

# 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 이용한 얼굴 텍스처 생성

(Face Texture Generation using an Image Alignment  
Algorithm based on Variable-Sized Blocks of Cylindrical  
Prototype Model)

이 중 재<sup>†</sup> 최 형 일<sup>\*\*</sup>  
(Joong Jae Lee) (Hyung Il Choi)

**요약** 본 논문에서는 실감 있는 3차원 얼굴모델 생성에 필요한 얼굴 텍스처를 생성하기 위해 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 제안한다. 이 알고리즘은 실린더형 3차원 모델에 대한 2차원 영상을 영상의 상관관계를 이용하는 블록 정합 알고리즘을 사용해서 정렬하는 방법으로서 블록 정합을 수행할 때 동일한 블록 크기를 사용하는 기존 방법과는 달리 모델의 곡률을 고려해 가변적인 크기의 블록을 사용한다. 그리고 영상 모자이크 기법은 사용해서 정렬된 영상을 하나의 텍스처 영상으로 생성한다. 이때 중첩된 영역에 따라 선형적인 가중치를 부여하면서 크로스디졸브 기법을 이용해 영상을 스티칭한다.

**Abstract** We propose an image alignment algorithm based on variable-sized blocks of cylindrical prototype model to generate a face texture for a realistic 3D face model. This is a block matching algorithm which aligns 2D images of a 3D cylindrical model using the correlation between them. While matching blocks, it does not use same sized blocks but variable sized blocks with considering a curvature of 3D model. And we make a texture of aligned images using a technique of image mosaic. For this purpose, we stitch them with assigning linear weights according to the overlapped region and using the cross-dissolve technique.

## 1. 서론

컴퓨터와 멀티 미디어 기술의 발전과 함께 다양한 사용자 인터페이스 기술이 개발되고 있으며 이런 기술의 발전방향은 사용자에게 보다 친근감 있는 인터페이스를 제공하는데 있다.

특히 3차원 얼굴모델 생성 기법은 이런 연구의 대표적인 예로서 가상현실, 방송, 통신, 지능형 인터페이스 에이전트 등 그 필요성과 활용성이 매우 크다.

3차원 얼굴모델을 생성하는 기존 방법은 크게 장비를 이용하는 방법과 2차원 얼굴영상을 이용하는 방법으로 나눌 수 있다. 장비를 이용하는 방법은 3차원 스캐너와 같은 하드웨어 장비를 사용해 3차원 정보를 획득한 뒤 디지털화해서 실제 사람에 근사한 모델을 생성하는 방법으로 쉽게 모델을 생성할 수 있지만 고가의 장비를 사용해야 한다는 단점이 있다[1]. 2차원 얼굴영상을 이용하는 방법으로는 정면과 측면영상을 이용하는 방법, 2차원 영상을 이용해서 3차원 모델의 기하학적 구조를 복원하고 실린더형 텍스처 맵을 추출하는 방법, 4방향 또는 8방향 얼굴영상을 이용하는 방법 등이 있다[2][3][4][5]. 정면과 측면영상을 이용하는 방법은 2차원 얼굴영상에서 눈, 코, 입, 턱과 같은 특징점들을 추출한 뒤 실제 모델의 프로파일과 형판정합을 해서 일반모델을

\* 본 논문은 2000년도 교내 학술연구비를 지원 받아 수행하였음.

† 비 회 원 : 송신대학교 컴퓨터학부

hic@computing.soongsil.ac.kr

\*\* 총신회원 : 송신대학교 컴퓨터학부 교수

lihop@vision.soongsil.ac.kr

논문접수 : 2001년 2월 5일

심사완료 : 2001년 8월 24일

변형하고 정면과 측면영상을 텍스처 매핑시킴으로써 모델을 생성한다. 그러나 이 방법은 두 장의 얼굴영상만을 사용하기 때문에 정면과 측면을 제외한 나머지 방향에서는 사실감이 많이 떨어진다. 2차원 얼굴영상을 이용해서 3차원 모델의 기하학적 구조를 복원하고 실린더형 텍스처 맵을 추출하는 방법은 서로 다른 방향에서 5대의 카메라로부터 촬영한 영상에 특징점들을 표시한 후 이 점들의 좌표를 카메라 모델에 적용해서 각각의 카메라 파라미터를 추출한다. 추출된 카메라 파라미터를 이용해서 2차원 영상에 표시한 특징점들에 대응되는 3차원 좌표를 계산하고 특징점들을 제외한 나머지 점들에 대해서는 보간법을 사용하여 3차원 좌표를 얻어냄으로써 3차원 모델의 기하학적 구조를 복원한다. 그리고 복원된 3차원 모델과 각 방향에서 촬영한 2차원 영상을 이용하여 실린더형 텍스처 맵을 추출한다. 그런데 이 방법에서 정확한 텍스처 맵을 추출하기 위해서는 3차원 일반모델을 특정한 모델로 변형하는 작업이 필수적이며 그 결과 또한 정확해야 한다. 즉 정확하게 변형된 3차원 모델의 기하학적 정보를 활용해야만 텍스처 맵을 추출할 수 있다는 단점이 있다. 4방향 또는 8방향 얼굴영상을 이용하는 방법은 각 방향에서 촬영한 2차원 얼굴영상에 대해 3차원 일반모델을 프로젝션 시킨 뒤 2차원 상에서 모델을 변형하고 다시 3차원으로 복원함으로써 실제 사람과 근사한 모델을 얻는다. 그리고 이때 사용한 2차원 얼굴영상을 파노라미 기법을 이용해 텍스처를 생성한 뒤 변형된 3차원 모델에 매핑해서 최종적인 얼굴 모델을 생성한다. 그러나 이 방법은 얼굴 텍스처를 생성하기 위해 파노라미 기법중 영상 스티칭 기법만을 사용하므로 정확한 각도에서 촬영을 해야만 하는 제한이 있다. 즉 정확한 각도에서 촬영을 하지 않으면 부자연스러운 텍스처가 생성되고 또 이 텍스처를 모델에 매핑했을 경우 실감 있는 3차원 모델을 생성할 수 없다. 따라서 본 논문에서는 기존 연구의 텍스처 생성 시 3차원 모델 정보를 활용해야한다는 점과 얼굴을 촬영할 때 정확한 각도에서의 촬영이 필요하다는 제한사항을 개선할 수 있는 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 제안한다. 본 논문에서 제안한 영상정렬 알고리즘은 얼굴을 촬영할 때 약간의 오차가 발생하더라도 영상간의 상관관계(correlation)를 이용해 영상을 정확하게 정렬할 수 있으므로 자연스러운 텍스처를 생성할 수 있다. 또한 텍스처 생성시 3차원 모델 정보를 이용하는 기존방법과는 달리 2차원 영상정보만으로도 텍스처를 생성할 수 있다는 장점을 가진다.

그림 1은 본 논문에서 제안한 얼굴 텍스처 생성방법

에 대한 전체 개요도 이다. 그림 1에서 볼 수 있듯이 본 논문에서 제안한 방법은 크게 입력 영상 정렬 단계, 영상 모자이크 단계의 두 단계로 구성된다. 입력 영상 정렬 단계에서는 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘을 사용해서 입력 영상을 정렬한다. 이것은 누상을 실린더형으로 가정했을 때 각각 다른 각도에서 촬영한 2차원 얼굴영상을 정렬하는 방법으로서 블록정합 알고리즘을 사용한다. 그런데 기존 블록정합 방법에서는 동일한 크기의 블록을 사용해서 정합을 하는 반면 본 논문에서는 얼굴모델의 곡률에 따른 가변적인 블록 크기로 분할 및 정합을 시도한다. 그리고 블록에 대한 칼라 특징과 텍스처 특징을 추출한 후 후보블록에 대한 정합 유사도를 계산한다. 이때 정합 유사도 값이 가장 높은 후보블록을 정합된 블록으로 선택하고 이로부터 변위 벡터를 추출한다.

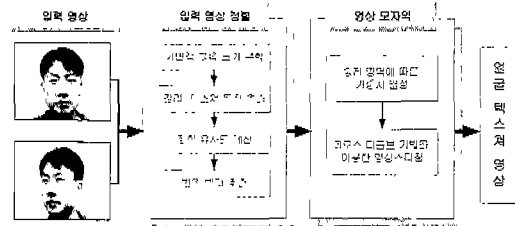


그림 1 전체 개요도

영상 모자이크 단계에서는 정렬된 입력 영상에 대해 중첩 영역에 따라 선형적인 가중치를 부여하고 크로스 디졸브 기법을 이용해 영상을 스티칭함으로써 얼굴 텍스처를 생성한다.

2장에서는 본 논문에서 제안한 실린더형 원형모델 기반의 영상정렬 알고리즘을 설명하고 3장에서는 정렬된 영상을 모자이크해서 얼굴 텍스처를 생성하는 방법에 대해서 설명한다. 4장에서는 실험결과를 설명하며 5장에서는 결론에 대해서 기술한다.

## 2. 실린더형 원형모델 기반의 영상 정렬 알고리즘을 이용한 입력 영상 정렬

파노라마 기법을 사용해서 자연스러운 얼굴 텍스처 영상을 생성하기 위해서는 입력 영상을 정확하게 정렬해야만 한다. 기존 방법에서는 이 과정을 정해진 각도에서 정확하게 촬영을 한다는 전제로 생략할 수 있다. 그러나 부정확한 각도에서 촬영을 할 경우에 텍스처 생성 결과에 오류가 발생하게 된다. 따라서 본 논문에서는 기

존 방법의 문제점을 보완하기 위해 사람의 두상을 실린더형 원형모델로 정의하고 촬영된 2차원 얼굴영상을 정렬할 수 있는 알고리즘을 제안한다. 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상 정렬 알고리즘은 입력 영상을 정렬하기 위해 영상간의 상관관계를 이용하는 블록정합 알고리즘을 사용한다. 기존의 블록정합 알고리즘은 동일한 크기의 블록을 사용해서 정합을 시도하는 반면 본 논문에서는 얼굴모델의 기하학적 특징, 얼굴모델의 곡률, 을 고려해 정합 시 가변적인 크기의 블록을 사용하여 정합을 시도함으로써 정확한 정렬을 할 수 있도록 한다.

2.1 가변적 블록 크기 분할

블록 정합 알고리즘은 두 영상을 동일한 크기의 블록으로 분할한 뒤 임의의 기준 블록에 대한 후보 블록 중에서 정합유사도가 가장 높은 블록을 찾는 방법이다[6]. 본 논문에서는 블록 정합 시 동일한 크기의 블록을 사용하는 기존 방법과는 달리 모델의 곡률을 고려해서 가변적인 크기의 블록을 사용해 정합을 수행한다. 이것은 실린더형 3차원 모델을 촬영한 2차원 영상에서는 모델의 곡률 때문에 동일한 크기의 블록을 사용할 경우 비교하려는 영역간에 오차가 발생하게 되고 이런 문제점을 해결하기 위해서는 모델의 곡률에 따라 가변적인 크기의 블록을 적용해야 하기 때문이다.

본 논문에서는 실린더형 3차원 모델에 대한 2차원 영상의 블록 크기를 식 (1)과 같이 정의한다. 식 (1)에서  $\theta_{Div}$ 는 3차원 모델을 동일한 각도로 나누는 분할각도를 의미하고  $R$ 은 모델의 반지름의 길이를 의미한다. 그리고  $k_0$ 는 3차원 모델을  $\theta_{Div}$ 로 분할했을 때 생성되는 내접 삼각형의 밑변의 길이를 나타내며  $k_0$ 를 제외한  $k_n$ 은 2차원 영상에 프로젝션된 블록의 크기를 의미한다.

식 (1)에서  $n$ 은 3차원 모델에 대해 임의의 한 방향에서 볼 수 있는  $0^\circ$ 에서  $180^\circ$ 까지의 시계를 분할각도  $\theta_{Div}$ 로 분할했을 때 2차원 영상에 생성되는 블록열의 인덱스를 의미한다.

$$\begin{cases} BW = k_n \\ k_0 = 2R \sin\left(\frac{\theta_{Div}}{2}\right) \\ k_n = k_0 \cos\left[\left(|n| - \frac{1}{2}\right) \times \theta_{Div}\right] \end{cases} \quad (1)$$

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

그림 2는 실린더형 모델을 동일한 각도  $\theta_{Div}$ 로 분할했을 때 모델에 대한 2차원 영상에서 모델의 곡률에 따라 영상의 중심을 기준으로 영상의 좌우측으로 갈수록

블록의 크기가 점차 작아지는 가변적인 크기의 블록이 생성되는 결과를 보여준다.

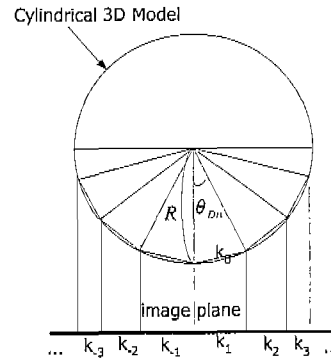


그림 2 모델의 곡률에 따른 가변적 블록 크기

이때 블록의 크기  $k_n$ 에서 첨자  $n$ 이 음수값을 가지면 2차원 영상에서 중심을 기준으로 좌측영역의 블록을 의미하고 이에 반해서 양수값을 가지면 우측영역의 블록을 의미한다.

그림 3은 각각 기존의 블록 정합 알고리즘과 본 논문에서 제안한 방법으로 영상을 분할한 결과를 비교해서 보여준다. 그림 3에서 (a),(c)는 기존 블록 정합 알고리즘에서 사용하는 동일한 크기로 영상을 분할하는 방법을 보여주며 (b),(d)는 본 논문에서 제안한 방법인 모델의 기하학적 성질, 즉 곡률을 이용해서 가변적인 크기로 영상을 분할한 결과를 보여준다.

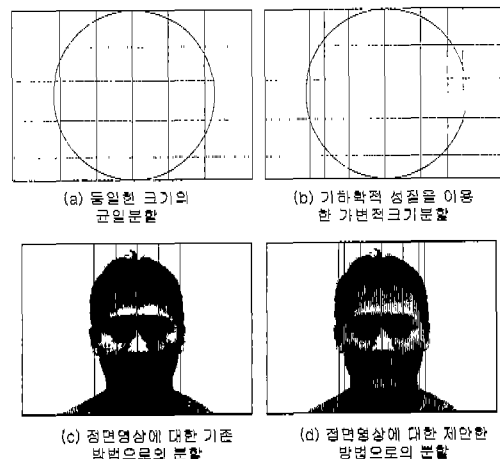


그림 3 정면영상에 대한 블록분할 방법에 따른 블록 크기 비교

2.2 가변적 정합 블록 크기 조절

본 논문에서는 블록 정합을 수행하기 위해서 가변적인 크기의 블록으로 영상을 분할했기 때문에 기존 블록에 대해서 정합을 시도하고자 하는 후보 블록들의 크기 역시 가변적으로 조절한다.

본 논문에서는 실린더형 3차원 모델에 대한 2차원 영상의 임의의 점  $I_p$ 에서의 블록 크기인  $BW(I_p)$ 를 식 (2)와 같이 정의한다. 식 (2)에서  $\theta_{Div}$ 는 분할각도를 나타내고  $\theta_c$ 는 모델의 중심  $c$ 에 대해서 점  $c_0$ 와 점  $p$ 가 이루는 각도를 나타낸다. 그리고 점  $I_p, I_{p'}, I_c$ 는 각각 점  $p, p', c$ 를 2차원 영상에 투영시킨 점들이다. 또한  $R$ 은 3차원 모델의 반지름의 길이를 의미하며  $D$ 는 점  $I_p$ 와  $I_c$ 사이의 픽셀거리를 의미한다.

$$\begin{aligned}
 BW(I_p) &= |I_{p'} - I_p| = |I_p - I_c| - |I_p - I_c| \\
 &= R \sin \theta - D \\
 &= R \sin(\theta_{Div} + \theta_c) - D \\
 \theta_c &= \sin^{-1}\left(\frac{D}{R}\right) \\
 D &= |I_p - I_c|
 \end{aligned}
 \tag{2}$$

식 (2)에서 알 수 있듯이 2차원 영상의 임의의 점  $I_p$ 에서의 블록 크기  $BW(I_p)$ 는 점  $I_p$ 와  $I_p$ 사이의 거리로 정의하는데 이것은 그림 4의 (a)에서 볼 수 있듯이  $I_p$ 에서의 블록 크기  $BW(I_p)$ 가 모델상의 한 점  $p$ 에서 분할각도  $\theta_{Div}$ 만큼 떨어진 점  $p'$ 을 찾아 이 점을 2차원 영상에 투영시킨 점  $I_{p'}$ 을 찾음으로써 구할 수 있다.

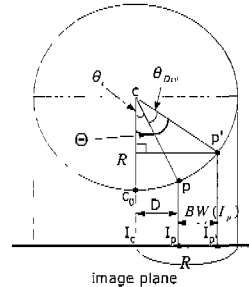
그림 4의 (a)는 식 (2)에 의한 실린더형 3차원 모델의 곡률에 따른 가변적인 블록 크기 조절방법을 보여주며 (b)는 이 방법을 이용해 모델의 반지름  $R$ 이 62픽셀, 분할각도  $\theta_{Div}$ 가  $20^\circ$ 일 때 2차원 영상에서 중심으로부터의 거리에 따라서 가변적으로 조절되는 블록의 크기를 보여준다. 그림 4의 (b)에서 확인할 수 있듯이 중심으로부터 거리가 멀어질수록 블록의 크기가 점차 줄어드는데 이것은 모델의 기하학적 특징인 곡률에 따라서 2차원 영상에서는 블록의 크기가 가변적으로 조절됨을 말한다.

2.3 정합 유사 함수의 정의

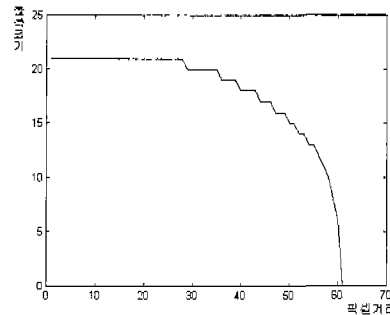
본 절에서는 블록 사이의 정합 척도로 사용되는 정합 유사 함수를 설명한다.

2.3.1 블록에 대한 칼라 및 텍스처 특징 추출

본 논문에서는 입력영상으로부터 영상간의 상관관계를 잘 반영할 수 있는 칼라 특징과 텍스처 특징을 사용해 블록 정합을 수행한다. 영상에 대한 칼라 특징은 명암의 밝기 변화에 민감한 RGB 칼라모델 대신 조명의 변화에 안정적인 YIQ 칼라 모델을 사용한다[7]. 그리고



(a) 블록 크기 결정



(b) 실험결과( R=62픽셀,  $\theta_{Div}=20^\circ$  )

그림 4 가변적 블록 크기 조절

블록 정합 시 색상분포가 균일한 영역에서 발생할 수 있는 정합 오류 발생의 문제점을 보완하고 영상의 특징을 잘 반영할 수 있는 텍스처 정보를 사용한다.

식 (3)에서  $M_{color}(B_i)$ 은  $i$ 번째 블록의 칼라 특징에 대한 유사도를 나타내는 값으로 0과 1사이의 값을 가지며 이 값이 작을수록 비교하는 블록사이의 유사도가 커진다. 식 (3)에서  $C_{B_i}^0$ 은 각도 영상의  $i$ 번째 블록에 대한 칼라 특징을 나타내며,  $M$ 은 블록의 너비,  $N$ 은 블록의 높이를 나타낸다.

$$\begin{aligned}
 M_{color}(B_i) &= \left| \frac{C_{B_i}^0 - C_{B_i}^1}{C_{max}} \right| \\
 C_{B_i}^0 &= \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(x+i, y+j),
 \end{aligned}
 \tag{3}$$

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

식 (4)에서  $M_{texture}(B_i)$ 은  $i$ 번째 블록의 텍스처 특징에 대한 유사도를 나타낸다. 본 논문에서는 영상의 텍스처 정보를 획득하기 위해서 가보 웨이블릿 커널을 사용하고 구체적으로 웨이블릿 커널을 영상에 씌운 뒤 컨볼루션(convolution)해서 영상에 대한 텍스처 정보를 얻어낸다. 이 방법은 기존의 텍스처 추출 방법 보다 더 구

분력이 뛰어난 텍스처 정보를 얻을 수 있기 때문에 텍스처에 기반한 영상처리에 더 널리 사용되고 있다[8].

$$M_{texture}(B_i) = \left| \frac{T_{B_i}^{\theta_i} - T_{B_i}^0}{T_{max}} \right|$$

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

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

$$\psi_{m\theta}(x, y) = Real\ part + i \cdot Imaginary\ part \quad (4)$$

$$= \frac{(k_m \cos k_\theta)^2 + (k_m \sin k_\theta)^2}{\sigma^2}$$

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

$$\cdot [\cos \{xk_m \cos k_\theta + yk_m \sin k_\theta\}$$

$$+ i \cdot \sin \{xk_m \cos k_\theta + yk_m \sin k_\theta\}] - \exp \left\{ - \frac{\sigma^2}{2} \right\}$$

식 (4)에서  $\psi_{m\theta}$ 은 가보 웨이블릿 커널을 나타내며 이때  $m$ 과  $\theta$ 는 각각 커널의 크기와 방향을 결정하는 인덱스를 나타낸다.

### 2.3.2 정합 유사함수와 변위 벡터 추출

본 논문에서는 블록 사이의 정합 척도로 사용하는 정합 유사 함수를 식 (5)와 같이 정의 한다. 식 (5)에서  $SF(i, j; u, v)$ 는 칼라 특징과 텍스처 특징을 동시에 고려한 정합 유사도를 의미하고 0과 1사이의 값을 가진다. 이때  $\alpha, \beta$ 는 각 특징이 정합 유사도 계산에 기여하는 정도를 나타내는데 본 논문에서는 실험적으로  $\alpha$ 는 0.4,  $\beta$ 는 0.6을 사용한다.

$$\mathbf{d}(i, j) = (u^*, v^*) \text{ where } SF(i, j; u, v) \text{ is maximized} \quad (5)$$

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

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

식 (5)에서  $(i, j)$ 는 기준 블록의 위치를 나타내며  $(u, v)$ 는 기준 블록과 정합된 블록사이의 변위를 나타낸다. 식 (5)에서 알 수 있듯이 두 영상을 정렬하는데 필요한 기준 블록과 정합된 블록간의 변위는 기준 블록에 대한 후보 블록들의 정합 유사도를 계산한 뒤 이중 최대 정합 유사도를 가지는 블록을 정합된 블록으로 결정하고, 기준 블록에 대한 이 블록의 상대적 위치를 변위벡터  $(u^*, v^*)$ 로 정의한다. 여기서  $u^*$ 는 x축으로의 변위를 의미하고  $v^*$ 는 y축으로의 변위를 의미한다.

그리고 기준블록과 후보블록의 정합 유사도를 구할 때 기준블록에 대해서 생성되는 후보 블록의 개수는 식 (6)과 같으며 이때 생성된 후보블록의 크기는 식 (2)로부터 구할 수 있다.

$$NCB(k_n) = R \times \left[ 1 - n \times \left( \frac{2\theta_{Div}}{\pi} \right) \right] \quad (6)$$

식 (6)에서  $NCB(k_n)$ 는  $n$ 번째 기준블록인  $k_n$ 에 대해

서 생성되는 후보블록의 개수를 의미하고  $R$ 은 모델의 반지름,  $\theta_{Div}$ 는 분할각도를 나타낸다. 그리고 식 (6)은 얼굴영상에서 블록을 정합할 때 임의의 기준블록에 대한 탐색영역을 얼굴영역의 반에 해당하는 부분만을 고려하는데 이것은 촬영한 입력영상간에 각도차이를 90° 범위 이내로 했기 때문이다.

식 (6)에서 알 수 있듯이 중심에서 멀어지는 기준블록일수록 생성되는 후보블록의 개수가 줄어들며 이웃하는 기준블록간에는 단위 분할각도  $\theta_{Div}$ 에 대한 블록개수 만큼의 후보블록개수 차이가 있음을 알 수 있다.

그림 5는 첫 번째 기준블록인  $k_1$  대한 후보블록 생성 과정을 보여주고 있다. 그림 5에서와 같이 실린더형 모델 상에서 분할각도  $\theta_{Div}$ 만큼씩 일정하게 떨어진 점은 2차원 영상에 프로젝션시켜 후보블록을 생성한다.

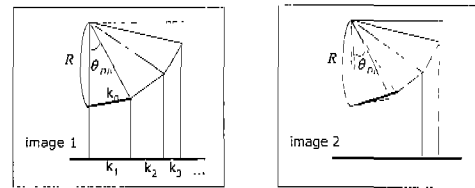


그림 5 기준블록에 대한 후보블록 생성과정

### 3. 영상 모자이크

본 장에서는 실린더형 원형모델 기반의 영상정렬 알고리즘을 이용해 두 영상을 정렬한 뒤 하나의 텍스처 영상을 생성하는 영상 모자이크 기법에 대해서 설명한다. 본 논문에서는 얼굴 텍스처 영상을 생성하기 위해 영상 모자이크 기법을 사용하며 이때 크로스디졸브(cross-dissolve) 기법을 적용한다. 크로스디졸브 기법은 영상들이 서로 자연스럽게 융합하도록 사용하는 방법으로서 컴퓨터 비전 또는 컴퓨터 그래픽스 분야에서 연구되고 있는 모핑 기술에서 많이 이용한다[9].

두 영상이 정렬된 상태에서는 같은 위치에 해당하는 픽셀의 칼라값을 적절한 가중치에 따라 혼합함으로써 최종 텍스처를 생성할 수 있다. 특히 두 영상을 스티칭해서 자연스러운 하나의 텍스처 영상을 생성하기 위해서는 두 영상이 중첩되는 영역에 대한 처리가 매우 중요하다. 본 논문에서는 중첩된 영역을 스티칭할 때 보다 자연스러운 텍스처를 생성하기 위해서 중첩된 영역의 크기를 고려해 가중치를 부여하는 방법을 사용한다. 이것은 중첩된 영역의 크기에 따라 각각의 영상에 대해서 반비례관계를 가지는 선형적인 가중치를 부여하는 방법이다.

3.1 크로스 디졸브 기법을 이용한 영상 스티칭

본 논문에서는 영상 모자이크 기법을 사용해 생성되는 얼굴 텍스처 영상을 식 (7)과 같이 정의한다.

$$T = \begin{cases} I_1, & \text{if } Visible(P, I_1) \text{ and } \neg Visible(P, I_2) \\ I_2, & \text{if } \neg Visible(P, I_1) \text{ and } Visible(P, I_2) \\ w_1 \cdot I_1 + w_2 \cdot I_2, & \text{if } Visible(P, I_1) \text{ and } Visible(P, I_2) \end{cases} \quad (7)$$

식 (7)에서  $T$ 는 텍스처 영상을 의미하고  $I_1, I_2$ 는 입력영상을 의미한다. 그리고  $P_i$ 는  $i$ 번째 픽셀을 의미하며  $w_1, w_2$ 는 두 영상을 스티칭할 때 각각  $I_1, I_2$ 에 부여하는 가중치를 의미한다. 식 (7)에서 알 수 있듯이 두 영상이 중첩되지 않은 영역에 대해서는 픽셀  $P_i$ 가 영상  $I_1$ 과  $I_2$ 중에서 어떤 영상에서 보이는 픽셀인지에 대한 여부를 확인해서 텍스처 영상에 사용할 픽셀값을 결정한다. 그리고 그림 5와 같이 두 영상이 중첩된 영역에 대해서는 식 (8)에서 정의한 가중치 부여함수에 따라 영상  $I_1$ 과  $I_2$ 에 대해 각각 가중치  $w_1, w_2$ 를 부여해서 영상을 스티칭한다.

3.2 중첩 영역에 따른 가중치 결정

본 논문에서는 두 영상을 스티칭할 때 정렬된 두 영상의 변위에 따라 발생하는 중첩된 영역을 고려해 가중치를 부여한다.

그림 6은 두 영상의 변위  $D$ 만큼 입력영상을 정렬했을 때 발생하는 중첩되는 영역관계를 보여주며  $D$ 가 커질수록 중첩되는 영역은 작아지는 것을 알 수 있다.

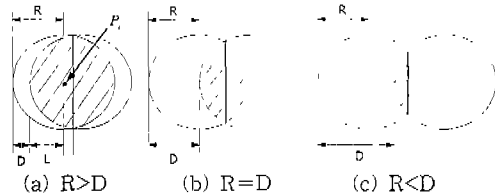


그림 6 두 영상의 중첩 시 발생하는 영역관계

본 논문에서는 두 영상을 스티칭할 때 부여하는 가중치들 식 (8)과 같이 정의한다. 식 (8)에서  $R$ 은 모델의 반지름을 나타내고  $L$ 은 그림 6에서 볼 수 있듯이 중첩된 영역의 좌측 경계로부터 임의의 픽셀  $P_i$ 까지의 거리를 나타내며  $D$ 는 정렬된 두 영상간의 변위를 나타낸다. 식 (8)에서 알 수 있듯이 본 논문에서 사용하는 가중치 부여함수는 중첩된 영역의 중심을 기준으로 한 거리  $L$ 에 대해서 반비례하는 선형적인 가중치를 사용한다.

$$w_1 = \begin{cases} \frac{2L}{2R-D}, & \text{if } D > R \text{ or } D < R \\ \frac{2L}{R}, & \text{if } D = R \end{cases} \quad (8)$$

$$w_2 = 1 - w_1 \quad \text{where } 0 \leq w_1, w_2 \leq 1, w_1 + w_2 = 1$$

그림 7은 중첩된 영역에 대해서 식 (8)에 의해 두 영상  $I_1$ 과  $I_2$ 에 부여하는 가중치  $w_1$ 과  $w_2$ 를 보여준다. 그림 7에서 알 수 있듯이 가중치  $w_1$ 과  $w_2$ 사이에는 반비례 관계가 있으며 각각의 가중치는 중첩된 영역의 중심으로부터의 거리  $L$ 에 반비례한다.

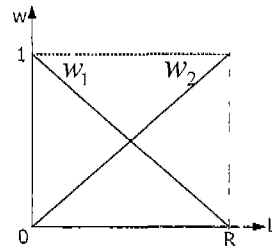


그림 7 두 영상의 중첩에 따라 부여하는 가중치

4. 실험 및 결과

본 장에서는 본 논문에서 제안한 실린더형 원형모델

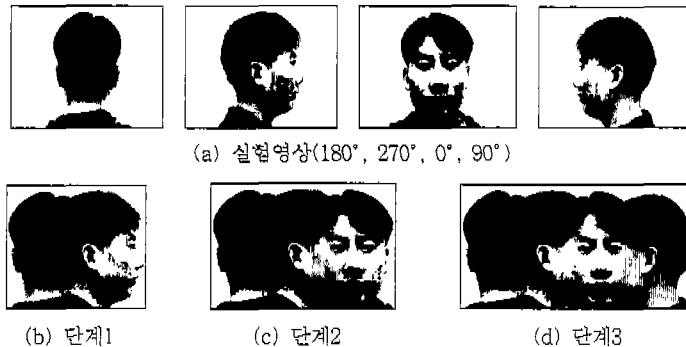
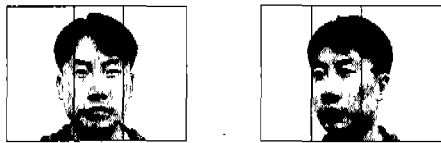


그림 8 4방향 얼굴영상을 이용한 텍스처 생성과정

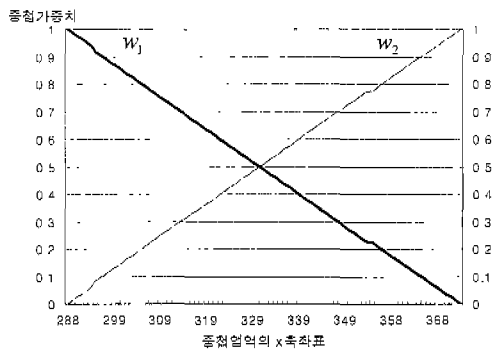
기반의 영상정렬 알고리즘을 이용한 얼굴 텍스처 생성에 대한 실험결과를 기술한다. 실험에 사용한 영상은 Kodak사의 DC220 디지털 카메라로 촬영하였으며 실험 영상의 크기는 340×240으로 정규화 하였고 전처리들 통해서 배경영역을 제거한 영상을 사용하였다.

그림 8은 (a)에서 사람의 얼굴을 90° 간격으로 4방향에서 촬영한 실험영상을 보여주고 (b),(c),(d)에서는 단계별로 생성되는 텍스처 영상을 보여준다.

그림 9는 정렬된 두 영상을 보자의 기법을 사용하여 스티칭할 때 중첩된 영역에 대해서 픽셀의 x좌표에 따라서 부여하는 가중치를 보여준다. 그림 9의 (a), (b)에서 각각 두 개의 실선으로 표시한 안쪽부분이 두 영상이 중첩되는 곳을 나타내고 (c)는 중첩영역에 따라 부여되는 가중치  $w_1$ ,  $w_2$  를 보여준다. 그림에서 볼 수 있듯이 각각의 가중치는 중첩영역에 따라서 선형적으로 부여되며 서로 반비례하는 관계를 가진다. 그리고 그림 9의 (d)는 (c)와 같은 가중치에 따라 (a), (b) 두 영상을 스티칭한 결과를 보여준다.



(a) 정면영상(0°) (b) 측면영상(45°)



(c) 중첩영역에 부여되는 가중치  $w_1$ ,  $w_2$



(d) 생성된 얼굴 텍스처

그림 9 중첩영역을 스티칭할 때 부여되는 가중치

그림 10은 4방향 그리고 8방향 얼굴영상을 이용해서 생성된 얼굴 텍스처 결과를 보여준다. 그림 10에서 (a), (b)는 기존 방법으로 텍스처를 생성한 결과이고 (c), (d)는 본 논문에서 제안한 방법으로 텍스처를 생성한 결과이다. 그림 10에서와 같이 정확한 각도에서 촬영한 얼굴 영상에 대해서는 두 방법 모두 거의 비슷한 텍스처를 생성하고 있음을 알 수 있다.



(a) 4방향(기존 방법) (b) 8방향(기존 방법)



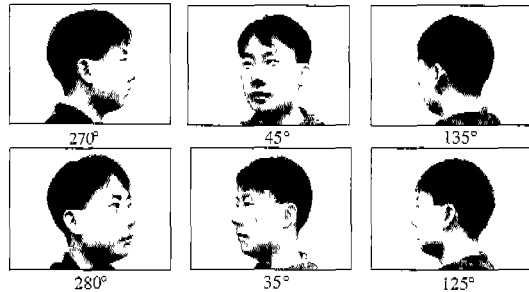
(c) 4방향(제안한 방법) (d) 8방향(제안한 방법)

그림 10 얼굴영상을 이용한 텍스처 생성결과

그림 11은 부정확한 각도에서 촬영한 실험영상을 이용해 각각 기존 방법과 본 논문에서 제안한 방법을 사용해서 생성한 텍스처 결과를 비교하는 장면을 보여준다. 그림 11의 (a)는 정해진 각도에서 정확하게 촬영된 영상과 각각의 영상에 대해 10° 차이가 나는 부정확한 촬영영상을 보여주고 (b)는 기존 방법을 사용해서 생성된 텍스처, (c)는 제안한 방법을 사용해서 생성된 텍스처 결과를 보여준다.

그림 11의 (b)에서 볼 수 있듯이 기존 방법으로 생성한 텍스처는 눈 영역이 희미해지고 귀 영역에서는 두 개의 귀가 나타나는 고스트 현상등의 문제점으로 인해서 부자연스러운 텍스처가 생성된다. 이에 반해서 그림 11의 (c)는 본 논문에서는 제안한 방법으로 기존 방법이 가지고 있는 문제점이 해결되었음을 볼 수 있다. 이것은 두 영상의 상관관계를 이용해 일력영상을 정렬한 뒤 스티칭함으로써 촬영오차가 발생해도 자연스러운 텍스처 결과를 얻을 수 있기 때문이다.

그리고 그림 12는 기존 방법과 본 논문에서 제안한 방법으로 생성한 그림 11의 (b),(c) 텍스처 결과를 더 효과적으로 비교해 볼 수 있도록 3차원 얼굴모델에 생성된 텍스처를 매핑한 결과를 보여준다.

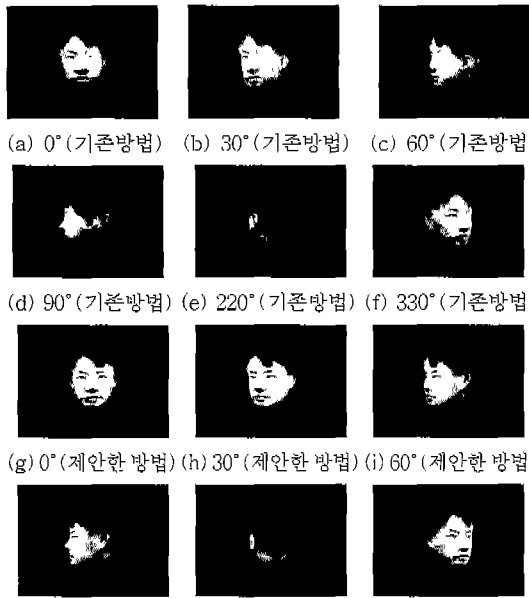


(a) 10° 의 촬영오차가 있는 실험영상



(b) 기존 방법 (c) 제안한 방법

그림 11 부정확한 촬영 시 텍스처 생성결과 비교



(a) 0°(기존방법) (b) 30°(기존방법) (c) 60°(기존방법)  
(d) 90°(기존방법) (e) 220°(기존방법) (f) 330°(기존방법)  
(g) 0°(제안한 방법) (h) 30°(제안한 방법) (i) 60°(제안한 방법)  
(j) 90°(제안한 방법) (k) 220°(제안한 방법) (l) 330°(제안한 방법)

그림 12 3차원 모델에 텍스처를 매핑한 결과

그림 12에서 (a)~(f)는 기존방법으로 생성한 텍스처를 3차원 얼굴모델에 매핑한 결과이고 (g)~(l)은 본 논문에서 제안한 방법으로 텍스처를 매핑한 결과이다. 그림 12에서 알 수 있듯이 부정확한 각도에서 촬영한 2차

원 영상으로부터 생성된 텍스처를 3차원 모델에 매핑한 결과, 기존방법에서는 왼쪽 눈 영역이 희미하고 귀 영역에서는 두 개의 귀가 나타나는 고스트 현상을 발견할 수 있다. 그러나 본 논문에서 제안한 방법으로 생성한 텍스처를 매핑한 결과에서는 이런 문제점이 해결되었음을 알 수 있다.

### 5. 결론

본 논문에서는 실린더형 3차원 모델을 촬영한 2차원 영상을 효과적으로 정렬할 수 있는 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘과 이를 이용해서 얼굴 텍스처 영상을 생성하는 방법을 제안하였다. 실린더형 원형모델의 가변적 블록크기를 기반으로 하는 영상정렬 알고리즘은 영상의 상관관계를 이용해 입력영상을 정렬하는 알고리즘으로서 기존의 동일한 크기의 블록을 사용해 정렬을 시도하는 블록 정합 알고리즘은 개선해 정합 시 가변적인 블록 크기를 사용하였다. 이것은 실린더형 3차원 모델의 기하학적 특징인 곡률을 고려한 것으로 블록 정합 시 동일한 영역을 비교하기 위해서는 영상의 중심을 기준으로 해서 영상의 좌우측으로 갈수록 블록 크기가 작아지는 가변적인 블록 크기를 사용해야 하기 때문이다. 그리고 본 논문에서는 영상의 상관관계를 구하기 위해서 입력영상의 특징을 잘 반영할 수 있는 칼라 특징과 텍스처 특징을 사용하였다.

본 논문에서는 정렬된 입력영상을 하나의 텍스처로 생성하기 위해서 영상 모자이크법을 사용하였다. 영상 모자이크 기법으로 두 영상을 스티칭할 때 크로스디졸브 기법을 사용하였으며 이때 중첩된 영역에 대해서는 중첩된 영역의 중심으로부터의 거리에 반비례하는 가중치를 부여하는 선형적인 가중치 부여함수를 사용하였다.

본 논문에서 제안한 영상정렬 알고리즘은 얼굴을 촬영할 때 약간의 오차가 발생하더라도 영상간의 상관관계(correlation)를 이용하여 영상을 정확하게 정렬할 수 있으므로 자연스러운 텍스처를 생성할 수 있다. 또한 텍스처 생성시 3차원 모델정보를 이용하는 기존방법과는 달리 2차원 영상정보만으로도 텍스처를 생성할 수 있다는 장점을 가진다.

향후 연구과제로는 본 논문에서 제안한 영상정렬 알고리즘이 기존 방법에 비해 영상을 정렬하는데 필요한 시간 복잡도가 높다는 문제점을 개선하는 것과 정렬된 두 영상을 스티칭할 때 3차원 모델의 기하학적 특징을 고려해서 폐색되는 영역을 자동적으로 찾는 동시에 제외시킬 수 있는 방법에 관한 연구가 필요하다.



참고 문헌

[1] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters, "Realistic Modeling for Facial Animation," Proceedings of SIGGRAPH 95. In Computer Graphics, pp. 55-62, 1995.

[2] Takaaki Akimoto, Yasuhito Suenaga, "3D Facial Model Creation Using Generic Model and Side View of Face," In IEICE TRANS. On INF & SYST., Vol E75 D, No.2, pp. 191-197, 1992.

[3] Takaaki Akimoto, Yasuhito Suenaga, "Automatic Creation of 3D Facial Models," In IEEE Computer Graphics & Applications, pp.16-22, Sep 1993.

[4] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, "Synthesizing Realistic Facial Expressions from Photographs", Siggraph proceedings, pp. 75-84, 1998.

[5] Ilan Tac-Woo, "3D face modeling system for realistic facial expression animation," MD Thesis, Department of Computer Science, Korea Advanced Institute of Science Technology, 1998.

[6] Kyoung Won Lim, Byung Cheol Song, and Jong Beom Ra, "Fast Hierarchical Block Matching algorithm Utilizing Spatial Motion Vector Correlation," Proceedings of Visual Communications and Image processing, Vol. 3024, pp. 284-291, 1997.

[7] Yu-ich Ohta, Takeji Kanade, Toshiyuki Sakai, "Color Information for Region Segmentation," In Computer Graphics and Image Processing, Vol. 13, pp. 222-241, 1980.

[8] W. Y. Ma and B. S. Manjunath, "Texture Features and Learning Similarity," IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June, 1996.

[9] G. Wolberg Image morphing: a survey. The Visual Computer, 14(8/9) : pp. 360-372, 1998.



최형일

1979년 2월 연세대학교 전자공학과 졸업(공학사). 1982년 6월 미시간대학교 전산공학과 졸업(공학석사). 1987년 6월 미시간대학교 전산공학과 졸업(공학박사). 현재 숭실대학교 미디어학부 교수. 관심분야는 컴퓨터 비전, 패턴 인식, 워터마킹, 인터페이스 에이전트 등임.



이중재

1999년 2월 숭실대학교 전자계산학과 졸업(공학사). 2001년 2월 숭실대학교 대학원 컴퓨터학과 졸업(공학석사). 2001년 3월 - 현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 컴퓨터 비전, 영상처리, 패턴인식, 3D 모델링 등임.