

# 스포츠 동영상의 모자익을 위한 이동계수의 계층적 향상

이재철\* · 이수중\*\* · 고영훈\*\*\* · 노홍식\*\*\*\* · 이완주\*\*\*\*\*

## 요 약

스포츠 중계 영상과 같은 동영상은 카메라의 줌과 팬에 의한 화면 전역 이동이 크게 발생하며, 적은 수의 특징점 및 다수의 불규칙 운동 개체들로 인하여 이동 해석이 매우 어렵다. 본 논문에서는 이러한 다이나믹한 동영상의 전역 이동 계수를 추출하기 위한 새로운 알고리즘을 제안하고, 이를 개체 추적 시스템에서의 이동 보상과 파노라마 합성에 이용할 수 있도록 하였다.

인접 프레임 사이의 이동을 연속적으로 적용할 때, 누적되는 오차를 최소화하기 위하여, 다중 프레임 간의 이동으로부터, 사이 프레임들 간의 이동을 보간하는 방법을 제안하였다. 여기서 아핀 변환 또는 원근 투영 변환을 하나의 정방 행렬로 보고, 미소 이동을 갖는 행렬의 거듭제곱으로 분할하는 문제를 제시하고, 뉴턴 랩슨 방법을 벡터와 행렬 행태로 변형하여 풀이하는 방법을 제안하였고 매우 효율적으로 계산할 수 있음을 보였다. 이를 다수의 프레임 사이에서 재 계산된 이동과의 오차의 분할에 이용하고, 같은 방법을 계층적으로 수행하는 이동 계수의 향상 방법을 제안하였다. 제안된 방법은 방송용 특수 효과의 연출과 비디오 인덱싱 등의 응용 분야에 활용될 수 있다.

## 1. 서론

디지털 동영상에 대한 컴퓨터 시각(computer vision) 기술은 고성능 마이크로 프로세서와 대용량 메모리, 광대역 통신 등의 발달로 인하여 급속도로 성장하고 있다. 동영상의 전역 이동 해석은 카메라의 이동에 의한 영상의 변화를 몇 개의 계수로 모델링하는 것이다[1][2][3]. 전역 이동을 추정하는 것은 매우 다양한 응용 분야에서 독자적인 연구가 진행되었다.

고정된 배경에서 촬영된 영상이 아닌, 물체의 배경이 이동하는 경우는 추적이 어렵게 된다. 배경의 이동은 전역 이동으로 볼 수 있으며, 전역 이동은 주로 카메라의 움직임, 회전이동 및 초점거리 변화에 의하여 발생한다. 움직이는 물

체들과 배경을 분리하고 추적하기 위해서는 전역이동의 보상이 필요하다[4][5][6]. 따라서 능동 카메라 개체 추적 시스템의 성능은 일차적으로 카메라 이동의 추출에 근거하므로, 다른 장비의 도움 없이 영상으로부터 카메라의 이동을 추출해야만 하는 경우, 전역 이동 계수의 추출은 매우 중요하다[7][8].

스포츠 중계 영상에서는 운동 선수나 공과 같은 물체가 추적되기 때문에, 카메라의 팬(pan)에 의한 평행이동과 초점거리 변화에 의한 줌(zoom) 이동이 빈번하게 발생한다. 화면의 개체들은 그 크기가 작고 형태 변화가 심하며 빠르게 이동한다. 배경의 형태는 일정한 규칙 내에서 정해지지만, 실험실 환경과 같이 엄격하게 제한되지 않으므로 배경의 색상, 무늬와 조명 조건이 매우 다양하다. 또한 영상 내에 가용한 특징점의 수가 매우 적다. 이러한 특성을 가진 영상에 대한 컴퓨터 시각적 해석은 어려운 문제

\* 가천길대학 인터넷 무악과 겸임교수

이다[9][10].

본문에서는 기존에 제시된 프레임간의 이동 추정 방법은 특징점이 거의 없고, 잡음이 많으며, 불규칙적으로 운동하는 많은 물체들이 존재하는 영상에서 전역 배경 이동 계수를 1회 통과(1 pass) 방식으로 비교적 정확하게 계산할 수 있었다.

그러나 화소 단위 이하의 정밀도를 갖기 어려우며, 많은 잡음 요소들이 존재하기 때문에 프레임간의 이동 추정 과정에서 약간의 오차는 피하기 어렵다. 특히 카메라의 팬 속도가 빠른 경우, 움직임에 의한 번짐(Motion Blur) 현상에 의하여 영상이 심하게 훼손되고, 이러한 경우 기존의 정합 방법들로는 매우 큰 오차 발생이 불가피하다.

인접 프레임간 계산된 이동 변환을 연속적으로 적용하면, 그 오차가 누적되어 장면이 진행되는 동안 처음과 마지막 사이의 이동을 계산하면 상당히 큰 오차가 발생한다. 특히 몇몇 프레임의 치명적인 오차들이 전체적으로 균형을 파괴하여, 배경 보상 값을 크게 흐트리거나, 합성된 파노라마 모양의 왜곡을 일으킨다.

여러 프레임에 걸친 연속적인 이동 상에서 발생 가능한 누적 오차를 줄이기 위하여, 아핀 변환 행렬의 제곱근을 이용한 보간(interpolation) 방법을 제안하였다. 여기서 정의된 아핀 변환 행렬의 제곱근을 빠르고 정확하게 계산하기 위하여 뉴턴 랩슨(Newton Raphson) 알고리즘을 행렬 방정식에 적용하는 방법을 제안하였다.

## II. 오차 분할을 이용한 이동계의 수의 계층적 향상

### 2.1. 아핀 변환 행렬의 이분할

평행이동 변환과 회전변환 및 확대 축소, 켄트러짐 변환은 아핀 변환으로 일반화가 가능하다. 아핀 변환은 동차 좌표계(homogeneous coordinate system)를 이용하면 선형 변환 행렬로 표현할 수 있다.

$$\begin{matrix} x' \\ y' \\ 1 \end{matrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} x \\ y \\ 1 \end{matrix} \quad (1)$$

연속적인 아핀 변환 A 와 B를 다음과 같이 행렬 곱에 의하여 표현할 수 있다.

$$A \cdot (Bx) = (AB)x \quad (2)$$

k번째 영상과 k+2번째 영상 사이에 어떤 변환 행렬을  $M_k$  라고 할 때, 이를 k와 k+1 사이의 변환  $L_k$  과 k+1과 k+2사이의 변환  $L_{k+1}$  의 곱으로 다음과 같이 표시할 수 있다.

$$M_k = L_{k+1}L_k \quad (3)$$

여기서,  $L_k$  에 대한 부가적인 정보가 없을 경우  $L_k = L_{k+1}$  로 가정하고 이를 풀이하면,

$$\begin{aligned} M_k &= L_k^2 \\ L_k &= (M_k)^{1/2} \end{aligned} \quad (4)$$

행렬에 대한 제곱근의 정의는 일반적으로 널리 사용되지 않으나, 여기에서는 하나의 선형 변환 행렬을 2개의 동일한 행렬 곱으로 표현할 수 있을 때, 이를 이등분한 변환 행렬로 정의하였다. 이는 Cholesky decomposition method와는 다르다.

대수적인 방법으로 해를 구하기 위하여, 주어진 변환 행렬 M과 미지 계수 행렬 L는 다음과 같을 때,

$$M = \begin{bmatrix} \bar{a} & \bar{b} & \bar{c} \\ \bar{d} & \bar{e} & \bar{f} \\ 0 & 0 & 1 \end{bmatrix} L = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

L의 제곱을 정리하면,

$$\mathbf{L}^2 = \begin{bmatrix} a^2 + bd & ab + be & ac + bf + c \\ ad + de & bd + e^2 & cd + ef + f \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

이다. 식 (4)의 계수 비교를 이용한 연립 방정식은 다음과 같이 구성된다.

$$\begin{aligned} \bar{a} &= a^2 + bd \\ \bar{b} &= ab + be \\ \bar{c} &= ac + bf + c \\ \bar{d} &= ad + de \\ \bar{e} &= bd + e^2 \\ \bar{f} &= cd + ef + f \end{aligned} \quad (7)$$

이 방정식은 6원 2차 연립 방정식으로서, 이 방정식의 일반해는 대단히 복잡하여 비실용적이다. 따라서, 수치해석적으로 제곱근을 풀이하기 위하여 Newton Raphson 방법을 응용하였다.

스칼라 값의 제곱근은 다음  $x$ 에 관한 방정식의 해이다.

$$f(x) = x^2 - y = 0 \quad (8)$$

Newton Raphson 방법은 다음과 같은 반복식을 이용하여,  $x_0$ 가 해에서 가까운 경우  $x_i$ 는 매우 빠른 속도로 근에 접근한다.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (9)$$

식 (8)의  $f(x)$ 와  $f'(x)$ 의 미분을 식 (9)에 대입하여, 다음과 같은 반복식을 얻는다.

$$x_{i+1} = x_i - \frac{x_i^2 - y}{2x_i} \quad (10)$$

식 (5)의 아핀 변환의 계수를 다음과 같이 백

터로 정리하면,

$$\begin{aligned} \mathbf{x} &= [a \ b \ c \ d \ e \ f]^T \\ \mathbf{y} &= [\bar{a} \ \bar{b} \ \bar{c} \ \bar{d} \ \bar{e} \ \bar{f}]^T \end{aligned} \quad (11)$$

이다. 식 (6)에서의 아핀 변환의 곱 행렬을 벡터로 정리하면,

$$\mathbf{sqr}(\mathbf{x}) = \begin{bmatrix} a^2 + bd \\ ab + be \\ ac + bf + c \\ ad + de \\ bd + e^2 \\ cd + ef + f \end{bmatrix} \quad (12)$$

이다. 여기서,  $\mathbf{sqr}(\cdot)$ 은 아핀 변환 행렬의 제곱에 대한 계수들의 벡터형 정렬이다. 제곱근을 구하기 위한 식은 주어진  $y$ 에 대하여

$$f(\mathbf{x}) = \mathbf{sqr}(\mathbf{x}) - \mathbf{y} = \mathbf{0} \quad (13)$$

을 만족하는 계수 벡터  $\mathbf{x}$ 를 찾는 일이다.

6개의 계수에 대하여, 각각에 대한 Newton Raphson 알고리즘을 적용하면, 다음과 같은 반복식을 얻는다.

$$\mathbf{x}_{i+1} = \mathbf{x}_i - f(\mathbf{x}_i) \cdot / f'(\mathbf{x}_i) \quad (14)$$

여기에서  $\cdot /$ 는 벡터의 대응 요소간 나눗셈이다. 또한,  $f'(\mathbf{x}_i)$ 는 각 요소에 대한 편미분으로 구성된 벡터로 다음과 같다.

$$f'(\mathbf{x}_i) = \begin{bmatrix} D_a(a_i^2 + b_i d_i - \bar{a}) \\ D_b(ab_i + b_i e_i - \bar{b}) \\ D_c(ac_i + b_i f_i + c_i - \bar{c}) \\ D_d(ad_i + d_i e_i - \bar{d}) \\ D_e(b_i d_i + e_i^2 - \bar{e}) \\ D_f(cd_i + e_i f_i + f_i - \bar{f}) \end{bmatrix} = \begin{bmatrix} 2a_i + D_a(b_i d_i) \\ a_i + e_i + b_i D_b(a_i + e_i) \\ a_i + 1 + c_i D_c a_i + D_c(b_i f_i) \\ a_i + e_i + d_i D_d(a_i + e_i) \\ 2e_i + D_e(b_i d_i) \\ e_i + 1 + f_i D_f e_i + D_f(c_i d_i) \end{bmatrix} \quad (15)$$

여기서  $D_k(\cdot) = \frac{\partial}{\partial k}(\cdot)$ 로 정의된 미분연산자이다. 식 (12)와 (13)에서 아핀 계수들은 상호 독

립이 아니므로, 식 (15)의 우측에서 남은 미분 항은 직접 계산하기 곤란하므로, 따라서 다음과 같이 이전 변수와 근사식을 이용하여 풀이한다. 여기서 다시 발생하는 미분 항은 무시한다.

$$\begin{aligned}
 D_a(b_i d_i) &= b_i D_a d_i + d_i D_a b_i \\
 &\approx b_i D_a d_{i+1} + d_i D_a b_{i+1} \\
 &\approx 2b_i d_i
 \end{aligned}
 \tag{16}$$

$$\begin{aligned}
 b_i D_b(a_i + e_i) &\approx b_i D_b(a_{i+1} + e_{i+1}) \\
 &\approx 2b_i d_i
 \end{aligned}
 \tag{17}$$

위와 같은 방법으로 모든 요소를 정리하면 반복식의 미분 항은 다음과 같다.

$$f'(x_i) \approx \begin{bmatrix} 2a_i + 2b_i d_i \\ a_i + e_i + 2b_i d_i \\ a_i + 1 + b_i d_i \\ a_i + e_i + 2b_i d_i \\ 2e_i + 2b_i d_i \\ e_i + 1 + b_i d_i \end{bmatrix}
 \tag{18}$$

식 (18)를 식 (14)에 대입하여, 벡터 반복식을 정리하면 다음과 같다.

$$\begin{bmatrix} a_{i+1} \\ b_{i+1} \\ c_{i+1} \\ d_{i+1} \\ e_{i+1} \\ f_{i+1} \end{bmatrix} = \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{bmatrix} - \begin{bmatrix} (a_i^2 + b_i d_i - \bar{a}) / (2a_i + 2b_i d_i) \\ (a_i b_i + b_i e_i - \bar{b}) / (a_i + e_i + 2b_i d_i) \\ (a_i c_i + b_i f_i + c_i - \bar{c}) / (a_i + 1 + b_i d_i) \\ (a_i d_i + d_i e_i - \bar{d}) / (a_i + e_i + 2b_i d_i) \\ (b_i d_i + e_i^2 - \bar{e}) / (2e_i + 2b_i d_i) \\ (c_i d_i + e_i f_i + f_i - \bar{f}) / (e_i + 1 + b_i d_i) \end{bmatrix}
 \tag{19}$$

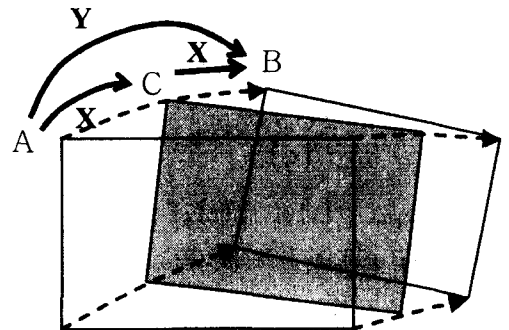
## 2.2. 아핀 모델 이동의 보간

### 2.2.1. 단일 프레임 이동 삽입

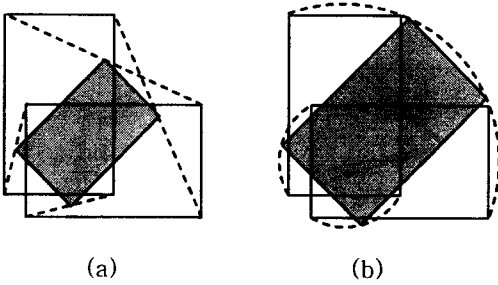
아핀 변환 행렬의 제공근을 계산한 결과를 이용하면, 두 프레임 간격의 이동에 대한 아핀 변

환이 주어 졌을 때, 중간의 프레임간 이동에 대한 아핀 변환을 계산할 수 있다. 그림 1에서 직사각형 A와 B 사이의 아핀 변환 Y에 대하여  $Y = X^2$  를 만족하는 아핀 변환 X는 직사각형 A를 C로 변환 할 수 있고, 또한 C는 X에 의하여 B로 변환 될 수 있다.

(그림 2)에서는 반시계 방향으로 90도 회전하는 직사각형의 예이다. (a)에서는 일반적인 키 프레임 애니메이션등에서 사용되는 방식으로, 사각형의 대응점 벡터를 2등분한 점을 얻고, 이들을 이용하여 원래 벡터를 분할하는 방법이다. 그림에서 보는 바와 같이 평행 이동이 아닌 회전이동에 대하여, 물체의 형태는 유지하지만 크기의 변화가 관찰된다. 회색으로 채워진 직사각형은 반시계 방향으로 45도 회전과 동시에 크기가 약 70% 정도로 축소되었다. (b)에서는 90도 회전 변환의 행렬의 제공근을 계산하여, 이 변환을 가로로 누인 직사각형에 적용하였다. 변환된 직사각형은 반시계 방향으로 45도 회전하였고 그 크기는 변화가 없다. 회전 이동에 대하여 크기 변화가 관찰되어서는 안되므로, 변환 행렬의 제공근을 이용한 삽입(interpolation)의 장점을 알 수 있다.



(그림 1) 직사각형 A의 아핀 변환 Y에 의한 이동: BY의 제공근 변환 X에 의한 이동: C



(그림 2) 직사각형의 회전 이동과 보간  
 (a) 꼭지점 이동 벡터의 중점 이용  
 (b) 이동 행렬의 제곱근 이용

2.2.2. 다수 프레임 내 이동의 삽입

주어진 하나의 변환 사이에 여러 프레임이 존재하는 경우는 변환 행렬의 n제곱근을 이용하여 삽입할 수 있다. 즉,  $Y = X^n$  를 만족하는 X를 찾는다.

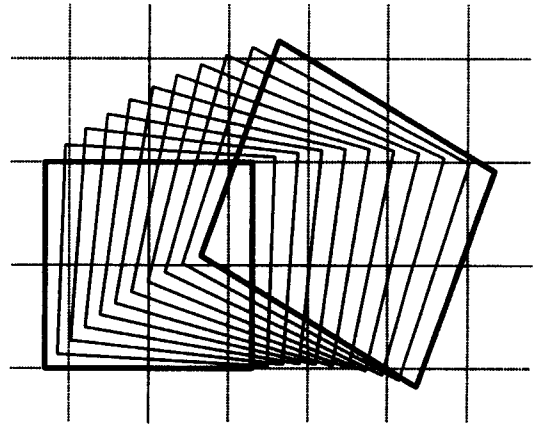
식 (9)을 확장하여, 어떤 변환 행렬 Y에 대한 n제곱근을 다음과 같이 계산하려면 다음과 같은 반복식을 이용할 수 있다.

$$X_{i+1} = X_i - \frac{1}{n}(X_i^n - Y) \cdot X_i^{1-n} \tag{20}$$

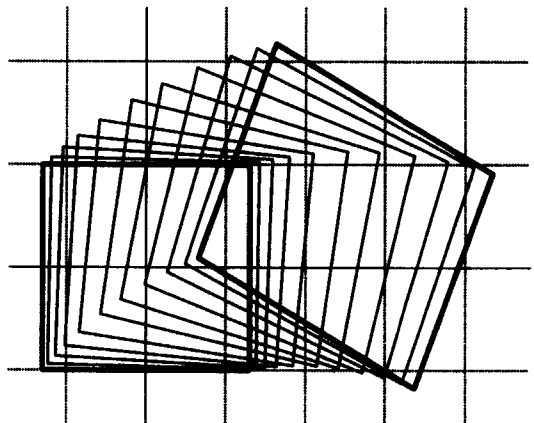
일반적으로 n의 값이 커지면,  $X_i$ 는 I에 접근한다. 따라서, 반복 계산식의 초기값은  $X_0 = I$  로 놓으면 충분히 해에 근접하므로,  $X_i$ 는 빠른 속도로 수렴하여 근을 얻을 수 있다.

j번째 프레임과 k번째 프레임에 사이에 대한 어떤 변환 행렬이 알려져 있을 때, 중간 프레임들 사이의 이동 변환이 동일하다고 보면 앞서 제안된 변환 행렬의 n제곱근 방법( $n=k-j-1$ )을 이용하면 (그림 3)과 같은 균일한 이동 결과를 얻을 수 있다. 이 그림에서는 굵은 색으로 표현된 두 영상 프레임이 존재하고, 프레임 거리는 10이다. 좌측으로 2.5, 상측으로 0.5 만큼 평행이동 하고, 시계방향으로 25도 회전하며, 크기는

15% 증가하였다. 그림 3에서는 내삽된 프레임들은 계산된 X의 연속적인 행렬 곱에 의하여 이동이 계산되었다.



(그림 3) 다수 프레임 간 이동의 균일 보간  
 (a) frame distance = 10, motion = [1.042, 0.486, 2.500, 0.486, 1.042, 0.500] Computed interframe motion = [1.013, 0.044, 0.222, 0.044, 1.013, 0.095]



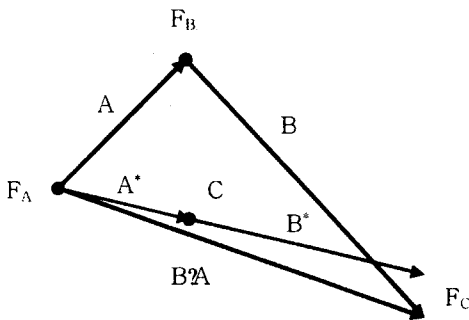
(그림 4) 다수 프레임간 이동의 불균일 보간 (b)

이와 같은 계산은 실제 카메라가 등속 운동을 하는 경우 매우 좋은 근사 방법이 된다. 그러나 실제 카메라로 취득한 영상은 그림과 같은 균일한 이동으로 완전하게 모델링할 수 없다. 관성과 같은 카메라의 물리적 특성과 카메라 조작자

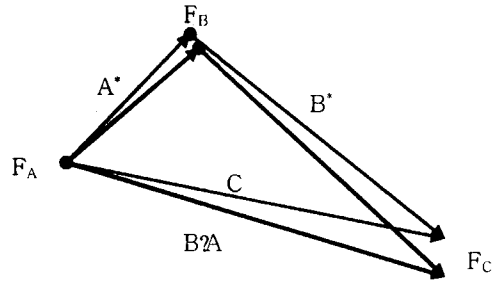
의 습관 및 취득하려는 장면의 움직임 정도에 따라서 서로 균일하지 않은 이동이 관찰된다. 즉, 실제 카메라 운동을 반영하여 가속과 감속에 의한 그림 4와 같은 방법으로 이동을 모델링하는 경우를 고려해야 한다. 그 이유는 j와 k가 가까운 거리라고 하더라도, 균일한 이동으로 가정하는 경우 상당한 오차가 발생할 수 있기 때문이다.

2.2.3. 오차의 분할을 이용한 보정 방법

임의의 구간에서 보간을 수행한다면, 다음과 같은 예를 고려해 볼 수 있다. 그림 5에서 프레임  $F_A$ 와  $F_B$ 에서 측정된 이동 변환 행렬이 A이고  $F_B$ 와  $F_C$ 에서 측정된 이동 변환 행렬이 B이다. 그러나 이 두 행렬의 합성 행렬  $B \cdot A$ 는 실제로 측정된 변환 행렬과 오차가 있다.  $C = B^* \cdot A^*$ 를 만족하면서, C를 A와 B의 크기의 비로 분할한 벡터  $A^*, B^*$ 는 그림과 같이 원래의 A, B와 상당한 차이가 있다. 만일 A와 B에서 C에 수직인 성분이 중요한 의미를 가진다면, 예시된 보간 방법의 사용은 곤란하다. 즉 이러한 문제를 고려하여 그림 6과 같이, 오차  $C - (B \cdot A)$ 를 분할하여 A, B에 각각 더한 값으로  $A^*, B^*$ 의 값을 정할 수 있다. 이러한 경우는 보간이



(그림 5) C의 분할을 이용한 A와 B의 보간  $A^*, B^*$



(그림 6)  $B \cdot A$ 와 C 사이의 오차를 이용한 A와 B의 보정  $A^*, B^*$

라는 용어 보다는 보정이 적합하므로, 이후는 보정(Correction)을 사용한다.

위와 유사한 방법으로 이동 변환 행렬의 보정 방법을 얻을 수 있다. 주어진 행렬 A, B, C와 보정 할 행렬  $A^*, B^*$ 를 다음과 같은 행렬 방정식으로 관련 짓는다.

$$\begin{aligned} A^* &= X_A \cdot A \\ B^* &= B \cdot X_B \end{aligned} \tag{21}$$

즉, 보정 행렬  $A^*$ 는 A에 어떤 미지 행렬의 곱이다.  $C = B^* \cdot A^*$  이므로 식 (21)을 대입하면,

$$C = B \cdot X_B \cdot X_A \cdot A \tag{22}$$

이다. 여기서  $X = X_B \cdot X_A$ 로 놓으면 X는 다음과 같이 계산할 수 있다.

$$X = B^{-1} \cdot C \cdot A^{-1} \tag{23}$$

여기서 계산된 X를  $X_A$ 과  $X_B$ 로 분리하면, 식 (23)에 의하여  $A^*$ 와  $B^*$ 를 정할 수 있다

즉 행렬 A, B의 속도 크기  $v_A, v_B$ ,

$$\begin{aligned} v_A &= \|A\|_v \\ v_B &= \|B\|_v \end{aligned} \tag{24}$$

이다. 이를 충분히 큰 상수  $k_{large}$ 의 곱으로 두 수의 정수비는

$$\begin{aligned} k_A &= \begin{bmatrix} k_{large} & v_A \\ k_{large} & v_B \end{bmatrix} \\ k_B &= \begin{bmatrix} k_{large} & v_A \\ k_{large} & v_B \end{bmatrix} \end{aligned} \tag{25}$$

와 같다. 이 정수 비를 이용하여, 식 (23)에서 계산한  $X$ 를 분할하고 식 (21)에 대입한다.

$$A^* = AX^{\frac{k_A}{(k_A+k_B)}}$$

$$B^* = X^{\frac{k_B}{(k_A+k_B)}}B \tag{26}$$

그림 7은 이 방법을 이용하여 다음과 같은 아핀 변환 행렬의 보정을 수행한 예이다. 사용된 이동 변환 행렬은 다음과 같다.

$$A = \begin{bmatrix} 1.0833 & -0.1910 & 2.5000 \\ 0.1910 & 1.0833 & 2.3000 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

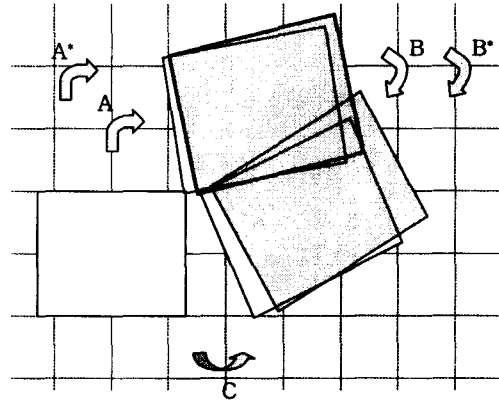
$$B = \begin{bmatrix} 0.9659 & -0.2588 & 1.5000 \\ 0.2588 & 0.9659 & -2.3000 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0003 & -0.5878 & 3.6286 \\ 0.5878 & 1.0003 & 0.8291 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

보정된 이동 변환 행렬은 다음과 같다. 그림에서는 굵은 테두리의 직사각형이 보정된 프레임이다.

$$A^* = \begin{bmatrix} 1.1037 & -0.2546 & 2.7007 \\ 0.2546 & 1.1037 & 2.3888 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$B^* = \begin{bmatrix} 0.9772 & -0.3072 & 1.7235 \\ 0.3072 & 0.9772 & -2.3348 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$



(그림 7) 아핀 변환 행렬 보정의 예

### 2.3. 원근 투영 변환으로의 확장

일반적인 경우를 고려하면 원근 투영 모델에 대하여 해석이 가능해야 한다. 원근 투영 변환은 아핀 변환과 마찬가지로  $3 \times 3$  행렬과 동차 좌표계를 이용하여 정의할 수 있다.

$$x' = \frac{ax + by + c}{gx + hy + 1}, \quad y' = \frac{dx + ey + f}{gx + hy + 1} \tag{27}$$

여기서,

$$(x', y', 1)^T = (x'', y'', z'')^T = (x''/z'', y''/z'', 1)^T$$

이므로, 변환 행렬  $T$ 에 의한 좌표변환 표현은 다음과 같다.

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{x}'' = T\mathbf{x} \tag{28}$$

연속적인 투영 변환을 다루기 위해서는 투영 행렬이 동차 좌표계의 성질 때문에 정규화가 필요하다. 즉, 변환행렬  $T_1$ 과  $T_2$ 의 곱을  $T_{12}$ 로 정의할 때,  $T_{12}$ 의 3행 3열 값  $t_{33}$ 은 정규화 요소이다.

$$\mathbf{T}_1 \mathbf{T}_2 = \mathbf{T}_{12} = \begin{bmatrix} a_{12} & b_{12} & c_{12} \\ d_{12} & e_{12} & f_{12} \\ g_{12} & h_{12} & i_{12} \end{bmatrix} \equiv \begin{bmatrix} a_{12}/i_{12} & b_{12}/i_{12} & c_{12}/i_{12} \\ d_{12}/i_{12} & e_{12}/i_{12} & f_{12}/i_{12} \\ g_{12}/i_{12} & h_{12}/i_{12} & 1 \end{bmatrix} \quad (29)$$

역행렬 계산은 다음과 같다.

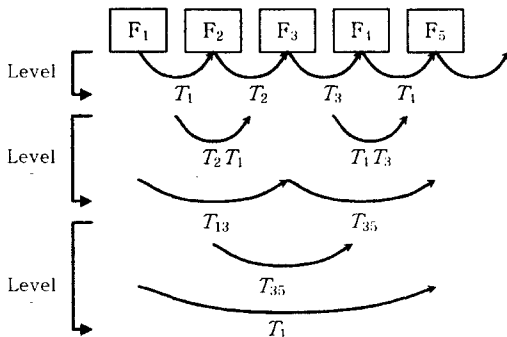
$$\mathbf{X}^{-1} = \frac{1}{ae-bd} \begin{bmatrix} e-fh & -b+ch & bf-ce \\ -d+fg & a-cg & cd-af \\ dh-eg & bg-ah & 1 \end{bmatrix} \quad (30)$$

따라서, 투영변환에 관한 이동 보간을 수행할 수 있다. 여기서는 정규화에 필요한 다수의 나눗셈 연산 수행 때문에 계산상의 오차가 발생하기 쉬우므로, 변환의 화면 깊이 정보  $g$ 와  $h$ 의 값이 커지면, 수렴 속도가 저하된다.

### 2.4. 오차 분할을 이용한 계층적 향상 방법

여러 프레임 간격 사이에서 이동을 계산하는 방법은 다음과 같은 장단점을 고려할 수 있다.

- 블록 또는 화소들의 이동 범위가 커지면 상대적으로 정밀한 이동 계수를 얻을 수 있다. 따라서 프레임 간격을 가능한 한 띄울 필요가 있다.



(그림 8) 계층적인 이동계수 향상 과정

- 프레임 간격이 멀어지면, 이동 벡터의 탐색

범위가 넓어져야 하므로 많은 계산량이 필요하고, 프레임간 겹치는 화소의 수가 줄어들게 되어 안정적인 해를 얻을 가능성이 줄어든다.

파노라마 자체의 합성은 많은 영상을 사용할 필요가 없이, 최소한도로 이어지는 소수의 영상만을 선택할 수 있다. 그러나 파노라마와 영상 내의 개체들과의 관계를 알기 위해서는 모든 프레임에 대해서 이동을 계산해야 한다. 추적 문제에 관해서도 모든 프레임의 이동이 필요하다. 따라서, 프레임 간격에 따른 정밀도를 고려하는 것과는 관계없이 모든 프레임 사이의 이동 계산은 필수적이다.

(그림 8)은 과정을 요약한 것이다. 단계 1에서는 매 프레임  $F_i$ 에 대하여 그 다음 프레임  $F_{i+1}$  사이의 이동 변환 행렬  $T_i$ 를 계산한다.

단계 2에서는 2프레임 간격의 이동 변환 행렬을 계산하고, 단계1의 결과를 수정한다. 연속된 두 이동 행렬의 곱  $T_{i+1}T_i$ 는  $i$ 번째와  $i+2$ 번째 프레임 사이의 이동으로 볼 수 있지만, 누적 오차가 존재한다. 이를 제거하기 위하여, 다음과 같은 이동 개선(refinement) 과정으로  $T_{i+2}$ 를 계산한다.

- ① 주어진 이동 행렬(ex:  $T_{i+1}T_i$ )을 이용하여 영상의 각 블록에 대한 이동 벡터를 계산한다.
- ② ①에서 계산된 이동 벡터 변위를 중심으로  $F_i$ 와  $F_{i+2}$ 에 대하여 BMA를 수행한다.
- ③ 이동행렬을 계산하고 이를  $T_{i+2}$ 로 놓는다.
- ④  $T_i$ 과  $T_{i+1}$ 는  $T_{i+2}$ 를 이용하여, 보간법에 의하여 수정한다.

이 과정으로 2프레임 간격의 이동 행렬과 하위 단계에 속하는 1프레임 간격의 이동 행렬들이 수정된다.



단계 3에서는 단계 2와 같은 방법으로 4프레임 간격의 이동  $T_{i:i-4}$  를 계산하고 하위 단계에 속한 모든 이동 행렬들을 수정한다.

이러한 방법으로  $n$ 번째 단계에 이르면,  $2^{n-1}$  프레임 간격의 이동이 얻어진다.

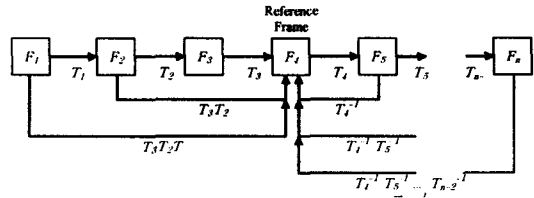
$$T_i^{-1} = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \frac{1}{a_i e_i - b_i d_i} \begin{bmatrix} e_i & -b_i & b_i f_i - c_i e_i \\ -d_i & a_i & c_i d_i - a_i f_i \\ 0 & 0 & a_i e_i - b_i d_i \end{bmatrix} \quad (32)$$

단,  $a_i e_i - b_i d_i \neq 0$  이다.

$j < k$  일 때,  $k$ 번째 프레임에서  $j$ 번째 프레임으로의 변환  $T_{kj}$  는

$$T_{kj} = T_j^{-1} T_{j+1}^{-1} T_{j+2}^{-1} \cdots T_{k-2}^{-1} T_{k-1}^{-1} \quad (33)$$

이다.



(그림 9) 연속 행렬 곱에 의한 기준 프레임으로의 좌표 변환

### III. 이동 보상과 파노라마 합성

이동 계수를 파노라마 합성에 적용하고 그 결과를 제시하였다. 아핀 모델로 표현된 프레임간 전역 이동 모델은 식 (32)와 같이  $3 \times 3$  행렬로 표현된다. 그리고 두 아핀 변환 행렬의 곱은 아핀 변환 행렬이다. 따라서 특정 기준 프레임에 대하여 상대적인 변환은, 프레임간 변환 행렬의 연속적인 곱으로 쉽게 계산할 수 있다. 그림 9 는 이러한 행렬의 연속 곱을 이용한 기준 프레임으로의 좌표 변환을 설명한다. 여기서는 4번째 프레임을 기준 프레임으로 설정하였다. 기준 프레임 이전의 프레임간 변환은 가까운 프레임의 변환 행렬로부터 시작해서, 왼쪽 프레임으로 가면서 각각 오른쪽에 행렬을 곱해 나가는 방법이다.  $T_i$  를  $i$ 번째 프레임에서  $i+1$ 번째 프레임으로의 아핀 변환 행렬로 정의한다.

$i < j$  일 때,  $i$ 번째 프레임에서  $j$ 번째 프레임으로의 변환  $T_{ij}$  는

$$T_{ij} = T_{j-1} T_{j-2} \cdots T_{i+2} T_{i+1} T_i \quad (31)$$

이다. 또한, 기준 프레임 이후의 변환은 기준 오른쪽 프레임의 변환 행렬로부터 시작해서, 오른쪽으로 한 프레임씩 전진하면서 각각의 왼쪽에 역행렬을 곱해 나간다. 임의의 아핀 변환에 대한 역행렬은 마찬가지로 아핀 변환이다.

합성된 파노라마 영상의 비교를 위하여 ORAD사의 제품 Digital Replay™의 데모 영상을 분석하였다. 그림 10의 (a)는 경기장 폭의 60% 이상이 나타난 파노라마 영상으로 가운데 흰 점선은 추적된 공의 궤적을 표시한다. 이 영상은 다른 그림에 비해 화면상에 등장하는 선수의 수가 적고 카메라의 줌인/아웃이 거의 발생하지 않았다. 그림에도 불구하고 (b)의 그림을 관찰하면, 영상의 기하학적 왜곡이 발생함을 알 수 있다. 기하학적으로 평행한 세 선의 연장선이 경기장의 중앙부로 올수록 치우친 형태이다. 제안된 알고리즘으로 합성된 파노라마 방법으로 연장선을 그

리면 그림 11과 같다. 여기에서는 중앙부에서도 거의 일치된 위치상에 존재함을 알 수 있다. 이러한 결과가 나온 이유는 그림 10을 합성하기 위해 계산된 전역 이동 계수의 부정확성 때문이다. 특히 기울어짐 계수(tilt parameter)의 계산이 이동 모델에 포함되지 않았거나, 거의 관찰되지 않았음을 의미한다. 파노라마 모자이크 합성에 관한 대부분의 문헌에서 아핀 변환 이상의 모델을 차용하기 때문에 포함되었을 가능성이 크다. 인접한 프레임간 고차항 이동 계수의 크기는 일반적으로 매우 작기 때문에 수치적으로 안정된 값을 얻기 어렵다. 그러나 본 논문에서 제안된 오차분할을 이용한 이동 계수의 계층적 향상 방법은 이러한 어려운 문제에 대하여 우수한 결과를 나타내었다.



(a)



(b)

(그림 10) ORAD사의 축구 파노라마 합성 영상  
(a) 원영상 (b) 기하학적 왜곡의 발생을 표시



(그림 11) 제안된 알고리즘으로 합성된  
파노라마 영상

## IV. 결론

본 논문에서는 스포츠 중계 동영상의 전역 이동 계수를 추출하기 위한 알고리즘을 제안하였고, 이를 이동 보상과 파노라마 합성에 이용할 수 있는 방법을 제안하였다. 기존의 여러가지 이동 해석 방법들이 있지만, 대상 영상이 축구 경기와 같은 특징점이 적고, 독립 개체가 많으며, 번짐에 의한 영상의 훼손 가능성이 높은 특별한 영상에 대해서 자동으로 수행되며 방송중계라는 실시간적 운용에 적합한 방법이 없었다.

제안된 오차 분할을 이용한 이동계수의 계층적 향상 방법은 인접 프레임 사이의 이동을 연속적으로 적용할 때, 누적되는 오차를 최소화하기 위하여 제안되었다. 아핀 변환 또는 원근 투영 변환은 3x3 크기의 행렬로 표현된다. 이러한 변환 행렬의 제공근을 구하는 문제를 정의하였고, Newton Raphson 방법을 행렬 형태로 변형하여, 매우 빠르고 안정적인 방법으로 해를 얻었다. 이 방법을 이용하여 다중 프레임 간의 이동으로부터, 사이 프레임들의 이동을 보간하는 방법을 제안하였다. 보간은 등간격으로 수행되거나 또는 별도로 정의된 속도 모델에 의하여 계산하는 방법을 제안하였다. 또한 기존에 계산된 변환 행렬의 크기만을 이용하거나 기존 변환을 합성한 행렬과의 오차를 분할하여 보정하는 방법을 제안하였다. 오차를 분산시키는 과정에서 다시 변환행렬의 고차 제공근을 구하는 문제를 풀게 된다. 이러한 부분 알고리즘들은 각 장마다 수행된 시뮬레이션에 의하여 검증되었다.

제안된 알고리즘을 이용하여, 실제 축구 경기 방송용 영상을 입력하여 파노라마를 합성한 결과, 비록 많은 양의 이동과 번짐이 발생하고 선수들의 움직임이 존재하는 경우라도, 비교적 좋

은 영상을 얻을 수 있었다. 그러나 영상에 따라서 확대 축소 이동을 정확하게 추정하지 못하여, 기하학적 왜곡이 발생하였다. 제안된 알고리즘을 적용하면 여러가지 개선 효과를 나타내었다. 특히 카메라의 좌우 팬이동에 의한 가로선이 곡선화 되는 현상이 많이 줄어들게 되었다. 이는 기존 방법에서 볼 수 없었던 결과로써 기하학적 왜곡을 억제한 파노라마 영상을 합성한 것을 의미한다. 이와 같이 본 논문에서 제안된 전역 이동 추정 방법은 파노라마 합성과 이동 보상을 이용한 방송용 특수 효과의 연출등에 사용이 가능하며, 향후 비디오 인덱싱 및 저전송률 동영상 부호화 등에 활용될 것으로 보인다.

## 참고문헌

- [1] Demin Wang; Limin Wang(1997), Global motion parameters estimation using a fast and robust algorithm, *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5), 823-826.
- [2] Y. T. Tse, and R. L. Baker(1991), Global zoom/pan estimation and compensation for video compression, *IEEE Proc. ICASSP'91, IV*, 2725-2728.
- [3] Frédéric Dufaux and Janusz Konrad(2000), Efficient, robust, and fast global motion estimation for video coding, *IEEE Transactions on Image Processing*, 9(3), 497-501.
- [4] MPEG 7 Requirements Group(1998), Applications for MPEG 7, *Doc. ISO/MPEG N2084*, MPEG San Jose Meeting.
- [5] M. Irani, B. Rousso, S. Peleg(1997), Recovery of ego-motion using region alignment, *IEEE Transactions on PAMI*, 19(3), 268-272.
- [6] M. Irani, P. Anadan, A unified approach to moving object detection in 2D and 3D scenes, *IEEE Transactions on PAMI, Scheduled*.
- [7] Y. Rosenerg, M. Werman(1998), Real-time object tracking from a moving video camera: A software approach on a PC, *IEEE Workshop on Applications of Computer Vision*, 238-239.
- [8] D. Kato, M. Yamada, K. Abe, A. Ishikawa, K. Ishiyama, and M. Obata(1997), Analysis of the camerawork of broadcasting cameramen, *Journal of SMPTE*, 108-116
- [9] Earl Gose, Richard Johnsonbaugh and Steve Jost(1996), *Pattern recognition and image analysis*, Prentice-Hall.
- [10] Ramesh Jain, Rangachar Kasturi and Brian G. Schunck(1995), *Machine Vision*, McGraw-Hill.

## Hierrachical manner of motion parameters for sports video mosaicking

Jae-Cheol Lee\* · Soo-Jong Lee\*\* · Young-Hoon Ko\*\*\* ·  
Heung-Sik Noh\*\*\*\* · Wan-Ju Lee\*\*\*\*\*

### Abstract

Sports scene is characterized by large amount of global motion due to pan and zoom of camera motion, and includes many small objects moving independently. Some short period of sports games is thrilling to viewers, and important to producers. At the same time that kinds of scenes exhibit exceptionally dynamic motions and it is very difficult to analyze the motions with conventional algorithms. In this thesis, several algorithms are proposed for global motion analysis on these dynamic scenes. It is shown that proposed algorithms worked well for motion compensation and panorama synthesis.

When cascading the inter frame motions, accumulated errors are unavoidable. In order to minimize these errors, interpolation method of motion vectors is introduced. Affine transform or perspective projection transform is regarded as a square matrix, which can be factorized into small amount of motion vectors. To solve factorization problem, we proposed the adaptation of Newton Raphson method into vector and matrix form, which is also computationally efficient. Combining multi frame motion estimation and the corresponding interpolation in hierarchical manner, enhancement algorithm of motion parameters is proposed, which is suitable for motion compensation and panorama synthesis. The proposed algorithms are suitable for special effect rendering for broadcast system, video indexing, tracking in complex scenes, and other fields requiring global motion estimation.

Key words : motion compensation, panorama, motion interpolation, Newton Raphson algorithm