
지역 인테그럴 히스토그램을 사용한 빠르고 강건한 전경 추출 방법

↓

Fast foreground extraction with local Integral Histogram

↓

장동헌, DongHeon Jang*, 김향화, XiangHua Jin**, 김태용, TaeYong Kim***

↓

요약 ~ 본 논문에서는 비전 기반 게임 인터페이스를 위한 배경영역으로부터 전경영역을 추출하기 위해 빠르고 강건한 새로운 방법을 소개한다. Background Subtraction 방법은 추적하고자 하는 이미지의 특징을 추출하기 전에 필수적으로 거쳐야 하는 전처리 과정이다. 이를 위해 본 논문에서는 이미지를 지역 셀로 나누어 가우시안 커널이 적용된 Local Histogram을 계산하고 히스토그램의 Bhattacharyya 거리를 계산하여 전경확률을 결정한다. 이처럼 지역적 히스토그램에 기반한 방법은 급격한 조명변화나 잡음 또는 작은 배경오브젝트의 움직임에 부분적으로 강간함을 보인다. 히스토그램을 계산하는데에서 Multi-Scaled Integral Histogram을 사용하여 잡음을 억제하면서 계산의 속도를 높였다.

↓

Abstract ~ We present a new method of extracting foreground object from background image for vision-based game interface. Background Subtraction is an important preprocessing step for extracting the features of tracking objects. The image is divided into the cells where the Local Histogram with Gaussian kernel is computed and compared with the corresponding one using Bhattacharyya distance measure. The histogram-based method is partially robust against illumination change, noise and small moving objects in background. We propose a Multi-Scaled Integral Histogram approach for noise suppression and fast computation.

↓

핵심어: 전경추출; 히스토그램; 게임인터페이스

↓

↓

본 연구는 ITRC(Information Technology Research Center)와 서울시 산학연 협력사업의 지원으로 수행되었음.

*주저자 : 중앙대학교 첨단영상대학원 영상공학과 석사과정

**공동저자 : 중앙대학교 첨단영상대학원 영상공학과 석사과정

***교신저자 : 중앙대학교 첨단영상대학원 교수; e-mail: kimty@cau.ac.kr

1. 서론

최근의 비디오 게임 환경은 모션센서와 같은 입력 장치를 채택함으로써 게임에 더욱 몰입할 수 있는 경험을 유저들에게 제공하고 있다. 이에 비전기반 게임 인터페이스는 카메라에서 얻어진 한 장의 이미지로 처리되므로 데이터 글로벌과 기타 장비를 유저가 몸에 걸치지 않아도 되며 다양한 형태로 응용 가능하다는 장점이 있다. 유저의 포즈를 인식하기 위해 전경 영역을 추출하는 전처리 과정인 Background subtraction 은 이전에도 여러 가지 방법으로 시도된 적이 있다. Color intensity 임계치값을 이용한 방법이 가장 기초적인 방법이며[7] 주변픽셀간의 관계를 고려하지 않고 단일 픽셀과의 차를 구하므로 잡음이나 그림자, 조명의 변화, 배경의 갱신 등을 고려하지 않은 문제가 있다. [8]에서는 배경의 변화에 대해 적응적 계수를 두었으나 갑작스런 조명의 변화에 민감하다. 배경의 다양성을 모델링하기 위해 [9]에서는 각 픽셀을 여러 개의 가우시안으로 모델링 하였으며 각 가우시안은 여러 프레임간의 픽셀변화를 국부적으로 수용하도록 수렴해간다. 이러한 방법은 조명의 변화나 노이즈에 강건함을 보이지만 각 가우시안이 수렴하기 위해서 일정시간 동안 초기화 과정이 필요하다. 최근에는 Optical flow 나 Color co-occurrence 와 같은 방법들이 소개되었으나 이들은 실시간 처리에 적합하지 못하다.[11,12]

이처럼 Background subtraction은 전경영역 추출을 위해 가장 직관적인 방법이나 정확한 전경영역을 분리해내기 위해서는 카메라 내부 잡음과 외부 잡음, 조명의 변화, 전경물체의 그림자 혹은 배경 물체의 작은 움직임과 같은 모든 환경을 고려해야 한다. 그러므로 단순히 배경과 전경의 픽셀간의 차분을 구하는 것보다 주변영역의 상관관계 혹은 컬러 정보들을 고려하는 것이 안정적이고 강건한 차영상을 구할 수 있다. 컬러 히스토그램은 이미지의 일정 영역 안에서 컬러들의 분포를 표현한다. 히스토그램을 이용하여 이미지를 검색하거나 오브젝트를 트래킹하는 방법은 여러 논문에서 다양한 방법으로 제안되었다. [1,13] 본 논문에서는 전경 이미지와 배경이미지를 지역 셀로 나뉜 로컬 히스토그램을 각각 계산하여 그 차분을 전경영역 확률로 계산한다. 이처럼 분포간의 차분 혹은 거리를 구하는 것은 복합적인 정보를 이용하므로 잡음이나 배경물체의 흔들림에 강건함을 보인다. 하지만 일정 영역안의 분포를 계산하므로 픽셀단위보다는 해상도가 떨어지게 되나 이는 게임에서 처리하기 적당한 수준의 대강의 포즈를 인식하는데 적당하다. 본래 일정 영역안의 픽셀들의 컬러 히스토그램을 계산하기 위해서는 각 픽셀이 히스토그램의 어떤 bin에 속하는지 quantization의 과정이 필요하며 이때 조명에 의한 픽셀의 작은 컬러 변화가 quantization error를 발생시키는데 본 논문에서는 이를 극복하기 위해 [3, 4]에서 사용되었듯이 한 픽셀의 히스토그램 bin 카운터를 증가 시킬 때 가우시안 커널을 적용하여 여러

bin에 가우시안 분포적으로 영향을 줌으로써 이러한 에러를 최소화 하였다. 하지만 수십 개의 히스토그램을 매 프레임당 실시간으로 구하는 과정은 많은 계산 시간을 필요로 한다. 이러한 문제를 극복하기 위해 인테그럴 히스토그램을 사용한다. 인테그럴 히스토그램은 한번의 스캔을 거쳐 해당 픽셀 위치에 누적된 히스토그램 정보를 저장하여 메모리 소모는 많지만 어떤 크기의 영역이던지 상수시간안에 똑같은 결과를 얻는 것이 가능하다.[2] 여기서 상수시간은 히스토그램의 bin의 개수에 비례한다. 또한 인테그럴 히스토그램을 활용하여 Multi-scaled 접근을 통해 노이즈나 움직이는 오브젝트에 대한 반응을 억제하고 속도를 높이는 방법을 제안한다. Multi scale로 이미지의 전체적인 크기에서부터 작은 크기로 히스토그램 차분을 구해나감으로써 필요하지 않은 세부 부분은 큰 영역에서 걸러지고 이는 결과적으로 지역적인 노이즈를 억제하고 계산 속도를 높인다. 재귀적으로 다양한 scale 영역의 히스토그램의 크기를 비교하는 계산 또한 인테그럴 히스토그램으로 인해 별다른 부하 없이 처리가 가능하다. 최종적으로 추출된 전경영역들의 셀들은 정해진 포즈와 매칭 시키기 위해 미리 훈련된 뉴럴 네트워크의 입력으로 연결되며 뉴럴 네트워크의 출력은 실제 게임에서 캐릭터의 방향을 조절하는 값으로 매핑 되었다.



그림 1. 제안하는 비전 게임 인터페이스의 계산 과정과 그 결과물.

전체적인 처리과정과 각 처리과정에 해당하는 결과물은

그림 1에 나타나 있으며 결과적으로 3D 공간의 입력을 카메라 입력으로 제어함으로 제안된 방법의 우수성을 보인다.

2. 강건하고 빠른 히스토그램 계산

2.1 로컬 커널 히스토그램

지역 히스토그램을 계산하는데 있어서 본 논문에서는 HSV 컬러 공간에서 Hue값만을 사용하는데 이는 히스토그램 차원을 줄이는 동시에 조명의 변화에 대해 강건해지기 위함이다.[5] Hue값은 HSV 컬러 공간의 정의에 따라 밝기값이 낮을 때 정의가 되지 않으므로 다음과 같은 임계값을 두어 픽셀의 값이 조건을 만족 할 때만 히스토그램 binning에 참가하도록 한다.

$$h(x_k) \begin{cases} h_j, & \text{if } (V(x_k) > T_v) \\ h_{undefined}, & \text{if } (V(x_k) \leq T_v) \end{cases} \quad (1)$$

한 픽셀 x_k 에 대한 Histogram bin을 하나로 제한하는 것은 양자화 에러에 대한 주된 이유이다. 본 논문에서는 다음과 같은 1D Gaussian kernel 에 의해 히스토그램 bin을 여러 bin에 분산시킨다. 이는 서론에서도 소개했듯이 quantization 에러에 대해 강건해 지는 영향이 있다.

$$d_h = H_k - H_j \quad (2)$$

$$G_j(H_k) = \frac{1}{2\pi\sigma_c} \exp\left(-\frac{d_h^2}{2\sigma_c^2}\right)$$

n_u 를 (1)식에서 선택되지 않은 픽셀의 개수라 할 때 $n \times n$ 로컬 영역의 각 히스토그램 값은 다음과 같은 수식에서 얻을 수 있다.

$$h_j = \sum_{k=1}^{n^2 - n_u} G_j(H_k) \quad (3)$$

위와 같은 히스토그램을 전경영역과 배경영역에 대해 모두 계산하고 지역 l 에서의 전경영역을 계산하기 위해 전경영역 히스토그램 H_l^f 과 H_l^b 의 Bhattacharyya 거리를 다음과 같이 계산한다.

$$P_l = \sum_{j=1}^b \sqrt{h_j^f h_j^b} \quad (4)$$

이 거리는 곧 l 영역이 전경영역일 확률이 되며 이 값은 0~1사이 값으로 정규화 된 값이다.

2.2 인테그럴 히스토그램

인테그럴 히스토그램[2]은 에서 소개된 강건하고 빠른 얼굴 검출에서 사용된 인테그럴 이미지[6]의 확장이다. 각 픽셀 위치에서 (0,0)에서 (x1,x2)에 이르기까지의 누적 히스토그램을 (x1,x2)에 저장한다. 이러한 wavefront scan은 그림 2에서 보여지듯이 주변에 계산된 3곳의 히스토그램값과 현재 픽셀의 히스토그램 bin 값을 더함으로써 빠르게 계산할 수 있으며 $H(x1,x2)$ 는 다음과 같은 식에 의해 계산된다.

$$H(x1,x2) = H(x1-1,x2) + H(x1,x2-1) - H(x1-1,x2-1) + Q(I(x1,x2)) \quad (5)$$

여기서 Q 는 가우시안 커널을 적용한 quantization 함수이다. 한번의 wavefront 스캔을 거친 후 왼쪽위, 오른쪽위, 왼쪽아래, 오른쪽 아래로의 4점으로 정의되는 임의의 사각형 영역 히스토그램은 다음 식과 같이 각 모서리에 저장되어 있는 누 히스토그램값을 더하고 뺄으로써 상수시간 안에 계산할 수 있다.

$$H_l = H(\text{right}, \text{bottom}) - H(\text{right}, \text{top}) - H(\text{left}, \text{bottom}) + H(\text{left}, \text{top}) \quad (6)$$

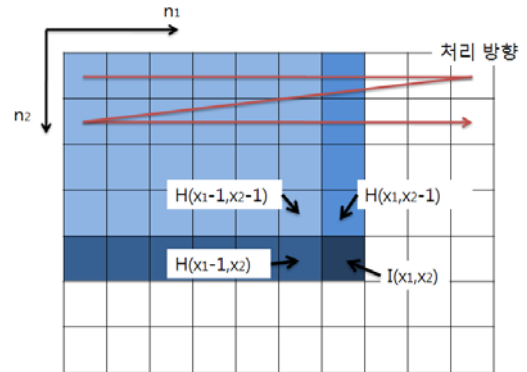


그림 2. 인테그럴 히스토그램을 계산하기 위한 스캔 전처리 과정

스캐닝시 각 픽셀의 위치에는 하나의 누적된 히스토그램 정보를 담고 있으므로 이미지를 $N \times N$ 이고 B 개의 히스토그램을 가지며 한 개의 bin 의 범위가 0~255라 할 때 인테그럴 히스토그램의 소비되는 메모리 총 사용량 M 은 다음과 같다.

$$M = N \times N \times B(\text{byte}) \quad (7)$$

본 논문에서 사용된 320x240 의 크기에 quantization 크기를 8개로 했을때 사용되는 메모리는 612KB에 달하며 이는 최근 하드웨어를 고려해 볼때 충분히 감당할 만한 수준이다.

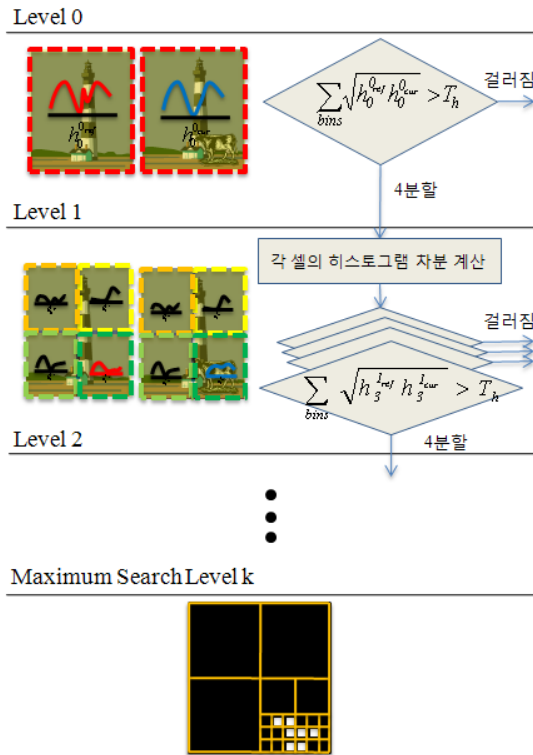


그림 3. Multi-scaled 히스토그램 차분 알고리즘

3. Multi-Scaled Background Subtraction

이미지를 M개의 column 과 N개의 row로 나눌 때 $M \times N$ 개의 셀들의 히스토그램을 계산하여야 한다. Multi-Scale Histogram Intersection Representation 은 한 이미지를 스케일이 다른 여러 윈도우의 히스토그램으로 표현하여 이미지 검색을 성공적으로 수행하였다.[13] 다시 말하면 국부적인 히스토그램과 그 영역을 포함하는 더 큰 스케일의 히스토그램, 혹은 전역적인 히스토그램들을 복합적

으로 구성하여 하나의 이미지의 특징을 나타내었다. 그와 비슷하게 본 논문은 국부적인 영역에서의 컬러 변화가 전역적인 히스토그램에 미치는 영향을 고려하는 Multi-scaled 차영상 방법을 히스토그램을 사용하여 적용한다. 이는 2절에서 설명하였듯이 이미지 전체 영역 Scale에서 시작하여, 즉 전체 영역의 히스토그램 차이를 구한 후에 그 차이가 임계값을 넘었을 경우 해당 영역을 4분할 하여 더 높은 스케일의 히스토그램 즉 더 잘게 나뉜 영역의 히스토그램의 차이를 재귀적으로 비교하는 방법이다. 4분할 되는 각 국부적인 영역을 본문에서는 셀(Cell)이라 정의한다. 재귀적인 검사이므로 Maximum search level k를 둬서 얼마나 작은 셀로 나뉘어 검사할 것인지 결정한다. 이때 스케일에 따른 히스토그램 영역의 크기는 인테그럴 히스토그램에 따라 계산 시간은 상수시간이다. 그림 3은 제안된 Multi-scaled 히스토그램 차분방식을 도식화 하여 보여준다. 그림에서 전경 이미지에서 우측 하단에 전경 오브젝트가 위치할 경우 Maximum Search Level 의 결과에서 보듯이 흰색으로 검출된 부분이 원하는 전경 오브젝트이다. 이는 셀 대 셀 계산으로도 같은 결과가 도출될 수 있으나 결과에서 보듯이 전역적인 히스토그램 변화에서부터 문턱치 값을 적용하므로 Level 1에서 전경 오브젝트가 위치하지 않은 3개의 셀들은 미리 걸러지게 되어 더 이상 3개의 셀을 4분할하지 않으므로 계산 시간이 단축되며 그 영역에서의 잠재적인 잡음이 억제할 수 있다.

4. 실험

본 논문에서 제안하는 비전 인터페이스를 구현하기 위한 가우시안 커널이 적용된 인테그럴 히스토그램을 이용하여 Multi-scaled 히스토그램 차분방법을 실제 실험을 통하여 그 유용성을 보인다. 제안된 방법은 OpenCV[15] 라이브러리를 사용하여 구현되었으며 기존에 라이브러리에서 제공하는 영상처리 기법들에 2,3절에서 제안된 방법들을 추가적으로 구현 하였다. 카메라는 Point grey research lab에서 제공하는Dragonfly[16]를 사용하였으며 이는 320x240 크기의 RGB 3채널의 컬러 이미지를 60 fps 로 제공해준다. 테스트 PC는 AMD 3200+, 2 GB RAM의 사양을 갖추었으며 카메라와 Firewire 로 연결되어 있다.

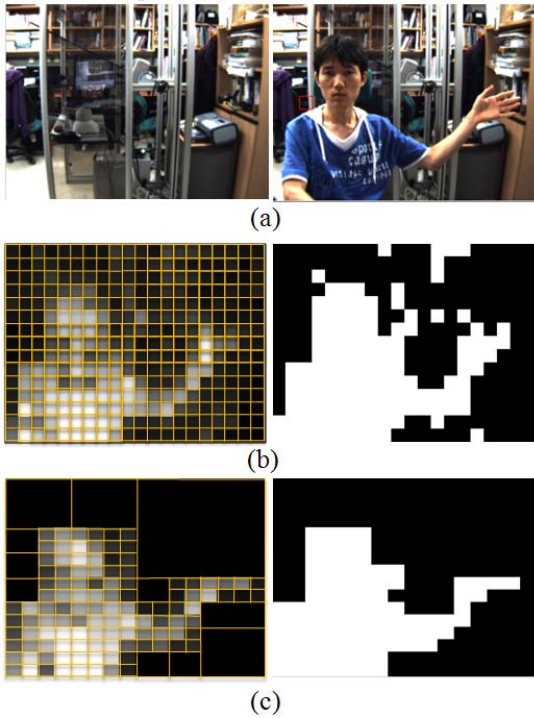


그림 4. (a) 배경과 전경 이미지 (b) $n=16, d=2$ 일 때 셀대 셀 차분결과 (c) $k=4, Th=0.03$ 일 때 Multi-scaled 차분 결과

320x240 이미지의 경우 $M = N = 2^k$ 가 되며 k 가 4 일 때 20x15 크기의 윈도우로 16x16 개의 셀로 나눌 수 있다. 픽셀대 픽셀간 계산을 하듯이 셀대셀 계산을 할 경우 256번의 히스토그램 차분을 계산하여야 하지만 그림 4에서 보여지는 것과 같이 Multi-Scaled 접근방법을 통해 배경영역에 대해 변화된 부분만을 재귀적으로 검사할 수 있게 되며 이것은 3절에서 설명되었듯이 셀대셀 차분에서 잠재적으로 발생될 수 있는 지역적 잡음을 억제하는 결과를 얻었다.

이렇게 얻어진 각 셀들의 전경 확률 값은 미리 정의된 포즈를 분류하기 위해 뉴럴 네트워크의 입력으로 전달된다. 제안된 4개의 포즈와 뉴럴 네트워크의 구조는 그림 5와 같다. 각 포즈를 훈련시키기 위한 트레이닝 데이터는 손으로 구성된 Ground truth 데이터에 랜덤 가우시안 노이즈 함수를 적절히 적용하여 만들어 내었으며 각 포즈당 50개를 두어 구성된 뉴럴 네트워크를 오류 역전파 방법으로 학습시켰다.

뉴럴 네트워크의 출력은 곧바로 그림 6에서 보이듯이 Torque 3D로[14] 구현된 비행 시뮬레이션의 입력으로 전달되었으며 1인치 카메라의 Pitch와 Yaw를 제어하는 게임 인터페이스로 매칭시켰다. 구현된 최종 비전기반 인터페이스와 게임은 기존의 키보드 및 마우스로 제어하는 것과 같은 결과를 보였다.

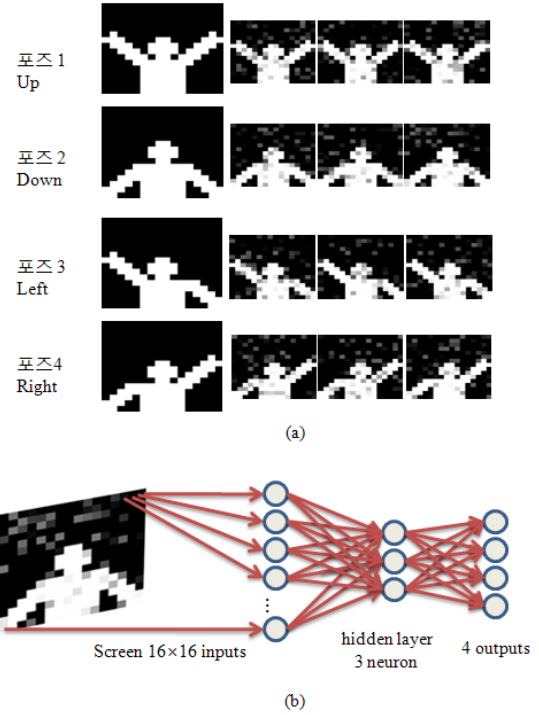


그림 5. (a) 제안된 4가지의 포즈와 그 트레이닝 데이터 (b) 인식을 위한 뉴럴 네트워크 구조

5. 결론

본 논문에서는 대강(Coarse)의 전경 오브젝트를 추출하기 위해 이미지를 일정 크기의 영역으로 나누었다. 배경과 전경의 각 영역의 차분을 구하기 위해 히스토그램 방법을 사용하였으며 히스토그램 quantization 에러를 최소화하기 위해 가우시안 커널을 적용하였다. 빠른 히스토그램 계산을 위해 인테그럴 히스토그램 방법을 도입하였으며 이것은 또한 Multi-scaled 히스토그램 차분을 실시간으로 계산 가능하게 해주었다. Multi-scaled 방법으로 국부적인 잡음 혹은 오류를 억제하였으며 이는 배경오브젝트의 작은 움직임과 같은 주변 환경의 변화에 강건함을 보였다.

제안된 방법에서 히스토그램 계산에 참여할지를 결정하는 임계치 값과 Multi-scaled 히스토그램 차분방식에서 더 높은 Scale로 검사할지 결정하는 임계치가 사용되었다. 앞으로의 연구에서는 이러한 고정값을 어떻게 적응적으로 결정할지에 대해 주안점을 두고자 한다. 또한 HSV 컬러 영역에서 Hue 영역에 대한 히스토그램은 부분적으로 조명에 강건함을 보이지만 역시 갑작스런 조명의 변화에서는 어느 정도 잡음이나 오인식이 발생하였다. 이는 컬러 정보를 더 활용하거나 경계선에 대한 히스토그램을 구하는 방식으로 해결되어 질 수 있으며 현재 연구가 진행 중이다.

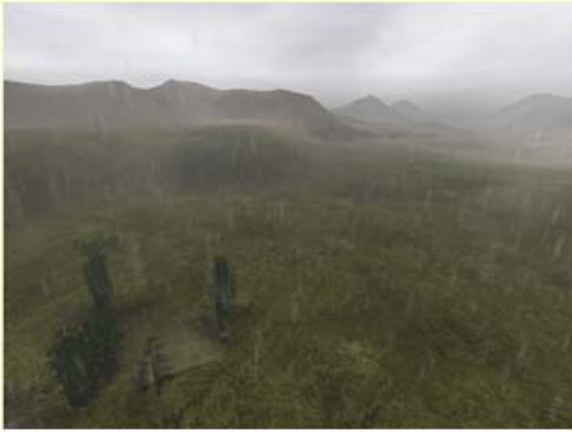


그림 6. Torque3D 로 구현된 비행 시뮬레이션 테스트 게임



참고문헌

- [1] Mason, M., Duric, Z., "Using histograms to detect and track objects in color video", In *Applied Imagery Pattern Recognition Workshop, AIPR 2001 30th*, IEEE, 10-12 Oct. 2001, 154-159.
- [2] F. Porkili, Integral histogram: A fast way to extract histograms in cartesian spaces, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [3] Philippe Noriega, Benedicte Bascle, and Olivier Bernier, "Local kernel color histograms for background subtraction", In *INSTICC Press*, editor, VISAPP, volume 1, pages 213,219, 2006.
- [4] Philippe Noriega and Olivier Bernier, Real Time "Illumination Invariant Background Subtraction Using Local Kernel Histograms". In *British Machine Vision Association(BMVC)* 2006
- [5] G. Bradski, "Real time face and object tracking as a component of a perceptual user interface", In *Proc. IEEE WACV*, pages 214,219, 1998.
- [6] P. Viola and M. Jones, "Robust real time object detection", In *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, 2001.
- [7] S. Di Zeno, "A Note on the Gradient of a Multi-Image", In *CVGIP: Graph. Mod. and Image Proc.*, 33, pp.116-125, 86.
- [8] J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians", in *Second IEEE Workshop on Visual Surveillance Fort Collins, Colorado, Jun. 1999.*, pp. 74-81.
- [9] Stauer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", In *Computer Vision and Pattern Recognition Fort Collins, Colorado, Jun. 1999*, pp. 246-252.
- [10] M. Harville, G. Gordon and J. Woodfill, "Foreground segmentation using adaptive mixture model in color and depth", In *Proc. IEEE Workshop Detection and Recognition of Events in Video*, July 2001, pp. 3-11.
- [11] L. Wixson, "Detecting salient motion by accumulating directional-consistent flow", In *IEEE Trans. Pattern Anal. Machine Intell.*, Volume 22, pp. 774-780, Aug. 2000.
- [12] L. Li, W. M. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection in changing background based on color co-occurrence statistics", In *Proc. IEEE Workshop Applications of Computer Vision*, Dec. 2002, pp. 269-274.
- [13] Gargi, U., Kasturi, R., "Image database querying using a multi-scale localized color representation," In *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries, 1999. (CBAIVL apos;99)*, 1999, pp. 28-32.
- [14] <http://www.garagegames.com/products/torque/tge/>
- [15] <http://www.opencv.org>
- [16] <http://www.ptgrey.com>