DirectX를 이용한 실시간 영상 모자익

정 민 영[†] · 최 승 현[††] · 배 기 태[†††] · 이 칠 우[††††]

요 약

본 논문에서는 일반 PC에서 방사형으로 배치된 카메라를 통해 획득되는 비디오 영상을 하나의 대형 고해상도 영상으로 만드는 실시간 영상 모자익 기법에 관해 기술한다. 제안된 방법은 먼저 위상 상관 알고리즘을 사용하여 인접하는 두 영상간의 수평 및 수직 이동거리를 산출한 다음, Levenberg-Marquardt 방법을 사용하여 카메라 사이의 정확한 변환 행렬을 계산한다. 마지막으로 DirectX의 텍스처 매핑 함수에 변환행렬을 적용하여 입력영상들을 하나의 대형 영상으로 합성한다. 이 방법의 특징은 일반 개인용 컴퓨터에서 널리 사용되고 있는 그래픽 API DirectX를 영상 합성과정에 이용하기 때문에 특별한 장치와 기계어 수준의 프로그래밍 없이도 실시간 영상 모자익을 구현할 수 있다는 것이다.

Real-Time Image Mosaic Using DirectX

Min-Yeong Chong[†] · Seung-Hyun Choi[††] · Ki-Tae Bae[†††] · Chil-Woo Lee[††††]

ABSTRACT

In this paper, we describe a fast image mosaic method for constructing a large-scale image with video image captured from cameras that are arranged in radial shape. In the first step, we adopt the phase correlation algorithm to estimate the horizontal and vertical displacement between two adjacent images. Secondly, we calculate the accurate transform matrix among those cameras with Levenberg-Marquardt method. In the last step, those images are stitched into one large scale image in real-time by applying the transform matrix to the texture mapping function of DirectX. The feature of the method is that we do not need to use special hardware devices or write machine-level programs for implementing a real-time mosaic system since we use conventional graphic APIs (Application Programming Interfaces), DirectX for image synthesis process.

키워드 : 영상 모자익(Image Mosaic), 위상 상관(Phase Correlation), 레번버그-마쿼드법(Levenberg-Marquardt method), 다이렉트 액스(DirectX), 실시간(Real-Time)

1. 서 론

영상 모자익은 고해상도의 대규모 영상을 획득하는 영상 기반 렌더링 기법 중의 하나로서, 가상현실, 대화형 멀티 플래이어 게임, 가시화 분야 등과 같은 많은 응용분야에 사용되고 있다[1]. 여러 카메라로부터 획득되는 영상들을 하나의 커다란 모자익 영상으로 정합하고 합성함으로써 높은 해상도를 가진 영상을 완성시킬 수 있다[7, 17]. 또한 임의의 시점에서 촬영한 영상들을 하나로 통합시킴은 물론 계산에 의해 시점을 달리하여 표현할 수 있기 때문에 건축내부와 같은 복잡한 장면들을 그래픽모델을 사용하지 않고 간단히 나타낼 수 있다[12].

영상 모자익에 있어서 중요한 요소기술은 두 영상의 중첩 영역을 찾아 정합하는 기술과 중첩영역을 기준으로 여러 영상을 불연속점 또는 왜곡 없이 매끄럽게 합성하는 기술이다. 따라서 모자익 기술은 근본적으로 컴퓨터비전에서 사용하는 투영변환(perspective transformation) 이론과 컴퓨터그래픽스에서 사용하는 영상워핑(image warping) 혹은 영상 렌더링(image rendering) 이론을 결합한 기술이라 할 수 있다.

일반적으로 영상모자익 처리에는 임의의 영상을 새로운 영상으로 변환시키기 때문에 각종 파라메터를 고려해야 하고, 부동 소수점 좌표 계산을 동반하기 때문에 계산이 복잡하여 처리속도가 느리다는 문제가 있다[8, 11, 15]. 영상정합에 있어서 이러한 문제를 개선하고 고품질의 모자익 영상을 얻기 위해 영상간의 상관을 이용하거나[6, 21], 특징점간의 거리제약[14, 16, 20], 또는 광류를 이용하는 방법[2]들이 고안되었다.

그러나 영상 모자익은 궁극적으로 크기가 큰 영상의 모든 픽셀들의 변환을 필요로 하기 때문에 실시간 처리가 매우 어렵다. 물론 컬러 정보의 보간(Interpolation) 수준에 따라 속도와 품질의 Trade off는 존재할 수 있지만, 영상의 픽셀 수만큼의 부동 소수점 좌표에 대한 계산이 수행되어야 하기 때문에 실시간 처리에는 문제가 있다[9, 10].

※ 본 연구는 한국과학재단 지정 전남대학교 "고품질 전기 전자 부품 및 시스템 연구센터"의 연구비 지원에 의해 수행되었음.
† 정 회 원 : 광주여자대학교 멀티미디어학과 교수
†† 정 회 원 : (주) 3R Game 연구원
††† 준 회 원 : 전남대학교 대학원 컴퓨터공학과
†††† 정 회 원 : 전남대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2003년 9월 8일, 심사완료 : 2003년 10월 24일

이러한 문제점을 해결하기 위해 콜롬비아 대학에서는 전방향 카메라라는 어안 렌즈(Fish-eye)를 고안하여 영상 합성에 이용하였다[3]. 180° 회전할 수 있는 어안 렌즈 2개를 사용하여 전반적인 절차를 단순화시킴으로써 영상합성에 있어 발생할 수 있는 누적 오류를 최소화하였다. 또한 Szeliski[4]는 Stitcher라 불리는 360° 회전 파노라마 생성기를 고안하였는데, 카메라를 삼각대에 고정시켜놓고 주변 환경을 360° 회전하여 찍은 다수의 영상을 합성하여 원통형 회전영상을 만드는데 사용하였다. 그러나 이러한 방법들은 특수한 장비에 의존한다는 면에서 일반적이지 못하고 2개의 렌즈만을 사용하기 때문에 고성능의 카메라를 사용하지 않는다면 해상도 측면에서 장점을 기대하기 힘들다[13].

실시간 처리는 초당 수십 프레임을 처리하는 것이므로, 영상의 실시간 처리를 위해서 대부분 영상처리 및 그래픽스 라이브러리에서는 기계어 수준의 SIMD(Single Instruction Multiple Data) 코딩을 하는 것이 일반적이다. SIMD 코딩은 병렬처리 기반으로 다량의 반복적 작업을 효율적으로 처리할 수 있게 하는 기법이다. 이를 이용하면 획기적으로 계산 속도를 향상시킬 수 있는 것이 사실이지만 기계어 수준의 언어를 사용하여 복잡한 응용프로그램을 개발하는 것은 결코 쉬운 일이 아니다.

DirectX는 윈도우즈 운영체제 기반의 응용프로그램에서 그래픽 영상을 생성하고 관리하는 응용 프로그램 인터페이스(API ; Application Programming Interface)이다. 이것은 소프트웨어 개발자가 하드웨어중심 저급 코드를 작성하지 않고도 3차원 그래픽 가속 칩과 같은 특별한 하드웨어 특징에 접근하게 만드는 윈도우기반 PC의 표준 개발 플랫폼을 제공한다. 따라서 우리는 이것의 텍스처 매핑 기술을 사용하여 실시간 영상 모자익을 구성할 수 있다.

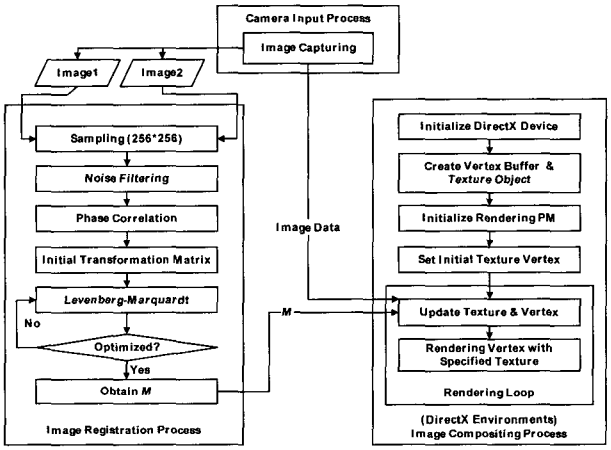
본 논문에서는 이러한 사실들을 기반으로 일반 PC에서 방사형으로 배치한 카메라를 통해 획득되는 다수의 영상을 DirectX를 이용하여 고속으로 합성하는 방법을 제안한다. 이 방법의 특징은 카메라와 영상보드 이외의 특별한 별도 하드웨어를 사용하지 않고도 고속 모자익 처리가 가능하고, 기계어 수준의 언어로 복잡한 응용프로그램을 개발하지 않아도 된다는 것이다. 인접하는 영상간의 대응점을 빠른 속도로 추출하기 위해 주파수기반의 위상 상관 방법을 이용하여 영상간 수직 및 수평 이동 거리를 계산하여 초기 변환 행렬로 하고, 회전과 크기를 고려하여 오차를 최소화하기 위한 Levenberg-Marquardt 알고리즘을 사용하여 최적의 변환행렬을 구한다.

본 논문의 구성은 다음과 같다. 2절에서는 두 영상간의 기하학적 변환이론과 정확한 변환 행렬을 구하기 위한 최적화 알고리즘에 대해 기술하고, 3절에서는 DirectX를 이용한 고속 영상 합성 방법에 대해 설명하고, 4절에서는 실험 및 고찰에 대해 기술하며, 마지막으로 5절에서 결론을 맺고 향후 연구 과제에 대해서 언급한다.

2. 위상상관에 의한 영상변환 및 최적화

2.1 고속 영상 모자익 프로세스

일반적인 영상 모자익은 영상 정합 프로세스와 영상 합성 프로세스로 구성된다. 영상 정합은 대응점을 사이의 변환행렬을 계산하여 두 개 이상의 영상을 맞추는 작업이고, 영상 합성은 명도의 불연속과 기하학적 왜곡 없이 중첩영역의 대응관계를 기반으로 영상들을 붙이는 작업이다. (그림1)은 카메라 입력 프로세스를 추가한 본 논문의 전반적인 영상 모자익 프로세스를 나타낸 것이다. 카메라 프로세스는 영상 캡처 보드로부터 입력된 영상을 영상 정합 프로세스와 영상 합성 프로세스로 넘겨주는 역할을 한다. 영상 정합 프로세스는 입력 영상들에 대한 변환 행렬을 계산하는 과정이며, 영상 합성 프로세스는 카메라로부터의 입력 스트림(stream)과 계산된 변환행렬을 받아들여 DirectX 함수를 이용하여 모자익 영상을 만들어내는 과정이다.



(그림 1) 고속 영상 모자익 과정

본 논문에서는 전체 프로세스에서 모든 입력 영상이 인접 영상과 최소한 1/4 이상의 중첩된 영역이 있는 것으로 가정한다. 이러한 제약사항은 입력영상의 예기치 않은 회전 및 크기 변화 요소를 줄이고, 영상사이의 안정적인 대응관계를 산출해내게 하기 위한 것이다.

2.2 입력 영상의 전처리

본 논문에서는 인접하는 영상간의 대응점을 빠른 속도로 찾기 위해 주파수기반의 위상 상관 알고리즘을 사용한다. 이 방법은 FFT(Fast Fourier Transform) 알고리즘에 기반을 두기 때문에 먼저 입력 영상을 256×256 크기로 샘플링을 통해 조정하고, 평균 필터를 적용하여 고주파 잡음을 제거한다. 다른 응용분야에서처럼 컬러 카메라를 사용하여 입력영상을 얻지만, 정합 프로세스에서는 컬러정보를 사용할 필요가 없으므로 입력영상을 색상과 채도, 그리고 명도의 구성요소로 이루어진 HSI 컬러모델로 변환하여 명도(In-

tensity) 정보만으로 구성된 영상으로 변경하고 명도정보만 영상 정합 프로세스에 사용한다. 영상정합을 한 후에는 완전한 컬러 영상들을 영상 합성 프로세스에 적용하여 하나의 모자익 영상을 만들게 된다.

2.3 위상 상관법을 이용한 영상 정합

영상 모자익을 위한 입력영상을 촬영할 때 카메라의 움직임이 적다고 가정하면 카메라 관련 운동 변수는 수평 또는 수직 요소에 의해 좌우되게 된다. 고종호[5]가 제안한 방법은 어느 정도의 회전에 대해서도 수용하는 방법을 제안하였지만 특별한 환경이 아니라면 큰 의미가 없고 계산시간 측면에서 실시간 모자익에 적합하지 않다. 따라서 본 논문에서는 모자익이 최소한의 영상정보를 이용해 가능한 넓은 시야를 표현하기 위한 기술이라고 가정하여 영상 사이의 주요 운동 요소인 수평 및 약간의 수직 운동 요소를 주요 변수로 고려하여 위상 상관법(Phase Correlation Method)에 의해 매칭 관계를 구한다.

위상 상관법은 Reddy와 Chatterji[6]에 의해 제안된 것으로, 퓨리에(Fourier) 변환에 기반을 둔 주파수간 상관 관계를 이용해 운동 요소를 계산하는 방법이다. 이 방법은 아핀 변환을 기반으로 병진 이동, 회전, 크기에 대한 계산이 가능하다. 그러나 여기서는 먼저 원근 변환을 사용하기 때문에 카메라의 작은 움직임을 고려하면 병진 이동만이 의미가 있다. 회전과 크기에 관련된 변환행렬의 최적화는 2.4절에서 기술한다.

공간영역에서 인접하는 영상을 f_1, f_2라 하고, f_1에 대해 f_2가 (x_0, y_0) 만큼의 병진 이동을 갖는다면 두 영상간의 관계는 다음과 같이 표현될 수 있다.

$$f_2(x, y) = f_1(x - x_0, y - y_0) \tag{1}$$

이것을 대응되는 퓨리에 변환 관계로 나타내면

$$F_2(\xi, \eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} F_1(\xi, \eta) \tag{2}$$

이 된다.

두 영상에 대한 정규화된 크로스적 스펙트럼(cross-power spectrum)을 취하면 다음과 같이 간단한 지수 방정식이 된다.

$$\frac{F_1(\xi, \eta) F_2^*(\xi, \eta)}{|F_1(\xi, \eta) F_2(\xi, \eta)|} = e^{j2\pi(\xi x_0 + \eta y_0)} \tag{3}$$

여기서 $*$는 공액복소수를 나타낸다. 퓨리에 이동 정리(Fourier Shift theorem)에 의하면 크로스적 스펙트럼의 위상(Phase)은 두 영상간 위상 차이와 동일하다. 이 점을 이용하여 식 (3)의 역 퓨리에 변환을 하면 다음과 같이 된다.

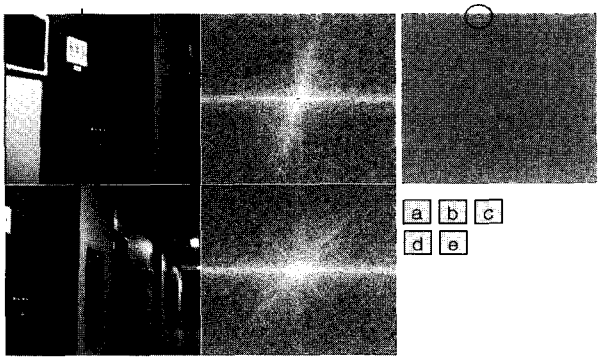
$$\mathcal{F}^{-1}[e^{j2\pi(\xi x_0 + \eta y_0)}] = \delta(x_0, y_0) \tag{4}$$

대응되는 영상이 수평 및 수직 운동 요소만 갖고 있고 Outlier에 의한 간섭이 없을 경우 (x_0, y_0) 위치에 하나의 강한 임펄스만 존재해야 할 것이다.

그러나 실제 영상에서는 잡음 요소 때문에 이런 결과를 항상 얻을 수 없고, 결과 영상에 많은 수의 임펄스가 나타나게 된다. 하지만 대응영상의 중첩영역이 최소 1/4 이상이 된다면 가장 큰 운동 요소의 임펄스 값이 가장 크게 나온다. 간단한 임계치 적용법을 사용하면 (x_0, y_0)를 구할 수 있다. 이는 두 영상간의 글로벌 모션에 해당된다.

(그림 2)는 간단한 실험으로부터 얻은 결과를 보여준다. (그림 2)(c)에서 동그라미로 체크해 둔 부분이 임펄스(Impulse)가 모여 있는 부분이다. 이 때 전체 영역에서 가장 강한 임펄스 요소의 위치가 바로 병진 이동 영상의 글로벌 모션 벡터가 된다.

(그림 2)(a)에서 크로스 표시는 영상공간에서 실제로 병진 이동된 거리를 나타낸다. (그림 2)(a)의 크로스 표시와 (그림 2)(c)의 임펄스 위치를 비교해 보면 얼마나 정확히 병진 이동량을 계산했는지 알 수 있다.



(그림 2) (a) 전 처리된 왼쪽 영상, (b) (a)의 FFT 결과,
(c) (a)와 (b)의 역 퓨리어 변환 영상,
(d) 전 처리된 오른쪽 영상, (e) (d)의 FFT 결과.

2.4 초기 변환 행렬 계산

초기 중첩영역은 앞서 기술한 위상 상관 알고리즘을 이용해서 쉽게 얻을 수 있다. 좌측 상단 기준 영상 좌표계를 사용했을 때 처음 영상((그림 2)(a))의 중첩영역은 (x_0, y_0)에서 그 영상의 우측 하단 $(x_{\max}, y_{\max})$까지의 사각 영역이 된다. 중첩 영역의 넓이와 높이를 (x_{rect}, y_{rect})이라 하면 대응 영상((그림 2)(d))의 중첩영역은 대각 좌표 (0, 0)에서 (x_{rect}, y_{rect})까지의 사각 영역이다. (x_{rect}, y_{rect})는 다음과 같이 계산된다.

$$\begin{aligned} x_{rect} &= x_{\max} - x_0 \\ y_{rect} &= y_{\max} - y_0 \end{aligned} \tag{5}$$

식 (5)에서 구한 초기 운동 요소를 사용하면 식 (11)과 같은 초기 변환 행렬 $\boldsymbol{M}_{init}$를 구성할 수 있다.

2차원 좌표계의 임의의 점 (x, y)와 새로운 점 (x', y')를 각각 homogeneous 좌표 $(x, y, 1)$과 (X', Y', D)로 나타내면 (X', Y', D)의 경우 Cartesian 좌표 $\left(\frac{X'}{D}, \frac{Y'}{D}\right)$이 되므로 식 (6)과 같은 관계가 성립한다.

$$x' = \frac{X'}{D}, \quad y' = \frac{Y'}{D} \tag{6}$$

$\boldsymbol{p} = [x, y, 1]^T$, $\boldsymbol{p}' = [X', Y', D]^T$라 하면 2차원 투영 변환에 의해 다음과 같은 식이 성립된다[19].

$$\boldsymbol{p}' = \boldsymbol{M}\boldsymbol{p} \Leftrightarrow \begin{bmatrix} X' \\ Y' \\ D \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{7}$$

여기서 $\boldsymbol{M}$은 변환행렬, $\boldsymbol{M}$의 요소 m_0, m_1, m_3, m_4는 회전과 크기 요소, m_2, m_5는 이동요소, m_6, m_7는 비례변환요소, m_8은 1을 나타낸다. 식 (7)을 풀어쓰면 식 (8)을 얻을 수 있고 이를 식 (6)에 대입하면 식 (9)와 식 (10)을 구할 수 있다.

$$\begin{bmatrix} X' \\ Y' \\ D \end{bmatrix} = \begin{bmatrix} m_0 x + m_1 y + m_2 \\ m_3 x + m_4 y + m_5 \\ m_6 x + m_7 y + 1 \end{bmatrix} \tag{8}$$

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1} \tag{9}$$

$$y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1} \tag{10}$$

초기 변환행렬 $\boldsymbol{M}_{init}$는 m_2, m_5가 각각 수평 및 수직 이동요소로 식 (7)의 x_0, y_0에 해당되므로 식 (11)과 같이 구성할 수 있다.

$$\boldsymbol{M}_{init} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

2.5 Levenberg-Marquardt 방법을 이용한 변환 행렬의 최적화

초기 변환 행렬에는 병진이동 요소만 포함되어 있다. 그러나 카메라의 상태 및 영상 취득 환경에 따라 크기 및 회전 요소를 고려하지 않을 수 없다. 또한 크기와 회전 요소의 계산결과 오차가 병진 운동과 전체 운동 변수에 영향을 미칠 수도 있다. 이러한 오차를 최소화하기 위해 본 논문에서는 비선형 계열의 최적화 알고리즘인 Levenberg-Marquardt 방법[18]을 사용한다. Levenberg-Marquardt 방법은 국부적 명도 오차와 오차의 기울기를 기반으로 최소치에 접근해 가는 방법이다. 이 방법은 중첩영역에서 명도 오차의 합 E가 최소가 되게 하는 새로운 변환 행렬 M을 계산하기 위해 미지의 운동 요소 $\{m_0, \cdots, m_7\}$에 대한 e_i의 편미분 값을 사용한다. 중첩영역에서 명도 오차의 합 E와 중첩영역의 각 대응점 쌍의 명도 값 오차 e_i는 다음과 같이 정의한다.

$$\boldsymbol{E} = \sum_i \boldsymbol{e}_i \tag{12}$$

$$\boldsymbol{e}_i = \boldsymbol{I}'(x_i', y_i') - \boldsymbol{I}(x_i, y_i) \tag{13}$$

여기서, $I(x_i, y_i)$는 (x_i, y_i) 픽셀의 명도값(Intensity)을 나타낸다.

변환행렬의 최적화란 식 (12)의 값이 최소가 되는 변환행렬 M을 구하는 것이다. 여기에서 8개의 대응되는 운동요소의 편미분값은 식 (9), 식 (10), 식 (13)으로부터 다음과 같이 구할 수 있다.

$$\left.\begin{aligned} \frac{\partial e}{\partial m_3} &= \frac{\partial e}{\partial m_0} = \frac{\partial x}{\boldsymbol{D}} \frac{\partial \boldsymbol{I}'}{\partial x'} \\ \frac{\partial e}{\partial m_4} &= \frac{\partial e}{\partial m_1} = \frac{\partial y}{\boldsymbol{D}} \frac{\partial \boldsymbol{I}'}{\partial x'} \\ \frac{\partial e}{\partial m_5} &= \frac{\partial e}{\partial m_2} = \frac{1}{\boldsymbol{D}} \frac{\partial \boldsymbol{I}'}{\partial x'} \\ \frac{\partial e}{\partial m_6} &= -\frac{x}{\boldsymbol{D}}\left(x' \frac{\partial \boldsymbol{I}'}{\partial x'} + y' \frac{\partial \boldsymbol{I}'}{\partial y'}\right) \\ \frac{\partial e}{\partial m_7} &= -\frac{y}{\boldsymbol{D}}\left(x' \frac{\partial \boldsymbol{I}'}{\partial x'} + y' \frac{\partial \boldsymbol{I}'}{\partial y'}\right) \end{aligned}\right\} \tag{14}$$

Levenberg-Marquardt 알고리즘에서는 식 (14)를 식 (15)에 대입하여 헷세 행렬 $\boldsymbol{A}$와 가중 기울기 벡터 $\boldsymbol{b}$를 계산한다.

$$\boldsymbol{A} = a_{kl} = \sum_i \frac{\partial e_i}{\partial m_k} \frac{\partial e_i}{\partial m_l}, \quad \boldsymbol{b} = b_k = -2\sum_i e_i \frac{\partial e_i}{\partial m_k} \tag{15}$$

여기서 i는 1부터 4까지의 대응점을, k와 l은 0부터 7까지의 운동 요소를 나타내는 첨자이다. 이 결과를 이용하여 Levenberg-Marquardt 알고리즘의 변위 계산식 $\triangle \boldsymbol{m} = (\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}\boldsymbol{b}$에 대입하여 운동 변수 $\boldsymbol{m}$을 새로 계산하는 것이 가능하다. 여기에서 λ는 반복 계산에 대한 안정화 변수이다. 식 (12)에 의해 오차의 합을 계산했을 때 이전 단계보다 값이 크면 λ를 증가시키고, 그렇지 않으면 감소시켜 새로운 $\triangle \boldsymbol{m}$을 계산해 최적의 변환행렬 $\boldsymbol{M}$을 구한다. 이 알고리즘은 오차의 합이 임계치보다 작아질 때까지 반복적으로 새로운 값을 계산하여 결국 최적의 변환행렬 $\boldsymbol{M}$을 계산해 낸다.

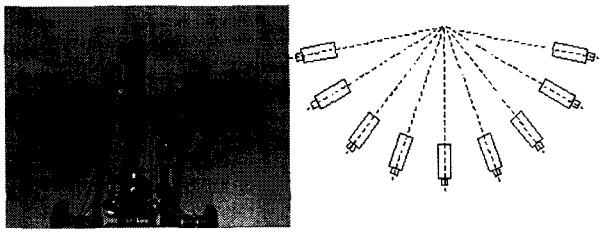
3. DirectX를 이용한 고속 영상 합성

3.1 고속처리를 위한 하드웨어 설정

컴퓨터를 이용한 영상합성에서는 영상이 많은 픽셀 데이터를 가지고 있고 그 데이터에 대한 수많은 부동소수점 연산을 필요로 하기 때문에 고속처리가 매우 중요하다. 앞서 언급한 바와 같이 많은 사양의 하드웨어 시스템이 영상 합성과 영상 생성을 위해 개발되었다. 그러나 일반적인 응용

소프트웨어를 위해 특별한 하드웨어 시스템을 사용하게 된다면 영상 시스템은 고가가 될 수밖에 없다. 본 논문에서는 저비용으로 대규모 실시간 모자익 영상을 생성할 수 있는 방법에 대해 기술한다.

본 논문에서는 고속 영상합성을 위해 영상 캡쳐 보드로부터 입력되는 영상을 영상 정합 프로세스의 샘플링 루틴과 DirectX의 렌더링 루프에 직접 전달한다. 먼저 4대 이상의 카메라 입력을 30fps로 처리 할 수 있고 최대 640×480의 해상도를 제공하는 영상캡쳐보드를 이용하여 영상을 입력한다. 카메라는 변환 관계 계산에서 기하학적 오류를 줄이기 위해 (그림 3)(a)와 같이 방사형으로 배열한다. (그림 3)(b)는 이상적인 카메라 배열을 보여주고 있다.

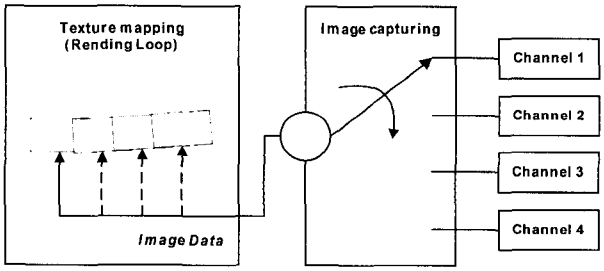


(a) (b)

(그림 3) 카메라 배열 형태

카메라로부터 입력되는 RGB 영상 데이터는 텍스처 갱신 요청(Texture Update Call)에 의해 채널별로 해당 텍스처 버퍼에 입력된다. 이 때 DirectX의 화면 렌더링 속도와 영상캡쳐보드의 캡쳐속도가 현격히 차이가 나기 때문에 화면 갱신은 DirectX 렌더링 속도에 따르고 영상 업데이트는 캡쳐보드의 속도에 의해 영향을 받는다.

(그림 4)는 4대의 카메라로부터 영상을 획득하여 순차적으로 DirectX의 렌더링 루프에 전송하는 영상 캡쳐 구조를 나타낸 것이다. 이것은 카메라로부터 입력되는 영상이 렌더링 루프에 병렬로 전달되는 것이 아니고 직렬로 전달된다는 것을 의미한다.



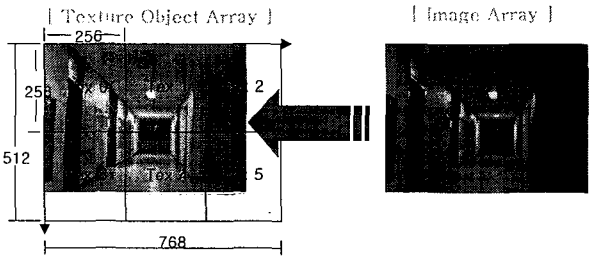
(그림 4) 영상 캡쳐 구조

3.2 DirectX를 이용한 영상 생성

DirectX의 그래픽 기능 중 Direct3D는 3차원 공간에서 각종 변환을 수행하는 API이다. 본 논문에서 고려하는 2D 기반의 모자익 기술을 Direct3D에 적용하려면 3차원 공간에 2차원 평면을 정의하고 매핑해야 한다. 먼저 3차원 공간의 원점을 기준으로 수평축을 따라 텍스처들을 매핑할 수 있는 3차원 기반 2차원 가상 평면을 정의한다. 그런 다음, 입력 영상을 텍스트 객체의 텍스트 공간에 채우고, 최종적으로 텍스처 객체를 가상 평면에 매핑시킨다.

DirectX의 텍스트 매핑 기능은 사각형 데이터 집합을 사각형이 아닌 임의의 다각형 영역에 매핑을 할 수 있도록 텍스처 내부의 데이터 요소(Texel)들의 기하변환이 아주 용이하고, 그래픽스 보드의 성능에 의존하여 동작하기 때문에 시스템 성능에 덜 영향을 받고 빠르게 수행된다는 장점이 있다. 다만 2의 자승의 크기의 정방형 영상에서 텍스처 매핑 기술이 최적의 성능을 발휘 할 수 있다는 것이 그 한계이다.

또한 일반적인 텍스처 매핑 방법은 하나의 영상을 표현하기 위해 하나의 텍스처 공간을 사용하기 때문에 메모리 낭비를 초래 할 수 있고, 텍스처 매핑의 속도에 직접적으로 영향을 주게 된다. 이에 비해 (그림 5)와 같은 분할 텍스처 매핑 방법은 메모리를 여러 개의 텍스처 공간으로 분할하여 각 텍스처 공간 하나 하나가 입력 영상의 한 부분으로 표현될 수 있기 때문에 메모리 공간을 절약할 수 있고 DirectX가 처리 할 수 있는 최적의 크기로 매핑을 수행할 수 있다.

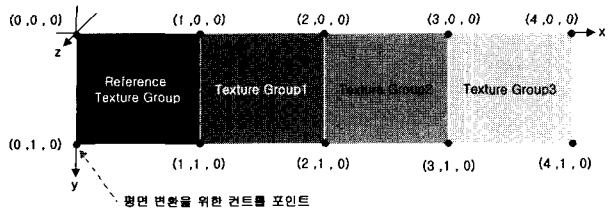


(그림 5) 분할 텍스처 매핑

따라서 본 논문에서는 한 장의 입력 영상을 256×256 크기로 고정된 다수의 텍스처 객체에 할당하고 그것을 가상 평면에 매핑하는 분할 텍스처 매핑 기법을 사용한다.

텍스처의 기하학적 변환은 정점(Vertex)의 변환에 의해 수행된다. 정점은 3차원 공간상에 드로잉 프리미티브(Drawing Primitive)를 구성하는 것으로서, 텍스처 매핑시 텍스처를 3차원 공간에 위치시키는 점이라고 할 수 있다. DirectX에서는 FVF(Flexible Vertex Format) 플래그(Flags)를 통해 다양한 형태의 정점 형식을 지원하고 있다. 본 논문의 구현에 사용하는 FVF 플래그는 다음과 같다.

```
D3DFVF_XYZ | D3DFVF_DIFFUSE | D3DFVF_TEX1
• D3DFVF_XYZ : x, y, z 좌표
• D3DFVF_DIFFUSE : Color Component
• D3DFVF_TEX1 : Texture Component
```

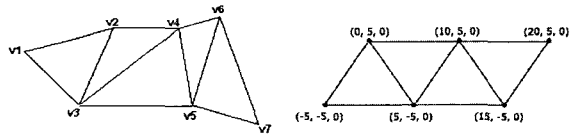


(그림 6) 텍스처 변환을 위한 정점 정렬

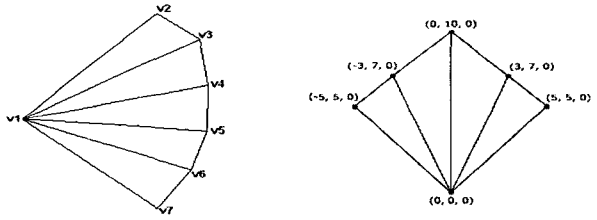
(그림 6)은 텍스처 매핑시 각 텍스처당 초기 정점들의 위치를 정의하고 있다. 이 정점들이 각 텍스처 그룹의 평면 변환을 위한 제어점이 된다. 텍스처 그룹을 하나의 텍스처로 가정한다면 각 텍스처는 4개의 정점으로 정의된다. 이 때 각 정점은 텍스처 데이터의 위치정보 (u, v)를 포함하게 된다. 텍스처 좌표는 (그림 6)에서 볼 수 있듯이 (0.0, 0.0)~(1.0, 1.0)까지의 정규화된 좌표로 표현된다. 이를 이용해 텍스처를 정점 위치에 따라 3차원 공간에 렌더링 하게 되는데 이 때 텍스처를 드로잉하는 방법은 다음과 같다.

- 점 : D3DPT_POINTLIST
- 선 : D3DPT_LINELIST, D3DPT_LINESTRIP
- 삼각형 : TRIANGLELIST, TRIANGLESTRIP, TRIANGLEFAN

이것은 드로잉 함수에 의해 정의되는 최소단위의 드로잉 프리미티브들이다. 텍스처 매핑은 삼각형의 프리미티브를 사용한다. 환경에 따라 다르지만 텍스처 매핑시에는 (그림 7), (그림 8)과 같은 스트립(Strip)형과 팬(Fan)형을 사용하는 것이 일반적이다.



(그림 7) 삼각 스트립(Triangle Strip)



(그림 8) 삼각 팬(Triangle Fan)

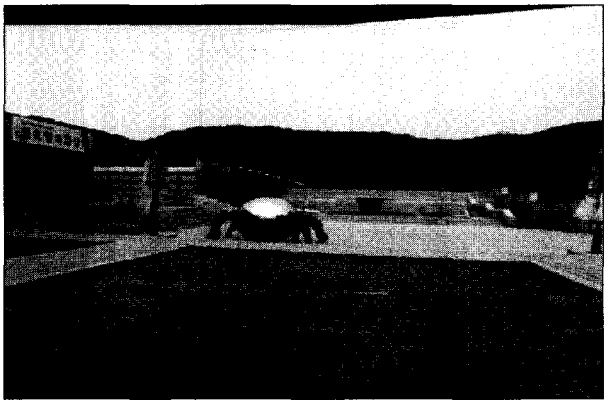
위의 방법들은 최소의 정점을 가지고 다양한 형태의 다면체를 표현하기 위해 고안된 렌더링 규칙들이다. 삼각형을 기반으로 하나의 사각형 텍스처를 표현하기 위해서는 6개의 정점이 필요하다. 그러나 스트립 방법을 사용하면 4개의 정점에 의해 표현이 가능하다. 본 논문에서는 스트립 방법을 사용하여 하나의 텍스처에 4개의 정점을 할당하여 2개의 삼각형으로 구성된 텍스처를 렌더링한다.

4. 실험 결과

실험은 보통의 개인용 컴퓨터(Pentium Ⅳ 1.4GHz CPU Clock, 256M MRAM)을 사용하여 구현하였다. 입력영상은 4대의 Secure CCN-256IA 카메라를 이용하여 취득하였고 VC++6.0을 이용하여 프로그래밍 하였다.

약간의 회전과 수평 및 수직 운동이 일어난 두 장의 영상에 대해 DirectX를 사용하지 않고 영상 정합 실험을 해보았다. 640×480의 크기와 1024×768 크기의 두 가지 경우에 대해 실험했고 정합결과는 (그림 9)와 (그림 10)에 보인다. 각각의 영상이 어떻게 매핑이 되는지 보이기 위해 매핑되는 부분의 주변에 안내선을 그었다.

(그림 9) 한 쌍의 640×480 정지영상에 적용한 결과(복도)



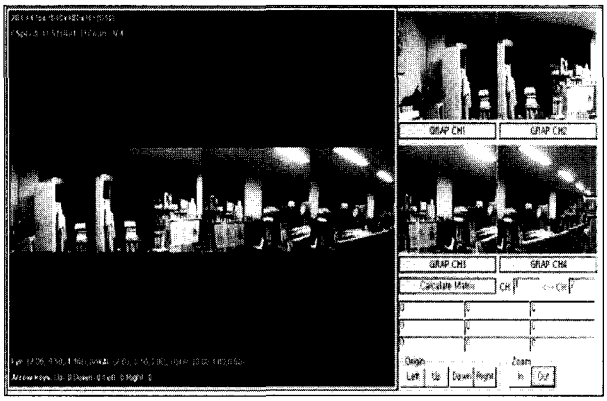
(그림 10) 한 쌍의 1024×768 정지영상에 적용한 결과(공원)

<표 1>은 (그림 9)와 (그림 10)을 얻는데 소요된 시간이다. (그림 9)의 실험 결과는 중첩영역이 큰 이유로 변환 행렬을 최소화하는 단계에서 상당히 많은 시간을 소모한 것을 보여준다. (그림 10)의 실험 결과는 상대적으로 큰 크기 때문에 최종 합성시간이 2배 이상 증가한 것을 보여준다. 그러나, (그림 9)와 (그림 10) 모두 변환 관계 계산에 소요되는 시간은 1초 내외라는 것을 알 수 있다.

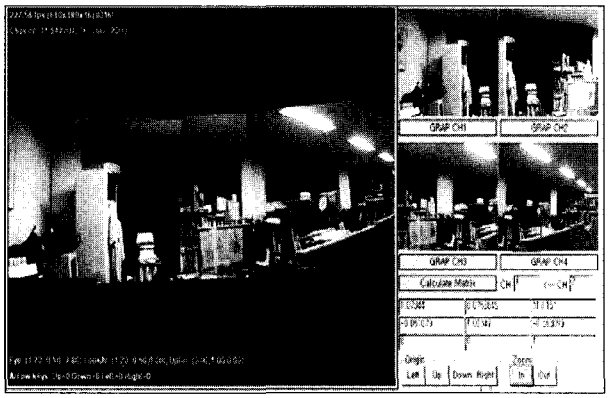
〈표 1〉 프로세스별 실행 속도 비교

Processes / Sample Size	Phase Correlation (sec)	Levenberg Marquardt (sec)	Color Composite (sec)	Total (sec)
640×480	0.218	1.238	0.938	2.395
1024×768	0.237	0.787	2.655	3.681

(그림 11)과 (그림 12)는 DirectX를 이용한 실시간 모자익의 결과를 보여준다. (그림 12)는 실시간 모자익 최종 결과를 나타내고 있다.



(그림 11) 실시간 모자익 초기 화면



(그림 12) 실시간 모자익 최종 결과 영상

<표 2>는 DirectX 환경하의 실시간 영상 처리 속도를 보여준다. <표 2>에서 DirectX 렌더링 속도가 영상의 입력 처리와 변환 행렬 계산과 같은 영상의 갱신 처리와 상관없이 렌더링 루프에서 화면이 갱신되는 시간에만 좌우되므로 매우 빠르게 모자익 영상을 생성할 수 있다는 것을 알 수 있다.

〈표 2〉 DirectX 하에서 영상 처리 속도

캡쳐 채널의 수	4 channels
DirectX 렌더링 속도	227.54 fps
영상 캡쳐 속도	30 fps

5. 결론 및 향후 연구 계획

본 논문에서는 방사형으로 배치한 여러 개의 카메라를 통해 획득되는 비디오 영상을 하나의 고해상도 영상으로 합성하는 고속 영상 모자익 방법에 대해 기술하였다. DirectX를 이용하여 카메라와 영상캡쳐보드 이외의 특별한 별도 하드웨어를 사용하지 않고 기계어 수준의 프로그래밍 노력 없이도 실시간으로 영상 모자익을 만들 수 있게 하였다. 또한 인접하는 영상간의 대응점을 주파수기반의 위상 상관법을 이용하여 신속히 검색하고, 이를 이용하여 초기 변환 행렬 모델을 구성한 다음, Levenberg-Marquardt 알고리즘을 사용하여 정확한 변환 행렬을 계산한다.

<표 2>의 결과를 보면 DirectX 렌더링 속도가 영상 갱신 처리와 상관없이 화면 갱신 시간에만 좌우되어 빠른 영상 모자익이 이루어진다는 것을 알 수 있다. 또한, DirectX를 사용한 실시간 합성 실험 결과 카메라와 대상간에 거리가 충분하다면 좋은 결과가 도출된다는 것을 알 수 있었다. 따라서 본 논문의 결과는 영상의 대응점 검색시 특징점 기반 방법의 검색 속도의 문제점을 비교적 노이즈나 명도변화에 덜 민감한 주파수 기반의 위상 상관법을 통해 극복하고, 영상 합성시 계산시간 문제를 DirectX의 텍스처 매핑기법을 통해 개선시킴으로써 저비용 고속 모자익 처리에 기여할 수 있다는 점에서 그 의의가 있다.

(그림 12)에서 원이 가리키는 영역은 카메라간의 운동오차로 인해 영상이 왜곡된 부분을 보여주고, 사각형으로 체크해 놓은 것을 보면 카메라의 초점이동에 따른 영상간의 차이를 보여주고 있어서 카메라의 국소 이동을 가정한 것에 대한 한계를 극복하지 못하고 있다. 이는 3차원 투영의 차이에서 오는 것이므로 향후 실세계의 3차원 운동에 대한 추가적 연구가 필요하다.

참 고 문 헌

[1] H.-Y. Shum and R. Szeliski, "Panoramic image mosaicing," Technical Report MSR-TR-97-23, Microsoft Research, 1997.

[2] Bruce D. Lucas and Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proceedings of Imaging Understanding Workshop, pp.121-130, 1981.

[3] Shree K. Nayar, "Omnidirectional Video Camera," Proc. of DARPA Image Understanding Workshop, New Orieans, May, 1997.

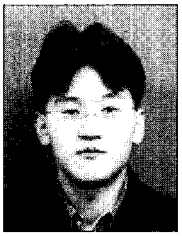
[4] R. Szeliski, "Video mosaics for virtual environments," IEEE Computer Graphics and Applications, pp.22-30, March, 1996.

[5] 고종호, "하우스도프 거리를 이용한 영역정합기반 자동영상 모자이킹", 전남대학교 컴퓨터공학과 석사학위 논문, 2001.

[6] B. Srinivasa Reddy and B. N. Chatterji, "A FFT-Based Technique for Translation, Rotaion, and Scale-Invariant Image Registration," IEEE Trans. Image Processaing, Vol.5 No.8, 1996.
[7] R. Szeliski, "Video mosaics for virtual environments," IEEE Computer Graphics and Applications, pp.22-30, March, 1996.
[8] R. Szeliski, "Image Mosaicing for Tele-Reality Applications," CRL Technical Report 94/2, Digital Equipment Corporation, 1994.
[9] M. Uyttendaele, A. Eden and R. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '2001), Kauai, Hawaii, Vol.II, pp. 509-516, December, 2001.
[10] J. Davis, "Mosaics of Scenes with Moving Objects," CVPR pp.354-360, 1998.
[11] S. E. Chen, "QuickTime VR - an image-based approach to virtual environment navigation," Computer Graphics (SIGGRAPH'95), pp.29-38, August, 1995.
[12] R. Szeliski, "Image-based modeling and rendering," In Fourth International Workshop on Cooperative and Distributed Vision, Kyoto, Japan, pp.413-429, March, 2001.
[13] S. B. Kang, R. Szeliski and J. Chai, "Handling occlusions in dense multi-view stereo," Technical Report MSR-TR-2001-80, Microsoft Research, September, 2001.
[14] S. M. Smith and J. M. Brady, "SUSAN - A New Approach to Low Level Image Processing," Technical Report TR95 SMS1c, 1995.
[15] I. Cohen, G. Medioni, "Detecting and Tracking Moving Objects for Video Surveillance," IEEE Proc. Computer Vision and Pattern Recognition, Jun., 1999.
[16] Naoki Chiba, Hiroshi Kano, Minoru Higashihara, Masashi Yasuda, and Osumi, "Feature-Based Image Mosaicing," MVA '98 IAPR Workshop on Machine Vision Applications, pp. 5-10, 1998.
[17] Mann, S. and Picard, R. W, "Virtual bellows : constructing high-quality images from video," in first IEEE International Conference on Image Processing (ICIP-94), Austin, Texas, Vol.1, pp.363-367, November, 1994.
[18] W. H. Press, B. P. Flannery, A. A. Teukolsky and W. T. Vetterling, "Numerical Recipes in C : The Art of Scientic Computing," Cambridge University Press Cambridge, England, Second edition, 1992.
[19] R. Hartley and A. Zisserman, "Multiple View Geometry," Cambridge University Press, 2000.
[20] 이동훠, "방사그리드형 모자익을 위한 영상 정합 기법", 전남대학교 컴퓨터공학과 석사학위 논문, 2002.
[21] 최승현, "위상 상관을 이용한 자동영상 모자익", 제15회 신호처리 합동 학술대회, p.289, 2002.

정 민 영

e-mail : mychong@mail.kwu.ac.kr
1991년 숭실대학교 전자계산학과(학사)
1993년 숭실대학교 대학원 전자계산학과 (공학석사)
2002년 전남대학교 대학원 컴퓨터공학과 (박사수료)
1993년~1996년 숭실대학교 중앙전자계산소 연구원
1997년~1999년 광주여자대학교 정보전산원장
1996년~현재 광주여자대학교 멀티미디어학과 교수
관심분야 : 컴퓨터비젼, 멀티미디어 데이터베이스, 컴퓨터그래픽스, 소프트웨어공학



최 승 현

e-mail : okdds74@hotmail.com
2001년 호남대학교 정보통신공학과(학사)
2003년 전남대학교 대학원 컴퓨터공학과 (공학 석사)
2003년~현재 (주)3R Games 연구원
관심분야 : 실사기반 렌더링 및 3차원 그래픽스

배 기 태

e-mail : bkt2002@empal.com
1997년 호원대학교 전자계산학과(학사)
1999년 전남대학교 대학원 컴퓨터공학과 (공학 석사)
2003년~현재 전남대학교 대학원 컴퓨터공학과(박사 수료)
관심분야 : 컴퓨터비젼, 멀티미디어 데이터베이스, 패턴인식, 컴퓨터그래픽스, 영상처리

이 칠 우

e-mail : leecw@chonnam.ac.kr
1986년 중앙대학교 전자공학과(학사)
1988년 중앙대학교 대학원 전자공학과 (공학석사)
1992년 동경대학교 대학원 전자공학과 (공학박사)
1992년~1995년 이미지 정보과학 연구소 수석 연구원겸 오사까대학 기초공학부 협력연구원
1995년 리츠메이칸 대학 특별초빙강사
1996년~현재 전남대학교 전자컴퓨터정보통신공학부 교수
관심분야 : 컴퓨터비젼, 멀티미디어 데이터베이스, 컴퓨터그래픽스