텍스쳐 매핑을 이용한 고속 영상 모자익

최경숙, 이칠우
전남대학교 컴퓨터공학과

High-Speed Image Mosaics Using Texture mapping

Kyoung-Sook Choi, Chil-Woo Lee
Dept. of Computer Engineering, Chonnam Nat'l University

요 약

본 논문은 연속된 이미지로부터 고속 영상 모자익을 구성하는 새로운 접근을 기술한다. 제안하는 알고리듬은 전체 영상이 아닌, 중첩영역에서 특징점 및 대응점을 찾아 초기 변환 행렬을 구함으로 좀더 정확한 변환식을 표현하도록 하고 계산량를 감소시킨다. 또한 고속으로 모자익을 수행하도록 하기 위해 OpenGL 기반 텍스쳐 매핑을 이용하였다. 기존의 방법은 모든 영상의 픽셀에 변환식을 곱함으로 인해 많은 계산시간을 초래했다. 본 논문에서 제안하는 방법은 OpenGL기반 텍스쳐 매핑을 이용해 영상의 각 버텍스에 변환식을 곱함으로서 계산시간을 단축시켰다. 그 결과, PC에서 카메라로부터 영상을 받아들여 고속으로 모자익을 구성할 수 있다.

1. 서론

영상 모자이킹이란 내용이 다른 여러 장의 영상을 모아 하나의 통합된 영상으로 재구성하는 기술을 말한다. 이 기술은 우주/항공사진의 해석, 컴퓨터그래픽스의 페인팅 시스템, 사진의 가공 등 영상을 다루는 분야에서 폭 넓게 사용되어 왔다. 최근에 들어 시야각이 좁은 비디오 카메라를 이용한 가상현실 시스템의 제작, 고해상도 영상의 합성, 영상압축과 같은 멀티미디어 응용시스템의 제작을 위한 중요한 기술로 자리잡아 가고 있다.

모자이킹의 대표적인 방법으로 파노라마 영상 생성법이 있다[1]. 이 방법은 고정된 위치에 카메라를 설치하고 수평방향으로 회전시키면서 촬영한 영상을 하나의 영상으로 통합하여 원통형의 벽에 매핑 시킴으로써 관찰자에게 전 방위의 시야를 제공하는 방법이다. 하지만, 이 방법은 카메라가 자유롭지 못하고, 영상 획득시 고려해야 할 제한된 조건이 많이 존재한다.

카메라를 전후좌우로 이동시키면서 촬영한 여러 영상을 한 영상으로 합성하기 위해서는 영상들간의 변환관계를 계산하여야만 한다. 변환 관계는 영상들간의 대응점을 이용하여 계산이 가능하며 카메라의 이동량이 많을 경우 대응점을 찾기란 매우 곤란한 작업이 된다. 이 때문에 초기 연구의 경우에는 수작업을 통해 대응점을 지정하였다. 그러나 수십장의 영상을 합성할 경우 수작업을 통해 대응점을 지정한다는 것은 매우 어려운 일이며 가장 현실감 있는 3차원 합성 영상을 제공하기 위해서는 자동적으로 대응점을 찾아 촬영당시의 카메라 파라미터를 완전히 복원하는 것이 필요하다.

모자이킹 기술은 근본적으로 그래픽스에서 사용하는 영상 워핑(Image Warphing) 이론과 컴퓨터 비젼에서 사용하는 투영변환 (Perspective Transformation) 이론을 결합하는 것이다. 이 기술은 임의 영상을 새로운 영상으로 변환시키기 때문에 각종 파라미터를 고려해야 되나 파라미터가 많으면 많을수록 계산이 복잡하여 처리시간이 길어질 뿐 만 아니라, 국소 극소점에 수렴할 가능성이 있어 바람직하지 않다. 이러한 문제점을 해결하기 위해, Szeliski[2]는 두 영상 사이의 회전만을 고려한 3-파라미터 회전 모델을 제시하였다. 이 모델은 두 영상의 초점 거리를 임의로 가정하고, 그 값은 별도의 계산이 필요한 최소화 방법[3]에 의해 초기 추정치로부터 반복적으로 개선된다. Chiba[4]는 다중 광류(optical flow) 추정법을 이용하여 새로운 모자이킹하는 방법을 소개하였다. 먼저, 초기 중첩영역을 구하고, 광류측정의 한 방법인 Lucas-Kanade 방법을 이용 대응점을 찾는다. 이로부터 변환행렬을 구

해 에피폴라 기하에 적용시켜 영상을 모자킹하는 방법을 제안하였다.

본 논문에서는 카메라로 얻은 영상으로부터 특징점을 찾아 영상을 합성하는데, OpenGL의 텍스쳐 매핑을 이용해서 고속 모자익하는 방법을 제안한다.

기존의 방법은 영상의 모든 픽셀에 변환식을 곱함으로 많은 계산 시간이 초래된데 비해, 본 논문에서 제안하는 방법은 OpenGL의 텍스쳐 매핑 이용해 텍스쳐된 영상의 각 버텍스에 변환식을 곱함으로서 계산시간을 단축시켰다. 그 결과, 3D그래픽가속기 없이 PC에서 카메라로부터 영상을 받아들여 고속으로 모자익을 구성할 수 있다.

본 논문의 구성은 아래와 같다. 2장에서는 특징점 추출 및 대응점 선택, 그리고 3장에서는 얻어진 대응점을 이용해서 평면 사영 변환식을 유도한다. 4장에서 텍스쳐 매핑을 이용한 영상합성, 5장에서는 제안한 방법을 적용한 실험결과를, 6장에서는 결론 및 향후 연구방향에 대해서 기술한다.

2. 특징점 추출 및 대응점 선택

서로 중첩된 영역을 지니도록 고정된 카메라로부터 영상을 얻는다. 이 경우 카메라가 고정되었음으로 중첩된 영역은 대략 파악할 수 있다. 중첩된 영역내 대응 관계로부터 한 쌍의 영상간의 변환 행렬을 구한다.

2.1 특징점 추출

두 영상의 중첩된 영역에서 특징점(코너점)을 얻기 위해 본 논문에서는 Hessian Matrix를 이용한다. 복잡한 영상의 경우 수많은 특징 점을 가지므로 특징점 선택에 어려움을 가지게된다. 이를 개선하기 위해 Sobel 오퍼레이터를 이용하여 영상의 에지 성분을 강조한 뒤 여기에 Hessian Matrix를 적용하여 영상의 밝기가 급격히 변하는 점을 특징점 후보로 선택한다. Hessian Matrix는 식 (1)과 같다. 여기에서, I_{xx} 와 I_{yy}는 각각 함수 I의 x방향과 y방향의 2차 미분이며, I_{xy}와 I_{yx}는 함수 I의 x방향과 y방향의 2차 미분이다.

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad (1)$$

$|H| \geq 0$ 인 경우 코너

2.2 특징점 선택 및 대응점

에지 검출 후 Hessian Matrix를 적용해도 많은 특징점들이 발생한다. 그러므로 임의의 특징 점을 선택하는데 어려움이 발생한다. 많은 특징 점을 이용하면 좀 더 정확하게 영상 모자이킹을 수행할 수 있지만, 특징 점을 최소 4개 이상 선택 시 최소이승오차 알고리즘을 적용하면 계산량이 증가하게 된다. 그래서 본 논문에서는 초기 변환 행렬을 구하기 위해 5×5 의 사각형 영역을 선정하고, 이 영역 내에서 가장 많은 특징점을 지닌 부분영역 4개를 선택하여 계산량을 줄인다. LSE를 적용해 최적의 $\boldsymbol{M}$을 구할 수 있는 4쌍의 대응점을 선택한다. 4개의 특징 점에 대해서 첫 번째 영상과 연속되는 두 번째 영상의 좌표를 원실험 영상에 적용하여 최초의 $\boldsymbol{M}$값을 구하기 위해 투영변환 행렬에 적용한다.

3. 평면 사영 변환식

한 평면 위의 점들이 시선 각이 다른 두 영상으로 투영되는 평면 투영변환의 경우를 고려해 보자. 그림1은 이런 경우의 변환 관계를 나타낸다.

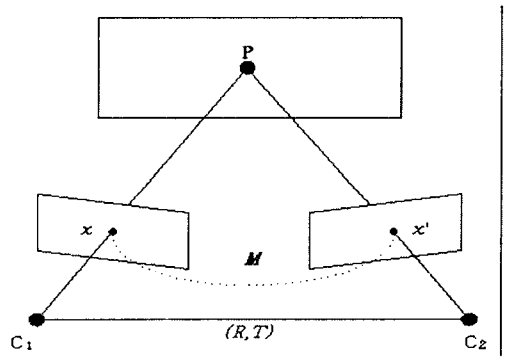

그림1 두 영상으로 투영되는 평면투영변환 관계

두 개의 서로 다른 지점 C_1과 C_2로부터 평면상에 있는 점 P를 볼 때 투영변환 행렬 $\boldsymbol{M}$을 이용하여 좌표 x에서 x'의 균일 좌표를 얻어 변환 얻을 수 있다. 2차원 평면 투영 변환 행렬은 식 (2)과 같다.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (2)$$

그리고 변환 매트릭스 $\boldsymbol{M}$은 식(3)과 같이 정의한다.

$$\boldsymbol{M} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \quad (3)$$

여기에서 m_0, m_1, m_3, m_4는 회전과 크기를 가지는 요

소이고, m_2, m_5는 이동을 가지는 요소이며, m_6, m_7와 m_8은 각각 비례변환의 요소와 1을 나타낸다.

위 식(2) 의 관계를 이용하여 식 (4)와 같은 방정식으로 4쌍의 초기 대응점을 구할 수 있다.

$$x' = \frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1}, \quad y' = \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1}, \tag{4}$$

초기 행렬 변환 행렬 $\boldsymbol{M}$은 대응점을 통해 구할 수 있다. 이렇게 구해진 초기 변환행렬은 완전한 대응점을 토대로 구해진 것이 아니므로, 오차를 갖게 된다. 따라서, 비선형 최소화 기법인 Levenberg-Marquardt 알고리즘을 사용하여 최적의 변환관계를 얻는다[3].

초기 변환행렬 $\boldsymbol{M}$으로 전체 영상에 대해서 사용되었던 기존 모자이크 방식과는 달리 본 논문에서는 중첩된 영역에 대해서만 알고리즘을 수행하기 때문에 계산속도가 개선된다.

4. 텍스쳐 매핑을 이용한 영상합성

기존의 방법은 변환식을 구한 후, 영상의 모든 픽셀에 변환식을 곱함으로 인해 많은 계산시간을 초래했다. 그러나 본 논문에서는 OpenGL의 텍스쳐 매핑을 이용하여 영상 모자익을 하고자 한다.

텍스쳐 매핑은 수행시간이 많이 걸렸는데, 그래픽 보드에서 텍스쳐 매핑을 하드웨어적으로 수행해주는 기능을 제공함으로 OpenGL, DirectX의 표준을 사용한다면 하드웨어의 지원을 받을 수 있다.

따라서 제안하는 방법은 식(6)에서와 같이 모든 영상의 픽셀이 아닌, OpenGL의 텍스쳐 매핑을 이용함으로 텍스쳐 매핑된 영상의 각 버텍스에만 변환식을 곱한다. 이렇게 하므로 계산시간을 단축시킬 수 있다.

◆ **기존의 방법**

$$I_2' = I_2(x, y) \times M \tag{5}$$

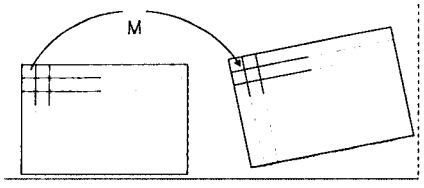

그림 2 모든 픽셀에 변환식 곱

◆ **제안하는 방법**

$$V_2' = V_2 \times M \tag{6}$$

V_2 : 영상2의 버텍스 좌표

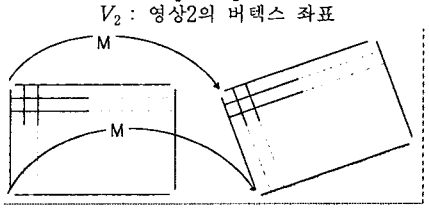

그림 3 버텍스에 변환식 곱

4.1 영상블렌딩

카메라나 비디오로부터 얻어진 영상은 시선방향과 광원의 영향에 의해 영상들 사이에 밝기 차이가 나타나게 되며, 이러한 이유로 재구성된 영상은 경계선(visible edge)을 가지게 된다. 이를 제거하기 위해서 각 영상에 가중 평균(weighted average) 알고리즘을 이용하여 눈에 보이는 경계선을 줄인다. 본 논문에서는 전체영상이 아닌 중첩영역을 포함한 최소한의 영상을 얻어, 그 부분만 가중값을 적용해 블렌딩을 한 후, 텍스쳐 매핑한다. 그러므로 계산 시간을 단축할 수 있다.

4.2 가중 평균 함수

가중 평균 함수(w)는 식 (7) 과 같고, 식 (8)을 이용하여 영상 모자이킹을 수행한다.

$$w(x', y') = w_i(x')w_i(y') \quad \text{식 (7)}$$

$$I' = \frac{\sum_k w_k(x_i, y_i) I_k(x_i, y_i)}{\sum_k w_k(x_i, y_i)} \quad \text{식 (8)}$$

그림 5는 중첩영역을 포함한 최소한의 영상을 얻어, 그 부분만 가중값을 적용해 블렌딩한 결과 이다.

따라서,그림 4에 그림 5를 텍스쳐 매핑함으로 그림10과 같은 결과를 얻을 수 있다.

그림 4 블렌딩 전-텍스쳐매핑 결과

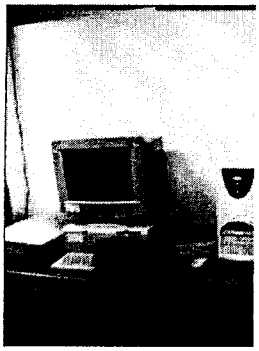

그림 5 중첩영역을 최소한
포함한 영상 블렌딩한 결과

5. 실험

실외 및 실내에서 촬영한 영상을 대상으로 실험을 수행하였다. 실험에 사용된 영상은 모두 카메라로부터 획득한 영상이고, 크기는 640×480 화소이다. 촬영 장비는 Kodak DC120 디지털 카메라이고, 영상 모자익에 관한 실험은 PentiumⅢ 500MHz의 CPU와 128MB의 메모리를 장착한 PC 환경에서 Visual C++6.0과 OpenGL을 사용하였다. 사용 OS는 Windows 98이다.

5.1 실험 영상

그림 6, 7은 디지털 카메라로 얻은 실험영상이다

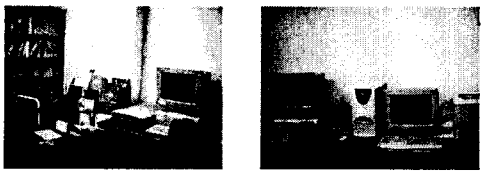

그림 6 실험영상 #1

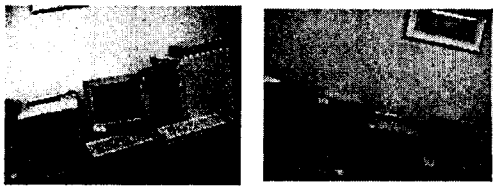

그림 7 실험영상 #2

5.2 실험 결과

그림9,10,11,12은 그림6,7의 실험영상에 대하여 모자익한 결과를 보여준다.

◆ 기존의 방법

그림 9 실험 영상1 의 기존의 방법 결과

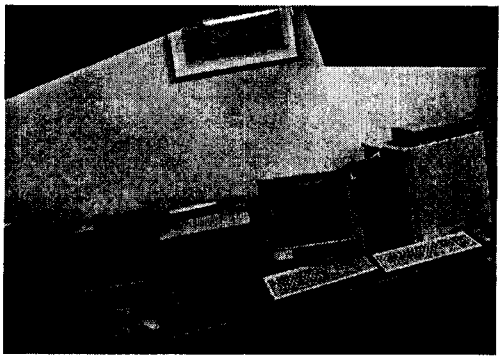

그림 10 실험 영상2의 기존의 방법 결과

◆ 제안한 방법

그림 11 실험 영상1의 제안한 방법 결과

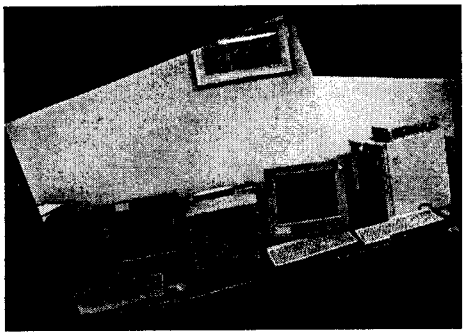

그림 12 실험 영상2의 제안한 방법 결과

5.2 기존 방법과 비교

영상을 합성하는데 있어서 기존의 방법과는 차별화된 텍스쳐 매핑이라는 방법을 이용하였다. 기존의 방법은 8개의 자유도(파라미터)를 갖는 사영변환식을 구한후, 영상의 모든 픽셀에 변환식을 곱함으로 인해 많은 계산시간을 초래했지만, 본 논문에서 제안하는 방법인 OpenGL의 텍스쳐 매핑을 이용함으로 매핑된 영상의 각 버텍스에 변환식을 곱하고 중첩영역을 포함한 최소한의 영역만을 블렌딩함으로 계산 시간을 단축됨을 확인 할 수 있다.

◆성능평가

표1은 제안한 방법과 기존방법의 영상 모자익 수행시 계산 시간을 비교한 것이다.

본 실험에서 두 실험 결과를 비교해 보았을 때, 제안하는 방법이 기존 방법보다 계산시간이 빠름을 볼 수 있다. 시각적으로도 좋은 결과를 보임을 확인할 수 있다.

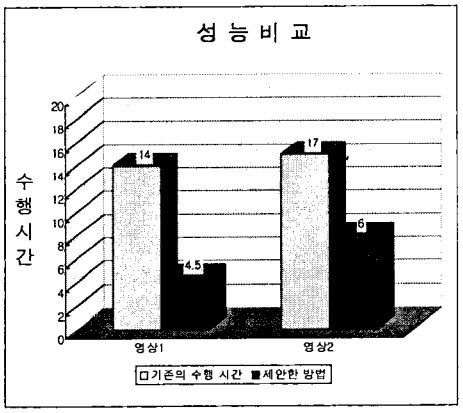

표 1 기존의 방법과 제안한 방법의 비교

6. 결론 및 향후 연구방향

본 논문에서, 텍스쳐 매핑을 이용해서 고속 모자이킹하는 방법을 제시했다. 기존의 방법에 비해, 본 논문에서 제안하는 방법 즉, OpenGL의 텍스쳐 매핑을 이용함으로 매핑된 영상의 각 버텍스에 변환식을 곱해 계산시간을 단축시킴을 확인할 수 있었다.

앞으로 영상의 경계선을 제거하기 위한 알고리듬을 개발할 것이다. 또한 지금까지 영상으로부터 이동과 회전에 관한 2차원 정보만을 얻어 depth값을 고려하지 않고 영상을 재구성했는데, 영상을 기반해서 물체를 표현하되 시퀀스로부터 물체의 움직임 정보를 이용하여 3차원정보(depth)를 고려할 것이다. 그러므로 더욱 정확한 영상으로 재구성을 가능하도록 모자익을 구성할 것이다.

[참고문헌]

[1] W. Puech, A. G. Bors, J.M. Chassery, I. Pitas, "Mosaicing of Paintings on Curved surfaces," IEEE Workshop on Applications of Computer Vision, Sarasota, USA, pp.50-55, 1996

[2] R. Szeliski, "Image Mosaicing for Tele-Reality Applications",CRL Technical Report 94/2, Digital Equipment Corporation, 1994

[3] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes in C : The Art of Scientice Computing" Cambridge University Press Cambridge, England, second edition, 1992

[4] Naoki Chiba, Hiroshi Kano, Minoru Higashihara, Masashi Yasuda, and Osumi, "Frature-Based Image Mosaicing", MVA'98 IAPR Workshop on Machine Vision Applications, pp5-10, 1998

[5] R. Szeliski, "Video Mosaics for Virtual Environments", IEEE Computer Graphics and Applications, (c) IEEE Vol. 16, No. 2: MARCH 1996, pp. 22-30, 1996

[6] Burt P.J., and Adelson, E.H., "A multiresolution spline with applications to image mosaics", ACM Transactions on Graphics, Vol. 2, No. 4, pp. 217-236, October 1983.