

# 로봇 축구를 위한 소프트웨어 구조 및 비전 처리

한국과학기술원 최승환 · 김종환

## 1. 서론

로봇 축구는 1995년 KAIST의 김종환 교수에 의해 체계적인 경기 규칙 제정과 함께, 기존의 작업 로봇이나 서비스 로봇 등의 영역을 넘어 엔터테인먼트 로봇 톱스라는 새로운 장을 열었다(1). 이 로봇 축구는 MiroSot(Micro robot Soccer Tournament)이라는 대회 이름으로 한 팀이 3대의 로봇으로 구성된 3:3 경기를 시작으로 매년 로봇축구 기술의 발전과 함께 그 규모와 박진감은 배가되었으며, 그 결과 2001년에는 5:5 경기가(5), 2004년에는 11:11 경기로 발전되어 왔다. 로봇 축구의 범주는 로봇의 종류와 크기에 따라 KheperaSot, MiroSot, NaroSot, QuadroSot, RoboSot, SimuroSot, HuroSot으로 나뉘며, 각 범주는 또한 MiroSot의 경우처럼 로봇의 수에 따라 세부 종목으로 나뉜다. 이들에 관한 자세한 소개는 <http://www.fira.net>에 나와 있다.

로봇 축구란 사람이 아닌 로봇이 미리 정해진 경기 규칙에 따라 축구를 하는 것이다. 사람처럼 로봇이 스스로 경기장을 누비며, 공을 찾고 상대방을 피하여 드리블과 패스를 하며, 상대방의 골대에 슈트를 넣을 수 있도록 하는 것이 궁극적인 목표이다. 로봇 축구도 여러 대의 로봇이 서로 협동을 하면서 경기를 하는 것이기에 로봇 개체들 간의 행동이론을 다루는 다개체 시스템(Multi-agent system)에 대한 연구가 필요하다(2, 3). 다개체 시스템에서의 개체란 일반적으로 목적과 행동 능력, 그리고 자기가 속한 환경에 대한 정보를 소유하고 있는 한 개체를 의미하며, 여기서 '행동'이란 개체의 활동 방식을 의미한다. 다개체 시스템을 구현하기 위해서는 다수의 개체들로 이루어지는 시스템의 구현 원리를 정하고, 시스템 내에서 개체들이 독자적인 행동을 하면서 동시에 서로 협력할 수 있도록 하는 동작 매커니즘을 마련해주어야 한다(7). 따라서 다개체 시스템의 대표적인 예들 중 하나가 바로 로봇 축구 시스템이다. 로봇이 축구를 하기 위해서는 복합적인 기술이

요구된다. 로봇 축구 시스템은 일반적으로 주 컴퓨터(Host computer)에서 내부적으로 동작하는 영상 처리 시스템(Vision system)과 로봇의 전략 및 제어 시스템(Strategy and control system), 그리고 외부적으로 통신 시스템(Communication system), 로봇 시스템(Robot system)으로 구성된다.

경기장에 위치한 다수의 로봇들은 경기장 위에 설치된 카메라와 연결된 주 컴퓨터 내의 비전 프로그램에 의해 위치정보가 파악되고, 이 정보를 받은 주 컴퓨터의 전략 및 로봇 제어 프로그램은 각 로봇의 역할이나 로봇이 움직여 가야할 곳 등을 계산하여 그 결과를 무선 송신기를 통해 로봇들에 전달하여 로봇들을 제어하게 된다. 즉, 주 컴퓨터가 계산된 로봇들의 왼쪽과 오른쪽 바퀴 속도 정보를 연속적으로 로봇에 보내어 로봇을 제어하게 된다. 이와 같은 시스템의 경우 로봇의 구조는 간단하지만 로봇 자체에는 센서 등이 내장되지 않아 정확한 제어를 위해 고속 영상 처리 시스템 및 주 컴퓨터의 뛰어난 계산 능력, 고속 통신 등이 요구된다. 이 시스템의 소프트웨어인 비전 시스템은 모두 주 컴퓨터에 집중되어 중앙 집중 시스템의 장단점을 모두 갖는다(6).

본 논문에서는 로봇 축구 시스템 중 비전 처리 부분을 중심으로 지금까지 널리 알려진 방법과 이를 보완하기 위한 방법에 대해 알아보도록 한다. 2장에서는 로봇 축구 시스템의 전반적인 구조에 대해 설명하도록 한다. 3장에서는 소프트웨어 구조, 4장에서는 비전처리부분에 대해 보다 자세한 논의를 하도록 하며, 5장에서 결론을 맺는 것으로 한다.

## 2. 로봇 축구 시스템의 구조

로봇 축구 시스템은 크게 비전 처리 과정과 전략 생성 과정으로 나눌 수 있다. 우선 카메라를 통해 들어온 영상이 프레임 그레이버를 거쳐 호스트 프로그램에 입력이 되면 비전 처리 과정을 거쳐 각 로봇과 공의 위

치를 알아내게 된다. 이렇게 알아낸 위치 정보를 바탕으로 적절한 전략을 수립하고 이 전략에 따른 각 로봇의 이동 속도를 정하게 된다. 이 속도 정보는 R/F 통신을 통해 각 로봇에게 전달되며 로봇은 이 속도 명령에 따라 움직이게 된다. 정확하고 정교한 로봇 제어를 하기 위해서 이상의 과정을 가능하면 빠른 시간 내에 수행해야 한다. 때문에 이 시간을 줄이기 위한 노력은 계속되고 있으며 대다수의 팀들이 16ms 내외에서 모든 과정을 처리하고 있다. 그림 1은 3대의 로봇에 대한 로봇 축구 시스템을 나타내고 있다.

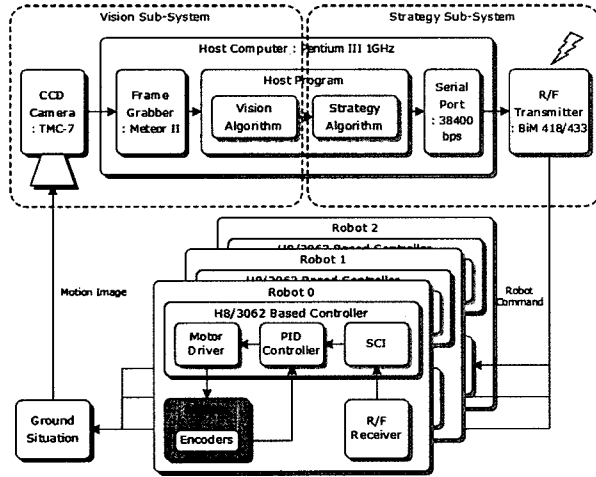


그림 1 로봇 축구 시스템

### 3. 소프트웨어 구조

그림 2는 로봇 축구에서 사용되고 있는 제어 프로그램의 전체 구조를 나타내고 있다. OnReady() 함수는 시스템의 비전 처리, 전술, 통신을 순차적으로 실행하는 함수로서, 경기 중에는 무한루프를 통해 계속 실행된다. 이 OnReady() 함수는 FindObject() 함수, My\_Strategy() 함수와 Send\_Command() 함수로 구성되어 있다. 먼저 FindObject() 함수에서 각 로봇과 공의 위치와 방향을 계산한다. 이와 관련된 내용은 4장에서 자세히 살펴보도록 한다. 경기를 위한 전술 및 로봇 제어 프로그램은 My\_Strategy()라는 함수로 구현되었다. 마지막으로 Send\_Command() 함수는 My\_Strategy() 함수에서 전술에 따라 구한 각 로봇들의 오른쪽 왼쪽 바퀴의 속도 정보를 정해진 프로토콜에 따라 로봇에게 보낸다.

My\_Strategy() 함수에서 다시 Game()이라는 함수가 호출되는데 이 함수는 크게 5가지의 상태로 나누어진다. 이 5가지 상황은 Kick\_OFF, Penalty\_Kick, Free\_Kick, Free\_Ball, Goalie\_Kick, Defense이다. Kick\_OFF 상황은 경기를 시작할 때와 골이 났을 때

수행하는 모드이고, Penalty\_Kick은 상대방으로 페널티킥을 할 때 사용하는 모드이다. Free\_Kick 역시 상대방 골대에서 프리킥을 할 때 사용하는 모드이고, Goalie\_Kick은 자기 진영의 골대 영역에서 상대방 쪽으로 공을 찰 때 사용하는 모드이다. Free\_Ball은 프리볼 상태일 때 적용하는 모드이다. 이 이외에는 모두 Defense 모드를 사용한다. 즉, 상대방이 페널티킥을 할 때나 프리킥을 할 때도 포함된다.

Kick\_OFF 모드에서는 로봇과 공의 조건을 판단한 후 로봇1이 Velocity() 함수로 공을 차게 하고 다른 로봇들은 Position() 함수로 지정된 위치에 있게 한다. 공이 움직이기 시작하면 다시 조건을 판단한 후 Defense 모드로 모드가 바뀐다.

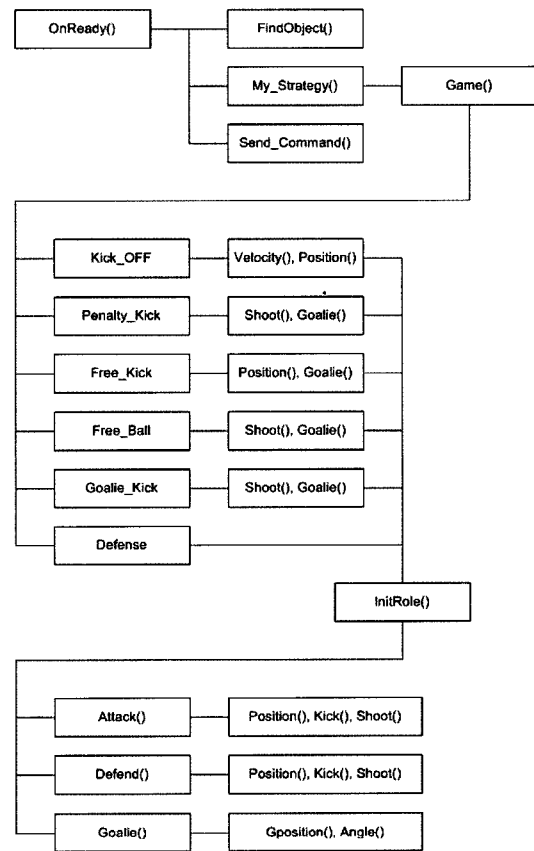


그림 2 로봇 제어 프로그램

Free\_Kick 모드도 로봇과 공의 조건을 판단한 후 로봇1이 공을 차게 할 때 Position() 함수를 이용하고 Goalie() 함수를 이용해 골키퍼 로봇을 움직이게 한다. 다시 조건을 조사한 후 Defense 모드로 모드가 바뀐다.

Free\_Ball, Penalty\_Kick, Goalie\_Kick의 경우 조건을 판단하는 부분만 다르고 이용하는 함수는 같다. 조건 판단 후 Shoot() 함수와 Goalie() 함수를 호출한다. 다시 조건을 판단한 후 Defense 모드로 모드가 바뀐다.

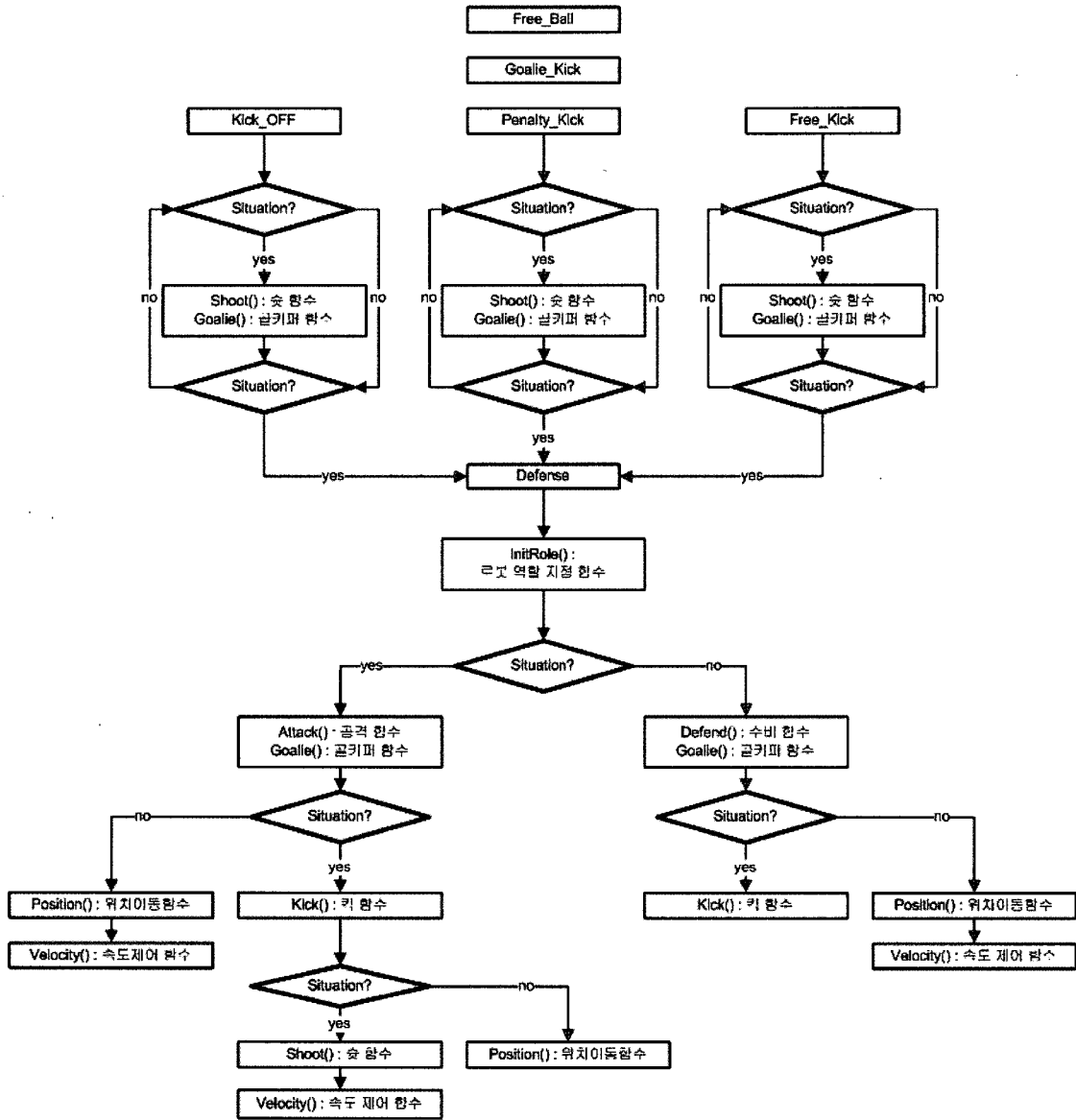


그림 3 로봇 제어 함수의 순서도

그림 3은 그림 2의 로봇제어 프로그램을 각각의 조건에 따른 함수 호출의 순서도로 나타낸 것이다. InitRole() 함수 내에서는 상황을 판단한 후 공격을 할 것인지, 수비를 할 것인지를 결정한 후 그에 따른 함수를 호출한다. Kick() 함수 내부에서도 다시 조건을 검사한 후 Shoot() 함수와 Position() 함수를 호출한다. Position(), Shoot(), Goalie() 함수는 필요한 로봇들의 속도를 계산한 후 Velocity() 함수를 호출한다. Velocity() 함수는 계산한 속도 값을 모터의 PWM 값으로 변환하여 로봇들에게 보낼 데이터 전송 배열에 저장한다.

이상의 소프트웨어 구조는 초창기부터 현재까지 꾸준히 사용되고 있다. 하지만, 3:3 경기에서 5:5, 11:11 경기로 로봇의 수가 늘어나면서 그 내부 알고리즘은

큰 변화가 생기고 있다. 그 한 예가 어떤 한 상황이 공격 상황인지 수비 상황인지 파악하느냐 하는 문제이다. 3:3 경기의 경우에는 상대적으로 고정적인 위치를 갖고 있는 골키퍼 로봇을 제외하면 두 팀에서 각각 2대의 로봇의 위치와 공의 위치만을 고려하면 비교적 쉽게 공격과 수비의 위치를 판단할 수 있었다. 하지만, 5:5, 11:11 경기는 로봇의 수가 많아진 만큼 경기 중 가능한 상황 또한 복잡해졌다. 때문에 모든 로봇의 정확한 위치를 고려하는 것 보다는 공 주변에 있는 로봇들만 고려하거나 두 팀의 전체적인 로봇 분포를 분석하는 것이 보다 효과적이라 할 수 있다. 추후 로봇 축구 시스템의 발전을 위해 소프트웨어의 구조도 이러한 알고리즘을 보다 쉽게 수행하기 위한 적절한 변화가 필요할 것이다.

#### 4. 로봇 축구 시스템의 비전 처리

비전 시스템은 일반적으로 카메라로부터 얻어진 2차원 영상으로부터 여러 가지 유용한 정보들을 얻는 것으로, 의료 분야에서의 CT 촬영, 이동 로봇에서의 스테레오 비전, 위성 자료에 대한 기상 예측 등에 사용한다. 로봇 축구를 위해서는 각 로봇의 위치와 방향, 그리고 공의 위치 정보가 필수적이다.

비전 처리는 크게 두 부분으로 나누어 생각할 수 있다. 하나는 카메라 영상에서 각 로봇의 위치 정보를 알아내는 과정으로 경기 중에 반복적으로 수행되기 때문에 수행 속도가 무엇보다도 중요하다. 다른 하나는 경기 전에 이루어지는 색상 추출 과정이다[4].

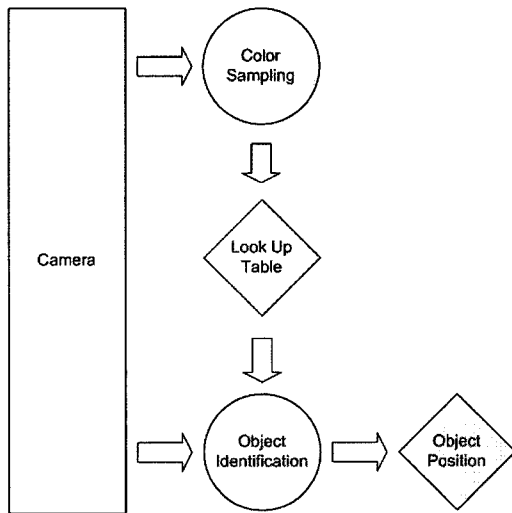


그림 4 비전 처리 과정

##### 4.1 색상 추출

로봇 축구 시스템은 로봇에 붙어 있는 색 패치를 통해 로봇의 위치를 파악하기 때문에 패치를 구성하는 여러 색상을 적절한 자료구조로 표현해야 한다.

대표적인 색 표현 방식으로 RGB, YUV가 있다. 여기서는 빛의 3원색인 적색, 녹색, 청색에 의해 색을 정의하는 색 모델인 RGB를 중심으로 설명하도록 한다.

하나의 특정 색상은 RGB 공간상에 한 점에 대응된다. 하지만, 실제로 같은 색상이라 하더라도 카메라에 들어오는 영상에는 여러 요인에 의해 조금씩 다른 색으로 나타나게 된다. 때문에 화면상에서 같은 색으로 인식되어야 하는 색상의 집합을 구하고 모든 원소가 포함되는 일정한 영역이 같은 색으로 인식되도록 해야 한다. 그림 5는 이 처럼 하나의 색상을 나타내는 4개의 집합 원소의 예를 보여주고 있다. 이처럼 RGB 공간을 2차 평면으로 표현한 그래프를 이용하여 보다 직관적으로 이해할 수 있다. 그림 6은 기본적으로 적색,

녹색, 청색에 대해 각각 최소값과 최대값을 정의하고 해당 구간에 있는 색은 하나의 색상으로 인식하는 방법이다. 이는 6개의 임계값으로 한 가지 색상을 표현할 수 있는 장점이 있다.

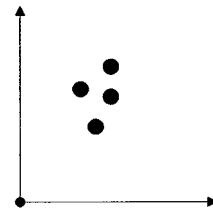


그림 5 색상 집합의 원소

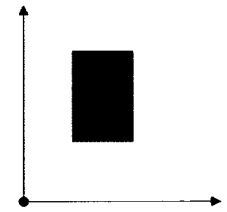


그림 6 임계값 모델

하지만 이러한 방법은 임계값 부근이 매끄럽지 못해 불필요한 값이 같은 색상으로 인식되는 경우가 많이 발생한다. 이러한 문제점을 해결할 수 있는 방법이 그림 7, 그림 8이 나타내고 있는 구름형, 타원형 모델이다.

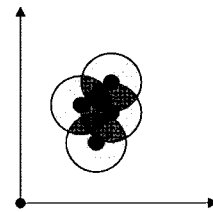


그림 9 구름형 모델

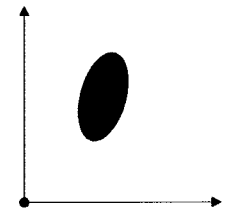


그림 9 타원형 모델

구름형 모델은 각각의 색상 집합 원소와 일정 거리 안에 있는 값을 같은 색상으로 인식하는 방법이다. 타원형 모델은 색상 집합 원소를 적절히 포함할 수 있는 방정식을 구하여 만들어진다.

$$x = \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \quad \mu = \text{sample color data의 평균값} \quad (1)$$

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \quad (2)$$

$$d = V^T C^{-1} V$$

단,  $d$ 는 동일한 색상으로 인식될 공간의 크기,  $V$ 는  $x - \mu$ 이다.

두 모델 모두 비전 처리 과정에서의 계산 부담을 줄이기 위해 사전에 색상을 나타내는 3차원 배열에 동일 색상에 대한 마스크 값을 0, 1로 계산해두게 되며 이 배열을 Look Up Table이라 부른다.

두 모델은 서로 상반되는 성질이 있다. 구름형 모델은 초기 색상 집합의 원소 개수가 충분히 많아야 하지만 타원형 모델은 적은 수 색상 정보만으로도 쉽게 동일 색상 공간을 만들어낼 수 있다. 하지만 충분히 많은 원소만 주어진다면 구름형 모델이 타원형 모델보다 더 구체적으로 동일 색상을 표현할 수 있다.

위에서 설명한 방법으로 하나의 동일 색상은 어느 정도 안정적으로 표현할 수 있다. 하지만 실제 영상 처리 과정에서는 위치에 따른 조명 차이, 인접한 색상에 따른 색상 변화에 의해서 동일 색상도 여러 가지 색상으로 바뀌게 된다. 때문에 그림 9, 그림 10과 같이 비슷한 여러 가지 색상 정보를 하나로 모아야한다.

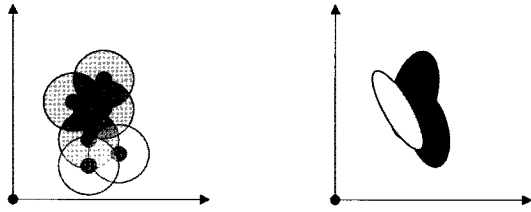


그림 11 다중 구름형 모델      그림 11 다중 타원형 모델

다중 구름형, 다중 타원형 모델은 Look Up Table을 합치는 간단한 방법으로 쉽게 만들어 낼 수 있다. 또한 이 방법을 응용하면 두 색상 합치는 것뿐만 아니라 한 색상 공간에서 다른 색상 공간을 빼는 연산도 가능하기 때문에 보다 정밀한 색상 공간을 만들어 낼 수 있다.

실제 프로그램에는 적은 수의 색상으로 만들어 낼 수 있는 다중 타원형 모델이 사용되었으며 다양한 조명 변화에도 색상을 정확히 찾는 것을 확인하였다.

#### 4.2 로봇 위치 인식 및 영상 왜곡 보정

색상 추출을 통해 한 색상에 대한 2진 영상을 만들어 낼 수 있다. 이렇게 만들어진 2진 영상에서 해당 색상의 물체를 인식하는 대표적인 방법으로 크기 필터링과 무게중심법이 사용된다.

카메라의 해상도나 조명의 불균일성 등에 의하여 2진 영상에는 잡음이 있을 수 있다. 이런 잡음은 불규칙적으로 발생하며 연결성분의 화소 또한 작은 값을 가진다. 라벨링 이후에 얻어진 연결성분에 대하여, 특정 화소수(N) 미만인 연결성분들을 제거하는 크기 필터링 과정을 거친다. 그림 11은 알파벳 i에 대해서 화소수가 10미만에 대해서 필터링을 한 결과이다.

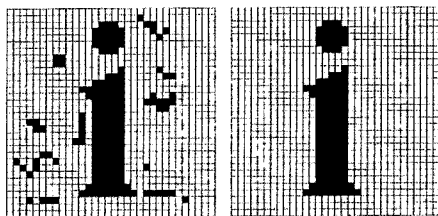


그림 12 필터링

이후 하나의 연결성분을 구성하는 모든 점 (x,y)에 대해 무게중심을 구하면 해당 물체의 위치를 구할 수 있다. 이와 같은 기본적인 이미지 분석방법을 이용하여

로봇의 위치를 인식하게 되는데 이때 구체적인 계산 방법은 패치 디자인에 따라 조금씩 차이가 난다.

그림 12와 그림 13은 각각 로봇 축구 초기에 많이 사용되던 디자인으로 대각 패치와 사각 패치이다.



그림 13 대각 패치

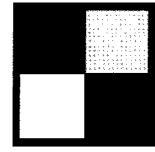


그림 14 사각 패치

대각 패치는 팀 색상(노란색 또는 파란색)의 중심 좌표를 로봇의 위치로 판단한다. 또한 동시에 팀 색상의 방향성을 판단하여 로봇의 방향도 판단할 수 있다 [8]. 끝으로 주변에 있는 삼각형의 ID색으로 각각의 로봇을 구별하게 된다. 하지만 ID색의 크기가 작기 때문에 경기장이 커지고 화면에 나타나는 로봇의 크기가 작아지면서 최근에는 거의 사용되지 않고 있다.

사각 패치의 경우는 팀 색상과 ID색 각각의 무게 중심점의 중심을 로봇의 위치로 판단하고 두 무게 중심의 기울기를 이용해 로봇의 방향을 판단한다. 또한 ID색으로 각각의 로봇을 구별한다[9]. 하지만 그림 14에 서처럼 여러 로봇이 이웃하게 되면 한 로봇을 나타내는 팀 색상과 ID색을 정확히 찾아내는 것이 힘들게 된다. 이런 이유로 역시 최근에는 사용하는 팀이 줄고 있다.

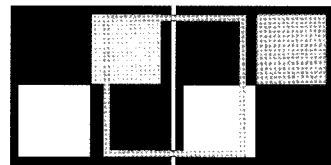


그림 15 사각 패치의 문제점

이러한 문제점을 극복하고자 만들어진 고안된 디자인이 ICRO 패치와 Soty-Cross 패치이다(그림 15, 그림 16).

ICRO 패치는 성균관 대학교 ICRO팀에서 만든 디자인으로 중심에 팀 색상이 위치하며 주변에 3개의 ID색이 존재하게 된다. 우선 팀 색상으로 로봇의 위치를 찾고 주변 ID색의 조합으로 로봇을 구별하고 방향도 계산하게 된다[9]. Soty-Cross 패치는 KAIST Soty팀에서 만들어진 디자인으로 삼각형의 팀 색상과 보조 팀 색상(흰색)이 서로 마주보고 있으며 그 양 옆으로 ID색이 존재한다. 팀 색상과 보조 팀 색상을 이용해 사각패치와 같은 방법으로 로봇의 위치와 방향을 찾고 좌우 ID색 조합에 따라 로봇을 구별하게 된다. 이 두 디자인은 겉보기에는 확연히 다르나 그 성격은 비슷하다 할 수 있다. 하나는 각각의 색상이 충분한 크기이기 때문에 색상이 주변 색상에 의해 영향 받을 가능성이

적다는 것이다. 다른 하나는 여러 로봇이 이웃하더라도 로봇을 구별하는데 어려움이 없다는 것이다.



그림 16 ICRO 패치

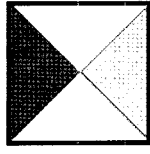


그림 17 Soty-Cross 패치

패치 디자인에 의해 각각의 로봇을 정확히 찾았다고 하더라도 한 가지 문제가 남아있다. 바로 줌렌즈에 의해서 일어나는 화면의 왜곡 현상이다.

이 문제는 카메라 영상을 영상 중심을 원점으로 하는 극좌표로 생각하여 해결할 수 있다. 이때 카메라 영상 전체에 대해 보정 연산을 하게 되면 연산 시간이 오래 걸린다. 때문에 우선 왜곡 영상에서 각 색상의 위치를 구한 후에 그에 대응되는 보정 위치를 역으로 구하게 된다. 원점에서의 거리의 제곱에 비례하여 왜곡이 심해진다고 하면 이를 이용하여 실제 위치를 역으로 계산할 수 있다.

$$D_{error} = -K_{error} (D_{camera})^2 \quad (3)$$

$$D_{real} = K_{zoom} (D_{camera} + D_{error}) \quad (4)$$

단,  $D_{camera}$ 는 카메라 영상 원점에서부터의 거리,  $D_{error}$ 는 왜곡된 거리,  $D_{real}$ 는 왜곡 보정 후 실제 거리,  $K_{error}$ 는 왜곡 보정 상수,  $K_{zoom}$ 는 화면 확대 상수를 뜻한다.

두 상수를 적절히 조절하면 렌즈에 맞는 왜곡 보정을 할 수 있다. 그림 17과 그림 18은 초기 왜곡된 화면과 왜곡을 보정한 후의 화면이다.

## 5. 결 론

로봇 축구 시스템은 하드웨어와 소프트웨어의 조합

으로 이루어진 시스템이다. 보다 작고, 빠르고, 튼튼하고, 정확히 움직이는 로봇을 만들기 위해 많은 연구와 그에 따른 많은 성과가 있었다. 또한 시간이 갈수록 보다 성능 좋은 카메라와 프레임 그레이버가 알려지고 사용되었다. 소프트웨어 시스템이라 할 수 있는 전략 생성 부분은 다개체 제어라는 연구 분야로 폭넓게 연구되고 있다.

본 논문에서는 로봇 축구 소프트웨어 시스템이 전반적인 구조를 알아보았고, 그 중에 특히 비전 처리 과정의 변화 과정 및 최근 문제점을 해결하는 방법에 대해 알아보았다. 비전 처리는 하드웨어와 소프트웨어에 걸쳐 이루어지기 때문에 로봇 축구 시스템의 가장 기본적인 면에서도 중요한 요소라고 할 수 있다. 바로 이러한 특징 때문에 비전 처리 과정은 카메라, 프레임 그레이버와 함께 발전되어왔다. 즉, 고성능의 최신 카메라 성능을 최대한 발휘할 수 있는 방향으로 비전 처리 기술이 발전하고 있는 것이다.

다중 구름형 모델, 다중 타원형 모델을 통한 색상 추출 과정은 조명 변화가 있는 환경에 맞는 좋은 모델이며 로봇 축구 시스템뿐 아니라 다른 여러 영상 처리 과정에도 응용될 수 있다. 로봇 축구에서 사용하는 다양한 패치 디자인과 발생된 문제점, 또 그에 따른 대안을 알아보았다. 끝으로 줌렌즈에서 발생하는 왜곡현상을 보정하는 방법을 살펴보고 이는 로봇축구 뿐 아니라 다른 여러 영상 처리 분야에서 이용될 수 있을 것으로 보인다.

## 참고문헌

- [1] J.-H. Kim, H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi and K.-O. Kim, "A Cooperative Multi-Agent System and Its

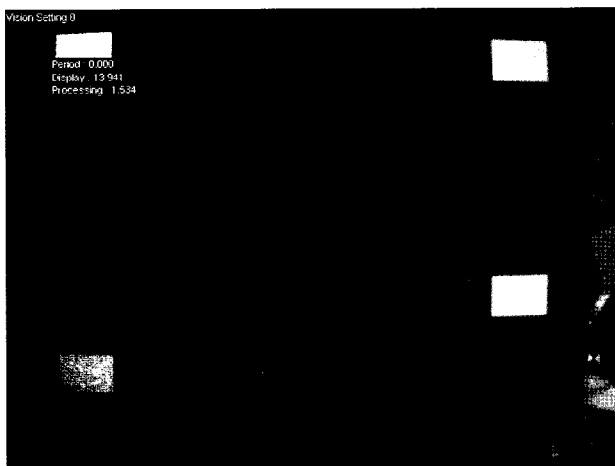


그림 18 왜곡된 화면

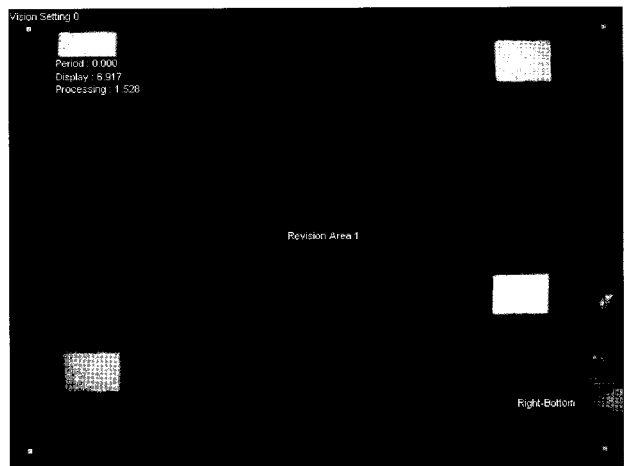


그림 18 보정 화면

- Real Time Application To Robot Soccer,” Proc. of the IEEE Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, 1997.
- [2] J.-H. Kim, H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi and K.-O. Kim, “A Cooperative Multi-Agent System and Its Real Time Application To Robot Soccer,” Proc. of the IEEE Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, 1997.
- [3] H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi, J.-H. Kim and J.-O. Kim, “Designing Distributed Control Architecture for Cooperative Multi-agent System and Its Real-Time Application to Soccer Robot,” Journal of Robotics and Autonomous System, Vol.21, No.2, pp.149-165, 10 Sep., 1997.
- [4] H.-S. Shim, M.-J. Jung, H.-S. Kim, Jong-Hwan Kim and P. Vadakkepat, “A Hybrid Control Structure for Vision Based Soccer Robot System,” Intelligent Automation and Soft Computing, Vol.6, No.1, pp.89-101, Jan., 2000.
- [5] Kang.-Hee Lee and Jong-Hwan Kim, “New MIROSOT : 5 vs 5,” Games International Symposium on Soccer Robots and Intelligent Systems,” in Proc. of FIRA RWC China’ 2001, Beijing, China, Aug., 2001.
- [6] J.-H. Kim, D.-H. Kim, Y.-J. Kim, K.-T. Seow, “Soccer Robotics,” Springer-Verlag, Jun., 2004.
- [7] 로봇 축구 공학, 김종환 외 7인 저, 2002, KAIST Press.
- [8] J.-H. Kim and Prahlad Vadakkepat, “Multi-Agent Systems: A Survey from the Robot-Soccer Perspective,” Intelligent Automation and Soft Computing, Vol.6, No.1, pp.3-18, Jan., 2000.
- [9] 김종환, 이강희, 이윤기, 구미희, 김태훈, 이재경, “Robot Soccer Technology,” 대한전자공학논문지, Vol.31, No.10, pp.1286~1300, 2004.
- [10] 이동훈, “MIROSOT의 영상시스템을 위한 새로운 칼라 패치”, 2005 로봇축구 공학 학술대회 논문집, 2005.

---

### 최승환



2005 한국과학기술원 전자전산학과  
전산학전공(공학사)  
2005~현재 한국과학기술원 전자전산학과  
전기 및 전자공학과 석사과정  
관심분야: 유비쿼터스 로봇, 유전자 로봇  
E-mail : shchoi@rit.kaist.ac.kr

### 김종환



1981 서울대학교 전자공학과(공학사)  
1983 서울대학교 전자공학과(공학석사)  
1987 서울대학교 전자공학과(공학박사)  
1988~현재 한국과학기술원 전자전산학과  
교수  
2003. 3~현재 호주 Griffith University  
겸임교수  
2005. 3~현재 필리핀 De La Salle Uni-  
versity 명예교수  
2003~현재 KAIST ITRC-지능로봇연구센터 소장  
1996~현재 IEEE Trans. on Evolutionary Computation,  
Founding Associate Editor  
관심분야: 유비쿼터스 로봇, 유전자 로봇  
E-mail : johkim@rit.kaist.ac.kr

---