

벡터장 기반 변형기술을 이용한 이미지 와핑 방법 : 텍스처 매핑에의 응용을 중심으로

(Image Warping Using Vector Field Based Deformation and Its Application to Texture Mapping)

서 헤 원 [†] 코디에 프레데릭 ^{**}
(Hyewon Seo) (Frederic Cordier)

요약 본 논문은 Von Funck 외가 제안한 벡터장 기반 변형기술에 기반한 새로운 텍스처 이미지의 와핑 방법을 제안한다. 사용자는 텍스처 이미지와 텍스처매핑이 될 3차원 물체 표면에 두 가지 방법으로 제한조건을 지정할 수 있다: 위치 제한조건은 이미지 상의 특징점의 위치를 특정위치 (물체 표면의 대응하는 특징점 위치)로 제한하는 데 사용하며, 그라디언트 제한조건을 사용하려는 특징점 이미지의 일부분에 대한 방향과 스케일에 대해 제어가 가능하다. 이러한 제한조건으로부터 그에 대응하는 C1 연속인 속도벡터장을 이미지 공간 상에 구축하고, 구축된 벡터장을 따라 텍스처 이미지 상의 모든 픽셀들을 이동시킨 결과로써 최종 텍스처 이미지를 획득한다.

본 연구에서 구현한 텍스처 매핑 애플리케이션에서는 우선 대상 3차원 메쉬를 최소제곱 등각매핑 (Least Squares Conformal Mapping) 방법을 사용하여 2차원 평면에 임베딩시킨다. 다음으로, 제안하는 이미지 와핑 방법을 적용하여 텍스처 이미지 상의 특징점이 임베딩된 메쉬의 대응하는 특징점과 일치되도록 하여 최종 텍스처맵을 획득한다. 이와 같은 벡터장기반 방법을 통해 기존의 텍스처매핑 방법보다 더욱 부드러운 이미지 와핑결과를 얻을 수 있었으며, 왜곡이 많이 생기는 요구조건을 만족하면서도 자기 오버랩이 생기지 않는 좋은 성질을 보임을 확인하였다.

키워드 : 텍스처매핑, 벡터장 기반 변형, 이미지 와핑

Abstract We introduce in this paper a new method for smooth foldover-free warping of images, based on the vector field deformation technique proposed by Von Funck et al. It allows users to specify the constraints in two different ways: positional constraints to constrain the position of a point in the image and gradient constraints to constrain the orientation and scaling of some parts of the image. From the user-specified constraints, it computes in the image domain a C1-continuous velocity vector field, along which each pixel progressively moves from its original position to the target. The target positions of the pixels are obtained by solving a set of partial derivative equations with the 4th order Runge-Kutta method.

We show how our method can be useful for texture mapping with hard constraints. We start with an unconstrained planar embedding of a target mesh using a previously known method (Least Squares Conformal Map). Then, in order to obtain a texture map that satisfies the given constraints, we use the proposed warping method to align the features of the texture image with those on the unconstrained embedding. Compared to previous work, our method generates a smoother texture mapping, offers higher level of control for defining the constraints, and is simpler to implement.

Key words : texture mapping, vector-field based deformation, image warping

· 본 연구는 학술진흥재단의 2007년 신진교수지원사업(D00401) 및 한국연구재단의 지원을 받아 수행된 기초연구사업(2009-0063172)의 결과물임

[†] 정희원 : 충남대학교 컴퓨터 공학과 교수
hseo@cnu.ac.kr

^{**} 정희원 : KAIST
frederic.cordier@uha.fr

논문접수 : 2009년 3월 9일

심사완료 : 2009년 7월 20일

Copyright©2009 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제36권 제5호(2009.10)

1. 서론

방사기저함수(Radial Basis Function)와 같이 특징점 제한조건을 허용하는 이미지 와핑 방법들이 많이 알려져 있으나, 이들 대부분은 일대일(bijective) 매핑을 보장하지 않기 때문에 심한 왜곡이 불가피한 제한조건이 주어지는 경우 오버랩(foldover)이 생길 수 있다는 문제점이 있다.

본 연구에서는 Von Funck 외[1]가 제안한 벡터장 기반 변형 방법을 도입하여 왜곡이 많이 생기는 요구조건을 만족하면서도 오버랩이 생기지 않는 이미지 와핑 방법을 개발하였다. 형상을 이루는 기하 원시요소들(geometric primitives)이 주어진 출발점(source)에서 목적점(target)까지 도달하는 과정을 단일(one-step) 프로세스로 보는 기존의 방법들과는 달리, 벡터장 기반 변형 방법은 시간에 따라 연속적으로 변하는 벡터장이 존재하여 그것에 의해 원시요소들의 점진적인 위치 변화가 일어나고, 그 결과 원시요소들의 집합으로 모델링된 형상에 변형이 일어난다고 본다. 즉, 출발형상에서 목적형상까지의 변형을 유도하는 벡터장이 시간의 흐름에 따라 업데이트되고, 이에 따라 형상을 점진적으로 변형하는 프로세스가 반복적으로 일어난다고 보는 것이다. 그 결과 왜곡이 심한 제한조건이 주어지는 경우에도 자기오버랩이 일어나지 않으면서 부드러운 변형을 유도해 낼 수 있다.

특히, 개발된 이미지 와핑 방법은 제한조건을 가지는 텍스처매핑 애플리케이션에 사용하는 것을 목표로 하였다. 텍스처 매핑은 3차원 모델 표면에 이미지를 매핑하여 효과적으로 외관의 표현을 향상시키는 데 널리 사용되고 있다. 텍스처 매핑의 가장 중요한 과정은 3차원 물체 표면의 매개화, 즉, 모델 표면전체와 텍스처 평면 사이의 일대일 대응을 찾아내는 작업이다. 이러한 대응 과정에는 종종 사용자가 지정한 특정 점들 사이의 대응조건을 만족해야하는 경우가 생긴다. 본 연구에서는 대상 3차원 물체를 최소제곱 등각매핑(Least Squares Conformal Mapping) 방법을 사용하여 2차원 상의 평면 임베딩(planar embedding)[2]을 생성한 후, 제안하는 이미지 와핑 방법을 사용하여 텍스처 이미지를 변형함으로써 제한조건을 만족하는 일대일 대응을 계산한다. 이 변형 과정에는 사용자가 미리 지정한 특징점쌍들을 제한조건으로 사용한다: 사용자는 텍스처 이미지 상의 특징 점이 메쉬 상의 특징점에 대응하도록 하는 특징점쌍을 지정하며, 이 때 메쉬 상의 특징점들은 내부적으로 플래너 임베딩을 통해 메쉬의 2차원맵으로 일대일 대응된다. 와핑 직후 텍스처 이미지와 메쉬의 2차원맵 사이의 일대일 대응으로부터 메쉬 상의 모든 점에 대해 텍

스처좌표를 결정할 수 있고, 이로부터 완성된 텍스처링 결과를 얻는다.

2. 기존연구

텍스처매핑과 관련하여 지금까지 많은 연구가 3차원 표면데이터의 글로벌 매개화, 즉 모델 표면전체와 텍스처 평면공간 사이의 일대일 대응을 찾아내는 문제에 초점을 맞추었다(자세한 문헌조사에 대하여는 [3]을 참조). 이러한 글로벌 매개화 방법을 확장하여 제한조건을 가지는 텍스처매핑에 사용할 수 있을 것이라고 생각할 수도 있다. 예를 들어, Desbrun 외[4]에서는 제한조건을 가지는 텍스처매핑을 위하여 매개화 목표함수로부터 제한조건을 포함하는 점들의 자유도를 제거하는 방법을 제안하였다. 그러나, 이와 같은 접근 방법은 일대일 매핑을 보장하지 않기 때문에 제한 조건이 많아질 경우 올바르게 동작하지 않을 수 있다.

몇몇 연구자들은 최소제곱법(least-squares minimization)을 사용하여 제한조건을 근사하게 만족하는 알고리즘을 제안하였다[5]. 이러한 방법들 역시 제한조건의 수가 적은 경우에는 잘 동작하지만, 제한조건의 수가 많아짐에 따라 잘 동작하지 않을 수 있다. 더욱이, 제한조건이 정확하게 만족되어야 하는 경우에는 이러한 근사화 방법을 사용할 수 없다. 이음부분에서 텍스처 이미지의 비연속성을 제거하고 완벽하게 병합된 텍스처 매핑을 얻으려는 경우와 같이 제한조건이 정확하게 만족되어야만 하는 경우가 있다. Gingold 외[6]는 제한조건을 정확하게 만족하는 방법을 제한하였으나, 일대일 매핑 계산을 보장하지는 않는다. 변형에 의한 벤딩에너지를 최소화하는 변형함수 기반 와핑 방법들[7] 역시 일대일 매핑 생성을 보장하지 않아 오버랩이 생길 수 있다.

다른 연구자들은[8-11] 슈타이너점(Steiner vertices)들을 임베드된 메쉬에 삽입함으로써 위치 제한 조건을 정확하게 만족시키는 방법들을 제안하였다. 본 논문에서 제안하는 방법은 이러한 방법들과 달리 단순하며, 임베드된 메쉬를 변경할 필요도 없다. 또한, 기존의 방법들이 위치제한 조건만을 다루는 반면에 본 논문에서 제안하는 방법은 그라디언트 제한조건까지도 허용함으로써 사용자에게 보다 많은 제어를 제공한다.

몇몇 연구자들은 제한 조건을 가진 3차원 물체 표면을 구와 같이 보다 단순한 공간으로 임베딩하는 방법을 개발하였다. Alexa 외[12]는 제한조건을 가지는 구매개화(spherical parameterization)를 계산하는 방법을 제안하였다; 제한조건은 두 개의 표면 사이의 특징점 대응의 형태로 정의되었다. 이 방법 역시 특징점이 너무 많은 경우 동작하지 않을 수 있다. Lin 외[13]는 에지 뒤집기(edge flipping)를 사용하여 이 문제를 해결할 것을 제

안하였다. 그러나, 텍스처매핑 과정에서 물체 표면의 기하 정보를 변경한다는 것이 단점으로 지적될 수 있다.

3. 벡터장 기반 이미지 와핑

주된 아이디어는 이미지의 와핑 계산 시 시간에 따른 벡터장함수를 이용하는 것이다. 사용자가 정한 제한조건을 반영하도록 C1 연속인 속도벡터장 $v(x,y,t)$ 을 생성한 후, 이미지 상의 모든 점 p_{src} 에 대해 현 위치에서의 벡터장을 따라 경로적분을 적용함으로써 와핑 후의 위치를 계산한다(그림 1). 즉, 2차원 상에서 정의된 시간 t 에 따른 경로함수 $p(t)$ 는 다음 식과 같이 정리된다:

$$\begin{cases} \frac{d}{dt} p(t) = v(p(t), t) \text{ for } t \in [0,1] \\ p(0) = p_{src} \end{cases} \quad (1)$$

위의 식 (1)은 직관적으로, 이미지 상의 한 점 p_{src} 가 와핑 과정 중에 시작 위치로부터 시작하여 속도벡터장의 흐름에 따라 이동 경로를 정한다는 것을 의미한다. 당연한 말이지만, 임의의 순간 한 점의 이동속도는 그 점의 위치에서 정의된 벡터장의 값과 일치하게 된다.

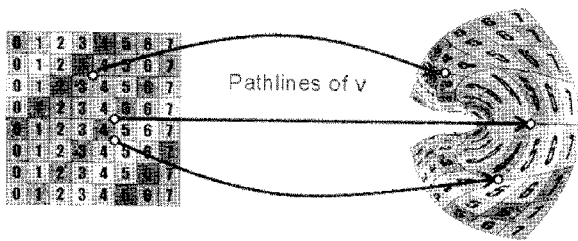


그림 1 벡터장 기반의 이미지 와핑의 한 예: 이미지 상의 점들은 각각의 출발점에서 목표점을 잇는 경로를 따라 이동한다.

이제, 사용자의 제한조건을 반영하도록 하는 속도벡터장의 구축에 대해 기술하겠다. 사용자는 와핑 전과 후의 이미지에 각각 특징점의 위치 P_i 와 Q_i 를 지정하는 방법으로 제한조건을 정의할 수 있다. (실제 애플리케이션에서는 사용자가 UI를 통해 Q_i 를 메쉬 상에 지정하지만, 내부적으로는 메쉬의 2차원 매핑을 통해 2차원 이미지 공간 상의 한 점으로 대응된다.) 따라서, 여기서 고려하는 텍스처 이미지 와핑문제는 다음과 같이 정리될 수 있다: 제한 조건을 만족시키도록, 즉 원래 이미지 상의 특징점들이 사용자가 지정한 위치로 매핑되도록 이미지를 변형하는 것이다. 본 연구의 중심 아이디어는 이미지 상에 정의된 특징점들을 원래 위치 P_i 에서 목적위치인 Q_i 까지 부드럽게 이동하면서(그림 2(a)), 동시에 주변의 다른 점들도 비슷한 속도(방향과 길이)로 이동하는 것이다. 이 때 이미지 상의 각 점의 속도는 위치와 시간에

따른 벡터장함수 $v(x,y,t)$ 로 정의된다(그림 2(b)). 본 보고서에서는 i 번째 특징점의 위치와 속도를 각각 $F_i(t)$ 와 $\frac{dF_i(t)}{dt}$ ($t \in [0,1]$)로 표기하도록 한다. $F_i(0)$ 와 $F_i(1)$ 는 각각 소스 위치 P_i 와 목적 위치 Q_i 를 나타낸다. 우리가 원하는 것은 이 매핑이 일대일 대응 성질을 만족시키는 것이므로, 특징점들의 지정 위치 또한 이 일대일 대응 성질을 만족시켜야 한다. 다시 말해, 사용자는 특징점들을 모두 서로 다른 위치로 지정하여야 하고, 2개 이상의 특징점들을 동시에 같은 위치에 놓는 것은 허용되지 않는다. (모든 $i \neq j$ 에 대하여 $F_i(t) \neq F_j(t)$, $t \in [0,1]$)

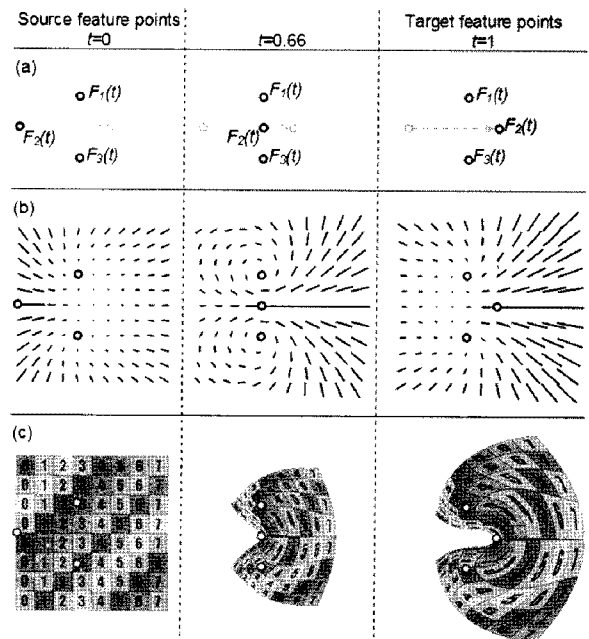


그림 2 벡터장 기반의 와핑: (a) 특징점들 위치의 선형 보간(소스에서 목적까지). (b) 텍스처 점들의 이동 속도를 표현하는 벡터장 계산. (c) 텍스처 이미지의 와핑 결과

3.1 위치 제한조건이 포함된 벡터장

2개 이상의 특징점들의 위치제한조건을 사용하여 이미지의 일부를 스케일하고 회전할 수 있는 기능은 이미지 와핑의 중요한 기능 중의 하나이다. 두 점에 대한 위치 제한조건이 주어지면, 이것으로부터 이미지 와핑에 사용될 2차원 유사도 변환(similarity transformation, 위치이동과 회전만을 포함하는 변환)을 정의할 수 있다. 또한, 2개 이상의 특징점이 지정되었을 때에는 가능한 모든 특징점쌍들에 대해 유사도 변환을 정의한 후, 이들을 블렌딩하여 전체적인 와핑을 계산하도록 한다.

2개의 특징점 i 와 j 가 주어졌다고 할 때, 시간 t 에서

의 이들의 좌표값 $F_i(t)$ 와 $F_j(t)$ 로부터 로컬좌표계를 정의하고, 이 좌표계를 기준으로 이미지 상의 한 점 (x,y) 의 좌표값 $x_{p,i,j}$, $y_{p,i,j}$ 을 다음과 같이 계산할 수 있다 (그림 3).

$$p = F_i(t) + x_{p,i,j} \cdot (F_j(t) - F_i(t)) + y_{p,i,j} \cdot R_{90}(F_j(t) - F_i(t)) \text{ with}$$

$$R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2)$$

이 로컬좌표계는 $F_i(t)$ 와 $F_j(t)$ 가 서로 다른 위치에 있다는 가정 하에 정의됨에 유의한다. 식 (2)의 양변을 시간에 대해 1차 미분하면 특징점쌍 i 와 j 의 속도로부터 $p=(x,y)^T$ 의 속도를 유도하는 데 사용할 수 있다. 즉, 특징점 i 와 j 의 속도로부터 유도되는 점 $p=(x,y)^T$ 의 속도 벡터장 함수 $\mathbf{v}_{ij}(x,y,t)$ 는 다음과 같이 정의된다:

$$\mathbf{v}_{ij}(x,y,t) = \frac{d}{dt} F_i(t) + x_{p,i,j} \cdot \frac{d}{dt} (F_j(t) - F_i(t)) + y_{p,i,j} \cdot R_{90} \frac{d}{dt} (F_j(t) - F_i(t)) \quad (3)$$

이 벡터장은 이미지 상의 모든 점들의 속도를 정의하되, $F_i(t)$ 와 $F_j(t)$ 의 속도로부터 정의되는 유사변환에 따라 움직이도록 하는 효과를 가져온다. 즉, (3)식의 벡터장에 의해 계산된 점 p 의 변형 후 위치는 $F_i(t)$ 와 $F_j(t)$ 에 의해 정의된 로컬좌표계 변하지 않는다(그림 3(a), (b), (c)).

3.1.1 텍스처 매핑에 적합한 성질

본 연구에서 제안하는 와핑 방법은 제한조건을 허용하는 텍스처 매핑에 적합한 몇 가지 성질을 만족시킴을 어렵지 않게 보일 수 있다. 첫째, 벡터장에 기반한 와핑은 부드러운 변형을 만들어낸다. 식 (3)은 x 와 y 에 대하여 2번미분에도 연속이다; 즉, 벡터장이 C^1 연속이라는 의미이다. 둘째, 소스 위치들을 정확하게 목적 위치로 매핑한다. $p=(x,y)^T$ 가 $F_i(t)$ 와 일치한다면, 상대적인 좌표 $x_{p,i,j}$ 와 $y_{p,i,j}$ 는 모두 0이 된다. 식 (3)으로부터 $\mathbf{v}_{ij}(x,y,t) = \frac{d}{dt} F_i(t)$ 이 되어 p 의 속도 또한 $F_i(t)$ 의 속도와 일치하게 되므로, p 는 $F_i(t)$ 에 매핑이 된다. 마찬가지로, p 가 $F_j(t)$ 와 일치한다면, 벡터장은 $\mathbf{v}_{ij}(x,y,t) = \frac{d}{dt} F_j(t)$ 가 되고 p 의 속도는 $F_j(t)$ 의 속도와 일치, $F_j(t)$ 에 매핑된다. 마지막으로, 제안하는 와핑 방법은 identity 속성을 만족시킨다. 특징점 i 와 j 의 소스 위치

와 목적 위치가 일치하면, $\frac{d}{dt} F_i(t)$ 와 $\frac{d}{dt} (F_j(t) - F_i(t))$ 모두 0이 되고 벡터장 또한 0이 되면서 이미지는 변형이 적용되지 않는다.

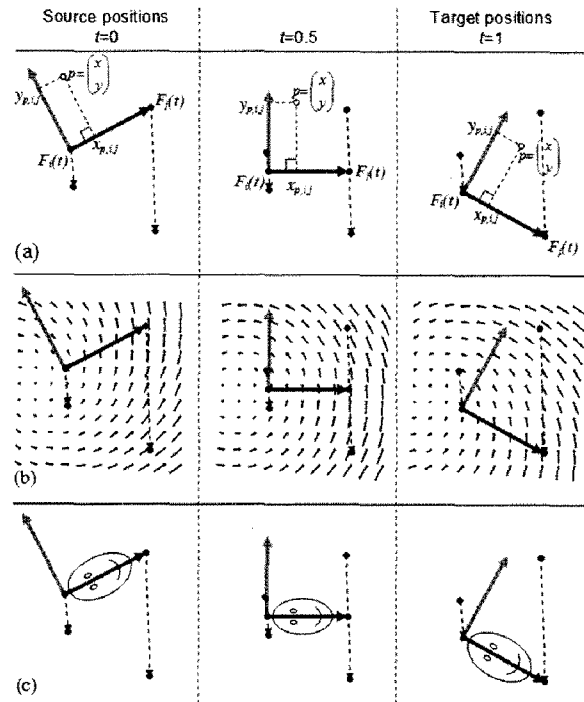


그림 3 (a) $F_i(t)$ 와 $F_j(t)$ 에 의해 정의되는 로컬좌표계. $F_i(t)$ 와 $F_j(t)$ 를 연결하는 선에 수직인 선을 2번째 축으로 사용한다. (b) $F_i(t)$ 와 $F_j(t)$ 의 속도로부터 유도된 속도벡터장의 모습. (c) (b)의 속도 벡터장을 사용한 이미지의 와핑 결과

3.2 그라디언트 제한조건이 포함된 벡터장

본 절에서는 3.1절에서 기술한 위치 제한조건에 더하여 그라디언트 제한조건을 고려하는 와핑방법을 기술한다. 그림 4에 보인 바와 같이, 사용자는 메쉬표면의 임의의 점에서 텍스처의 방향과 크기에 대한 제한 조건을 정의할 수 있다.

그라디언트에 대한 제한조건을 두기 위한 특징점은 앞서 설명한 특징점과 비슷하고, 다만 사용자가 텍스처 이미지의 위치 외에도 방향(orientation)과 스케일을 조절할 수 있다는 점이 다르다. 즉, 특징점 k 는 위치이동

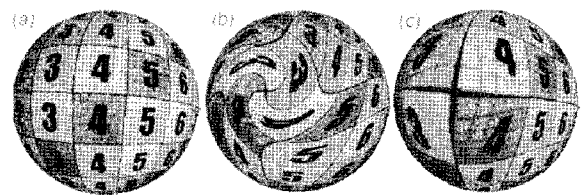


그림 4 그라디언트 제한조건을 포함하는 와핑의 예. (a): 제한조건이 없는 텍스처매핑의 예. (b): 한 점에서 텍스처의 방향에 대한 제한조건을 정의하였을 때 (시계 방향으로 180도 회전) 매핑 결과. (c) 역시 같은 점에서 텍스처의 스케일에 대한 제한조건

외에도 회전, 그리고 균등스케일로 이루어진 변환행렬에 의해 정의된다; $M_k(0)$ 가 텍스처이미지에 대응하는 행렬(소스행렬)이라 하고, $M_k(1)$ 가 대상메쉬의 평면 임베딩에 대응하는 행렬(목적행렬)이라고 하자. 제안하는 방법은 소스행렬과 목적행렬 사이를 보간한 $M_k(t)$ 를 계산한 후 이 로컬좌표계 안에서 이미지 상의 한 점 p 의 위치가 변하지 않도록 p 의 속도를 매 시간스텝마다 계산하는 것이다(그림 5).

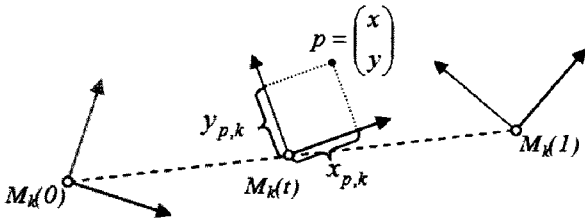


그림 5 $M_k(t)$ 에 의해 정의된 로컬좌표계에서의 p

우선 행렬 $M_k(0)^{-1}M_k(1)$ 을 극분해 방법(polar decomposition)을 사용하여 위치이동, 회전, 스케일링 변환행렬로 분해한다[14]:

$$M_k(0)^{-1}M_k(1) = \begin{pmatrix} 0 & 0 & T_x \\ 0 & 0 & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$M_k(t)$ 는 위 행렬의 각 성분을 보간함으로써 얻는다:

$$M_k(t) = M_k(0) \cdot T(t)R(t)S(t),$$

$$T(t) = \begin{pmatrix} 0 & 0 & tT_x \\ 0 & 0 & tT_y \\ 0 & 0 & 1 \end{pmatrix}, \quad R(t) = \begin{pmatrix} \cos(t\phi) & -\sin(t\phi) & 0 \\ \sin(t\phi) & \cos(t\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$S(t) = \begin{pmatrix} S' & 0 & 0 \\ 0 & S' & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

특징점 k 에 대응하는 벡터장은 다음과 같이 정의된다:

$$\mathbf{v}_k(x, y, t) = \frac{d}{dt} \left(M_k(t) \cdot \begin{pmatrix} x_{p,k} \\ y_{p,k} \\ 1 \end{pmatrix} \right) \quad (4)$$

위 식에서 $(x_{p,k}, y_{p,k})^T$ 는 로컬좌표계 $M_k(t)$ 에서의 점 $p=(x,y)^T$ 의 위치를 의미한다.

3.2.1 텍스처 매핑에 적합한 성질

식 (3)의 벡터장(위치제한조건과 연관됨)과 유사하게, 식 (4)에서의 벡터장(그라디언트 제한조건과 연관됨)에 기반한 와핑이 텍스처 매핑에 적합한 성질들을 만족하는지 확인해 보겠다. 우선, 부드러운 변형을 만들어내는 것은 쉽게 알 수 있다. 식 (4)는 x 와 y 에 대해 2번 미분 가능하고 연속이므로, 벡터장은 C^1 연속이다. 다음으로, 제안하는 와핑방법은 소스위치를 목적 위치로 보낸다.

$p=(x,y)^T$ 가 특징점 k 와 동일한 위치에 있다고 하면, 식 (4)로부터 $x_{p,k}$ 와 $y_{p,k}$ 는 0이 되고, 벡터장은 $M_k(t)$ 의 위치이동 성분과 동일하게 된다. 즉, p 는 특징점 k 의 목적 위치로 매핑된다. 마지막으로, 와핑방법은 identity 성질을 만족한다. 소스 행렬 $M_k(0)$ 과 목적행렬 $M_k(1)$ 이 동일하다면, $M_k(t)$ 는 t 에 대하여 일정한 상수(constant)가 되고, 벡터장 $\mathbf{v}_k(x,y,t)$ 는 0이 된다. 따라서, 이미지는 변형되지 않는다.

3.3 여러 벡터장의 결합

일반적으로는 텍스처 이미지에 2개 이상의 특징점들이 존재하므로, 이미지 상의 한 점의 위치는 복수 개의 특징점들에 의해 영향을 받게 된다. 점 $p=(x,y)$ 의 최종 속도는 따라서, 자신에게 영향을 주는 특징점들의 속도들의 가중치 결합으로 결정된다. 점 p 에 대한 한 쌍의 특징점 i 와 j 에 대해 가중치를 다음과 같이 정의한다:

$$\beta_{p,i,j} = \frac{1}{(d_{p,i} \cdot d_{p,j})^\alpha}$$

즉, p 가 두 특징점 중 어느 하나에라도 가까워짐에 따라 특징점 쌍의 영향력은 증가하도록 하였다. $d_{p,i}$ 와 $d_{p,j}$ 는 각각 p 와 특징점 i , 그리고 p 와 특징점 j 와의 거리를 나타낸다. 그라디언트 제한조건에 대한 가중치는 다음과 같이 정의한다:

$$\beta_{p,k} = \frac{1}{d_{p,k}^\alpha}$$

위 식에서 $d_{p,k}$ 는 $p=(x,y)^T$ 와 특징점 k 사이의 거리이다. 지수 α 는 사용자가 설정할 수 있으며 특징점이 텍스처 와핑을 얼마나 완만하게, 혹은 급격하게 유도할지를 제어하는 데 사용된다. 그림 6에 보였듯이, 동일한 특징점 제한조건에 대하여 α 가 증가할수록 이미지 전체의 와핑 정도가 크고 그 왜곡의 정도도 증가함을 알 수 있다.

이들 가중치는 각 점 $p=(x,y)^T$ 에서의 합이 1이 되도록 정규화하였다. 또한, p 가 특징점 l 과 동일한 위치에 놓여 있는 경우 ($d=0$)이 특징점과 연관된 벡터장의 계수는 1이 되며, 다른 모든 벡터장의 계수는 0이다. 이는 p 의 속도는 특징점 l 과 관련된 벡터장에 의해서만 결정됨을 의미한다.

n_{pos} 개의 위치 제한조건과 n_{grad} 개의 그라디언트 제한조건이 주어졌을 때, 전체 벡터장은 다음과 같이 표현할 수 있다:

$$\mathbf{v}(x, y, t) = \sum_{i,j=1 \text{ with } i \neq j}^{n_{pos}} (\beta_{p,i,j} \mathbf{v}_{i,j}(x, y, t)) + \sum_{k=1}^{n_{grad}} (\beta_{p,k} \mathbf{v}_k(x, y, t)) \quad (5)$$

p_{src} 의 와핑 후 위치, 즉 목적 위치는 $t=1$, $p(0) = p_{src}$ 를 (1)식에 대입하여 풀면 구할 수 있다. 본 연구에서는

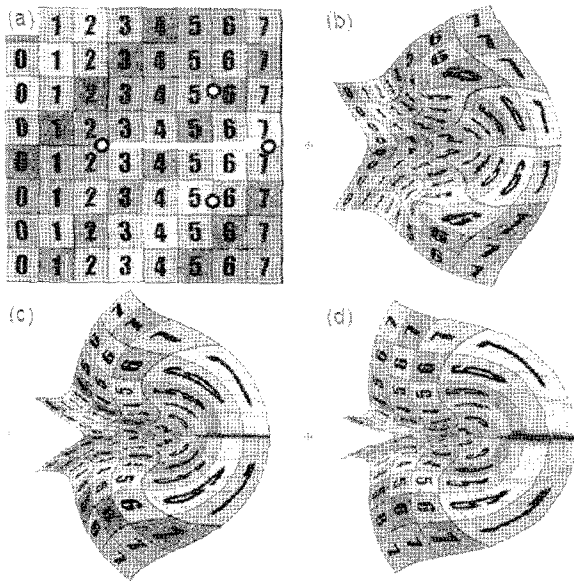


그림 6 소스 이미지(a)에 서로 다른 α 값을 적용한 와핑 결과. (b) $\alpha=1.0$ (c) $\alpha=2.0$ (d) and $\alpha=3.0$

([1]의 연구에서와 유사하게) 가변적인 스텝사이즈를 사용하는 4차원 Runge-Kutta 방법을 사용하여 구현하였다[14]. 한 적분(integration) 스텝은 각 점에서 속도장 벡터값으로부터의 위치계산($p(t)$)(1)식), 그리고 각각의 특징점쌍에 의해 정의된 로컬좌표계에서의 좌표값과 이를 이용한 가중치와 $v_{i,j}$ and v_k 의 업데이트(식 (3)(4)), 그리고 이들을 (5)식에 대입하여 다음 시간 스텝에서의 속도벡터장 계산을 포함한다. 이 때, 경로적분을 텍스처 이미지의 각 점에 대해 독립적으로 이루어지므로, 속도 향상 등을 위해 각각의 점에 대한 경로적분을 여러 쓰레드(thread)에 나누어 할당함으로써 병렬화하였다.

4. 구현 결과

사용자가 특징점들을 인터랙티브하게 추가, 제거할 수 있도록 하는 사용자 인터페이스를 함께 구현하였다. 전체적인 텍스처매핑은 다음의 단계로 구성된다: (1) 사용자가 삼각매쉬 위에 단순한 폐곡선을 그려 텍스처가 입혀질 영역을 선택한다. (2) 선택된 영역은 최소제곱 등각매핑(Least Squares Conformal Map) 방법[2]을 사용하여 2차원 평면에 임베딩된다. (3) 사용자가 특징점들을 매쉬와 텍스처 이미지 상에 지정하는 작업을 진행하는 동안 제한조건을 포함하는 텍스처매핑이 계산되고, 그 결과를 화면에 가시화한다.

그림 7에 제한된 방법을 다양한 복잡도를 가진 예제에 적용한 결과를 보였다. 삼각매쉬의 복잡도는 삼각형 5천~10만 개, 그리고 특징점의 개수는 60~80여 개로 다양하다. 가능한 모든 특징점쌍들에 대해 유사도변환을

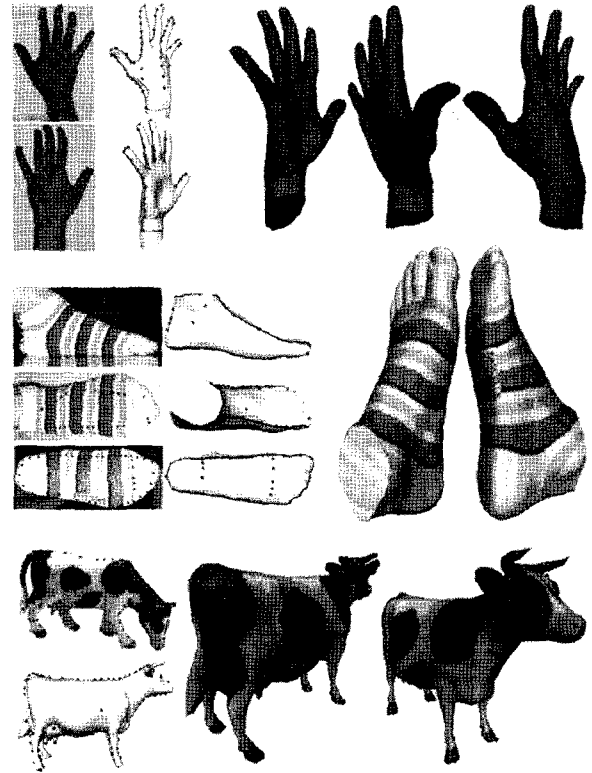


그림 7 다양한 복잡도의 모델들을 대상으로 제안하는 텍스처매핑 방법을 적용시킨 결과. 매쉬를 구성하는 삼각형의 개수와 매쉬상의 특징점의 개수는 각각 약 20000과 65(손모델), 약 5000과 73(발모델), 약 10만과 81(소모델)이다. 소모델의 경우 이미지 한 장을 매쉬의 좌우에 매핑하였다.

정의하여 블렌딩한 결과로 이미지와핑을 정의하므로 소요되는 계산시간은 특징점쌍의 갯수, 즉 특징점 개수의 제곱에 비례하게 된다. 특징점이 80여개 사용된 소모델의 경우 매핑시간은 인텔 펜티엄 2.4GHz PC에서 약 4초 정도가 소요되었다.

4.1 기존 텍스처매핑 방법과의 비교

제안된 알고리즘을 다른 제한조건을 허용하는 최근에 발표된 텍스처매핑 방법들과 비교해 보았다[9-11]. 기존의 알고리즘들과 비교했을 때 제안한 방법은 몇 가지 면에서 중요한 차이점들이 있다: 첫째, 제안한 방법은 보다 부드러운 텍스처 매핑결과를 생성한다. 이러한 결과는 삼각형의 개수가 상대적으로 적고 많은 양의 왜곡을 요구로 하는 제한조건에 대해 더욱 두드러진다. 이는 제안한 방법이 C^1 연속성에 기반한 와핑에 근거하기 때문이다. 그림 8은 동일한 모델에 대하여 Lee 외[11]의 방법과 본 연구에서 제안하는 방법을 각각 사용해 얻은 결과를 보인 것이다. 후자의 경우가 더욱 부드러운 와핑 이미지를 생성한 것을 볼 수 있다.

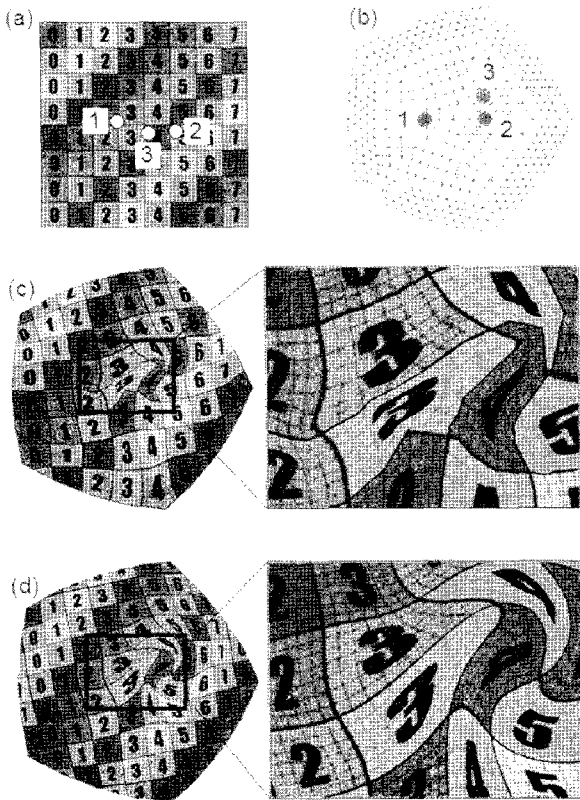


그림 8 Lee 외[11]의 방법과 제한하는 방법의 비교: (a) 입력 텍스처, (b) 입력메쉬, (c) [11]의 논문에서 발췌한 텍스처매핑 결과 (d) 제안하는 방법에 의한 텍스처매핑 결과

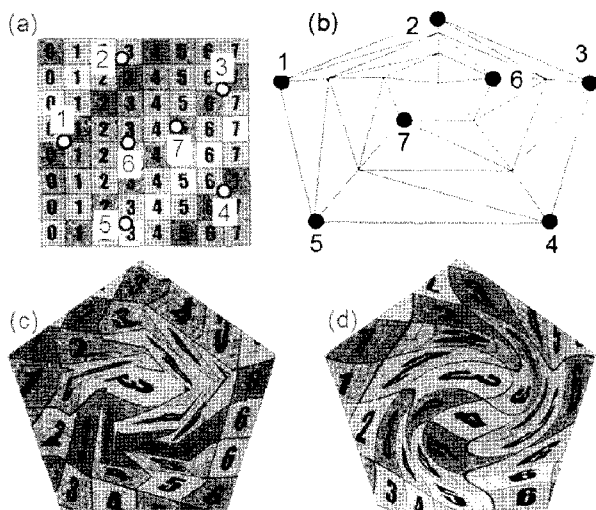


그림 9 Lee 외[11]의 방법과 제한하는 방법의 비교: (a) 입력 텍스처, (b) 입력메쉬, (c) [11]의 논문에서 발췌한 텍스처매핑 결과 (d) 제안하는 방법에 의한 텍스처매핑 결과

그림 9에는 또 다른 예를 보였다. 본 연구에서 제안하는 방법의 부드러운 와핑 결과가 더욱 두드러지게 나타

난 결과이다.

둘째, 제안하는 알고리즘은 구현이 용이하며, 기존의 방법들[8,9]과는 달리 겹침(foldover) 현상을 방지하기 위해 슈타이너점(Steiner vertices)을 삽입할 필요가 없다. 마지막으로, 기존의 방법들[9-11]이 위치 제한조건만을 허용하는데 반하여, 제안하는 방법은 그라디언트 제한조건을 허용함으로써 사용자가 텍스처의 방향과 스케일을 조정할 수 있다.

5. 결론

본 연구에서는 최근 관련 학계의 관심을 받고 있는 벡터장 기반의 변형 기술을 도입하여 2차원 텍스처매핑에 유용하도록 왜곡이 많이 생기는 요구조건을 만족하면서도 오버랩이 생기지 않는 이미지 와핑 방법을 개발하였다. 텍스처 이미지 상의 픽셀들이 사용자에게 주어진 제한조건을 만족하기 위해 주어진 출발점에서 목적점까지 도달하는 과정을 단일 프로세스로 보는 기존의 방법들과는 달리, 시간에 따라 연속적으로 변하는 속도벡터장을 구축하고 그것에 의해 각 픽셀들의 점진적인 위치 변화를 반복적으로 계산하였다. 그 결과 기존의 텍스처매핑 방법보다 더욱 부드러운 이미지 와핑결과를 얻을 수 있었으며, 왜곡이 많이 생기는 요구조건을 만족하면서도 오버랩이 생기지 않는 좋은 성질을 보임을 확인하였다. 기존의 이미지 와핑 방법들이 제한조건에 따라 겹침(foldover)현상을 피할 수 없었던 것과는 대조적으로 제안한 방법은 겹침(foldover)이 일어나지 않으며, 이는 벡터장 기반 변형 기술의 강점에 기인한다.

참고 문헌

- [1] Von Funck W., Theisel H., Seidel H.-P., "Vector field based shape deformations," *ACM Transactions on Graphics*, vol.25, no.3, pp.1118-1125 (2006).
- [2] Levy, B., Petitjean, S., Ray, N., and Maillot, J.-L. "Least squares conformal maps for automatic texture atlas generation," *In Proc. of SIGGRAPH 2002*, pp.362-371.
- [3] Sheffer A., Honmann K., Levy B., Desbrun M., Zhou K., "Mesh Parameterization: Theory and Practice," *Siggraph Course*, 2007.
- [4] Desbrun, M., Meyer, M., and Alliez, P. 2002. "Intrinsic parameterizations of surface meshes," *In Proc. of Eurographics 2002*, pp.209-218.
- [5] Levy, B., "Constrained texture mapping for polygonal meshes," *In Proc. of SIGGRAPH 2001*, pp.417-424.
- [6] Gingold, Y. I., Davidson, P. L., Han, J. Y., Zorin, D., "A Direct Texture Placement and Editing Interface," *In Proc. of UIST 2006*, Montreux, Switzerland, October 2006.

- [7] Schaefer S., McPhail T., and Warren J., "Image Deformation Using Moving Least Squares," *In Proc. of ACM SIGGRAPH 2006*, pp.533-540.
- [8] Eckstein, I., Surazhsky, V., and Gotsman, C. "Texture mapping with hard constraints," *Computer Graphics Forum*, vol.20, no.3, 2001.
- [9] Kraevoy, V., Sheffer, A., and Gotsman, C., "Matchmaker: constructing constrained texture maps," *In Proc. of SIGGRAPH 2003*, pp.326-333.
- [10] Zhou, K., Wang, X., Tong, Y., Desbrun, M., Guo, B., and Shum, H.-Y. 2005. "TextureMontage: Seamless Texturing of Arbitrary Surfaces From Multiple Images," *In Proc. of SIGGRAPH 2005*, pp.1148-1155.
- [11] Lee, T.-Y., Yen, S.-W., Yeh, I.-C., "Texture Mapping with Hard Constraints Using Warping Scheme," *IEEE Transactions on Visualization and Computer Graphics*, vol.14, no.2, March-April 2008, pp.382-395.
- [12] Alexa, M., Cohen-Or, D., and Levin, D., 2000. "As rigid-as-possible shape interpolation," *In Proc. of ACM SIGGRAPH 2000*, ACM Press/Addison-Wesley Publishing Co., pp.157-164.
- [13] Lin C.-H., Lee T.-Y., Chu H.-K., and Yao Z.-Y., "Progressive Mesh Metamorphosis," *Journal of Computer Animation and Virtual Worlds*, vol.16, no. -4, pp.487-498, 2005.
- [14] Press W., Flannery B. P., Teukolsky S., Vetterling W. T., *Numerical Recipes in C*, Cambridge University Press, 1992.



서혜원

1996년 KAIST 전산학과 학사. 1998년 KAIST 전산학과 석사. 1999년 MIRALab, University of Geneva 교환연구원. 2004년 스위스 University of Geneva 전산학 박사. 2004년~현재 충남대학교 전기정보통신공학부 조교수. 연구분야는 물리

기반 시각 시뮬레이션, 인간과 컴퓨터 상호작용, 가상 현실 등



코디에 프레데릭

프랑스 University of Claude-Bernard Lyon I 전산학 학사(1996년) 및 석사(1998년). 2004년 스위스 University of Geneva 전산학 박사. 2004년~2005년 KAIST 컴퓨터그래픽스 연구실 Post-doc 연구원. 2005년~2007년 KAIST 문

화기술대학원 초빙교수. 2007년~현재 프랑스 University of Haut Alace 조교수. 관심분야는 실시간 옷감 시뮬레이션, 스케치 입력 기반 모델링 및 텍스처매핑 등