

적응형 칼라 모델을 이용한 Active Contour

박헌근, 정명진
한국과학기술원 전자 전산학과

Active Contour using Adaptive Color Model

Hyun Keun Park, Myung Jin Chung
Department of Electrical Engineering & Computer Science, KAIST

Abstract - Active contour로 알려져 있는 snake는 반복적인 계산으로 이미지 상에서 찾고자 하는 물체의 외곽선에 수렴하는 contour로 이미지 상의 물체의 외곽선으로부터 발생하는 외부 에너지(external energy)와 contour 자체로부터 기인하는 내부 에너지(internal energy)를 최소화하는 방향으로 움직인다. 그러나 물체의 윤곽선으로부터 발생하는 외부 에너지는 찾고자 하는 물체뿐만 아니라 주위의 다른 물체로부터도 발생하므로 단일 추적하고자 하는 물체의 주변에 다른 물체들이 존재한다면 snake은 올바르게 동작하지 않게 된다. 본 논문에서는 이러한 단점을 극복하기 위하여 물체의 색상 정보를 이용하는 방식을 제안하였다. 물체의 색상 정보는 물체의 고유한 특성 중의 하나로 본 논문에서는 색상 정보를 이용하여 원래의 이미지를 찾고자 하는 물체의 색상과 얼마나 유사한가를 나타내는 확률 이미지로 변환하였다. 이렇게 변환된 확률 이미지 상에서 snake 알고리즘을 적용함으로써 배경의 다른 물체로부터 발생하는 외부 에너지를 효과적으로 제거할 수 있다. 또한 본 논문에서는 물체가 이동함에 따라 변화하는 색상 정보를 지속적으로 갱신함으로써 물체의 추적이 효과적으로 이루어지도록 하였다.

은 연구가 물체의 추적에 이러한 물체의 색상 정보를 이용하고 있다[4][5][6][7]. 그러나 물체의 모양이나 주변의 조명 상태, 물체의 반사도 등에 따라 동일한 색상 일지라도 실제 카메라로 받아 들인 이미지 상에서는 서로 다르게 나타나기 때문에 색상 정보를 효과적으로 사용하기 위하여는 이러한 변화를 적절히 모델링할 수 있는 방법이 있어야 한다[4][5]. 또한 기존의 색상을 이용한 알고리즘에서는 찾고자 하는 색상과 그렇지 않은 색상을 구분하는 경계값(threshold)를 결정하는 것이 중요한 문제 중의 하나였다[5]. 본 논문에서는 이러한 기존의 문제점들을 해결하기 위하여 카메라로부터 받아들인 이미지를 색상 정보를 이용하여 찾고자 하는 색상과 얼마나 유사한지를 나타내는 확률 이미지로 변환시킨 후 확률 이미지 상에서 snake 알고리즘을 적용하였다. 색상 정보를 이용하여 초기의 이미지로부터 확률 이미지로 변환하면 찾는 색상을 제외한 다른 색상들은 모두 제거되어 찾고자 하는 색상을 지닌 물체만 나타나므로 윤곽선을 확실히 분리해 낼 수 있다. 또한 확률 이미지 상에서 snake 알고리즘을 적용하면 물체의 외곽선을 찾는 데 경계값을 설정할 필요가 없으므로 기존의 색상 정보만을 이용하여 물체를 찾을 때 발생하는 경계값의 문제를 해결할 수 있다. 본 논문에서는 또한 찾아진 contour 안의 색상을 이용하여 물체의 색상이 지속적으로 갱신하여 주위 환경에 따른 색상의 변화에 대처하였다.

1. 서 론

물체의 윤곽선 추출, 물체의 추적 및 인식, 물체의 움직임 분석, 스테레오 비전의 매칭 등 다양한 분야에서 응용되고 있는 snake는 형태가 가변적인 contour로 외부 에너지와 내부 에너지로 구성된 전체 에너지가 최소화되는 방향으로 움직인다[1][2][3]. 전체 에너지 중 내부 에너지를 contour 자체의 모양에서 기인하며 내부 에너지를 최소화시키는 것은 contour의 각 노드 사이의 간격을 일정하게 유지시켜 주어 contour의 모양이 전체적으로 부드러운 형태를 유지하도록 하는 것과 contour의 각 노드 사이의 간격을 축소시켜 contour가 전체적으로 중앙으로 줄어들도록 하는 역할을 한다. 반면 외부 에너지는 이미지 상의 물체의 윤곽선으로부터 발생하는 것으로 외부 에너지를 최소화시키는 것은 contour를 물체의 윤곽선 방향으로 밀어 넣는 역할을 한다. 따라서 snake는 에너지 최소화 기법에 따라 외부 에너지로부터 발생하는 외력(external force)과 contour 자체로부터 발생하는 내력(internal force)에 의하여 이동하게 된다. 그러나 snake가 물체의 외곽선에 수렴하도록 만드는 외력의 경우 물체의 윤곽선으로부터 발생되기 때문에 우리가 추적하고자 하는 물체의 주변에 다른 물체들이 존재할 경우 이러한 물체로부터도 외력이 발생하게 된다. 따라서 원하지 않는 외부의 외력이 작용할 경우 snake 알고리즘을 적용하기 어렵게 된다. 본 논문에서는 이러한 snake의 단점을 극복하기 위하여 물체의 색상 정보를 이용하였다. 물체의 색상 정보는 물체의 고유한 성질 중의 하나로 물체의 특징을 나타내는 중요한 정보이다. 따라서 이러한 특성을 이용하면 물체를 추적하는데 많은 도움을 얻을 수 있으며, 또한 많

2. 본 론

2.1 Snake: Active Contour Model

Snake는 정의된 내부 에너지와 외부 에너지를 최소화 시키는 방향으로 움직이는 contour로 snake의 전체 에너지는 다음과 같이 정의된다.

$$E_{total} = \int (E_{ext}(v(s)) + E_{int}(v(s))) ds \quad (1)$$

위의 수식에서 $v(s) = (x(s), y(s))$ 로 contour를 나타내며 E_{ext}, E_{int} 는 각각 외부 에너지와 내부 에너지를 나타낸다.

$$E_{int}(v(s)) = \frac{\alpha |v_s(s)|^2 + \beta |v_{ss}(s)|^2}{2} \quad (2)$$

내부 에너지는 수식 (2)와 같으며 내부 에너지 중 $v_s(s), v_{ss}(s)$ 는 각각 $v(s)$ 의 s에 대한 1, 2차 미분이다. α, β 는 각각 각 항목에 대한 중요도를 나타낸다. 외부 에너지는 아래의 수식 (3)으로 표현되며 여기서 $E_{line} = I(x, y), E_{edge} = |\nabla I(x, y)|$ 를 각각 나타낸다. 본 논문에서는 $w_{line} = 0$ 로 하여 이미지의 밝기값의 미분치만을 사용하였다.

$$E_{ext} = w_{line} * E_{line} + w_{edge} * E_{edge} \quad (3)$$

그림 [1]은 초기 이미지에 대하여 밝기값의 미분을 구한 이미지이다. 그림 [1]에서 찾고자하는 물체는 우측 중앙의 분홍색 원형 물체이지만 외곽선을 구한 이미지에서는 많은 주변 물체의 외곽선이 나타남을 알 수 있다. 이러한 외곽선은 외부 에너지로 작용하여 contour가 올바른 외곽선에 수렴하는 것을 방해한다.

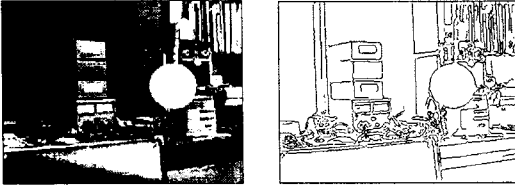


그림 [1]. 초기 이미지와 외곽선 추출

2.2 색상 모델링과 확률 이미지

물체의 색상은 물체의 고유한 성질 중의 하나로 물체를 추적하는데 색상 정보를 이용하면 물체를 주위 배경으로부터 쉽게 분리해 낼 수 있다. 그러나 실제 동일한 색상을 지닌 물체일지라도 물체의 모양이나 물체의 반사도, 카메라의 특성, 주위 조명의 변화 등에 의하여 이미지상에서는 서로 다른 색상으로 나타나므로 주변 환경이 변화하는 상태에서는 이미지 속의 물체의 색상도 변화하게 된다. 따라서 색상 정보를 이용하여 물체를 분리해내고자 할 경우 이러한 주변 환경의 변화에 따른 색상의 변화를 수용할 수 있는 방법이 필요하다.

본 논문에서는 물체의 색상에 대한 정보를 RGB 색상 공간상에서 3차의 정규 분포로써 모델링하고 주변 환경의 변화에 따라 적절히 변화시키므로써 색상의 변화를 수용할 수 있도록 하였다[5]. 이 방법에서 (r, g, b) 값을 갖는 이미지 상의 한 점은 R, G, B를 각각의 축으로 하는 RGB 색상 공간 상의 한점으로 변환된다. 만일 이미지 상에서 동일한 색상을 갖는 점들이 있다면 그 점들은 모두 RGB 색상 공간 상에서는 기하학적으로 동일한 점에 매핑이 될 것이다. 마찬가지로 비슷한 색상을 지닌 점들은 RGB 색상 공간 상에서 기하학적으로 비슷한 위치에 모여 있게 된다. 동일한 색상이 조명의 변화나 물체의 모양, 위치의 변화에 의하여 변화하는 경우 변화된 색상은 모두 원래의 색상과 비슷한 색상을 지니므로 RGB 색상 공간상에서 보면 모두 기하학적으로 비슷한 위치에 모여 있는 점들로 표현된다. 그림 [2]는 실제 획득된 영상에서 원형의 물체(분홍색)의 색상을 RGB 색상 공간상에서 표현한 그림으로 색상이 대부분 모여 있으며 타원형의 형태를 유지하고 있음을 알 수 있다.

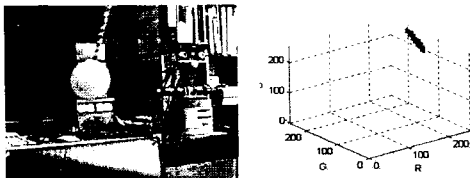


그림 [2]. RGB 색상 공간상에서의 색상의 분포

일반적으로 색상의 변화는 조명의 변화에 따른 변화가 가장 크므로 색상 공간상에서의 색상의 분포는 가장 밝은 값을 나타내는 (1, 1, 1)에서 가장 어두운 값을 나타내는 (0, 0, 0)의 방향으로 타원형의 튜브 형태로 나타나게 된다. 따라서 이러한 색상의 분포는 3차원의 정규분포로 적절히 모델링될 수 있다. 다음 수식(4)는 색상 공간에서 색상을 모델링한 정규분포의 식이다.

$$n(v) = \frac{1}{\sqrt{(2\pi)^N \det \Sigma}} \exp\left(-\frac{1}{2}(v - \bar{\mu})^T \Sigma^{-1}(v - \bar{\mu})\right) \quad (4)$$

수식(4)에서 $v = (r, g, b)^T$ 이며 $\bar{\mu}, \Sigma$ 는 각각 색상 분포의 평균값과 분산 행렬을 나타낸다.

$$\Sigma = \begin{bmatrix} \sigma_R^2 & Cov(R, G) & Cov(R, B) \\ Cov(G, R) & \sigma_G^2 & Cov(G, B) \\ Cov(B, R) & Cov(B, G) & \sigma_B^2 \end{bmatrix} \quad (5)$$

수식(4)을 사용하면 $v = (r, g, b)^T$ 3차원의 벡터로 표현된 이미지상의 모든 점을 1차원의 확률값 $n(v)$ 을 갖는 영상으로 변환할 수 있다. 이 영상을 확률 영상이라고 하자.

아래의 그림은 사전에 정의된 $\bar{\mu}, \Sigma$ 를 이용한 수식(4)를 이용하여 임의의 영상을 확률 영상으로 변환한 것이다. $\bar{\mu}, \Sigma$ 를 이미지 중앙의 원형 물체의 색상(분홍색)으로 설정하여 놓았기 때문에 변환된 확률 영상에서는 원형의 물체 부분은 큰 값을 갖으며 물체를 제외한 다른 부분은 작은 값을 갖는다. (그림[3]에서는 어두운 부분이 큰 값을 나타낸다.)

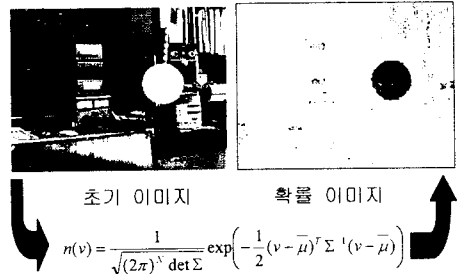


그림 [3]. 확률 이미지로의 변환

2.3 확률 이미지에서의 snake 알고리즘의 적용

확률 이미지는 이미지 상의 각각의 점들이 우리가 찾고자 하는 색상에 얼마나 유사한지를 나타낸다고 할 수 있다. 따라서 찾고자 하는 색상에 가까운 색상은 확률 이미지 상에서는 큰 값을 갖으며 찾고자 하는 색상과 관련이 적은 색상은 확률 이미지상에서 작은 값을 갖게 된다. 따라서 확률 이미지 상에서는 우리가 찾고자 하는 색상만이 뚜렷하게 나타나며 나머지 색상들은 작은 값을 갖는다. 그러므로 확률 이미지 상에서 물체의 외곽선을 구하면 찾고자 하는 물체의 외곽선만이 나타나며 주변이 배경 이미지상에서의 외곽선은 소거되는 효과가 나타난다. 본 논문에서는 확률 이미지 상에서 snake 알고리즘을 적용하여 이미지상의 배경 물체로부터 발생하는 외곽을 제거할 수 있다. 본 논문에서 사용된 색상 정보를 포함한 snake의 외부 에너지는 다음과 같이 표현된다고 할 수 있다.

$$E_{ext} = \left| \frac{1}{\sqrt{(2\pi)^N \det \Sigma}} \nabla \exp\left(-\frac{1}{2}(v - \bar{\mu})^T \Sigma^{-1}(v - \bar{\mu})\right) \right| \quad (6)$$

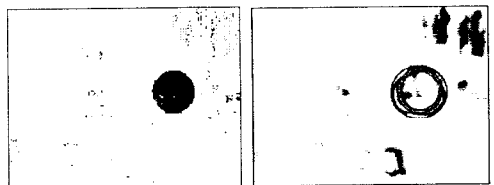


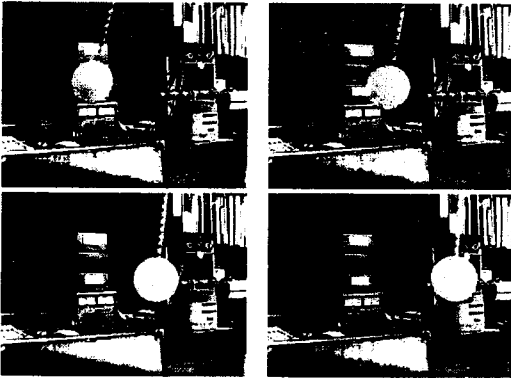
그림 [4]. 확률 이미지에서의 snake의 적용

그림(4)는 확률 이미지와 확률 이미지에 적용된 snake 알고리즘으로 배경 물체가 거의 제거되어 찾고자 하는 물체의 외곽선을 확실하게 구분할 수 있음을 알 수 있다.

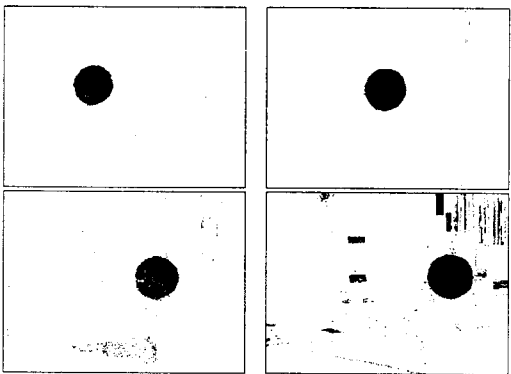
2.4 색상 모델의 갱신

본 논문에서 적용된 전체 알고리즘은 다음과 같다. 먼저 이미지상에서 찾고자 하는 물체의 색상 영역을 선택한다. 선택된 영역의 색상들로부터 평균값과 분산 행렬 μ, Σ 를 계산한다. 초기에 계산된 μ, Σ 와 수식(4)를 이용하면 초기 이미지를 확률 이미지로 변환할 수 있다. 변환된 이미지는 그림(3)과 같이 찾고자 하는 물체만이 또렷이 나타나는 확률 이미지일 것이다. 이 이미지에서 snake 알고리즘을 적용한다. 찾고자하는 물체만이 나타난 이미지이므로 쉽게 snake 알고리즘이 적용될 수 있을 것이다. 물체의 외곽선에 contour가 수렴을 하면 다음번 알고리즘 적용을 위하여 contour 내부의 점들을 이용하여 다시 μ, Σ 를 계산한다. 이 과정을 반복하면 물체가 이동하여 물체의 색상이 변화하더라도 μ, Σ 를 지속적으로 갱신하여 찾고자하는 물체의 contour를 추적할 수 있다.

2.4 실험



(a) 초기 이미지



(b) 확률 이미지



(c) snake가 적용된 이미지
그림 (5). 실험 결과

그림 (5)는 실험 결과를 보여 준다. 그림 (5)-(a)는 실험에 사용한 연속된 이미지중 10, 15, 20, 25번째 이미지이며 그림 (5)-(b)는 각각의 (a) 이미지에 대한 확률 이미지이다. 마지막 그림 (5)-(c)는 확률 이미지에 snake를 적용하여 물체의 외곽선을 찾은 이미지이다.

3. 결 론

본 논문에서는 snake 알고리즘을 적용하는데 있어서 원하지 않는 외부 물체의 영향을 줄이기 위하여 물체의 색상 정보를 이용하였다. 물체의 색상 정보는 물체의 고유한 특성으로 본 논문에서는 추적하고자 하는 물체의 색상 정보를 이용하여 초기 이미지를 색상값에 대한 확률을 나타내는 확률 이미지로 변환하여 확률 이미지 상에서 snake 알고리즘을 적용하였다. 이러한 방법은 기존의 snake 알고리즘의 문제점이었던 주위의 불필요한 외곽선들로부터 발생하는 외력을 제거할 수 있을 뿐 아니라 단순히 색상 정보만을 이용하여 확률 이미지 상에서 물체를 판별하는 방법에서 발생하는 경계값 설정의 문제를 해결할 수 있다. 또한 찾아진 물체의 내부의 색상값을 이용하여 다음 번 색상값의 변환에 이용되어지므로 항상 가장 최근의 추적하는 물체의 색상값을 묘사하게 되어 색상값이 변화하더라도 적절히 적용할 수 있게 된다. 또한 가장 최근에 변화된 색상값을 이용하는 이러한 방법은 확률 이미지 상에서 현재 추적하는 물체의 색상을 가장 큰 값을 갖도록 하기 때문에 단순히 밝기값을 이용한 이미지에서 외곽선을 찾는 것보다 효과적이라고 할 수 있다.

(참 고 문 헌)

- [1] M. Kass, A. Witkin and D. Terzopoulos, "Snake: Active Contour models", *Int. Journal of Computer Vision*, pp. 321-331, 1988
- [2] Gunn, S.R.; Nixon, M.S. "A robust snake implementation: a dual active contour", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol 19, pp. 63 -68, 1997. 1
- [3] Brigger, P.; Hoeg, J.; Unser, M., "B-spline snakes: a flexible tool for parametric contour detection", *IEEE Trans. Image Processing*, vol. pp. 1484 -1496 2000. 9
- [4] 박현근, 김도윤, 정명진, "적용형 칼라 Look-up Table을 이용한 물체의 추적", 대한전기학회 하계 학술 대회 논문집, pp. 2714-2716. 2000. 7
- [5] R.Schuster, "Color Object Tracking with Adaptive Modeling," *Proceedings of IEEE Symposium on Visual Languages*, pp.91-96, 1994
- [6] T.D.Grove, K.D.Baker, T.N.Tan, "Colour Based Object Trackng," *Int. Conf. Pattern Recognition*, vol. 2, pp. 1442-1444, 1998
- [7] D.-Y.Kim, H.K.Park, M.J.Chung, "Robust and Fast Color-Detection using a Look-up Table," *Proceedings of International Symposium On Artificial Life and Robots*, vol. 2, pp.650-653, 1999