

움직임 벡터와 GPU를 이용한 인간 활동성 분석

김선우* · 최연성**

Analysis of Human Activity Using Motion Vector and GPU

Sun-Woo Kim* · Yeon-Sung Choi**

요약

본 논문에서는 실시간 감시 시스템에서 인간의 활동성을 분석하기 위하여 움직임 벡터를 사용하며, 고속연산에 GPU를 활용한다. 먼저 가장 중요한 부분인 전경으로부터 적응적 가우시안 혼합기법, 두드러진 움직임을 위한 가중치 차영상 기법, 움직임 벡터를 이용하여 인간이라고 판단되는 블랍을 검출하고, 추출된 움직임 벡터를 이용하여 사람의 활동성을 분석한다.

본 논문에서는 사람의 행동을 크게 {Active, Inactive}, {Position Moving, Fixed Moving}, {Walking, Running}의 세 가지 메타 클래스로 분류하고 인식하였다. 실험을 위해서 약 300개의 상황을 연출하였으며, 약 86%~98%의 인식률을 보였다. 또한 1920×1080 크기 영상에서 CPU 기반은 4.2초 정도 걸렸는데, GPU 기반에서는 0.4초 이내로 빨라진 결과를 얻었다.

ABSTRACT

In this paper, We proposed the approach of GPU and motion vector to analysis the Human activity in real-time surveillance system. The most important part, that is detect blob(human) in the foreground. We use to detect Adaptive Gaussian Mixture, Weighted subtraction image for salient motion and motion vector. And then, We use motion vector for human activity analysis.

In this paper, the activities of human recognize and classified such as meta-classes like this {Active, Inactive}, {Position Moving, Fixed Moving}, {Walking, Running}. We created approximately 300 conditions for the simulation. As a result, We showed a high success rate about 86~98%. The results also showed that the high resolution experiment by the proposed GPU-based method was over 10 times faster than the cpu-based method.

키워드

Human Activity, Motion Vector, Intelligent Surveillance System, GPU
인간 활동성, 움직임 벡터, 지능형 감시시스템, GPU

1. 서론

지능형 감시시스템은 컴퓨터 비전에 있어서 가장 활발하고 흥미로운 분야 중 하나이다. 전경에서 객체를 분리, 검출, 추적 하는 것에서 그치지 않고, 영상에

서 벌어지는 상황을 이해하려는 노력이 이루어지고 있다. 즉, 객체(인간)가 어떠한 행동을 하고 있는지, 어떠한 행동이 앞으로 벌어질 것인지를 판단하고 예측할 수 있게 될 것이다.

최근 들어 컴퓨터의 성능이 급격히 향상됨에 따라

* 군산대학교 정보통신공학과(ssuny@kunsan.ac.kr)

** 교신저자(corresponding author) : 군산대학교 정보통신공학과(yschoi@kunsan.ac.kr)

접수일자 : 2014. 08. 12

심사(수정)일자 : 2014. 09. 25

게재확정일자 : 2014. 10. 17

서 대용량 비디오를 실시간 처리할 수 있는 방법들이 제안되고 있다. 그 중 하나가 대량의 행렬 및 벡터 연산 시 탁월한 성능을 발휘하는 GPU를 활용한 병렬 프로그래밍 프레임워크인 CUDA이다. CCTV 카메라 또한 많은 진화를 하고 있다. 최근 들어서는 해상도가 1280×720인 HD급 카메라와 1920×1080의 해상도를 가지는 Full HD 카메라도 많이 사용되며, 가격 또한 저렴해지고 있다.

본 논문에서는 기존에 주로 연구되어왔던 저해상도가 아닌 1280×720, 1920×1080의 고해상도에서 인간의 활동성을 인식할 수 있는 기법을 제안한다.

먼저 관건은 배경으로부터 블랍(인간 또는 물체)을 얼마나 정확하게 추출해내느냐는 것이다. 본 논문에서는 가중치 차영상, 적응적 가우시안 혼합 모델, 움직임 벡터를 이용하여 두드러진 움직임 정보를 검출하는 기법을 이용하였다[1-3].

영상 내의 움직이는 블랍을 실시간으로 검출하는 기법 중에서 가장 많이 사용되고 있는 방법은 광류와 배경 모델링 등이 있다. 광류를 이용하면 배경에 대한 정보가 없어도 독립적으로 움직이는 블랍을 검출할 수 있다. 하지만 많은 연산량이 필요하여 하드웨어의 도움이 필요하다. 따라서 본 논문에서는 하드웨어의 도움 없이 소프트웨어적으로 구현이 가능한 움직임 검출 기법을 사용하기 위해서 가중치 차영상, 적응적 가우시안 혼합 모델, 움직임 벡터를 사용하였다[4].

이렇게 추출된 블랍을 이용하여 인간의 활동성을 인식하는데, 본 논문에서는 크게 {Active, Inactive}, {Position moving, Fixed moving}, {Walking, Running}의 세 가지 경우로 제한하고 연구하였다. 각 단계를 구분하기 위해 특징 파라미터들을 사용한다[5-6].

본 논문의 구성은 다음과 같다. II장에서는 적응적 가우시안 혼합모델과 가중치 차영상을 이용하여 객체를 검출하고, 움직임 벡터를 추출하는 방법에 대해서 설명한다. III장에서는 고해상도 영상을 처리하기 위한 GPU에 대해서 설명하고, IV장에서는 움직임 벡터를 이용하여 사람의 활동성을 분석하는 방법을 제시하고, V장에서는 실험결과를 분석하고 결론을 맺는다.

II. 움직임 벡터 추출 방법

본 논문에서는 두드러진 움직임이 있는 물체를 배

경에서 분리하기 위한 방법으로 가우시안 혼합 모델을 통해 생성된 배경 모델에 적응적 차영상을 통하여 객체를 검출한다[7-9].

구현한 감시시스템의 비디오는 MPEG-4(ffmpeg) 형식으로 압축되어 저장, 전송되므로 MPEG-4 알고리즘에 채택된 그림 1의 EPZS(Enhanced Predictive Zonal Search) 함수에서 나오는 움직임 벡터의 값을 그대로 이용한다.

```
simpleme_epzs_motion_search(s, 0, &mx, &my, P,
pred_x, pred_y, rel_xmin, rel_ymin, rel_xmax,
rel_ymax, s->p_mv_table, (1<<16)>>shift, mv_penalty);
```

그림 1. EPZS 함수
Fig. 1 EPZS function

이 함수에서 사용되는 것은 mx와 my로 x와 y에 대한 움직임 벡터의 값이다. 고해상도의 경우 움직임 벡터를 추출하기 위해서는 연산량이 방대해지는데, 이럴 때 CPU의 부하를 경감하기 위해서 GPU를 활용한다.

움직임 벡터가 저장되는 변수를 $psMV_z$ 라 하면 $psMV_z[X][Y]$ 의 이중 배열 형태로 나타낼 수 있다. 여기에서 $psMV_z$ 는 식 (1)과 같다.

$$psMV_z = \sqrt{MV_x^2 + MV_y^2} \tag{1}$$

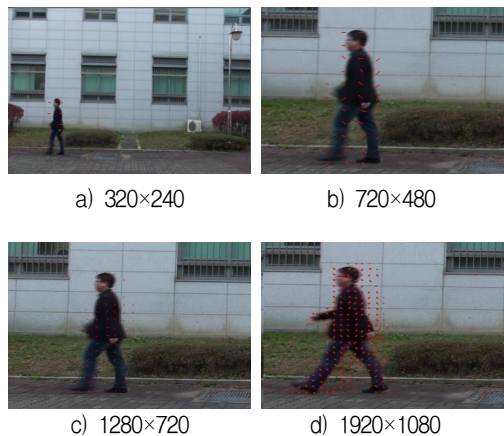


그림 2. 해상도별 움직임 벡터의 개수
Fig. 2 The number of motion vectors per each resolutions

그림 2는 해상도에 따른 움직임 벡터의 검출 개수의 차이를 보여준다. 해상도가 높을수록 움직임 벡터의 개수가 많이 추출되는 것을 확인할 수 있다.

표 1은 그림 2의 사람이 걷는 영상에 대해 여러 해상도에서 움직임 벡터를 추출한 결과이다. 가로와 세로는 최대 추출될 수 있는 개수이고, 벡터의 수는 움직이는 물체에 대해서 추출된 벡터의 수이다. 이 표를 통해서도 볼 수 있듯이 동일한 사람이라도 해상도가 높아질수록 움직임 벡터의 수가 많이 추출되고 이렇게 추출된 벡터 값들은 다양한 분석에 활용될 수 있다.

표 1. 그림 2에서의 움직임 벡터의 개수
Table 1. The number of motion vectors in Fig. 2

Resolution	X-axis	Y-axis	Number of vector
320×240	2	5	9
720×480	6	11	39
1280×720	8	16	81
1920×1080	11	24	167

III. GPU 활용방법

GPU는 컴퓨터 그래픽에서 요구되는 많은 양의 연산을 고속처리하기 위하여 사용되고 있다. GPU는 CPU보다 더 많은 ALU를 탑재하고 있기 때문에 데이터를 병렬적으로 처리한다. GPU의 모든 Thread는 동일한 순차 코드를 실행하지만 서로 다른 데이터를 가지고 연산을 수행하는 SIMD방식으로 동작된다.

제안 알고리즘은 GPU 병렬처리 구조에 적합하도록 설계하였고, CUDA C를 이용하여 구현하였다.

CUDA에서 프로그램 실행 단위는 스레드이고 다수의 스레드를 효율적으로 관리하고 실행하기 위하여 블록과 그리드의 기능을 제공한다. CUDA는 스레드가 모여 블록을 이루고 블록이 모여 그리드를 이루게 된다. 이를 그리드 블록 모델(Grid Block Model)이라 한다. 병렬처리하기 위해서는 커널 프로그램에 따라 적절한 블록당 스레드 개수와 블록의 개수를 할당하여야 한다.

하나의 이미지를 생성하기 위해서는 2차원 인덱스를 이용한다. 그래서 각 스레드는 특정(x,y) 인덱스를

가질 것이며, 출력 영상 내의 한 픽셀에 해당하는 위치로 쉽게 접근할 수 있다.

본 논문에서는 프로그램의 효율성을 높이기 위해서 입력되는 해상도에 따라서 블록과 스레드의 수를 적절히 변경하여 사용하였다. 스레드의 수가 512개를 초과할 수 없고, 한 개의 블록이 가지는 스레드의 개수는 128, 256, 512개로 지정하는 것이 좋다. 왜냐하면, GPU에서는 SM(Streaming Multiprocessor)이 32배수 단위로 동작하게 되어 있기 때문이다. 본 논문에서는 아래 표 2와 같이 입력 해상도에 따른 블록과 스레드의 수를 조정하였다[10].

표 2. 해상도별 블록과 스레드의 수
Table 2. The number of motion vectors in different resolutions

Resolution	Block	Thread per block
320×240	300	256
720×480	675	512
1280×720	1,800	512
1920×1080	4,050	512

그림 3은 본 논문에서 제안한 기법들의 연산을 GPU와 CPU에서 활용하는 방법에 대한 그림이다.

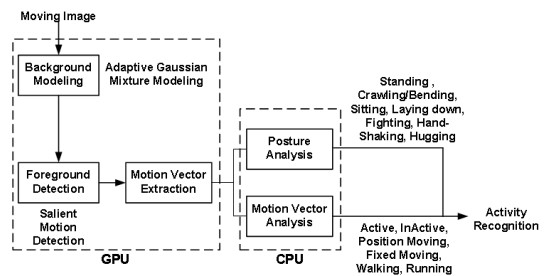


그림 3. GPU의 활용 방법
Fig. 3 The method GPU utilization

그림 3에서 보듯이 연산량이 많은 움직임 벡터와 적응적 가우시안 혼합 배경 모델링과 두드러진 움직임을 활용한 차영상을 구하는 부분을 GPU를 이용하여 처리한다. 추출된 움직임 벡터를 이용하여 활동성을 분석하는 부분은 연산량이 적어서 CPU에서 처리한다.

IV. 인간 행동성 분류 시스템

사람의 행동성 분류기는 그림 4와 같이 {Active, Inactive}, {Position moving, Fixed moving}, {Walking, Running}, {Standing, Bending, Laying down, Sitting}의 네 가지 경우로 제한하고 연구하였다. 하지만 본 논문에서는 움직임 벡터를 이용하여 분석할 수 있는 3단계까지 만을 제안한다. 마지막 4단계의 분류는 실루엣과 다른 기법들을 적용하여 연구하고 있기 때문에 본 논문에서는 언급하지 않는다.

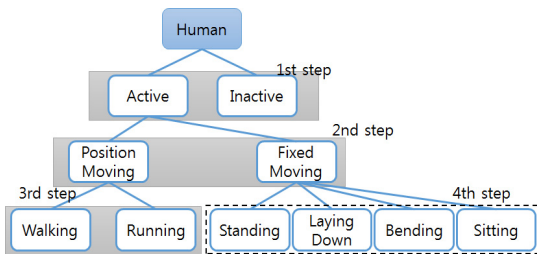


그림 4. 인간 활동성 분류기 구조
Fig. 4 Classifier structure of human activity

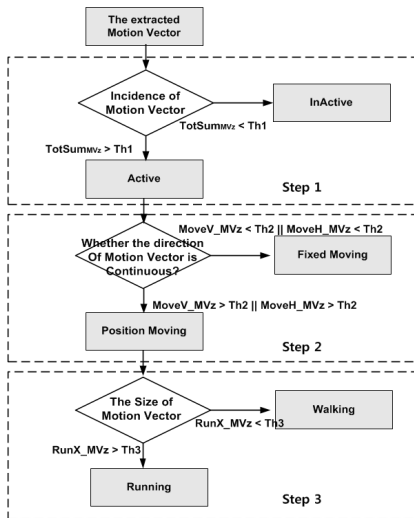


그림 5. 인간 활동성 분류 방법
Fig. 5 Flowchart for human activity classification

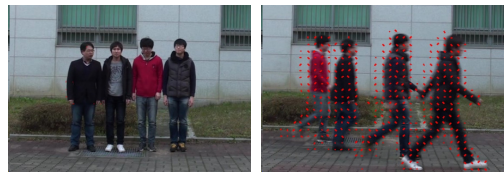
그림 5는 움직임 벡터를 이용한 인간 활동성을 분류하는 방법을 보여주는데, 이 부분은 GPU에서 추출된 움직임 벡터의 값들을 이용하여 CPU에서 처리한

다. 다양한 해상도에서 실험하였기 때문에, 해상도마다 기준 값들이 조금씩 변경되어야 한다.

먼저 배경에서 물체를 분리하고 난 후 사람이 움직이고 있느냐, 그렇지 않느냐를 판별한다. 사람의 움직임이 없다면 행동을 인식할 필요가 없기 때문에 제외시킨다.

1단계에서는 {Active, InActive}를 판별하기 위해서 전체 움직임 벡터의 발생 빈도를 검사한다. 전경 물체가 특정 임계값 이상의 움직임을 발생했다면 이것은 사람이 움직이고 있다는 것으로 활동적이라고 표시할 수 있다. 즉, $psMV_z$ 가 0보다 큰 값을 검사하여 특정 임계값 이상의 움직임 벡터가 검출되었다면 Active라고 판단하고 다음 단계로 넘어가게 된다. 그렇지 않은 경우는 정지된 물체로 배경에 속한 물체로 판단한다.

여기서 $TotSum_{mvz}$ 는 MV_zMV_z 가 0보다 큰 경우의 총합을 나타낸다. 이 단계의 임계값 $Th1$ 은 유동적으로 판단할 수 있겠지만 최소 3 이상의 값을 주어야 한다. 이 단계에서는 단지 활동이냐 그렇지 않느냐만 판단하는 것이다. 그림 6은 1단계 {Active, InActive}를 움직임 벡터로 나타낸 예이다.



a) Inactive Case b) Active Case
그림 6. 1단계 움직임 벡터의 예
Fig. 6 Example of motion vector in step 1

2단계에서는 1단계 분류에서 활동적이라고 판단된 전경 물체에 대해서 {Position Moving, Fixed Moving}을 분류한다. 이 단계에서는 움직임 벡터의 방향성을 검사한다.

MV_z 값의 분포가 위/아래/오른쪽/왼쪽에서 연속하게 나타나는지를 체크한다. 예를 들어 값의 분포가 오른쪽에서 왼쪽으로 3 이상 연속해서 두 개에서 세 개 이상 발생하면 이것은 사람이 걷거나 뛰고 있다는 Position Moving(위치가 이동하고 있다는 것)으로 판별한다. 그렇지 않으면 Fixed Moving(위치는 이동하지 않고 단지 움직이는 것)으로 판별한다.

그림 7의 a)는 이동하고 있다가 거의 정지하는 상황으로 거의 모든 위치에서 값이 나오지 않고 여러 군데에서 1에서부터 5정도의 값이 나오는 것을 알 수 있다.

그림 7의 b)의 열들을 보면 2번째 열의 Z 값들은 10->11->11->12->14 ->... 의 값을 보여주고 있고, 10번째 열의 Z 값은 16->17->18->20->18->...의 값을 보여주고 있다. 다른 열들의 값들을 보아도 추출된 벡터 값들이 연속해서 유사한 것을 알 수 있다.



(a) Fixed moving case

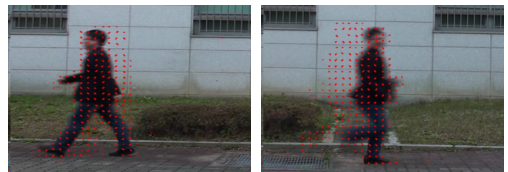


b) Position moving case

그림 7. 2단계 움직임 벡터의 예
Fig. 7 Example of motion vector in step 2

3단계에서는 위의 2단계 분류에서 Position Moving으로 판단된 물체에 대해서 {Walking, Running}을 분류하는 것으로, 움직임 벡터의 크기를 검사하여 이용한다. MV_z 값의 크기가 20 이상의 수가 많은 경우를 검사한다. 걸어가고 있는 사람의 경우에는 움직임 벡터의 그림을 보면 사람의 위치와 비교했을 때 사람 뒤로 3개 이하에서 벡터 값들이 나오는 것을 볼 수 있다. 반면 뛰어가는 사람의 경우에는 움직임 벡터의 길이도 사람의 블랍에 비해서 뒤로 4개 이상으로 나오는 경우가 많다. 이러한 특성을 이용하여 분류한다.

그림 8에서 보듯이 걷는 상황에서는 움직임 벡터의 값이 20 이상 나오지 않고, 추출된 사람 블랍의 위치와 비교하였을 때 사람 블랍 뒤로 최대 3개 이하까지만 나오는 경우가 많음을 확인할 수 있다. 하지만 뛰는 상황에서는 움직임 벡터의 값들이 대부분이 20 이상이고, 추출된 사람 블랍의 위치와 비교하였을 때 사람 블랍 뒤로 움직임 벡터의 값이 최소 4개 이상 나오게 된다.



(a) Walking image (b) Running image



(c) Motion vector of (a)



(d) Motion vector of (b)

그림 8. 3단계 움직임 벡터의 예
Fig. 8 Example of motion vector in step 3

이러한 방식으로 움직임 벡터의 값을 분석하여 사람의 활동성을 판단하였다. 1 단계에서는[Active] -> 2 단계에서는[Position Moving] -> 3 단계에서는 [Walking]으로 단계별로 판단이 이루어진다.

V. 실험 및 결론

본 논문에서 프로그램은 Visual Studio 2010, OpenCV 2.4.2버전을 사용하여 실험하였으며, CPU는 Intel I7-2600K 3.4GHz, GPU는 NVIDIA GTS 450을 사용하였다. 윈도우 계열의 OS에서 동일한 영상을 다양한 해상도로 변경한 후 실험하였다.

표 3. 수행시간 비교

Table 3. Compared of the execution time

Resolution	CPU(sec)	GPU(sec)+CPU(sec)
320×240	0.21	0.05
720×480	0.87	0.11
1280×720	2.76	0.23
1920×1080	5.57	0.35

표 3은 CPU만을 사용하여 처리했을 경우와 GPU를 함께 사용하여 처리했을 경우를 비교한 것이다. CPU만을 사용한 방법보다 속도가 약 5~10배 정도 향상된 것을 볼 수 있다. CPU 기반으로 연산을 하였을 때는 영상의 해상도가 2배 커질 때마다 전체 수행 시간은 4배 이상 늘어났는데, 본 논문에서 제한한 GPU 기반의 기법에서는 영상이 커져도 속도의 차이가 크지 않은 것을 알 수 있었다.

실험을 위해서 300여개의 다양한 상황을 연출하였으며, 이들 중에서 중복되는 상황들이 발생하기 때문에 Active 200번, Inactive 40번, Position Moving 150번, Fixed Moving 60번, Walking 85번, Running 85번의 상황이 발생되었고 이 중 두 명 이상의 사람이 다른 상황을 연출한 것은 20번 이었다.

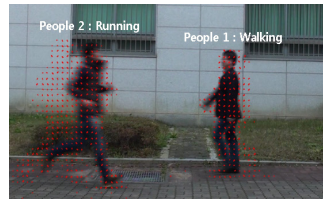
그림 9는 실제 실험한 영상인데, b), c), d)에서처럼 두 명 이상을 실험한 결과영상을 보면 활동적인 경우와 걷는 것과 뛰고 걷는 것이 정확히 구분되는 것을 확인할 수 있다. d)에서는 한 명은 걷고 있고 세 명의 그룹이 겹쳐서 뛰고 있는데, 이럴 경우에 뛰는 사람들의 수를 구분하지는 못하고, 여러 명일 경우에는 인간 그룹이라고 표시만 하였다.



a) Inactive & position moving situation



b) Active & walking situation



c) Walking & running situation



d) Walking & group running situation

그림 9. 모의실험 영상

Fig. 9 Simulation image

사람이 약간 빠르게 걷는 상황 같은 경우에는 어떤 경우에는 Walking으로 판단하지 못하는 경우도 발생한다. 이 부분은 향후 추가 연구가 필요한 부분이다.

표 4. 모의실험 결과

Table 4. The result of simulation

Situation	Simulation	Success	Fail	Success Rate (%)
Active	200	195	5	97.5
InActive	40	37	3	92.5

Position Moving	150	142	8	94.7
Fixed Moving	60	53	7	88.3
Walking	90	85	5	94.4
Running	90	83	7	92.2

표 4는 실험 결과를 표로 나타낸 것이다. 성공률은 88.3%~97.5% 까지 나왔다.

향후 과제로는 좀 더 다양한 실험을 통하여 각 단계별로 기준이 되는 임계값을 상황에 맞게 자동으로 설정할 수 있도록 기준을 정하는 것이 필요할 것이며, 움직임 벡터에서 오동작을 일으켰을 경우 보완할 수 있는 다른 특징을 찾아 접목시킬 필요가 있다. 실시간 영상감시시스템은 하나의 PC에서 16개 이상의 비디오를 처리해야 하므로 GPU의 활용을 극대화시킬 수 있는 병렬처리 알고리즘을 연구 개발 하여야 한다.

또한 고해상도에서 검출된 움직임 벡터를 활용하여 인간의 활동을 보다 정교하게 인식할 수 있어야 하며, 인간의 포즈를 인식하여 활동성을 세분화할 수 있는 방법도 연구할 것이다.

References

- [1] S.-W. Kim, T.-R. Ha, C.-B. Park, and Y.-S. Choi, "Salient Motion Information Detection Method Using Weighted Subtraction Image and Motion Vector," *J. of the Korean Institute of Maritime Information and Communication Sciences*, vol. 11, no. 4, 2007. pp. 779-785.
- [2] J.-J. Park, S.-W. Kim, Y.-S. Choi, C.-B. Park, and T.-R. Ha, "A Study On the Moving Object Tracking System Using Multi-feature Matching," *J. of the Korean Institute of Maritime Information and Communication Sciences*, vol. 11, no. 4, 2007, pp. 787-792.
- [3] H.-T. Kim, G.-H. Lee, J.-S. Park, and Y.-S. Yu, "Vehicle Detection in Tunnel using Gaussian Mixture Model and Mathematical Morphological Processing," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 5, 2012, pp. 967-974.
- [5] S.-H. Lee, "Fast motion Estimation with Adaptive Search Range Adjustment using Motion Activities of Temporal and Spatial Neighbor Blocks," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 5, no. 4, 2010, pp. 372-378.
- [6] S.-W. Kim, Y.-S. Choi, and H.-K. Yang, "Analysis of Human Activity Using Silhouette and Feature Parameters," *J. of the Korean Institute of Maritime Information and Communication Sciences*, vol. 15, no. 2, 2011, pp. 923-926.
- [7] S.-W. Kim, Y.-S. Choi, and H.-K. Yang, "Analysis of Human Activity Using motion Vector," *J. of the Korean Institute of Maritime Information and Communication Sciences*, vol. 15, no. 2, 2011, pp. 157-160.
- [8] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *In CVPR 99*, Fort Collins, FL, June vol. 2, 1999, pp. 246-252.
- [9] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, 2006, pp. 773-780.
- [10] J. Sanders and E. Kandrot, *CUDA by Example : An Introduction to General Purpose GPU Programming*. Boston : Addison Wesley, 2010.

저자 소개



김선우(Sun-Woo Kim)

2000년 군산대학교 정보통신공학과(공학사)

2002년 군산대학교 정보통신전파공학과(공학석사)

2008년 군산대학교 정보통신전파공학과(공학박사)

2010년~현재 군산대학교 정보통신기술연구소 선임연구원

※ 관심분야 : 영상처리, 컴퓨터 비전, 실시간 감시 시스템, 인간행동분석, 빅데이터, 사물 인터넷



최연성(Yeon-Sung Choi)

1982년 중앙대학교 전자공학과 졸업(공학사)

1984년 중앙대학교 전자공학과 졸업(공학석사)

1990년 중앙대학교 전자공학과 졸업(공학박사)

1988년~1990년 제주대학교 정보공학과 조교수

1992년~현재 군산대학교 정보통신공학과 교수

※ 관심분야 : 빅데이터, 사물 인터넷, 컴퓨터 비전, 인공지능, 인지과학, u-City