

타원체형 모델 기반의 영상정렬 알고리즘을 이용한 얼굴 텍스처 생성

(Facial Texture Generation using an Image Registration Algorithm based on Ellipsoidal Prototype Model)

이 중 재 [†] 노 명 우 [†] 최 형 일 ^{**}
(Joong Jae Lee) (Myung Woo Noh) (Hyung Il Choi)

요 약 본 논문에서는 얼굴 형태와 유사한 타원체형 모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 제안한다. 블록정합 수행 시 좌우 곡률만을 고려한 기존의 실린더형 원형모델 방법은 좌우 영상에 대하여 정확한 정렬을 수행하지만 상하 영상에 대해서는 사람의 두상모양이나 턱 구조의 특징을 반영하지 않았기 때문에 정렬오류가 발생한다. 본 논문에서 제안한 알고리즘은 타원체형 얼굴 모델의 상하 좌우의 곡률에 따라 가변적인 블록크기를 사용하는 블록정합 알고리즘으로서 영상의 상관관계를 이용하여 정확한 정렬을 할 수 있다. 그리고 정렬된 영상으로부터 얼굴 텍스처 영상을 생성하기 위하여 영상 모자이크 기법을 사용한다. 이때 중첩된 영역에 따라 선형적인 가중치를 부여하여 영상을 스티칭하고 부분적으로 나타나는 고스트 효과를 제거함으로써 더 실감 있는 텍스처를 생성한다.

키워드 : 타원체형 모델, 가변적 블록 크기, 상관관계, 얼굴 텍스처

Abstract In this paper, we propose an image registration algorithm based on variable-sized blocks of ellipsoidal prototype model which is similar in shape to human face. While matching blocks, the existing cylindrical prototype model which only takes into account left and right curvature can accomplish a correct alignment on left and right images. But, registration errors are produced from up and down images because the cylindrical prototype model not reflects characteristics of head shape and jaw structure of human. The proposed method is a block matching algorithm which uses variable-sized blocks with considering left-right and up-down curvature of ellipsoidal face model and can correctly align images by using the correlation between them. We then adapt image mosaic technique to generate a face texture from aligned images. For this purpose, we stitch them with assigning linear weights according to the overlapped region and remove ghost effects to make more realistic facial texture.

Key words : ellipsoidal prototype model, variable-sized block, correlation, face texture

1. 서 론

디지털 영상의 역사를 살펴보면 알 수 있듯이 컴퓨터 그래픽스 분야에서 3차원 모델링 기술은 컴퓨터 비전뿐만 아니라 가상현실, 원격화상회의 등 다양한 분야에서 활용되는 핵심기반 기술이다. 특히, 3차원 얼굴 모델링은 가상공간에서의 아바타 생성, 성형외과에서 수술

전 모의실험, 컴퓨터 게임, HCI, 3차원 얼굴 인식 등의 분야에서 널리 활용되고 있다. 그런데, 3차원 모델링 기술은 미래의 핵심산업임에도 불구하고 아직까지 시장형성 면에서는 초기단계에 불과하며 많은 연구를 필요로 하는 분야이다.

이중에서, 3차원 얼굴 모델링과 관련된 연구는 실감 있는 얼굴 텍스처와 모델 생성, 실시간 처리, 자동화 시스템 생성, 일반 모델에 적용적인 알고리즘 구현 등에 초점을 맞추고 있다. 하지만 현재까지 완벽하게 실감 있는 얼굴 모델을 만들어내지는 못하고 있다. 3차원 모델을 생성하는 방법 중 레이저스캐너나 모션캡처 방식 등 장비를 이용하는 방법은 상세한 기하 형태 및 실감 있는 텍스처 정보를 얻을 수 있다는 장점이 있지만, 반대

· 본 논문은 2003년도 교내 학술연구비를 지원 받아 수행하였음

[†] 비 회 원 : 송실대학교 컴퓨터학과
lijhop@vision.ssu.ac.kr
eduardo@vision.ssu.ac.kr
^{**} 종신회원 : 송실대학교 미디어학부 교수
hic@computing.ssu.ac.kr

논문접수 : 2004년 3월 23일
심사완료 : 2004년 11월 2일

한 데이터 양과 접합지점의 영역 손실, 많은 경비를 필요로 하는 단점이 있다[1].

2차원 얼굴 영상을 이용하는 방법으로는 정면과 측면 영상을 이용하는 방법, 2차원 영상을 이용하여 3차원 모델의 기하학적인 구조를 복원하고 실린더형 텍스처 맵을 추출하는 방법, 4방향 또는 8방향 얼굴영상을 이용하는 방법 등이 있다. 정면과 측면 영상을 이용하는 방법은 2차원 영상에서 특징점을 추출한 후 3차원 모델 윤곽과의 정합을 통해 일반모델을 변형하고 정면과 측면 영상을 텍스처 매핑하여 모델을 생성한다. 그러나 이 방법은 정면과 측면을 제외한 나머지 방향에서 부자연스러운 느낌을 받는다[2,3]. 2차원 영상을 이용하여 3차원 모델의 기하학적인 구조를 복원하고 실린더형 텍스처 맵을 추출하는 방법은 서로 다른 방향을 가진 5대의 카메라와 영상 특징점의 좌표를 통해 카메라 파라미터를 추출한다. 추출된 파라미터를 사용하여 2차원 영상의 특징점에 대응하는 3차원 좌표를 얻어냄으로써 3차원 모델의 기하학적인 구조를 복원한다. 그리고 복원된 3차원 모델과 2차원 영상을 이용하여 실린더형 텍스처 맵을 추출한다. 이 방법은 정확하게 변형된 3차원 모델의 기하학적인 정보를 획득하여야만 한다는 단점이 있다 [4-6].

4방향 또는 8방향 얼굴영상을 이용하는 방법은 여러 방향에서 촬영한 2차원 영상에 대해 3차원 얼굴모델을 2차원으로 프로젝션 시킨 후 2차원 상에서 모델을 변형하고 다시 3차원으로 복원을 함으로써 두상에 근사한 모델을 생성한다[7,8]. 그런데 이 방법에서 생성한 텍스처는 파노라믹 기법 중 영상 스티칭 기법만을 사용하여 변형된 3차원 모델에 매핑을 하므로 자연스러운 텍스처를 생성하기 위해서는 정확한 각도에서 촬영해야 하는 단점이 있다. 즉 부정확한 각도에서 촬영한 영상을 사용하였을 경우 부자연스러운 텍스처가 생성되고 이것을 모델에 매핑했을 경우 사실적인 3차원 모델을 생성할 수 없다. 이를 해결하기 위한 방법으로 얼굴의 곡률을 고려한 실린더형 모델의 가변적 블록크기를 이용한 영상정렬 알고리즘이 있다[9]. 이 방법은 영상정렬 시 얼굴의 좌우 곡률을 고려하므로 좌우 영상에 대해서는 정확한 정렬이 가능하다. 그러나 실린더형 모델의 상하 부분은 사람의 두상모양이나 턱 구조와는 달리 평평하므로 상하 영상을 정렬할 때 정렬오류가 발생한다. 본 논문에서는 이러한 문제점을 해결하기 위해 얼굴 형태와 가장 유사한 타원체형 모델을 사용함으로써 기존 방법에서 나타났던 텍스처 생성 오류를 해결한다.

본 논문에서 제안하는 실감 있는 3차원 얼굴모델 생성 시스템은 크게 입력 영상 정렬과 영상 모자의 두 단계로 구성된다.

입력 영상 정렬 단계에서는 타원체형 모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 사용해서 입력 영상을 정렬한다. 이것은 두상을 타원체형으로 가정하고 각각 다른 각도에서 촬영한 2차원 얼굴영상을 정렬하는 방법으로 블록정합 알고리즘(Block matching algorithm)을 사용한다. 블록정합을 할 때 얼굴모델의 상하좌우 곡률에 따라 가변적인 블록 크기로 분할하고 정합을 한다. 이때 얼굴영상의 각 블록에 대한 칼라와 텍스처 특징을 추출한 후 정합 유사도가 가장 높은 후보블록을 정합될 블록으로 선택하고 이로부터 변위 벡터를 계산한다. 영상 모자의 단계에서는 얼굴 텍스처 영상을 생성하기 위하여 영상 스티칭 기법을 사용하며 크로스디졸브 기법을 적용한다. 영상을 스티칭한 후 고스트 효과를 제거함으로써 실감 있는 얼굴 텍스처를 생성한다.

본 논문의 구성은 다음과 같다. 2장에서는 타원체형 모델을 사용하기 위하여 타원 파라미터를 추출하는 과정을 보여주고, 3장은 기존의 영상정렬 알고리즘의 문제점과 해결방법에 대하여 비교하여 설명한다. 4장은 정렬된 영상에 대해 영상 모자의 기법을 사용하여 얼굴 텍스처를 생성하는 방법에 대해 설명한다. 5장은 실험결과를 보이고 6장은 결론을 기술한다.

2. 타원 파라미터 추출

타원체형 모델을 사용하여 영상을 정렬하기 위해서는 타원 파라미터인 단축 r_x 와 장축 r_y 를 결정하여야 한다. 그림 1은 영상 프로젝션을 이용한 타원 파라미터 추출 방법을 보여준다. 그림 1에서 단축인 r_x 는 입력된 두상영상에 일계값을 사용하여 일정 값 이상의 경우에

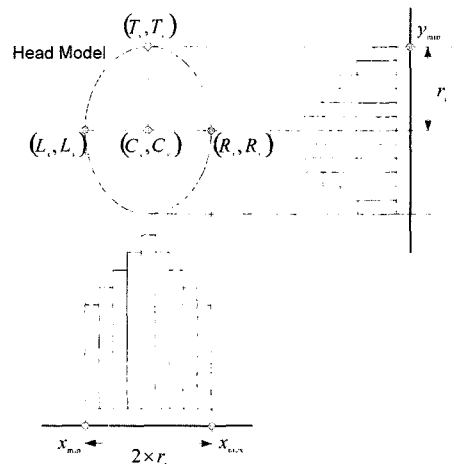


그림 1 영상 프로젝션을 이용한 타원 파라미터 추출

대응하는 출력 영상의 화소값을 1, 그 외의 경우는 0으로 하여 이진화한 후 수평으로 프로젝션 시켜서 결정하고 장축 r_y 는 목 영역과 턱 선의 경계가 불분명함을 고려하여 수직 프로젝션과 타원의 중심좌표 값을 이용하여 결정한다.

식 (1)에서 $I(x, y)$ 는 이진화된 두상영상이고 $P_h(x)$ 는 이진영상의 x 위치에서 y 축에 대한 검은색 픽셀의 총 개수이며 $P_h(y)$ 는 이진영상의 y 위치에서 x 축에 대한 검은색 픽셀의 총 개수를 의미한다. x_{min} 은 $P_h(x)$ 에서 0이 아닌 x 값 중 처음 나타나는 최소값으로 두상의 좌측경계가 되고 x_{max} 는 최대값으로 두상의 우측경계가 된다. y_{min} 은 상단경계가 되지만 y_{max} 는 목 영역을 포함하기 때문에 파라미터로 사용하지 않는다. 추출된 x_{min} 과 x_{max} 의 x 좌표 값과 y 좌표 값을 통해 타원의 중심 좌표 (C_x, C_y)를 획득할 수 있다. 획득된 타원 중심의 y 좌표인 C_y 와 y_{min} 의 차로 r_y 를 결정한다.

$$r_x = \frac{|x_{min} - x_{max}|}{2}, r_y = |y_{min} - C_y| \quad (1)$$

$$p_h(x) = \sum_y I(x, y), p_h(y) = \sum_x I(x, y)$$

$$\text{where } \begin{cases} x_{min} = \min_x \{p_h(x) > 0\}, x_{max} = \max_x \{p_h(x) > 0\} \\ y_{min} = \min_y \{p_h(y) > 0\}, y_{max} = \max_y \{p_h(y) > 0\} \\ C_x = \frac{x_{min} + x_{max}}{2}, C_y = \frac{L_y + R_y}{2} \end{cases}$$

3. 타원체형 모델의 영상정렬 알고리즘

파노라마 기법을 사용하여 자연스럽게 실감 있는 얼굴 텍스처 영상을 생성하기 위해서는 입력영상을 정확하게 정렬하여야 한다. 부정확한 각도에서 얻어진 얼굴 영상들을 정렬할 때 발생하는 텍스처 생성 결과 오류를 해결하기 위해 본 논문에서는 3차원 얼굴모델을 얼굴모양과 가장 유사한 타원체형 모델로 정의한다. 그리고 3차원 얼굴모델로부터 획득한 2차원 얼굴영상을 정렬하기 위하여 가변적 블록크기를 기반으로 하는 블록정합을 수행한다. 이때, 가변적 블록크기를 사용하는 이유는 얼굴모델의 기하학적인 특징인 곡률을 고려하기 때문이다. 그런데 실린더형 모델을 사용한 기존의 방법은 얼굴의 좌우 곡률만을 고려한 반면 본 논문에서 제안한 타원체형 모델은 좌우 곡률뿐만 아니라 상하 곡률까지도 고려하여 정합을 수행함으로써 더 정확한 정렬을 할 수 있다.

3.1 가변적 블록 크기 계산

블록정합 알고리즘은 입력된 영상들을 동일한 크기로 분할한 후 임의의 기준 블록에 대한 후보 블록 중에서 정합유사도가 가장 높은 블록을 찾아내는 정합 방법이

다. 3차원 얼굴모델을 촬영한 2차원 영상에 대해 동일한 크기의 블록을 사용한 일반적인 블록정합 알고리즘을 사용할 경우 모델의 곡률 때문에 영역간의 오차가 발생하게 된다. 또한 이를 개선한 실린더형 원형모델의 경우는 상하 곡률을 고려하지 않는 단점이 있다. 본 논문에서는 이러한 문제점을 해결하기 위해서 모델의 상하좌우 곡률에 따라 가변적인 블록크기를 적용한다.

타원체형 3차원 모델에 대한 2차원 영상의 블록 크기를 식 (3), (4)와 같이 정의한다. 식 (3)에서 BW 는 얼굴모델의 좌우 곡률을 고려한 가변적 크기의 블록너비이고 식 (4)에서 BH 는 얼굴모델의 상하 곡률을 고려한 가변적 크기의 블록높이가 된다. 임의의 분할각도 θ_{Div} 만큼 떨어진 타원상의 점 $p(x, y)$ 를 얻기 위하여 슈퍼타원 모델(Superellipse model)의 경계선 검출 함수를 사용한다[10].

슈퍼타원 모델이란, 원점에 중심을 두고, 매개변수에 의해 형태가 결정되는 타원을 말하고 그림 2에서 r_x 는 단축을, r_y 는 장축을 의미하며, ϕ 는 각도로서 0부터 2π 까지 값을 가질 수 있다.

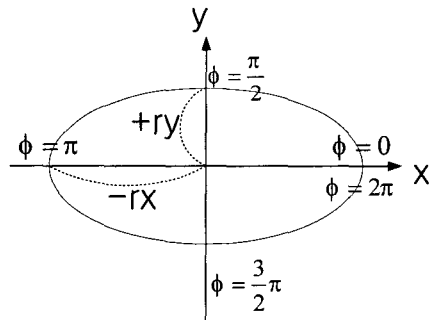


그림 2 슈퍼타원모델(Superellipse model)

식 (2)는 슈퍼타원을 표현하는 수식으로서, 입력 좌표인 $x(\phi), y(\phi)$ 의 각도를 변화시키면서 그 궤적을 표시하면 그림 2와 같이 나타난다. 해당 좌표인 $x(\phi)$ 와 $y(\phi)$ 는 슈퍼타원모델의 중심으로부터의 거리를 sin함수 및 cos함수를 이용하여 그 위치를 나타내며 각 좌표를 나타내는 $x(\phi)$ 와 $y(\phi)$ 의 n 값에 따라서 슈퍼타원모델의 형태가 결정되는데 타원의 경우는 $n=1$ 인 경우가 된다.

$$S_{n, r_x, r_y} = (x(\phi), y(\phi)) \quad (2)$$

$$x(\phi) = r_x \cos^n(\phi) \quad y(\phi) = r_y \sin^n(\phi)$$

$0 < n < 1$: 사각형, $n=1$: 타원, $n > 1$: 마름모

이렇게 얻어진 타원상의 점 $p(x, y)$ 를 2차원 영상에 프로젝션 시켜 블록의 크기를 구할 수 있다. 식 (3), (4)에서 θ_{Div} 는 3차원 모델을 동일한 각도로 나누는 분할

각도이고 n 은 3차원 모델을 임의의 한 방향에서 볼 수 있는 0° 에서 180° 까지의 시야를 분할각도 θ_{Div} 로 분할할 때 2차원 영상에 생성되는 블록 열과 행의 인덱스이다. r_x 는 타원체의 단축의 길이이고 r_y 는 장축의 길이이다. h_n 과 v_n 은 2차원 영상에 프로젝션된 각각의 블록 크기이며 x_n 과 y_n 은 2차원 영상에 프로젝션된 전체의 블록 크기를 의미한다.

$$\begin{cases} BW = h_n \\ h_1 = x_1 \\ h_n = x_n - x_{n-1} \quad (n \geq 2) \\ x_n = r_x \cos \left[\frac{3\pi}{2} \times (n \times \theta_{Div}) \right] \end{cases} \quad (3)$$

$$\begin{cases} BH = v_n \\ v_1 = y_1 \\ v_n = y_n - y_{n-1} \quad (n \geq 2) \\ y_n = r_y \sin (n \times \theta_{Div}) \end{cases} \quad (4)$$

where $0 < \theta_{Div} < \frac{\pi}{4}$, $1 \leq |n| \leq \frac{\pi}{2\theta_{Div}}$

그림 3은 타원체형 모델을 동일한 각도 θ_{Div} 로 분할했을 때 2차원 영상에서 모델의 곡률에 따라 영상 중심을 기준으로 영상의 상하 또는 좌우측으로 갈수록 블록의 크기가 작아지는 가변적인 크기의 블록이 생성됨을 보여준다. 그림 3에서 블록 열과 행의 인덱스 n 이 음수 값을 가지면 2차원 영상에서 중심을 기준으로 좌측이나 하단영역의 블록이 되고 양수 값이면 우측이나 상단영역의 블록이 된다.

그림 4는 기존의 블록정합 방법 알고리즘과 본 논문

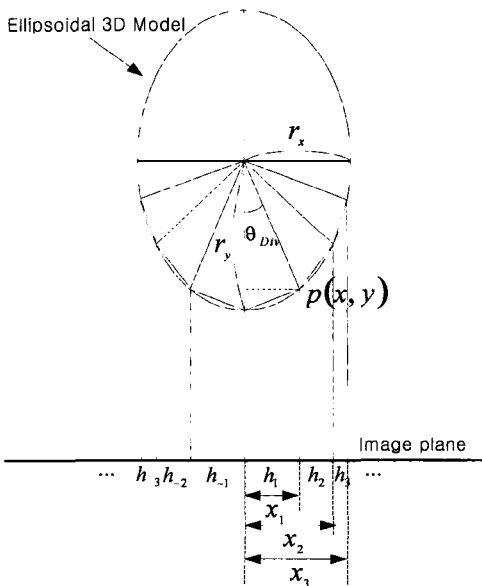
에서 제안한 방법으로 영상을 분할한 결과를 비교해서 보여준다. 그림 4에서 (a), (d)는 동일한 블록크기로 영상을 분할한 결과를 보여주며 (b), (e)는 실린더형 모델의 가변적 블록크기로 분할한 결과를 보여준다. (c), (f)는 본 논문에서 제안한 방법인 타원체형 모델의 기하학적 성질, 즉 곡률을 이용하여 가변적인 블록크기로 영상을 분할한 결과를 보여준다.

3.2 정합 유사 함수

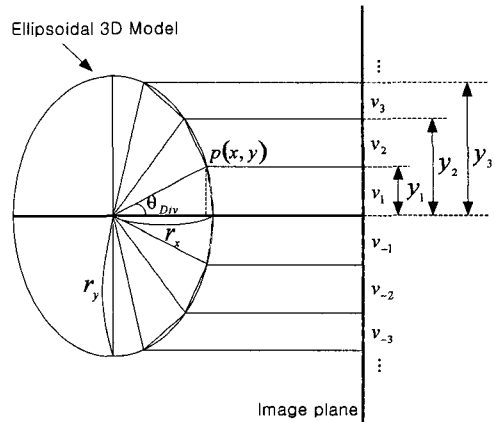
본 논문에서는 블록정합을 수행하기 위해 영상간의 상관관계를 잘 나타낼 수 있는 칼라특징과 텍스처 특징을 사용한다. 이들 특징은 블록 단위로 계산되며 각 특징의 기여도를 나타내는 가중치와 함께 정합 유사 함수를 정의하게 된다. 일반적으로 칼라 영상을 표현하기 위해 RGB 칼라 공간을 많이 사용하지만 영상 처리를 수행하려 할 때 세 가지의 R, G, B 칼라 값을 모두 처리해야 하므로 영상처리에 비효율적이다[11]. 본 논문에서는 조명의 변화에 강건한 YIQ 칼라 모델을 사용한다. 식 (5)는 RGB 칼라 모델을 YIQ 칼라 모델로 변환하는 식을 나타낸다.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.27 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5)$$

그리고 블록정합 시 색상분포가 균일한 영역에서 발생할 수 있는 정합 오류 발생 문제점을 보완하고 영상의 특징을 잘 반영하기 위해 텍스처 정보도 함께 사용한다. 식 (6)에서 $M_{color}(B_i)$ 는 i 번째 블록에 대한 칼라

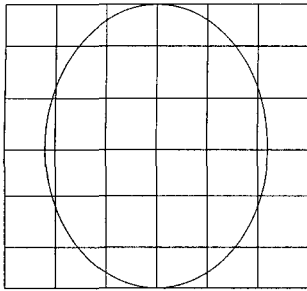


(a) 모델 윗면(블록너비)

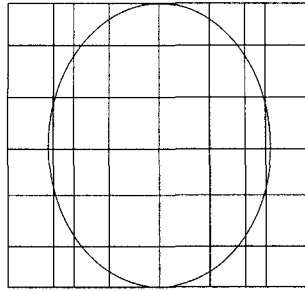


(b) 모델 정면(블록높이)

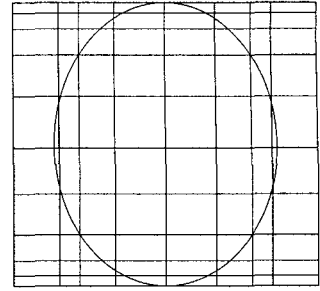
그림 3 곡률에 따른 가변적 블록 크기



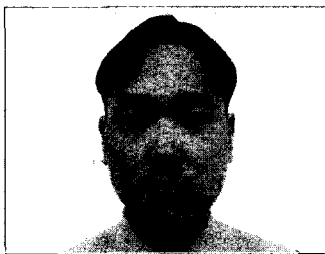
(a) 동일한 크기의 균일 분할



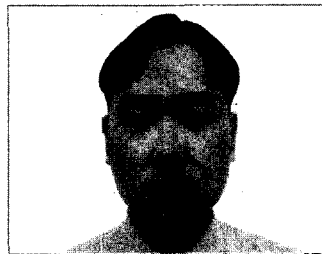
(b) 실린더형 모델의 가변적 크기 분할



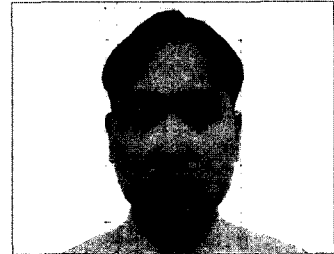
(c) 타원체형 모델의 가변적 크기 분할



(d) 동일한 크기의 균일한 방법에서의 분할



(e) 실린더형 모델을 이용한 방법에서의 분할



(f) 타원체형 모델을 이용한 방법에서의 분할

그림 4 블록분할 방법에 따른 블록 크기 비교

특징의 유사도를 나타내는 값으로 0과 1사이의 값을 가지며 이 값이 작을수록 비교하는 블록사이의 유사도가 커진다. $C_{B_i}^{\theta_1}$ 는 θ 각도 영상의 i 번째 블록에 대한 칼라 특징을 나타내고 M 은 블록의 너비, N 은 블록의 높이를 나타낸다.

$$M_{color}(B_i) = \left| \frac{C_{B_i}^{\theta_1} - C_{B_i}^{\theta_2}}{C_{max}} \right| \quad (6)$$

$$C_{B_i}^{\theta_1} = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x+i, y+j)$$

where $0 \leq M_{color}(B_i) \leq 1$

텍스처 특징은 가보 웨이블릿 변환(Gabor wavelet transform)을 수행하여 얻은 가보 웨이블릿 인수(Gabor wavelet coefficients)를 사용한다. 가보 웨이블릿 인수는 가보 웨이블릿 커널(Gabor wavelet kernel)을 영상에 컨볼루션(convolution)하여 구하는데 이는 영상의 텍스처 정보를 잘 반영하는 특징으로 뛰어난 텍스처 구분력을 가진다[12]. 식 (7)에서 $M_{texture}(B_i)$ 는 i 번째 블록의 텍스처 특징에 대한 유사도를 나타내며 $\psi_{m\theta}$ 는 커널의 크기 및 방향을 나타내는 색인 벡터(index vector)이다. 이때 m 은 크기 색인으로 사인 곡선의 빈도수(frequency)를 결정하고 θ 는 방향 색인으로 사인 곡선의 방향을 결정한다. 크기 색인은 m 이 클수록 낮은 빈도수를 결정한다.

$$M_{texture}(B_i) = \left| \frac{T_{B_i}^{\theta_1} - T_{B_i}^{\theta_2}}{T_{max}} \right| \quad (7)$$

$$T_{B_i}^{\theta_1} = \sum_{x=i}^{M-1} \sum_{y=j}^{N-1} I(x+i, y+j) \cdot \psi_{m\theta}(x, y)$$

where $0 \leq M_{texture}(B_i) \leq 1$

$$\psi_{m\theta}(x, y) = \text{Real part} + i \cdot \text{Imaginary part}$$

$$= \frac{(k_m \cos k_\theta)^2 + (k_m \sin k_\theta)^2}{\sigma^2} \cdot \exp \left\{ \frac{[(k_m \cos k_\theta)^2 + (k_m \sin k_\theta)^2] \cdot [x^2 + y^2]}{2\sigma^2} \right\} \cdot [\cos(xk_m \cos k_\theta + yk_m \sin k_\theta) + i \cdot \sin(xk_m \cos k_\theta + yk_m \sin k_\theta) - \exp \left\{ -\frac{\sigma^2}{2} \right\}]$$

블록 사이의 정합 척도로 사용되는 정합 유사 함수는 식 (8)과 같이 정의한다. 식 (8)에서 $SF(i, j, u, v)$ 는 칼라 특징과 텍스처 특징을 동시에 고려한 정합 유사도를 의미하며 0과 1사이의 값을 가진다. α 와 β 는 각 특징이 정합 유사도 계산에 기여하는 정도를 의미하며 실험적으로 α 는 0.4, β 는 0.6을 부여했다.

$$d(i, j) = (u^*, v^*) \quad (8)$$

where $SF(i, j, u, v)$ is maximized

$$SF(i, j, u, v) = 1 - (\alpha \cdot M_{color}(i, j, u, v) + \beta \cdot M_{texture}(i, j, u, v))$$

where $0 \leq \alpha, \beta \leq 1, \alpha + \beta = 1, 0 \leq SF(i, j, u, v) \leq 1$

식 (8)에서 (i, j) 는 기준 블록의 위치를 나타내며 (u, v) 는 기준 블록과 정합된 블록 사이의 변위를 나타낸다. 기준 블록과 정합된 블록간의 변위는 기준 블록에 대한 후보 블록들의 정합 유사도를 계산한 후 이중 최대 정합 유사도를 가지는 블록을 정합된 블록으로 결정하고, 기준 블록에 대한 이 블록의 상대적 위치를 변위 벡터 (u^*, v^*) 로 정의한다. 여기서 u^* 는 x 축으로의 변위를 의미하고, v^* 는 y 축으로의 변위를 의미한다.

그림 5는 첫 번째 기준블록의 너비 h_1 에 대한 후보블록의 너비를 생성하는 과정을 보여주고 있다. 그림 5에서와 같이 타원체형 모델 상에서 분할각도 θ_{Div} 만큼씩 일정하게 떨어진 점을 2차원 영상에 프로젝션 시켜서 후보블록의 너비를 계산한다. 그리고 같은 방법으로 후보블록의 높이도 계산할 수 있다.

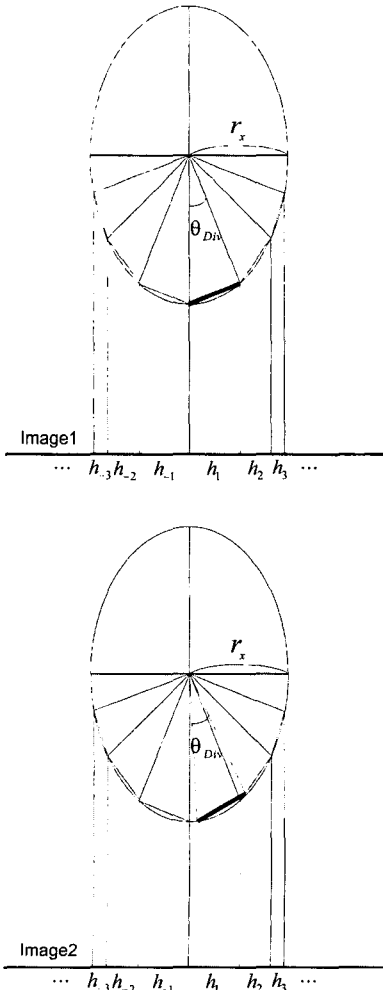


그림 5 기준 블록에 대한 후보블록 생성과정

여기서 기준블록과 후보블록의 정합 유사도를 구할 때 기준블록에 대해서 생성되는 후보블록의 개수는 식 (9)와 같으며 이때 생성되는 후보블록의 크기는 식 (2), (3)으로부터 구할 수 있다.

$$NCB(B_i) = NCBW(h_n) \times NCBH(v_n) \quad (9)$$

$$where \begin{cases} NCBW(h_n) = r_x \left[1 - n \times \left(\frac{2\theta_{Div}}{\pi} \right) \right] \\ NCBH(v_n) = r_y \left[1 - n \times \left(\frac{2\theta_{Div}}{\pi} \right) \right] \end{cases}$$

식 (9)에서 $NCB(B_i)$ 는 너비가 h_n , 높이가 v_n 의 크기를 갖는 i 번째 기준블록에 대해서 생성되는 후보블록의 개수를 의미하고 $NCBW(h_n)$ 와 $NCBH(v_n)$ 은 각각 생성되는 후보블록의 너비의 개수와 높이의 개수를 의미한다. 식 (9)에서 알 수 있듯이 중심에서 멀어지는 기준블록일수록 생성되는 후보블록의 개수가 줄어들며 이웃하는 기준 블록 간에는 단위 분할각도 θ_{Div} 에 대한 블록개수 만큼의 후보블록개수 차이가 있음을 알 수 있다.

4. 영상 모자이크

본 장에서는 타원체형 모델 기반의 영상정렬 알고리즘을 이용해 입력된 영상들을 정렬한 후 하나의 텍스처 영상을 생성하는 영상 모자이크 기법에 대해 설명한다. 본 논문에서는 얼굴 텍스처 영상을 생성하기 위하여 영상 스티칭 기법을 사용하며 영상들이 서로 잘 융합하도록 크로스디졸브(Cross-dissolve) 기법을 적용한다[13,14].

영상이 정렬된 상태에서 중첩된 영역에 위치하는 픽셀의 칼라 값을 적절하게 부여된 가중치에 따라 혼합함으로써 최종 텍스처를 생성할 수 있다. 본 논문에서는 보다 자연스러운 텍스처를 생성하기 위하여 중첩된 영역과 입력 영상간의 거리를 고려해 가중치를 부여하는 데, 이는 각각의 영상에 대해서 반비례 관계를 가지는 선형적인 가중치를 부여하는 방법이다.

4.1 영상 스티칭

영상 모자이크 기법을 사용하여 생성되는 얼굴 텍스처 영상을 식 (10)과 같이 정의한다.

$$if (Visible P \text{ unique in } I_i) \\ T = I_i(p) \\ else \\ T = \sum_{i=1}^n w_i(p) \cdot I_i(p) \quad (10)$$

식 (10)에서 T 는 텍스처 영상을 의미하고, I_i 은 입력 영상을 의미한다. P 는 중첩된 영역에 존재하는 임의의 픽셀을 의미하며, $w_i(p)$ 은 영상을 스티칭할 때 P 에 대한 i 번째 입력영상에 부여하는 가중치를 의미한다. 식 (10)에서 알 수 있듯이 영상이 중첩되지 않은 영역에 대해서는 픽셀 P 가 영상 I_i 중 어떤 영상에 보이는 픽

셀인지를 확인하여 텍스처 영상에 사용할 픽셀 값을 결정한다. 그리고 그림 6과 같이 중첩된 영상에 대해서는 식 (11)에서 정의한 가중치 부여함수에 따라 영상 I_i 에 대해 각각 가중치 w_i 를 부여하여 영상을 스티칭한다.

4.2 가중치 결정

영상을 스티칭할 때 정렬된 영상들은 중첩된 영역을 고려하여 가중치를 부여한다. 그림 6은 중첩된 영역의 임의의 픽셀 P 가 각 입력영상의 중점 C_i 와의 거리에 반비례하는 가중치를 사용하여 값이 결정됨을 보여준다.

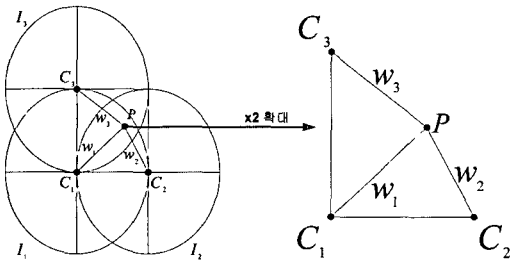


그림 6 영상의 중첩 영역에 부여되는 가중치

본 논문에서는 영상을 스티칭할 때 부여하는 가중치를 식 (11)과 같이 정의하며 이것은 거리에 반비례하는 가중치를 갖는 데이터간의 보간에 많이 사용되는 IDW (Inverse Distance Weighting)방법에 근거한 것이다 [15]. 식 (11)에서 r 은 거리에 따른 가중치 결정에 영향을 주는 인수로서 r 값이 커질수록 거리가 가까운 점들에 대해서는 좀 더 큰 값을, 거리가 먼 점들에 대해서는 상대적으로 더 작은 값을 가지도록 해준다. 본 논문에서는 r 값으로 2를 사용한다.

$$w_i(d) = \frac{d_i^{-r}}{\sum_{k=1}^n d_k^{-r}}, \quad d_i = \overline{C_i P} = \sqrt{(x_{C_i} - x)^2 + (y_{C_i} - y)^2} \quad (11)$$

where $0 \leq w_i \leq 1, \sum_{k=1}^n w_k = 1$

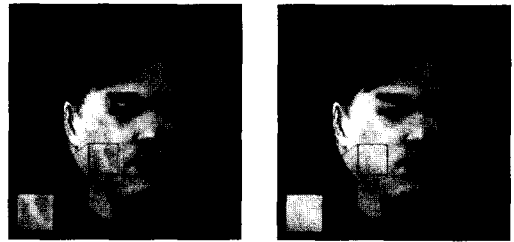
4.3 고스트 효과 제거

본 논문에서 제안한 스티칭 방법은 실감 있는 텍스처를 생성하지만 텍스처를 확대하여 보면 중첩영역의 경계 면에서 고스트 효과 즉 주름무늬가 발생함을 볼 수 있다. 이러한 고스트 효과를 제거하기 위해 3차 회선 보간법을 사용하여 영상의 경계지역을 매끄럽고 실감 있게 만들어준다[16,17]. 2차원 공간에서의 3차 회선 보간법은 식 (12)와 같이 정의된다. 식 (12)에서 $C_{(i+k, j+b)}$ 은 샘플 데이터에 의해 결정된 계수를 나타내며 h_x 와 h_y 는 샘플링 간격을 나타낸다.

$$I(x, y) = \sum_{k=-n}^n \sum_{l=-m}^m C_{(i+k, j+b)} f\left(\frac{x-x_i}{h_x}, \frac{y-y_j}{h_y}\right) \quad (12)$$

where $i = \text{int}\left(\frac{x}{h_x} + 0.5\right), j = \text{int}\left(\frac{y}{h_y} + 0.5\right)$

그림 7은 생성된 얼굴 텍스처에 나타나는 고스트 효과를 제거하지 않고 3차원 얼굴 모델에 매핑한 결과영상과 고스트 효과를 제거한 후 3차원 얼굴 모델에 매핑한 결과영상을 비교하여 보여준다. 그림 7에서 볼 수 있듯이 고스트 효과를 제거한 후 매핑했을 때 더욱 실감 있는 모델이 생성됨을 볼 수 있다.



(a) 고스트 효과를 제거하지 않은 영상을 매핑한 얼굴 모델

(b) 고스트 효과를 제거한 영상을 매핑한 얼굴 모델

그림 7 중첩 영역의 경계면 비교

5. 실험 및 결과

본 장에서는 다원체형 모델의 가변적인 블록크기를 기반으로 하는 영상정렬 알고리즘을 이용한 얼굴 텍스처 생성에 관한 실험결과를 기술한다. 실험에 사용할 영상은 Nikon사의 Coolpix 2500 디지털카메라로 촬영하였다. 실험영상의 크기는 420X315로 정규화 하였고 전처리를 통하여 배경영역을 제거하였다. 실험은 마이크로소프트의 비주얼 C++ 컴파일러를 사용하고 윈도우즈 2000 운영체제 환경에서 실험하였다.

그림 8은 본 논문에서 사용한 3차원 일반 얼굴모델로서 1504개의 점(Vertex)과 846개의 면(mesh)으로 구성되어 있다.



(a) 정면

(b) 측면

그림 8 3차원 일반 얼굴모델

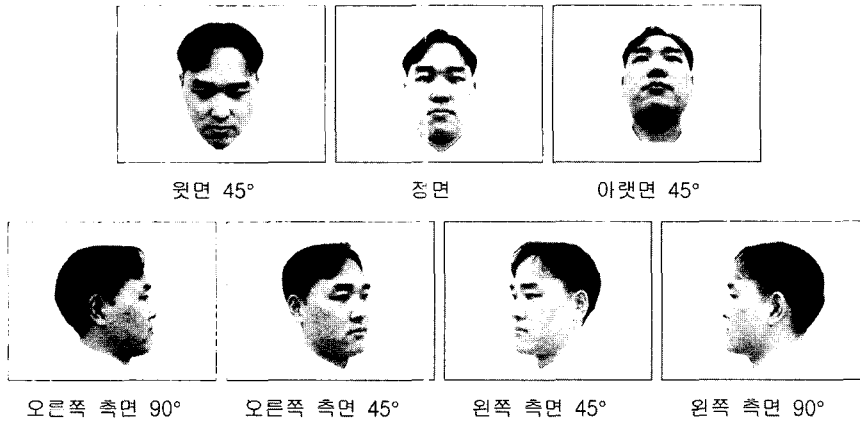


그림 9 실험영상

그림 9는 사람의 얼굴을 45° 간격으로 상하 2방향과 좌우 5방향으로 촬영하여 획득한 영상을 보여준다.

그림 10은 생성된 얼굴 텍스처 결과를 보여준다. 그림 10에서 (a)는 기존의 영상정렬 방법을 사용하여 얼굴 텍스처를 생성한 결과이고 (b)는 본 논문에서 제안한 타원체형 모델의 영상정렬 방법을 사용하여 생성한 얼굴 텍스처의 결과이다. 그림 10의 (a)에서 볼 수 있듯이 기존 방법으로 생성한 텍스처는 중첩영역의 경계가 희



(a) 동일한 크기의 블록을 이용한 방법으로 생성한 텍스처



(b) 타원체형 모델을 이용한 방법으로 생성한 텍스처
그림 10 텍스처 생성결과

미해지고 고스트 효과가 강하게 나타나는 등의 문제점으로 인하여 부자연스러운 텍스처가 생성된다.

표 1은 얼굴 텍스처를 생성하기 위해 영상을 각각 분할각도 10°, 15°, 30°로 분할했을 때 생성되는 블록의 개수와 크기를 보여주고 있다. 이때 영상의 중심을 기준으로 좌우 중 우측을, 상하 중 상단만을 고려한 결과이다. 또한 표 1에서는 각각의 기준블록에 대해서 정합 시 고려해야 하는 후보블록에 대한 너비와 높이의 개수를 보여주고 있다.

표 2는 분할각도에 따라 블록 정합의 성능을 비교한 결과로서 각 분할각도에 따라 생성되는 기준블록에 대한 평균제곱오차인 MSE(Mean Squared Error)와 정합 시에 발생하는 비교횟수에 따른 복잡도(Complexity)를 보여주고 있다. 표 2에서 알 수 있듯이 분할각도가 커질수록 MSE값은 감소하나 상대적으로 복잡도가 증가함을 알 수 있다. 여기서 분할각도가 작아질수록 MSE값이 증가하면서 정합 정확도가 떨어지는 이유는 지역적으로 균일한 색상분포와 텍스처값을 갖는 얼굴영상의 특성상 정합 특징으로 사용하는 칼라와 텍스처에 대한 구분력이 감소하기 때문이다. 본 논문에서는 정확도와 복잡도를 고려해 분할각도를 15°로 사용하였다.

그림 11은 기존의 영상정렬 방법들과 본 논문에서 제안한 타원체형 모델의 영상정렬 방법을 사용하여 획득한 결과영상의 정렬오류를 계산한 후 그래프로 비교하여 나타낸 것이다. 원 영상 텍스처의 픽셀 값과 정렬된 영상 텍스처의 픽셀 값을 비교하여 영상정렬 오류율을 얻었다.

그림 11의 그래프에서 알 수 있듯이 상하 부분의 곡률을 고려하지 않는 기존의 영상정렬 방법들은 상하 영상을 정렬하는 6번째 입력영상부터 급격하게 영상정렬의 오류율이 증가됨을 볼 수 있다. 반면 본 논문에서 제

표 1 분할각도에 따른 블록개수와 크기 비교

γ_x	62	index	x_n	h_n	y_n	v_n	$NCBW(h_n)$	$NCBH(v_n)$
γ_y	112	1	31	31	56	56	41	75
$\theta_{Div}(\text{dgree})$	30	2	53.69	22.69	96.99	40.99	21	37
$\theta_{Div}(\text{radian})$	0.523599	3	62	8.31	112	15.01	0	0

(a) $r_x=62, r_y=112, \theta_{Div}=30^\circ$

γ_x	62	index	x_n	h_n	y_n	v_n	$NCBW(h_n)$	$NCBH(v_n)$
γ_y	112	1	16.05	16.05	28.99	28.99	52	93
$\theta_{Div}(\text{dgree})$	15	2	31	14.95	56	27.01	41	75
$\theta_{Div}(\text{radian})$	0.261799	3	43.84	12.84	79.20	23.20	31	56
		4	53.69	9.85	96.99	17.80	21	37
		5	59.89	6.19	108.18	11.19	10	19
		6	62	2.11	112	3.816	0	0

(b) $r_x=62, r_y=112, \theta_{Div}=15^\circ$

γ_x	62	index	x_n	h_n	y_n	v_n	$NCBW(h_n)$	$NCBH(v_n)$
γ_y	112	1	10.77	10.77	19.45	19.45	55	100
$\theta_{Div}(\text{dgree})$	10	2	21.21	10.44	38.31	18.86	48	87
$\theta_{Div}(\text{radian})$	0.261799	3	31	9.79	56	17.69	41	75
		4	39.85	8.85	71.99	15.99	34	62
		5	47.49	7.64	85.80	13.80	28	50
		6	53.69	6.20	96.99	11.20	21	37
		7	58.26	4.57	105.25	8.25	14	25
		8	61.06	2.80	110.30	5.05	7	12
		9	62	0.94	112	1.70	0	0

(c) $r_x=62, r_y=112, \theta_{Div}=10^\circ$

표 2 분할각도에 따른 성능비교

θ_{Div}	MSE	Complexity
10°	9.02	17470
15°	6.85	10614
30°	5.26	3852

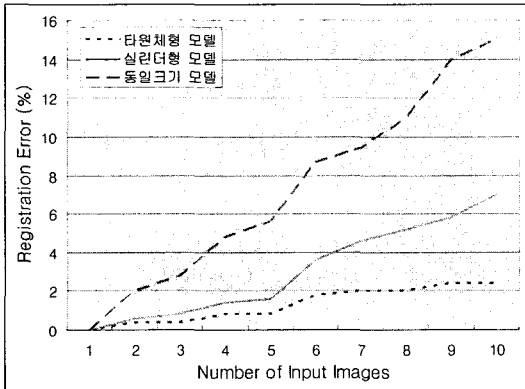


그림 11 영상정렬 오류 비교

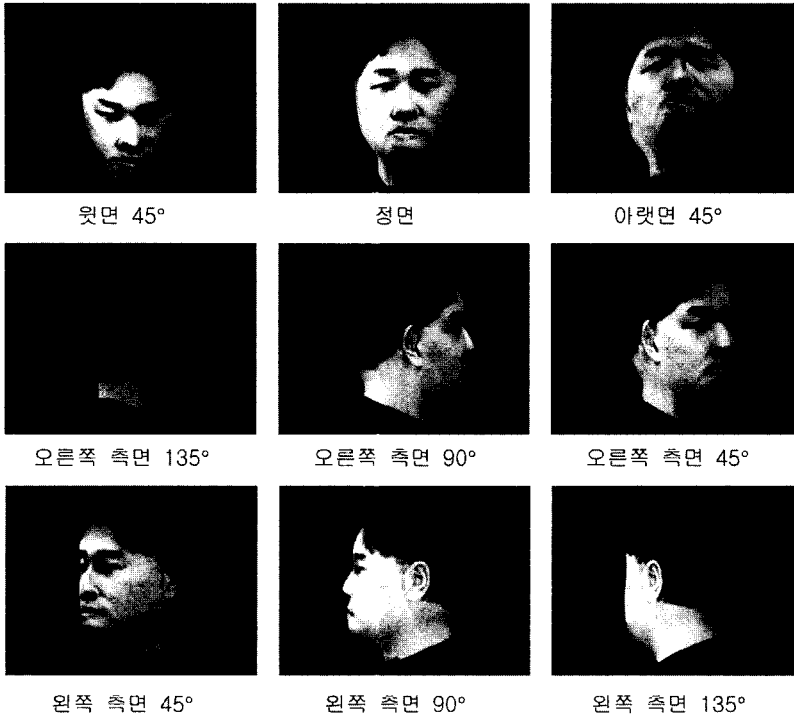
안한 방법은 상하 곡물에 대해서도 계산하므로 오류율의 변화가 상대적으로 작음을 알 수 있다.

그림 12는 기존의 방법과 본 논문에서 제안한 방법으로 생성된 그림 10의 (a), (b) 텍스처를 더 효과적으로 비교할 수 있도록 3차원 얼굴 모델에 텍스처를 매핑한 후 그 결과를 여러 각도에서 보여준다.

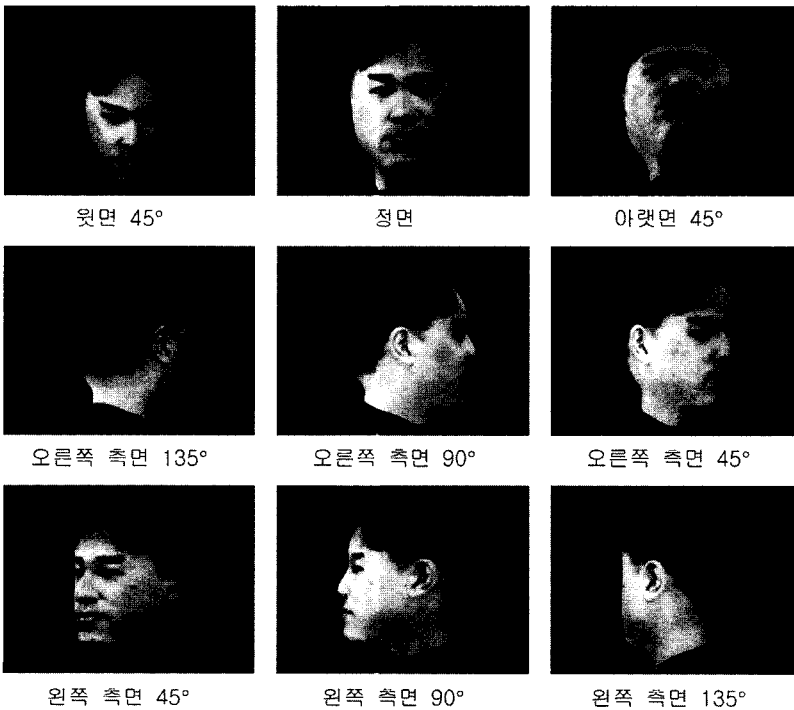
그림 12에서 (a)는 기존의 방법으로 생성된 텍스처를 얼굴 모델에 매핑한 결과이고 (b)는 본 논문에서 제안한 방법으로 생성된 텍스처를 얼굴 모델에 매핑한 결과이다. 그림 12에서 알 수 있듯이 기존 방법으로 생성된 텍스처를 얼굴 모델에 매핑하면 중첩영역의 경계지역은 희미해지고 뭉개져 보이는 고스트 효과가 강하게 나타나며 전체적인 얼굴 구조와 윤곽이 비뚤어져 보이는 등의 문제점으로 인하여 부자연스러운 결과물이 생성된다. 하지만 본 논문에서 제안한 방법으로 생성된 텍스처를 얼굴 모델에 매핑하면 위와 같은 문제점이 해결되고 자연스러운 결과물을 얻을 수 있다. 만약 더 세밀한 3차원 얼굴 모델을 사용한다면 시간 복잡도는 높아지겠지만 더 실감 있는 얼굴 모델을 얻을 수 있다.

6. 결론

본 논문에서는 타원체형 모델의 가변적인 블록크기를 기반으로 하는 영상정렬 알고리즘을 사용하여 얼굴 텍



(a) 동일한 크기의 블록을 이용한 방법으로 생성한 텍스처를 매핑한 결과



(b) 제안한 방법으로 생성한 텍스처를 매핑한 결과

그림 12 3차원 얼굴 모델에 텍스처를 매핑한 결과

스처 영상을 생성하는 방법을 제안하였다. 타원체형 모델을 사용하기 위하여 먼저 타원 파라미터인 장축과 단축을 구한 후 타원체형 모델의 기하학적인 특징인 곡률을 고려하여 영상을 정합하였다. 정합 시에는 상하좌우 곡률을 고려하고 영상의 상관관계를 이용하여 영상을 정렬하므로 영역간의 오차가 발생하는 문제점을 해결하였다. 영상간의 상관관계는 칼라특징과 텍스처 특징을 동시에 고려하므로 입력영상의 특징을 잘 반영할 수 있었다. 그리고 정렬된 입력영상을 하나의 텍스처로 생성하기 위하여 영상 모자의 기법을 사용하였다. 영상을 스티칭할 때 서로 잘 융합하도록 크로스디블브(Cross-dissolve) 기법을 적용하고 중첩된 영역에 대해서는 입력영상의 중첩폭의 거리에 반비례하는 선형적인 가중치를 사용하였다. 중첩영역의 경계 면에서 발생하는 고스트 효과는 3차 회선 보간법을 사용하여 제거하므로 실감 있는 텍스처를 생성하였다.

본 논문에서 제안한 알고리즘은 응용분야로서 얼굴 모델링뿐만 아니라 타원체형 모델의 기하학적인 특성을 갖는 모든 객체에 사용하여 자연스러운 텍스처를 획득할 수 있으므로 그 활용범위가 넓을 것으로 기대된다.

참 고 문 헌

- [1] Y. Lee, D. Terzopoulos, K. Waters, "Realistic Modeling for Facial Animation," In Computer Graphics, SIGGRAPH Proceedings, pp.191-198, Los Angeles, CA, August 1995.
- [2] W.-S. Lee, N. Magnenat-Thalmann, "Fast head modeling for animation," Image and Computing, Vol.18, pp.355-364, March 2000.
- [3] V. Lempitsky, D. Ivanov, Ye. Kuzmin, "Texturing Calibrated Head Model from Images," EG'2002, September 2002.
- [4] Frederic H. Pighin, Richard Szeliski, David Salesin, "Modeling and Animating Realistic Faces from Images," International Journal of Computer Vision 50(2), pp.143-169, 2002.
- [5] B. Tiddeman, N. Duffy, G. Rabey, "Construction and visualization of three-dimensional facial statistics," Computer Methods and Programs in Biomedicine, Vol.63, pp.9-20, September 2000.
- [6] Y. J. Chang, Y. C. Chen, "Facial model adaptation from a monocular image sequence using a textured polygonal model," Signal Processing: Image Communication, Vol.17, pp.373-392, 2002.
- [7] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. Salesin, "Synthesizing Realistic Facial Expressions from Photographs," SIGGRAPH98 Proceedings, pp.75-84, 1998.
- [8] Han Tae-Woo, "3D face modeling system for realistic facial expression animation," MD Thesis, Department of Computer Science, Korea Advanced Institute of Science Technology, 1998.
- [9] Joongjae Lee, Hyungil Choi, "An Image Registration Algorithm Based on Cylindrical Prototype Model," Lecture Notes in Computational Science-ICCS2001, pp.37-43, 2001.
- [10] Ryosuke Motani, "Estimating body mass from silhouette: testing the assumption of elliptical body cross-sections," Paleobiology, 27, pp.735-750, 2001.
- [11] Luis Torres, Jean-Yves Reutter, Luis Lorente, "The Importance of the Color Information in Face Recognition," Proceedings of the 1999 International Conference on Image Processing (ICIP '99), Kobe, Japan, pp.627-631, October 1999.
- [12] B. S. Manjunath, W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," IEEE Transactions on pattern analysis and machine intelligence, Vol.18, No.8, August 1996.
- [13] R. T. Whitaker, "A level-set approach to image blending," Image Processing, IEEE Transactions on, Vol.9, Issue.11, pp.1849-1861, Nov 2000.
- [14] Hendrik P. A. Lensch, W. Heidrich, Hans-Peter Seidel, "A Silhouette-Based Algorithm for Texture Registration and Stitching," Graphical Models, Vol.63, Issue.4, pp.245-262, July 2001.
- [15] W. J. Gordon and J. A. Wixom, "Shepard's method of Metric Interpolation to bivariate and multivariate interpolation," Mathematics of Computation, vol.32, pp.253-264, 1978.
- [16] C. Rocchini, P. Cignoni, C. Montani, R. Scopigno, "Multiple Textures Stitching and Blending on 3D Objects," 10th Eurographics Workshop on Rendering, G. Ward and D. Lischinsky ed., Granada(E), pp.127-138, June 21-23 1999.
- [17] M. Uyttendaele, A. Eden, R. Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," Computer Vision and Pattern Recognition, Proceedings of the 2001 IEEE Computer Society Conference on, Vol.2, pp.519-516, 2001.



이 중 애

1999년 2월 숭실대학교 전자계산학과 졸업(공학사). 2001년 2월 숭실대학교 대학원 컴퓨터학과 졸업(공학석사). 2001년 3월~현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 컴퓨터 비전, 증강 현실, 3D 모델링, 의료영상처리 등임



노 명 우

2002년 2월 성결대학교 멀티미디어학과 졸업(공학사). 2004년 2월 숭실대학교 대학원 컴퓨터학과 졸업(공학석사). 관심분야는 컴퓨터 비전, 영상처리, 패턴인식, 3D 모델링 등임



최 형 일

1979년 2월 연세대학교 전자공학과 졸업(공학사). 1982년 6월 미시간대학교 전산공학과 졸업(공학석사). 1987년 6월 미시간대학교 전산공학과 졸업(공학박사). 1987년 9월~현재 숭실대학교 미디어학부 교수. 관심분야는 컴퓨터비전, 패턴인식, 퍼지이론, 비디오검색, 인터페이스 에이전트 등