영역 최적화를 이용한 움직이는 물체의 추적

장진필, 윤태수, 양양규
동서대학교 인터넷공학부
xdms2000@hanmail.net

Moving Object Tracking Using Region Optimization

Won Dal Jang, Tae Sook Yun, Hwang Kyu Yang
Division of Internet Engineering, Dongseo University

요약
본 논문에서는 고정된 카메라로부터 입력된 연속 영상에서 움직이는 오브젝트의 영역을 추적하는 방법을 제안한다. 제안된 방법은 영상의 분석에 의해 오브젝트의 움직임을 관찰하는 단계와 연관된 검사 방법에 의하여 오브젝트의 영역을 찾아내는 단계가 포함되어 있다. 제안된 방법은 카메라로부터 입력된 연속영상들을 대상으로 실제현장 결과 오브젝트의 형태변화에 따른 고정된 최적영역을 찾는 것이 주목할 만하다.

1. 서론


본 논문에서는 연속 영상 속 오브젝트의 정확한 움직임을 추적하는 방법을 제안한다. 오브젝트의 움직임은 연속 영상에서 추적할 때, 오브젝트의 영역을 찾아내는 것이 단계이며 이는 영역 추적을 하기 위해서 오브젝트의 영역을 추적함으로 이루어진다. 제안된 방법의 첫 번째 단계

프레임 k

물적 영 감지

프레임 k+1

오브젝트 영역 찾기

영역 추적 카메라

그림 1. 제안된 방법의 개요

2. 움직임 감지

움직임 감지 단계에서는 시공간 정보를 이용한다. [5]
두 장치 연속 영상 속 오브젝트는 동일한 영상 기간 동안 움직임을 가질 것이다. 선형 프레임과 비교적 정확한 영역을 추적한다. 프레임의 영역 감지 및 추적 단계는 다음과 같다.

\[
I(x, y) - I_{k-1}(x, y) = I_{k+1}(x, y) - I_{k-1}(x, y)
\]

\[
I_{k+1}(x, y) - I_{k-1}(x, y) = I_{k+2}(x, y) - I_{k+1}(x, y)
\]

따라서 \( I \)와 \( I_{k+1} \)의 영역을 구하고 영역을 추적한다. 시공간 정보를 나타내는 영역을 다음과 같이 정의한다. \( t_k \)는 움직임을 이전화하는 업계이다.

\[
M_k(x, y) = \begin{cases} 1, & \text{if } (I_k(x, y) - I_{k+1}(x, y)) \leq t_k \\ 0, & \text{otherwise} \end{cases}
\]

시공간 정보를 나타내는 영역은 영역의 움직임에 따라 영역의 변화가 유의미한 변화로 인해 영역의 영역이 변한다. 영역의 변화 정보는 노이즈가 포함되어 있다.

따라서 노이즈를 효과적으로 제거하기 위해 영상을 이 
영역에서 적절한 업계값을 결정하는 것이 중요하다.
3. 오브젝트 영역 찾아기

이번 단계에서 추출한 육안영역(Mq) 육안영역에 의해 볼도 같이 변화한 영역들을 나타낸다. 이 단계에서는 육안영역 정보에 붙인 동작이 신호들은 연결성 검사를 수행하여 육안영상 연결된 영역들을 모두 찾아낸다.

![그림 2. 연결성 검사에 수행한 결과영상](image)

그림 2. 연결성 검사를 수행한 결과영상

그림 2는 가래로부터 입력된 영상과 본 단계에서 수행한 결과 영상을 나타낸다. 연결성 검사를 수행하여 육안영역 연결된 영역들을 모두 찾아낸다. 그림 (c)에서 추출된 연결된 영역들이 나타난다. 볼도근의 변화가 큰 영역들은 연결된 영역들로, 많은 영역들이 이 단계에서 구하려는 영역인 seeking 영역을 Box영역으로 판정한다. 연결된 영역들 Box에 포함되어 있을 경우 Box영역 feasible box에 포함된다. 여기서 C는 연결된 영역들의 집합이고 RC는 연결된 영역의 너비 c로 포함되는 최소 크기의 사각형(rectangle 영역)의 집합이다. rc는 Mq의 수직 위치 (xy) 및 너비 w, 높이 h로 표현되어진다.

4. 최적영역 찾기

마지막 단계에서는 오브젝트 영역을 오브젝트 영역 단계에서 최적화된 영역 도출하기 위해 사용한다. Mq에 있는 object와 Box 그리고 이 단계에서 육안영역 우주 영역 box와의 관계를 그림으로 표현해본다. 그림 3. 오브젝트 영역과 최적영역

![그림 3. 오브젝트 영역과 최적영역](image)

그림 3. 오브젝트 영역과 최적영역

그리고에서 바란작 사각형은 오브젝트 영역(Box)이고 이 영역 내부의 점으로 정하는 오브젝트(object) 영역 내부의 각점들은 최적 영역(Region) 영역으로 정한다. 영역에 노이즈가 심하게 존재하고 있지만, 최적영역을 찾는 영역 최적화(Region Optimization) 알고리즘은 다양한 영역 내 오브젝트 중심을 종합 해석을 포함하여 추출하기를 함으로써 줄어간다. 그림 4는 본 논문에서 제시한 영역 최적화 알고리즘을 나타낸다.

![그림 4. 영역 최적화(Region Optimization) 알고리즘](image)

그림 4. 영역 최적화(Region Optimization) 알고리즘

fBox: Box 영역 존재하는 오브젝트의 비율이며 수식 (7)과 같이 표현된다.

\[ f_{box}(Box) = \sum_{i=1}^{M} \sum_{j=1}^{N} M_i j \cdot w \cdot h \quad 0 \leq f_{box} \leq 1 \]  

\( \sigma \)는 Box가 결정지어주는 0과 1사이의 수치이며 영역 체크를 위한 상행영역, 하행영역, 좌방영역, 우방 영역은 그림 5에서 나타내기와 같이 \( a \)에 의해 결정된다. 이 값은 교차하는 최적영역을 풀어 냊을 수 있지만 도출되지 못하고 이 값이 너무 작아 주변 최적영역을 찾는 많은 시간이 소요된다.

![그림 5. 상,하,좌,우 방영역](image)

각 방향 영역의 볼에 대한 비율은 다음과 같이 계산한다.

\[ \text{방향비율} = \frac{\text{요소값}}{\text{하행영역}} \]
5. 실험결과 및 고찰

제안한 방법을 검증하기 위해서 본 논문에서 제시한 방법으로 연구실 내 정지된 카메라로부터 총 30 프레임으로 입력된 이미지(크기: 320×240)에서 오브젝트의 전체적인 움직임을 7 프레임 간격으로 추적해 보았다. 실험에 사용한 시스템은 Matrox meteо II 프레임 그레이버가 장착된 Pentium III 650 Mhz이고 프로그램 환경은 Visual C++ 6.0이다. 실험에 사용한 b0, a, σ는 각각 7, 4, 0.75으로 실험적으로 구했다. 그 결과가 그림 7에 나타나 있다. 그래프에서 알 수 있는 바와 같이 제안한 방법은 오브젝트의 전체적인 움직임을 잘 추적할 수 있었다. 전체적인 움직임을 확인해 보기 위해 최적영역의 중심점을 다음표에 나타내었다.

표 1. 최적영역의 중심좌표

<table>
<thead>
<tr>
<th>Frame</th>
<th>0</th>
<th>k</th>
<th>k+7</th>
<th>k+14</th>
<th>k+21</th>
<th>k+28</th>
<th>k+35</th>
<th>k+42</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x,y)</td>
<td>(231,177)</td>
<td>(381,150)</td>
<td>(196,189)</td>
<td>(454,190)</td>
<td>(459,139)</td>
<td>(169,208)</td>
<td>(89,201)</td>
<td></td>
</tr>
<tr>
<td>Frame</td>
<td>k+49</td>
<td>k+56</td>
<td>k+63</td>
<td>k+70</td>
<td>k+77</td>
<td>k+84</td>
<td>k+91</td>
<td>k+98</td>
</tr>
<tr>
<td>(x,y)</td>
<td>(71,190)</td>
<td>(88,180)</td>
<td>(48,145)</td>
<td>(45,170)</td>
<td>(36,130)</td>
<td>(23,149)</td>
<td>(13,176)</td>
<td></td>
</tr>
</tbody>
</table>

표에 나타난 값을 그래프로 그리보면 다음과 같다.

그림 6. 오브젝트의 전체적인 움직임

위의 그래프는 320×240 이미지 상에서 우측 왼쪽 움직인 오브젝트의 중심의 이동을 보여주고 있다. 그래프 상에 곡선이 심하게 굽어지는 부분이 나타날는데 이는 영상 속에 있는 오브젝트의 그림자에 의한 영향으로 추측된다.

6. 결론

본 논문에서는 카메라로부터 입력된 연속 영상에서 오브젝트의 전체적인 움직임을 추적하기 위해 제안한 방법을 제안하였다. 제안된 방법은 차별성을 이용하여 움직임을 감지하였고 연결성을 이용하여 최적영역을 찾아내었고 영역 추적을 이용하여 움직임을 추적할 수 있었다. 본 논문에서 제시한 최적영역을 이용하여 전체적인 움직임을 추적한 결과 개략적인 움직임을 잘 추출할 수 있었다. 향후개발로 영상 내부 오브젝트의 그림자를 제거하는 연구가 필요하다.

참고문헌