주행로봇을 위한 GPU 기반의 고속 인공표식 인식

GPU based Fast Recognition of Artificial Landmark for Mobile Robot

권오성 · 김영균 · 조영완 · 서기성*

Ohsung Kwon, Youngkyun Kim, Youngwan Cho and Kisung Seo*

서경대학교 전자공학과

요약

주행 로봇 환경에서 비전 기반의 물체 인식은 물체의 주변 요소와 동적인 환경에 대한 다양한 영상처리 문제를 포함한다. SURF(Speeded Up Robust Features)는 영상의 크기와 회전변화에 강인하게 물체를 인식하는 알고리즘으로 많은 연구자에 의해 사용되고 있다. 하지만 SURF 기반의 영상처리 방법은 고차원의 벡터 성분을 사용하기 때문에 연산 과정에서 많은 시간을 소비하며, 그로 인해 실시간 시스템에서 수행의 어려움을 가지고 있다. 본 연구에서는 이러한 문제점을 해결하기 위해서, 연산량이 많은 SURF 처리 과정을 GPU(Graphics Processing Unit)에서 수행하도록 하여, 보다 빠른 영상 인식을 구현하고자 한다. NVIDIA의 CUDA 라이브러리를 이용하여 GPU 상의 수행 프로그램을 구현하고, 실험을 통해 이동 로봇의 속도와 영상의 크기변화에 따른 표식의 인식률 및 수행시간에 대해서 CPU와 성능을 비교한다.

키워드 : 물체 인식, 이동 로봇, SURF, CUDA

Abstract

Vision based object recognition in mobile robots has many issues for image analysis problems with neighboring elements in dynamic environments. SURF(Speeded Up Robust Features) is the local feature extraction method of the image and its performance is constant even if disturbances, such as lighting, scale change and rotation, exist. However, it has a difficulty of real-time processing caused by representation of high dimensional vectors. To solve the problem, execution of SURF in GPU(Graphics Processing Unit) is proposed and implemented using CUDA of NVIDIA. Comparisons of recognition rates and processing time for SURF between CPU and GPU by variation of robot velocity and image sizes is experimented.

Key Words : Object recognition, mobile robot, SURF, CUDA

1. 서론

주행 로봇은 안정적인 이동성과 센서정보를 통해 다양한 작업 수행 능력을 가진다. 지역을 탐색하거나 SLAM(Simultaneous localization and mapping)과 같은 위치 인식 및 맵 생성, 또한 최근 서비스 로봇들과 같이 사람의 일을 대신하거나, 직접적인 교감을 주고받기도 한다. 이러한 작업은 대부분 외부 환경 및 특정 대상에 대한 정확한 인식을 기반으로 한다[1,2,3].

물체 및 표식인식은 특정 대상의 추종 작업이나 장애물 회피 등과 같은 다양한 비전 연구의 기초가 되며, 이동로봇이 외부정보를 인식하는 중요한 수단이 된다. 특히 특징 기반의 물체 인식은 영상이 가지고 있는 특징점을 추출하고 서로 다른 영상에 대해서 동일한 특징점을 찾아 매칭 시킴으로서 물체를 인식하는 방법으로 기존에 SIFT(Scale

Invariant Feature Transform)[4] 알고리즘이나 SURF(Speeded Up Robust Features)[5] 알고리즘이 많이 사용되고 있다.

SURF 알고리즘은 gray-level의 영상을 통해 크기와 회전 변환에 불변하는 특징점을 추출하고, 추출된 특징점의 표현자들을 생성하여 인식될 물체의 표현자들과 비교하여 물체를 인식하는 알고리즘이다. SIFT 알고리즘에 비해 고속 헤이시안 검출기를 사용하기 때문에 수행 속도를 크게 단축시켰지만 고차원의 벡터로 구성 되어 있어 많은 연산량을 필요로 하기 때문에, 주행 로봇과 같이 실시간성이 필요한 시스템에서의 적용이 어렵다.

본 논문에서는 이러한 문제를 해결하기 위하여 SURF 인식 알고리즘의 처리 과정을 CUDA(Compute Unified Device Architecture)[6] 라이브러리를 이용하여 GPU(Graphics Processing Units)에서 수행하도록 구현하였다. 제안된 기법을 검증하기 위하여 바이올로이드를 이용한 이동로봇을 구성하고, 주행 중 특정 표식의 인식률에 대해 기존 CPU에서의 연산결과와 비교 실험하였으며, 본 연구자에 의한 선행 연구[7]를 확장한 것이다.

접수일자 : 2010년 6월 11일

완료일자 : 2010년 9월 20일

교신저자 : 서기성

본 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2009-0071419)의 지원을 받아 수행되었음.

2. GPU와 CUDA

2.1 GPU (Graphic Processing Unit)

GPU(Graphic Processing Units)는 1999년 NVIDIA에서 발표한 그래픽스를 위한 계산 처리장치로, CPU와는 달리 구조적인 병렬화를 통해 쓰레드가 메모리를 처리할 때 ALU도 동시에 연산을 처리함으로써 메모리 액세스의 지연 시간을 줄이는 방식을 사용한다. 그림 1은 CPU와 GPU의 구조를 나타낸다. GPU는 메모리 연산지연을 줄이는 대신에 CPU보다 빠른 연산을 하기 위해서 메모리 캐시보다 더 많은 ALU를 가지고 있다. 이러한 구조적 특성 때문에 GPU의 처리속도가 빠르게 된다. 따라서 데이터 처리가 많은 분야에서는 기존의 CPU를 이용하는 방법보다 GPU를 이용하면 보다 좋은 성능을 얻을 수 있다[6,8].

그림 1. CPU와 GPU의 구조

Fig. 1. Architecture of CPU and GPU

2.2 CUDA

CUDA는 NVIDIA의 GPU를 범용목적으로 쉽게 사용할 수 있도록 도와주는 프레임워크이다. C/C++ 언어를 기반으로 사용할 수 있도록 구현되어 일반 사용자가 쉽게 접근할 수 있고, 그로 인해 많은 연구자들이 다양한 분야에 폭넓게 적용하고 있다[6].

CUDA의 프로그래밍 모델에서 GPU는 CPU의 계산을 도와주는 역할을 하며, 계산량이 많은 함수를 GPU에서 실행하도록 지정할 수 있다. 이러한 기능은 커널(Kernel)을 이용해서 수행한다.

3. 이동로봇 주행 중 실시간 인식을 위한 GPU기반의 SURF알고리즘 구현

3.1 주행중인 이동로봇의 실시간 인식

이동로봇이 자율 주행을 하기 위해서는 주변 환경에 대한 정확한 인지와 인지된 정보를 빠르게 처리하는 기술이 필요하다. 이러한 환경 정보를 인식하기 위하여 초음파 센서, 적외선 센서, 비전 센서 등이 사용되고 있으며, 특히 비전 센서는 측정 범위가 넓고 영상처리를 통해 다양한 정보를 얻을 수 있어 많은 연구에 사용되고 있다. 하지만 영상의 경우 잡음에 민감하고 많은 연산량을 필요하기 때문에 빠른 처리속도를 요구하게 된다.

최근 물체인식을 위해 크기와 회전 변화에 강인한 SURF 알고리즘이 많이 사용되고 있다. 하지만 SURF 알고리즘의 경우 추출된 특징점에 대해 고차원의 벡터연산을 수행하기 때문에 환경이 복잡하고, 영상의 크기가 커질수록 연산 수행이 증가하게 된다. 또한 환경 내에 고정된 물체를 인식하더라도 로봇의 주행 속도가 올라가면 그만

3.2 CUDA를 이용한 SURF 알고리즘 구현

그림 2. CUDA를 이용한 SURF 알고리즘 흐름도
Fig. 2. SURF algorithm flow chart using CUDA

CUDA는 NVIDIA에서 2007년에 발표한 GPU 프로그래밍 라이브러리로 GPU의 성능을 최대한 활용할 수 있는 도구로 알려져 있다. 본 논문에서는 SURF의 대규모 연산처리를 GPU의 Global 메모리를 이용하여 구현하였으며, SURF 연산에 의해 구해진 방향 성분을 기준으로 Texture 메모리에 할당된 블리 영상에 대해서 영역을 설정하고 4개의 Haar wavelet 특징 조합으로 구성된 특징 벡터를 사용하여 표현자들을 생성하게 된다. Texture 메모리는 각 쓰레드에 의해 블리된 영상을 마지막 단계인 Feature Vector Calculation에 빠르게 접근하기 위해서 할당하는 메모리이다. 캡을 통한 실제적인 영상의 획득은 Intel에서 제공하는 영상처리 라이브러리인 OpenCV[9]를 사용하였다. 그림 2는 CUDA를 이용한 SURF 알고리즘의 흐름도를 나타낸다.

3.3 이동로봇의 GPU 기반 인식 시스템 구성

그림 3은 GPU를 사용하여 특정 표식을 인식하는 시스템 구성도이다.

그림 3. GPU 기반의 인식 시스템 구성도
Fig. 3. Structure of GPU based Recognition system

GPU 기반 인식 시스템의 수행과정은 다음과 같다. 초기에 인식하고자 하는 인공 표식들의 영상을 캡을 통해 입력받아 CPU에 전송하게 된다. 입력된 영상 정보는 CPU를 통해 GPU로 전송되고, GPU는 3.2절에서 기술된 알고리즘을 통

해 SURF 작업을 수행한다. 수행된 결과는 다시 CPU로 전송되고 전송된 결과 값은 저장되어 비교 데이터로 사용된다. 이후, 이동로봇은 주행을 하면서 캠을 통해, 인공표식들이 포함된 주변 영상 정보를 입력 받고, 이들은 GPU 상에서 SURF를 통해 처리된다. 이의 결과를 CPU로 보내서 저장되어 있는 표식 정보와 매칭을 통해 인공 표식을 인식한다.

4. 실험 및 결과 분석

4.1 실험 환경

실험에 사용된 하드웨어는 Intel(R) Core(TM) i7 CPU 2.67GHz와 NVIDIA GeForce GTX 275 그래픽 카드를 한 시스템으로 구성하여 사용하였으며, 로보티즈사의 바이올로이드[10]를 통해 제작한 이동로봇과 Logitech사의 웹캠을 사용하였다(그림 4).

그림 4. 캠을 장착한 주행로봇
Fig. 4. Mobile robot with cam

실험에 사용된 인공표식의 이미지들은 아래 그림 5와 같다.

그림 5. 물체 인식을 위한 인공표식 이미지
Fig. 5. Artificial landmark images for object recognition

기본적인 영상의 획득 및 이미지 생성은 CPU에서 OpenCV 라이브러리를 사용하며, 아래 그림 6은 SURF 알고리즘에 의해 대상 표식을 인식한 결과 영상을 나타낸다.

그림 6. SURF 알고리즘의 결과영상
Fig. 6. Result image of SURF algorithm

4.2 실험 내용

4.2.1 SURF 수행 시간과 인식률 비교 실험

첫 번째 실험은 그림 5의 서로 다른 4개의 표식영상을 20cm 간격으로 배치하고, 표식과 로봇간 거리는 30cm 간격을 두고 영상인식에 대한 실험을 수행하였다(그림 7).

그림 7. 실험 환경
Fig. 7. Experimental environment

이 실험에서는 CPU와 GPU상에서의 해상도 변화에 따른 SURF 수행 시간을 비교 실험하였고 이동 로봇의 주행 속도에 따른 인공 표식의 인식률을 비교 실험하였다.

속도 측정은 한 실험 당 500번 수행해서 평균 시간을 계산하였다. 인식률 측정 방법은 이동로봇에 장착된 캠을 통해 입력된 영상에 대해 SURF 알고리즘을 수행하여 인식하고자 하는 인공 표식을 계속적으로 검출하고, 그 결과 입력 영상의 중앙에 인공 표식이 정확히 인식되었을 경우를 인식 성공률로 측정하였다.

4.2.2 주행 경로 비교 실험

두 번째 실험은 정해진 경로에 대해 20cm 떨어진 위치에 인공 표식을 배치하고, 이동로봇은 각각의 명령이 담긴 인공표식을 인식하여 주행하는 실험을 수행하였다(그림 8).

이 실험에서는 CPU와 GPU상에서의 이동 로봇의 주행 속도에 따른 주행 경로를 비교 실험하였다.

그림 8. 실험 환경
Fig. 8. Experimental environment

4.2.3 두 개의 캠을 장착한 주행 경로 비교 실험

세 번째 실험은 캠을 하나 더 추가하여 두 번째 실험과 비슷하게 정해진 경로에 대해 20cm 떨어진 위치에 인공 표식을 배치하고, 이동로봇은 각각의 명령이 담긴 인공표식을 인식하여 주행하는 실험을 수행하였다(그림 9).

주행로봇이 우측에 대한 인공표식의 영상만을 획득하는 두 번째 실험에 비해 이 실험에서는 좌, 우측의 영상을 획득할 수 있도록 좌측에도 캠을 장착하여 실험하였다.

이 실험에서는 두 개의 캠에서 획득되는 두 개의 영상에 대해서 CPU와 GPU상에서의 이동 로봇의 주행 속도에 따

른 주행 경로를 비교 실험하였다.

그림 9. 실험 환경

Fig. 9. Experimental environment

4.3 실험 결과

4.3.1 SURF 수행 시간과 인식률 비교 실험 결과

그림 10. 해상도 변화에 따른 SURF 수행 시간

Fig. 10. SURF processing time by resolution variation

그림 10은 영상의 크기 변화에 따른 CPU와 GPU상에서의 순수 SURF 알고리즘에 대한 수행시간이며, 그림 11은 영상 획득 및 CPU와 GPU사이의 데이터 전송시간을 포함한 전체적인 수행시간을 비교한 결과 그래프이다. 동일한 속도로 이동로봇을 주행시키며 320x240, 640x480, 800x600, 1280x960, 1600x1200 크기의 영상에 대해 각각 10번씩 실험을 수행하여 평균값을 측정하였다.

그림 10에서의 순수 SURF 처리시간은 해상도가 높아짐에 따라 거의 비례하여 CPU에 대한 GPU의 처리시간이 감소함을 알 수 있다.

그림 11. 해상도 변화에 따른 전체적인 수행 시간

Fig. 11. Total processing time by resolution variation

그림 11의 전체 수행 시간에서도 영상의 크기가 커질수

록 SURF에 의해 생성되는 특징점의 수와 연산되는양이 많아지기 때문에 점차 수행 시간의 차이가 커지며, 1600x1200 영상을 사용하였을 때 SURF의 수행시간이 CPU를 사용하였을 때보다 2배 이상 차이 나는 것을 확인할 수 있다. 하지만 전체적인 수행 시간에서는 GPU에서 연산을 처리하기 위한 데이터 전송시간이 추가되어 약 1.5배의 차이로 그 시간차가 감소하였다.

표식간 거리를 20cm 정도로 위치시켜 두었기 때문에 주행 속도가 조금만 빨라져도 표식을 금방 지나치거나, 이전 영상에 대한 인식연산이 지연되어 표식을 지나친 후 뒤늦게 인식이 되는 문제가 발생한다. 또한 캠의 각도에 따라 인공 표식의 면적이 다르게 보이기 때문에 추출할 특징점의 개수는 줄어들게 된다. 이로 인해서 적어진 특징점들을 빠르게 인식을 못하는 문제가 발생한다.

그림 12. 주행로봇 속도에 따른 인식률(캠 각도 0°)

Fig. 12. Recognition rates by velocity variation of mobile robot(cam angle 0°)

그림 13. 주행로봇 속도에 따른 인식률(캠 각도 20°)

Fig. 13. Recognition rates by velocity variation of mobile robot(cam angle : 20°)

그림 14. 주행로봇 속도에 따른 인식률(캠 각도 30°)

Fig. 14. Recognition rates by velocity variation of mobile robot(cam angle : 30°)

그림 12-15의 결과는 캠의 각도가 각각 0°, 20°, 30°, 40° 일 때 주행로봇의 속도가 5, 7, 9, 11, 13, 15, 17(cm)에서 입력된 영상의 중심에 표식이 인식된 결과를 나타낸다. 여기서, 캠의 각도란 이동로봇에 장착된 캠이 표식을 주시할 때, 정면에서 틀어진 각도를 의미한다. 즉, 정면이면 0 도이고, 정면에서 측면으로 틀어질수록 각도가 증가한다.

캠의 각도가 20° 인 경우부터 11cm/s부터 CPU의 인식률이 떨어지기 시작하며(그림 13), 각도가 30° 인 경우는 7cm/s부터 CPU의 인식률이 감소한다(그림 14).

그림 15. 주행로봇 속도에 따른 인식률(캠 각도 40°)
Fig. 15. Recognition rates by velocity variation of mobile robot(cam angle : 40°)

그림 15에서와 같이 캠의 각도가 40°일 때 GPU와의 인식률 차이가 가장 현저하게 나타나며, CPU의 경우 11cm/s 부터는 완전히 인식을 못하는 현상이 발생한다.

요약하면, 캠의 각도가 증가할수록 GPU와 CPU의 인식률의 차이가 커지는데, 이는 캠의 각도가 증가할수록 인공표식의 면적이 작아지기 때문에 인식의 어려움이 증가한다. 더불어 주행로봇의 속도가 증가할수록 이러한 현상은 심화된다고 볼 수 있다.

4.3.2 주행 경로 비교 실험 결과

그림 16. 주행로봇의 속도에 따른 주행 경로(단일 캠)
Fig. 16. Navigation paths by velocity variation of mobile robot(single cam)

그림 16은 CPU와 GPU상에서 주행로봇이 인공표식을 인식하면서 주행한 경로를 비교 실험한 결과이다. 그림에서 확인할 수 있듯이 GPU상에서의 주행은 정해진 경로를 크게 벗어나지 않은 것을 확인할 수 있다. 그에 비해 CPU상에서의 주행은 11cm/s의 속도를 제외한 실험에서 모두 정해진 경로를 벗어나서 주행한 것을 확인할 수 있다.

표 1은 그림 16의 결과 값을 수치로 표현한 것이다. 괄호 안의

수치들은 주행 로봇의 폭에 대한 오차의 비율을 계산한 값이다.

표 1. 주행경로의 오차(단일 캠)

Table. 1. Errors of navigation paths for mobile robot(single cam)

Velocity (cm/s)	TEST	1번째 인공표식 오차 (cm)	2번째 인공표식 오차 (cm)	3번째 인공표식 오차 (cm)
11	CPU	0.2 (0.03)	0.2 (0.03)	0 (0)
	CPU+GPU	0 (0)	0 (0)	0.3 (0.04)
13	CPU	6.5 (0.81)	6.8 (0.85)	3.2 (0.40)
	CPU+GPU	2 (0.25)	4.2 (0.53)	1.3 (0.16)
15	CPU	9.7 (1.21)	2.6 (0.32)	10.5 (1.31)
	CPU+GPU	5 (0.62)	4 (0.50)	2.8 (0.35)

4.3.3 두 개의 캠을 장착한 주행 경로 비교 실험 결과

그림 17. 주행로봇의 속도에 따른 주행 경로(두개의 캠)
Fig. 17. Navigation paths by velocity variation of mobile robot(dual cam)

표 2. 주행경로의 오차(두개의 캠)

Table. 2. Errors of navigation paths for mobile robot(dual cam)

Velocity (cm/s)	TEST	1번째 인공표식 오차 (cm)	2번째 인공표식 오차 (cm)	3번째 인공표식 오차 (cm)	4번째 인공표식 오차 (cm)
11	CPU	5 (0.63)	3 (0.38)	4.8 (0.60)	2.6 (0.33)
	CPU+GPU	0 (0)	0.5 (0.06)	2 (0.25)	1.5 (0.18)
13	CPU	9.5 (1.19)	7.4 (0.93)	10 (1.25)	11.2 (1.40)
	CPU+GPU	2.2 (0.28)	2.8 (0.35)	2.4 (0.30)	3.2 (0.40)
15	CPU	- (-)	- (-)	- (-)	- (-)
	CPU+GPU	6.2 (0.78)	5.5 (0.69)	6.3 (0.79)	5.7 (0.71)

그림 17은 두 개의 캠을 가지고 CPU와 GPU상에서의 주행로봇이 인공표식을 인식하면서 주행한 경로를 비교 실험한 결과이다. 캠의 추가로 인해 두 영상이 입력 값으로 들어오기 때문에 CPU상에서의 영상처리는 약 두 배의 연산 시간을 갖게 된다.

이로 인해서 캠이 하나일 때의 실험보다 큰 오차를 발생하게 된다. 이에 비해 GPU상은 병렬 구조의 형태이기 때문에 더 빠른 연산 시간을 갖게 되고 CPU상의 실험보다 작은 오차를 발생하게 된다. 특히 주행로봇의 속도가 15cm/s일 때 CPU상의 실험에서 인공표식을 인식 못하는 것을 확인할 수 있다.

표 2는 그림 17의 결과 값을 수치로 표현한 것이다.

5. 결 론

본 논문에서는 주행 중인 이동 로봇의 정확하고 빠른 표식 인식을 위하여 기존 CPU에서 연산되는 SURF 알고리즘을 GPU 상에서 동작하도록 구현하였다.

이를 위해 NVIDIA의 CUDA 라이브러리를 사용하여 GPU에서 동작하는 SURF 알고리즘을 구현하여, 기존 CPU에서 처리하던 고차원의 복잡한 연산을 GPU에서 처리할 수 있도록 하였다.

영상의 해상도와 이동로봇의 속도, 그리고 캠의 각도 및 개수를 다양하게 변화시키며, 표식의 인식 시간과 인식률에 대한 실험을 수행하였고, GPU를 사용한 방식이 기존 CPU만을 이용한 방법에 비해 주행 중인 로봇을 통한 인식률의 향상 및 인식 시간의 단축 결과를 얻을 수 있었다.

이를 통해 이동 중인 로봇에서의 물체 인식의 고속처리에 대한 가능성을 확인할 수 있었다.

향후 GPU 구조에 더욱 적합한 알고리즘으로의 개선과 좀 더 복잡하고 환경에서의 인식 속도 개선을 통해서 고속 주행 로봇에서의 강인한 인식 처리에 대한 적용이 필요하다고 사료된다.

참 고 문 헌

- [1] A. Gopalakrishnan and A. Sekmen, "Vision-based Mobile Robot Learning and Navigation," *ROMAN, IEEE International Workshop on Robots and Human Interactive Communication*, pp. 28-53, 2005.
- [2] H. Li and S. Yang 2003. "A Behavior-based Mobile Robot with a Visual Landmark-recognition System," *IEEE/ASME Transactions on Mechatronics*, Vol. 8, No. 3, pp. 390 - 400, 2009.
- [3] S. Muramatsu and D. Chugo, S. Jia, K. Takase, "Localization for Indoor Service Robot by using Local-features of Image," *ICROS-SICE International Joint Conference*, pp. 3251-3254, 2009.
- [4] D. G. Lowe "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-100, 2004.
- [5] H. Bay and T. Tuytelaars, and L. V. Gool, "Surf: Speeded Up Robust Features," *European Conference on Computer Vision*, Vol. 3951, pp. 404-417, 2006.
- [6] NVIDIA CUDA™ Programming Guide, NVIDIA Corporation, 2009, <http://developer.nvidia.com/object/cuda.html>
- [7] 권오성, 김영균, 조영완, 서기성, "GPU 이용한 주행로봇의 인공표식 인식," *한국지능시스템학회 춘계학술대회 학술발표논문집*, 제 20권, 제1호, pp. 123-126, 2010.
- [8] Zhichao Cao, Shiming Xu, Wei Xue and Wenguang Chen, "Improving Dense Linear Equation Solver on Hybrid CPU-GPU System," *10th International Symposium on Pervasive System, Algorithms, and Networks*, pp. 556-562, 2009.
- [9] OpenCV, <http://opencv.willowgarage.com/wiki/welcome>

[10] Bioloid Corporation, <http://www.robotis.com/>

저 자 소 개

권오성 (Ohsung Kwon)

2009년 : 서경대학교 전자공학과 공학사
2009~현재 : 동 대학원 전자공학과 석사과정

관심분야 : 주행이동로봇, SLAM

E-mail : s0nam00@nate.com

김영균 (Youngkyun Kim)

2008년 : 서경대학교 전자공학과 공학사
2008~현재 : 동 대학원 전자공학과 석사과정

관심분야 : GP, 영상인식, 지능로봇

E-mail : dudrbs072@naver.com

조영완 (Youngwan Cho)

1991년 : 연세대학교 전자공학과 공학사
1993년 : 연세대학교 전자공학과 공학석사
1999년 : 연세대학교 전자공학과 공학박사
2000~2003년 : 삼성전자 디지털 프린팅 사업부 책임연구원
2002~현재 : 서경대학교 컴퓨터공학과 조교수

관심분야 : 퍼지 및 신경망 시스템, 보행로봇, 적응 및 강인 제어, 보행로봇의 자세 제어, 무인 이동체 시스템

E-mail : ywch@skuniv.ac.kr

서기성 (Kisung Seo)

1986년 : 연세대학교 전기공학과 공학사
1988년 : 연세대학교 전기공학과 공학석사
1993년 : 연세대학교 전기공학과 공학박사
1993~1998년 : 서경대학교 산업공학과, 전자공학과 조교수

1999~2003년 : Michigan State University, Genetic Algorithms Research and

Applications Group, Research Associate

2002~2003년 : Michigan State University, Electrical & Computer Engineering, Visiting Assistant Professor

2003~현재 : 서경대학교 전자공학과 부교수

관심분야 : 지능로봇, GP, GA, 진화 디자인

E-mail : ksseo@skuniv.ac.kr