Visual pattern recognition by moment invariants", *IEEE Transactions on Information Theory*, IT-8, pp 179-187, 1962.