

개선된 패턴매칭을 사용한 이동물체 추적

신성환*, 이진한*, 이주일**, 최한고*
*금오공과대학교 전자공학부, **에이스이노테크 연구실

Tracking of a moving object using improved pattern matching

Seunghwan Shin*, Jinhan Lee*, Juill Lee**, Hango Choi*
*School of Electronic Engineering, Kumoh National Institute. of Technology
**Research Division, Ace Innotech

Abstract

본 연구에서는 개선된 영역기반의 패턴매칭 기법을 사용하여 이동물체의 탐색과 검출을 수행하였다. 시간에 따라 변화하는 이동물체의 안정된 추적을 위해 매 영상 프레임마다 이동물체의 윤곽선을 탐지하여 다음 영상에서의 템플릿으로 사용하기 위해 갱신하였으며, 패턴매칭의 연산속도 향상을 위해 패턴 정합률에 따라 영상을 다른 비율로 압축하여 추적하는 방법을 제안하였다. 기존의 영상파일 을 사용하여 시물레이션 한 결과 이동물체의 검출과 추적에 양호한 동작을 보여주었으며 제안된 방법의 실시간 동작 가능성을 조사하였다.

1. 서론

건물의 지하주차장, 은행의 무인 자동화 창구, 산업현장 등에서 보안과 범죄예방에 대한 관심이 크게 증대되면서 감시 카메라를 이용하여 경비인력 없이 외부 침입자를 식별하고, 이동 상황을 연속으로 감시하여 움직이는 물체를 지속적으로 추적할 수 있는 영상처리 시스템의 필요성이 크게 증가되고 있다.

최근에는 무인항공기를 이용하여 높은 고도에서의 영상 촬영과 관련되어 영상처리가 사용되고 있다. 군사용으로는 초소형 무인정찰기에 사용되고, 민수용으로는 농업분야의 공중 비료 살포 및 작물 생육 현황 조사, 재난사고 예방, 재해 현장 촬영, 등산 조난자 구호, 범죄자 추적 등에서 사용되고 있다. 특히, 이동물체를 식별하고 추적 가능한 시스템의 개발 가능성이 여러 분야에서 요구되므로 이와 관련된 연구가 꾸준히 이루어지고 있다.

고정된 위치에서의 카메라로부터 획득한 영상에서 이동물체를 식별하고 추적하는 감시 시스템에는 물체가 카메라 시야 영역에서 벗어나면 이동물체 추적을 더 이상 지속할 수 없다[1]. 이를 극복하기 위해 카메라 방향 구동이 가능한 능동 카메라를 통해 근거리에서 이동물체를 지속적으로 추적하는 방법이 개발되고 있다.

본 논문에서는 실시간 이동물체 추적을 위해 영상처리 시간을 단축시키기 위한 방법을 제안하고 이를 이용하여 영상처리 시스템을 설계하였다. 설계를 위해 능동카메라에서 획득한 영상을 이용하여 움직이는 물체를 검출하고 움직임 영역을 추출하여 이동물체를 연속적으로 추적할 수 있는 시물레이션을 구현하였으며, 샘플 데이터를 사용하여 실시간 동작 가능성을 조사하였다.

2. 제안된 이동물체 추적 알고리즘

본 논문에서는 다음과 같이 개선된 이동물체의 검출 및 추적 방법을 제안한다. 첫째, 초기 이동물체 추적을 시작할 때 이동물체의 검출을 위해서 사각형을 직접 이동물체에 대한 초기템플릿을 생성한다. 둘째, 시간에 따라 크기와 모양이 변화하는 템플릿의 검출을 위해서 매 프레임에서 이동물체의 윤곽선을 검출하여 템플릿을 갱신한다. 셋째, 영역기반의 정합 알고리즘의 속도개선을 위해 패턴 정합률에 따라 영상 압축률을 자동 조절한다.

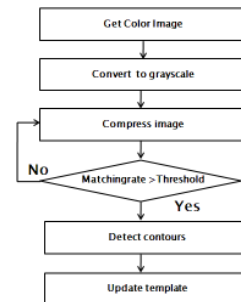


그림 1. 제안된 알고리즘 흐름도

전체적인 알고리즘 흐름은 그림 1에 도시하였다. 연속적으로 입력되는 영상을 통해 이동물체의 움직임을 검출한다. 참조 컬러영상에서 초기에 설정된 템플릿과의 정합하는 영상영역을 효과적으로 검출하기 위해 이미지 전처리과정을 거친다. 전처리 과정은 색상값 계산을 통해 컬러 영상을 그레이스케일로 변환한 후 템플릿 영역이 참조영상의 관심영역(Region of Interest) 내에서 비교하는데 영역기반 정합을 통해 가장 유사한 영역을 찾는다. 탐색된 유사영역의 중심점을 기준으로 이동물체의 윤곽선을 검출하고 윤곽선 외곽

을 에워싸는 최소 사각형 영상을 새로운 이동물체의 템플릿으로 정의하고 이전 템플릿과 대체함으로써 템플릿을 갱신한다.

2.1 그레이스케일 영상 변환

카메라 영상을 통해 이동물체의 움직임을 감지하고 추적하는 작업은 현실 세계에 존재하는 영상을 그대로 획득하므로 수많은 변수와 여러 가능성에 노출되어 있다. 때문에 움직임에만 전적으로 의존하는 시스템은 빛으로 주변 환경이 변화된 것처럼 보여 시스템에 오류를 일으킬 수밖에 없다. 따라서 움직임 외에 색상 값으로 계산하여 추적을 보정하는 작업이 필요하다. 본 시스템에서는 템플릿영역의 색 분석을 통해 RGB의 비율로 계산함으로써 보다 정교한 추적을 가능하게 한다. R, G, B 비율을 구하는데 식 2.1을 사용하였다[2].

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B} \quad (2.1)$$

입력영상인 컬러영상을 영상처리를 위해 그레이스케일로 변환하는데 식 2.2를 사용하였으며, 참조영상 전 영역에 적용하기 때문에 템플릿 영역이 그 외 영역에 비해 밝게 부각되는 효과를 갖는다. 템플릿 영역안의 윤곽선 영상을 뚜렷하게 하고 주변 노이즈 영향을 줄일 수 있다.

$$gray = \frac{r^2}{r^2+g^2+b^2}R + \frac{g^2}{r^2+g^2+b^2}G + \frac{b^2}{r^2+g^2+b^2}B \quad (2.2)$$

2.2 영역기반 정합(Region based pattern matching)

영역기반 정합은 화소의 밝기값(gray levels)을 정합요소로 하여 유사성 기준에 따라 행과 열을 탐색하면서 가장 유사한 영역을 찾는 것이다. 대표적인 유사성 기준에는 Cross-correlation 방법이 있다.

식 2.3은 Cross-correlation 연산을 나타내는 함수로서, $T(i,j)$ 는 템플릿 영상을 $M(i,j)$ 는 참조영상을, M, N 은 탐색영역내의 행과 열을 의미한다. 식 2.3의 결과는 템플릿 영상과 비교 대상 물체의 최소 경계영역 영상을 픽셀단위로 곱한 후 합산하여 구한다. 이 방법은 특히 잡음이 존재하는 영상 내에서 추적물체를 정확히 검출하는데 널리 사용된다. 식 2.3은 합산하여 얻은 결과중에 최대값을 갖는 물체의 최소 경계영역 위치를 추적물체의 현재 위치로 결정한다[1]. 이에 비해 본 연구에서 비교함수로 사용된 Correlation coefficient method인 식 2.4는 식 2.3의 상관계수를 정규화(Normalize)한 것이다.

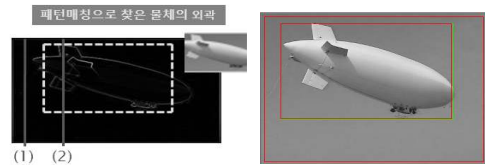
또한, 템플릿 비교함수를 적용하는데 탐색영역은 참조영상 $M(i,j)$ 의 템플릿에 따라 변하도록 하였다. 참조영상에서 템플릿과 같은 물체를 찾을 때 참조영상의 전 영역에 걸쳐 검색하면 연산량이 커지기 때문에 탐색하는 영역을 템플릿의 중심점을 기준으로 템플릿 영역의 4배 영역을 관심영역으로 설정하였다. 실제 계산에서는 참조영상 $M(i,j)$ 부분은 관심영역으로 대체되어 계산된다. 이는 이동물체의 움직임이 템플릿 크기 이상 이동하지 않는다는 가정을 전제하였다.

$$D(k,l) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} M(i+k,j+l) T(i,j) \quad (2.3)$$

$$D(k,l) = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} M(i+k,j+l) T(i,j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} M^2(i+k,j+l) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} T^2(i,j)} \quad (2.4)$$

2.3 가변 템플릿

시간에 따라 모양이나 크기가 변화하는 이동물체를 추적하기 위해 매 프레임마다 템플릿을 갱신하여야 한다. 만약 고정된 템플릿으로 갱신하게 되면 이동물체의 크기가 변해 물체의 일부만을 포함하는 템플릿으로 얻거나 배경영역이 많은 부분을 차지하는 템플릿을 얻게 되는 문제점이 있다. 이런 개선하기 위해 템플릿을 이동물체의 모양에 맞도록 높이와 넓이를 자동으로 변화시켜 템플릿을 갱신하는 것이다. 템플릿 물체의 Edge를 찾는 경우 그림 2에 나타난 것처럼 템플릿 영역에 물체의 윤곽이 제대로 나타내지 못하는 경우가 생긴다. 그러므로 패턴매칭 결과인 그림 2의 점선영역을 포함하는 (1)과 (2)사이에서 새로운 Edge를 구한다.



(a) Edge 탐지 전 (b) Edge 탐지 후
그림 2. 템플릿 검출

검출과정은 참조영상의 관심영역 내에서 Sobel mask를 적용하여 그림 2(a)와 같이 물체의 윤곽선 영상을 구하고 이전 프레임으로부터 구한 임계값을 이용하여 상하좌우 Edge를 찾는다. 구해진 윤곽선 영상에서 물체의 중심점에서 바깥방향으로 한 라인씩 검사하여 가장 바깥에 있는 라인을 찾아 Edge로 한다. 이때 검사하는 라인이 유효한 라인이 되려면 라인을 이루고 있는 픽셀의 밝기 값이 정해진 임계값 보다 큰 픽셀의 개수를 구하여 그 개수가 이전 프레임에서 구해진 임계값보다 크면 유효한 라인이 된다. 이때 배경영상에서 수목의 흔들림이나 미세한 윤곽을 갖는 잡음이 존재할 수 있으므로 바깥 선을 기준으로 4라인이 연속으로 유효할 때 Edge로 판별하였다. 물체의 중심점에서 상하좌우로 구한 각 Edge를 기준으로 직선으로 하는 사각형을 구하여(그림 2(b) 참조) 이를 템플릿으로 사용하였다.

2.4 영상 정합률에 따른 압축

참조영상의 관심영역에서 템플릿 영역과 가장 유사성이 높은 부분을 찾기 위해 패턴매칭을 사용하였다. 이 상관관계 판별은 Normalized cross-correlation 수식을 이용하였다. 이 수식은 영상의 유사관계를 정규화하여 0-1사이의 숫자로 표시한다. 표시된 수치는 정합률로서 0은 전혀 일치하지 않는 경우이고 1은 완전히 일치하는 경우를 의미한다.

패턴매칭을 영상전체에 걸쳐서 수행할 경우 많은 연산이 요구되므로 실시간이나 빠른 응답시간을 기대하는 시스템에서 사용하기가 부적절하다. 따라서 연산횟수를 줄이기 위해 영상을 압축하는 방법을 사용하였다. 보통 1/4 압축을 많이 사용하며, 1/16 압축도 사용가능하나 이러한 압축은 템플릿 영상의 높이와 넓이가 각각 1/4로 감소되므로 템플릿 패턴의 특성이 많이 축소되므로 복잡한 패턴에 대해서는 높은 정합률을 보장할 수 없다. 물론 물체의 움직임이나 크기 변화가 많지 않은 경우 비교적 간단한 패턴으로도 정합하는 부분을 쉽게 찾을 수 있다.

본 연구에서는 패턴매칭의 결과에 따라 압축률을 다르게 적용하는 방법을 제안하였다. 정합률을 따라 3단계로 구별하였는데, 높은 경우는 압축을 1/16로, 낮은 경우는 압축을 수행하지 않았으며 중간 경우는 1/4로 압축하는 방법을 사용하였다. 제안된 방법을 적용할 경우 움직임이 없는 구간에서는 1/16로 빠른 처리가 가능하고, 움직임이 많고 추적이 어려운 구간에서 압축을 하지 않음으로써 보다 정확한 추적이 가능하게 되기 때문에 패턴매칭 정확도와 속도를 동시에 향상시킬 수 있다.

3. 실험결과 및 분석

제안한 방법의 타당성을 검증하기 위하여 도로를 주행하는 자동차, 공중을 비행하는 기구[OK 항공촬영], 보행자에 대하여 약 1/30초의 프레임 속도로 촬영한 영상에서 이동물체의 추적실험을 수행하였다. 영상 사이즈는 320x240이며 각각 112 프레임, 1,000 프레임, 400 프레임 영상에 대하여 이동물체 추적을 수행하였다. 이 영상추적은 Intel Core2 duo 3.00GHz에서 Visual C++ 6.0을 사용하여 구현하였다.

초기 영상에서 템플릿은 마우스를 드래그하여 선택하였는데, 그림 3의 왼쪽은 이동물체의 윤곽선 영상이며, 오른쪽은 초기 템플릿 모델을 나타내는 영상을 각각 보여주고 있다. 제안된 방법에 대한 성능을 비교하기 위해 같은 영상에 대해 템플릿을 고정된 경우와 가변한 경우와 추적결과를 비교하였다.

먼저 고정된 템플릿으로 추적한 경우 카메라와 물체와의 거리에 따라 영상에서 물체의 크기가 변하게 된다. 그림 4에서 내부 사각형은 템플릿 영역을 외부 사각형은 탐색영역을 나타내는데, 초기에 템플릿 영역이 물체 크기에 맞게 설정되었으나 시간이 지남에 따라 물체의 원근에 변화가 있거나 방향이 변할 경우 템플릿 영역이 물체의 일부 또는 물체보다 더 큰 영역이 포함된다. 그림 4(a),(b)는 전자의 경우로서 물체 일부가 템플릿에 포함되었으며, 그림 4(c),(d)는 후자의 경우로서 배경영상이 템플릿의 상당부분에 포함된다.

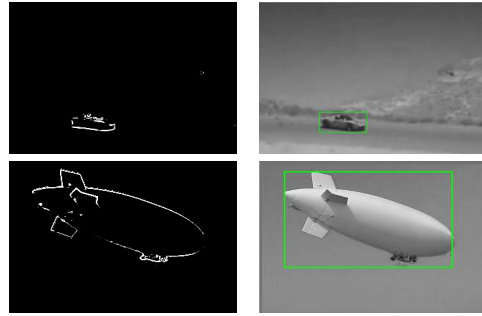


그림 3. 초기영상에서의 템플릿(1st frame)

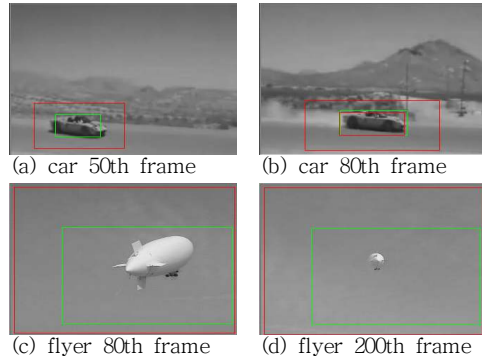


그림 4. 고정 템플릿을 사용한 추적 결과

가변 템플릿의 경우 매 프레임 윤곽선 영상에서 Edge를 구하여 템플릿을 갱신한다. 그림 5에서와 같이 변하는 물체의 크기에 맞게 템플릿이 설정됨을 알 수 있다. 그러나 가변 템플릿은 윤곽선 영상을 구한 후 원 영상에서 Edge를 구하기 때문에 속도 저하가 발생할 수 있다.

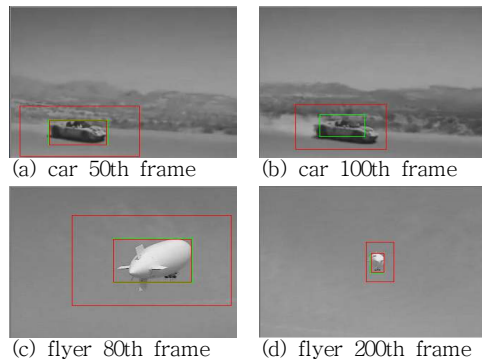


그림 5. 가변 템플릿을 사용한 추적 결과

그림 6은 가변 템플릿의 속도저하 문제를 해결하기 위해 본 연구에서 제안한 방법을 사용한 추적결과로서 이동물체를 빠른 시간 내에 추적함을 확인할 수 있었다. 시뮬레이션 결과에 의하면 3가지 영상파일에 대해 템플릿의 크기와 정합률에 따른 영상압축이 매 프레임마다 자동으로 변화하면서 물체의 검출과 추적이 양호하였다.

