

평면 투영 변환에 의한 영상 정렬

김동근*, 김주완**, 장병태**

*서남대학교 컴퓨터 정보통신학부

**한국전자통신 연구원

e-mail:dgkim@tiger.seonam.ac.kr

Image Alignment using Planer Projective Transformation

Dong-Keun Kim*, Ju-Wan Kim**, Byung-Tae Jang*

*Seo-Nam University

**Electronics and Telecommunications Research Institute(ETRI)

요약

본 논문에서는 영상의 일부가 겹치는 두 영상사이에 투영 평면 변환을 사용하여 보다 큰 장의 모자이크 영상으로 정렬하는 알고리즘을 제한한다. 먼저 블록 정합을 이용하여 초기 전역 이동을 계산하고, 4점을 이용하여 효율적인 투영 변환을 구하고, 두 영상사이에 겹치는 부분에서 RGB 컬러를 혼합하여 합성 영상을 생성하였다.

1. 서론

최근 제한된 시야를 갖는 카메라를 이용하여 얻은 여러 장의 영상을 보다 넓은 시야를 갖는 영상을 얻기 위한 영상 모자이크(image mosaics, stitching) 방법과, 비디오 영상의 관련된 연속된 영상 스트립을 커다란 한 장의 영상으로 합성하는 비디오 모자이크(video mosaics) 방법이 활발히 사용되고 있다 [1,2,3].

일반적인 모자이크 영상은 영상들 간에 변환을 구하여 정렬(alignment)하고, 영상들을 이음매 없이 합성(blending)하여 얻는다. 이러한 방법은 영상 워핑(warping)과 영상 합성(blending)을 사용하는 모핑(morphing)과 유사하나, 모자이크 영상은 영상들간에 변환을 자동으로 생성한다[1,11,12]. 영상들 사이의 변환으로는 이동(translation), rigid 변환(translation, rotation), affine 변환(translation, scaling, rotation), 평면 투영 변환(planar projective) 등이 사용된다. 이들 중 평면 투영 변환이 모자이크 영상의 영상 정렬을 위한 변환으로 주로 사용한다[1,2]. 평면 투영 변환을 계산하기 위한 방법으로 두 영상사이의 밝기 값의 차의 제곱을 최소로 하는 변환 계수를 구하기 위하여 비 선형 최소 자승 오차 최소화 방법인 Levenberg-Marquardt 방법을 주로 사용한다. 이 방

법은 영상 밝기 차의 제곱은 대부분의 경우 만족할 만한 결과를 가져오지만 카메라 위치, 빛 방향, 그림자 등에 의해 상관계수 보다 영향을 더 받는다.

본 논문에서는 수평 방향으로 획득한 영상에서 RGB 컬러 정규상관에 의한 블록 정합을 이용하여 초기 전역 이동을 계산하고, 4점을 이용하여 효율적인 투영 변환을 구하고, 두 영상사이에 겹치는 부분에서 RGB 컬러를 혼합(blending)하여 합성 영상을 생성하였다.

2. 4 점에 의한 평면 투영 변환

두 영상간에 평면 투영 변환은 식 (1)과 같다.

$$\begin{aligned} x' &= \frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1} \\ y' &= \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1} \end{aligned} \quad (1)$$

평면 투영 변환에서 라인은 평면 투영에서 라인으로 투영되며, 사변형(quadrilateral)을 사변형으로 변환한다. 식 (1)의 평면 투영 변환은 8개의 계수를 갖는다. 두 영상간에 평면 투영 변환을 구하기 위해서는 4개의 대응점(corresponding points)이 필요하

다. 이는 선형방정식을 풀어 계산할 수 있다. 역 투영 변환(inverse projective transformation) 역시 선형방정식의 해를 구하여 구할 수 있다. 그러나 단위 정사각형(unit square)에서 사변형(quadrilateral)으로의 평면 투영 변환 및 역 투영 변환은 선형방정식의 해를 계산하지 않고 간단히 계산할 수 있다[4, 5].

본 논문에서는 영상의 일부가 수평으로 겹치는 두 영상에서 기준 영상의 고정된 4점 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ 에 의한 직사각형의 정규 상관 계수(normalized correlation)에 의해 대응하는 탐색 영상의 4점 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 에 의한 임의의 사변형(quadrilateral)으로의 평면 투영변환과 역 투영 변환을 구하기 위하여 단위 정사각형에서 사변형으로의 평면 투영 변환과 이동(translation), 신축(scale)을 이용한다. 먼저 단위 사각형에서 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 에 의해 정의되는 사변형으로의 평면 투영 변환은 식(5)에 의해 아핀 변환과 투영 변환인 경우로 나누어 구할 수 있다.

$$\begin{aligned} dx_1 &= x_1' - x_2', & dx_2 &= x_3' - x_2' \\ dx_3 &= x_0' - x_1' + x_2' - x_3' \\ dy_1 &= y_1' - y_2', & dy_2 &= y_3' - y_2' \\ dy_3 &= y_0' - y_1' + y_2' - y_3' \end{aligned} \quad (5)$$

만약 $dx_3=0$ 이고 $dy_3=0$ 이면 아핀 변환(affine transformation)으로 변환 계수는 다음과 같다.

$$\begin{aligned} m_0 &= x_1' - x_0' \\ m_1 &= x_2' - x_1' \\ m_2 &= x_0' \\ m_3 &= y_1' - y_0' \\ m_4 &= y_2' - y_1' \\ m_5 &= y_0' \\ m_6 &= 0 \\ m_7 &= 0 \end{aligned} \quad (6)$$

만약 $dx_3 \neq 0$ 또는 $dy_3 \neq 0$ 이면 투영 변환으로 변환 계수는 다음과 같다.

$$\begin{aligned} m_6 &= \frac{dx_3 dy_3 - dx_2 dy_3}{dx_1 dy_2 - dy_1 dx_2} \\ m_7 &= \frac{dx_1 dy_3 - dy_1 dx_3}{dx_1 dy_2 - dy_1 dx_2} \\ m_0 &= x_1' - x_0' + m_6 x_1' \\ m_1 &= x_3' - x_0' + m_7 x_3' \\ m_2 &= x_0' \end{aligned} \quad (7)$$

$$\begin{aligned} m_3 &= y_1' - y_0' + m_6 y_1' \\ m_4 &= y_3' - y_0' + m_7 y_3' \\ m_5 &= y_0' \end{aligned}$$

위와 같이 구한 단위 정사각형에서 사변형으로의 평면 투영 변환을 이용하여 임의의 직사각형에서 사변형으로의 변환, M 을 구하기 위하여 먼저 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ 점에 의해 이루는 임의의 직사각형을 원점을 기준으로 하는 단위 사각형으로 변환하기 위해 식 (8) 같이 이동과 신축을 계산한다.

$$\begin{aligned} t_x &= x_0, & t_y &= y_0 \\ s_x &= \frac{1}{x_1 - x_0}, & s_y &= \frac{1}{y_2 - y_0} \end{aligned} \quad (8)$$

즉 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ 에서 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 로 평면 투영 변환은, 식 (1)의 변환을 적용하기 전에 먼저 (x_0, y_0) 를 원점으로 이동시키기 위하여 $(-t_x, -t_y)$ 만큼 이동하고, 단위 사각형을 만들기 위하여 (s_x, s_y) 만큼 신축시킨다. 이와 같이 이동 신축시킨 후에 위에서 단위 사각형에서의 평면 투영 변환 계수 $m_i, i=0, \dots, 7$ 를 식 (1)에 적용하여 계산할 수 있다. 또한 임의의 사변형 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 에서 단위 사각형으로의 역 투영 변환 M^{-1} 은 식 (9)과 같이 $m_i, i=0, \dots, 7$ 를 이용하여 간단히 계산된다.

$$\begin{aligned} x &= \frac{a_0 x' + a_1 y' + a_2}{a_6 x' + a_7 y' + a_8} \\ y &= \frac{a_3 x' + a_4 y' + a_5}{a_6 x' + a_7 y' + a_8} \end{aligned} \quad (9)$$

위에서의 계수 $a_i, i=0, \dots, 8$ 은 다음과 같다.

$$\begin{aligned} a_0 &= m_4 - m_7 m_5 & a_1 &= m_7 m_2 - m_1 & a_2 &= m_1 m_5 - m_4 m_2 \\ a_3 &= m_6 m_5 - m_3 & a_4 &= m_0 - m_6 m_2 & a_5 &= m_3 m_2 - m_0 m_5 \\ a_6 &= m_3 m_7 - m_6 m_4 & a_7 &= m_6 m_1 - m_0 m_7 \\ a_8 &= m_0 m_4 - m_3 m_1 \end{aligned}$$

즉 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ 에 의해 정의되는 직사각형으로의 역 투영 변환은 식 (9)을 적용한 후에, 식 (2)의 반대 과정인 $(1/s_x, 1/s_y)$ 만큼 신축시키고, (t_x, t_y) 만큼 이동시켜 얻는다.

3. 제안된 알고리즘

3.1 평가 함수

영상의 일부가 수평으로 겹치는 두 영상간의 평면 투영 변환을 구하기 위하여 정규 상관계수(normalized correlation) 식을 사용한다.

$$R = \frac{E[I(x, y)I'(x', y')] - E[I(x, y)]E[I'(x', y')]}{\sigma_{I(x, y)} \sigma_{I'(x', y')}} \quad (10)$$

즉 식 (10)은 기준 영상 $I(x, y)$ 과 변환 영상 $I'(x', y')$ 사이의 겹치는 부분에서의 상관계수이다. 두 영상간의 평면 투영 변환을 구하기 위해서는 식 (10)의 정규 상관계수를 최대화 하는 변환 계수 $m_i, i=0, \dots, 7$ 를 구하면 된다. 이것은 기준 영상 $I(x, y)$ 에서 4점을 직사각형으로 고정하면, 변환 영상 $I'(x', y')$ 에서 식 (10)의 상관계수를 최대화 하는 4점 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 을 구하는 것과 동일하다.

3.2 블록 정합에 의한 전역 이동

영상의 일부가 수평으로 겹치는 두 영상간의 평면 투영 변환의 초기해로 상관계수에 의한 블록 정합 방법을 사용하여 대략적인 전역 이동을 구할 수 있다. 본 논문에서는 대부분의 파노라마 영상에서처럼 기준 영상의 오른쪽이 탐색 영상의 왼쪽에 겹친다는 가정을 하고, 기준 영상의 오른쪽 25% 영역에서 3×3 격자를 일정 간격으로 설정한 후에, 이 격자의 9점 각각에 대해 탐색 영상의 왼쪽 60%영역을 탐색하여 블록 상관계수가 가장 큰 지점을 대응점으로 구한다.

수평 방향으로 움직이며 획득한 영상에서 제한된 블록(예, $7 \times 7, 11 \times 11$)을 사용하여 개별적으로 탐색한 대응점들은 대부분의 경우 만족할만하지만, 일부 유사 색상을 갖는 지점에서는 잘못 대응점을 찾을 수 있다. 기준 영상의 3×3 격자의 위치 관계와 블록 정합에 의해 찾은 탐색 영상의 대응점들의 관계를 이용하여 잘못 찾은 대응점을 제거할 수 있다. 본 논문에서는 블록 정합에 의해 찾은 3×3 격자에 대한 9개 쌍의 대응점 대응점들 각각이 올바른 지점이라 가정하고, 탐색 영상의 각 대응점을 중심으로 기준 영상에서와 같이 3×3 격자의 위치구조를 이용하여, 나머지 8지점을 생성한 후에 이점들에서의 블록 상관계수의 합을 구하여, 이 블록 상관계수의 합이 9쌍의 대응점 중에서 가장 높은 값을 갖는 지점을 선택하여 개별적으로 생성한 대응점을 연관성을 유지하게 할 수 있으며, 이 대응점들을 이용하면 효율적으로 기준 영상과 탐색 영상의 대략적인 이동(translation) 변환을 구할 수 있다.

3.3 Simulated Annealing(SA)에 의한 투영 변환 계산

블록 정합을 이용하여 기준 영상과 목적 영상사이의 이동에 대한 대략적인 대응점을 이용하여 투영 변환의 초기화로 한 다음 식 (10)을 최대화하는 투영 변환을 구한다. 즉 두 영상간의 상관계수가 최대가 되는 식 (1)의 투영 변환 계수를 구하기 위하여 본 논문에서는 8개의 투영 변환 계수를 구하는 대신 기준 영상에서 4점을 고정하고, 목적 영상에서 식 (10)의 상관계수를 최대화 하는 4점 $(x_0', y_0'), (x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ 을 구하는 방법을 사용하였다. 이렇게 구한 4점에 3.1 절의 방법을 적용하면 간단히 투영 변환을 구할 수 있다.

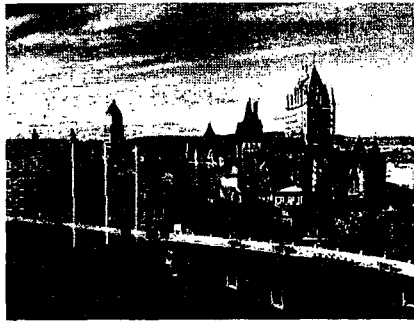
블록 정합에 의해 구해진 9개의 대응점들 중 외각의 4개의 대응점을 초기해로 SA를 사용하여 초기해 근처에서 식 (10)을 최대화 하는 목적 영상의 4점을 구하였다.

- ① 블록정합으로 구한 초기해 중에 외각의 모서리의 4점을 사용하여 식 (6), (7), (8)을 이용하여 투영 변환을 구하고, 이 변환에 의해 기준 영상과 목적 영상 사이의 겹쳐지는 영역에서의 식 (10)의 상관계수, oldR를 구한다.
- ② 초기해 4점을 근처에서 랜덤하게 perturb 시킨 다음, 식 (6), (7), (8)을 이용하여 투영 변환을 구하고, 두 영상 사이의 겹쳐지는 영역에서의 상관계수 newR를 구한다.
- ③ newR이 크면 perturb 시킨 4점을 새로운 해로 채택한다. 그렇지 않을 경우 확률적 ($p = \exp(-(oldR - newR)/T)$)으로 채택한다.

SA를 전역해(global solution) 근처에서 해를 구하기 때문에 빨리 수렴하게 할 수 있다.

4. 구현 및 결과

제안된 투영 변환 정합 알고리즘을 적용하여 영상 사이에 투영 변환을 구하고 합성하는 실험을 하였다. 전역 이동을 구하기 위하여 블록 정합과 SA 알고리즘으로 투영변환을 구할 때 평가함수 식 (10)의 정규상관 계수를 R, G, B 각각에 대하여 계산한 다음 합계를 사용하였다. 투영변환에 의해 겹치는 부분에서의 R, G, B 값은 선형 가중치 함수에 의해 두 영상의 값을 효과적으로 합성하였다. (그림 1)은 왼쪽 영상의 주어진 격자를 기준으로 제안된 투영 변환 알고리즘에 오른쪽의 대응점을 찾은 결과이다. (그림 2)는 대응점에 의해 구한 투영 변환을 이용하여 왼쪽 영상을 중심으로 오른쪽 영상을 선형 가중치에 의한 칼라 합성 결과이다.



(그림 1) 투영 변환 정합에 의한 대응점 결과



(그림 2) 투영 변환 정합에 의한 합성 결과

참고문헌

- [1] Richard Szeliski, Heung-Yeung Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps," SIGGRAPH 97, pp.251 - pp.258.
- [2] Richard Szeliski, "Video mosaics for virtual Environments," IEEE Computer Graphics and Application, 1996, pp.22-30.
- [3] F.Dornaika and R.Chung, "Image Mosaics under Arbitrary Camera Motion," ACCV 2000, vol.1, pp.484- pp.489.
- [4]. George Wolberg, Digital image warping, IEEE Computer Society Press, 1990.
- [5]. Randy Crane, Simplified approach to image processing
- [6]. W.H.Press et al, Neumerical Recipes in C: The Art of Scientific Computing, 2nd Edition, Cambridge Univ Press, 1992.

5. 결론

본 논문에서는 수평 방향으로 획득한 영상에서 RGB 컬러 정규상관에 의한 블록 정합을 이용하여 초기 전역 이동을 계산하고, 가우시안 피라미드 (Gaussian pyramid)에서 4점을 이용하여 효율적인 투영 변환을 구하고, 두 영상사이에 겹치는 부분에서 RGB 컬러를 혼합(blending)하여 합성 영상을 생성하였다.