

큐브맵을 사용한 구면 영상에서의 평면 텍스처 대치

박정현, 박종승
인천대학교 컴퓨터공학부
{wjdgus2951, jong}@inu.ac.kr

Planar Texture Replacement in Spherical Images using Cubemap

Jeong-Hyeon Park, Jong-Seung Park
Dept. of Computer Science & Engineering, Incheon National University

요 약

구면 파노라마 영상에서는 심한 구면 왜곡으로 인하여 기존의 평면 패턴의 탐색 방법인 SURF가 올바르게 작동되지 않는다. 평면 패턴은 구면 영상에서 왜곡되어 나타나므로 구면 파노라마 영상에서의 패턴 탐색 및 다른 텍스처로의 대치는 평면 영상의 경우와 다르게 처리되어야 한다. 본 논문에서는 구면 파노라마 영상을 큐브맵 영상으로 변환하고 구면 왜곡이 제거된 평면 영상에서 SURF를 사용하여 패턴을 탐색하고 해당 평면 텍스처를 대치한 후 다시 구면 파노라마 영상으로 변환하는 구면 파노라마 영상에서의 평면 대치 방법을 제안한다.

ABSTRACT

In spherical panoramic images, SURF, a feature description method for planar patterns, does not work correctly due to heavy spherical distortion. Since a plane pattern is distorted in a spherical image, the pattern search and replacement in a spherical panoramic image should be treated differently from the case of the planar image. This paper proposes a planar texture replacement method, which transforms a spherical panoramic image into a cubemap panoramic image, searches a pattern using SURF, replaces a plane pattern, and then converts it into a spherical panoramic image.

Keywords : Spherical panoramic image(구면 파노라마 영상), Pattern matching(패턴 매칭), Texture replacement(텍스처 대치), Cubemap(큐브맵)

Received: Nov. 11. 2017 Revised: Dec. 15. 2017
Accepted: Dec. 20. 2017
Corresponding Author: Jong-Seung Park(Incheon National University)
E-mail: jong@inu.ac.kr
ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

최근 증강현실 및 가상현실 콘텐츠에 대한 관심이 커지고 있으며 특히 구면 파노라마 영상의 사용이 점점 많아지고 있다. 구면 파노라마 영상은 증강현실과 가상현실에서 효과적으로 사용되는 영상으로써, 모든 방향의 장면을 한 영상으로 표현한다. 구면 파노라마 영상 획득에 관한 연구로 Fangi 등[1]은 여러 개의 평면 영상 또는 카메라를 이용해 구면 파노라마 영상을 생성하는 원리와 공식을 제안하였다. 최근에는 편리한 영상 취득을 위한 장비들이 상용화되어서 구면 파노라마 영상의 획득이 용이하게 되었다.

평면 영상에서의 평면 패턴을 매칭하는 방법에는 여러 가지가 존재한다. 관련 연구로 Lowe는 크기와 회전에 불변하는 특징을 추출하여 이를 이용해 평면 패턴을 매칭하는 SIFT를 제안하였다[2]. Bay는 SIFT 만큼의 성능을 유지하며 속도는 향상시킨 SURF를 제안하였다[3].

구면 파노라마 영상에서는 패턴이 왜곡되어 나타나므로 평면 영상에서 사용되는 알고리즘을 통한 패턴 매칭은 정확성이 매우 떨어진다. 김보성 등[4,5]은 평면 패턴의 기하 변환을 추정하여 구면 파노라마 영상에서 평면 패턴을 매칭하는 방법을 제안하였다. 김동룡 등[6]은 기존의 SURF 알고리즘을 통하여 패턴의 위치를 대략적으로 찾고 왜곡을 보정하여 패턴을 매칭 및 대치하는 방법을 제안하였다. Mota 등[7]은 구면 파노라마 영상의 특징점을 두 가지 형태로 추출하여 패턴 매칭하는 방법을 제안하였다. Liu 등[8]은 기존의 SIFT 알고리즘을 이용하여 구면 파노라마 영상에서의 패턴 매칭을 제안하였다. Tong 등[9]은 구면 파노라마 영상을 기반으로 하는 기술자 매칭 알고리즘을 제안하였다. 또한 Tong 등[10]은 이러한 매칭 기술들을 이용하여 구면 파노라마 영상에서의 물체의 위치를 찾는 방법을 제안하였다.

구면 파노라마 영상에서의 왜곡 보정에 대한 연구는 계속 진행되고 있다. Guan 등[11]은 구면 파

노라마 영상에서의 왜곡을 보정하는 새로운 렌더링 방법을 제안하였다. Grimm 등[12]은 구면 파노라마 영상과 큐브맵 영상과의 관계를 설명하였다. Kangni 등[13]은 큐브맵 영상에 대한 연구를 진행하였다. Zhou 등[14]은 구면 파노라마 영상을 큐브맵 영상으로 변환하여 물체를 탐색하는 방법을 제안하였다. 하지만 Zhou가 제안한 방법은 물체가 두 개의 면에서 나올 경우 탐색이 되지 않는 한계점을 가지고 있다.

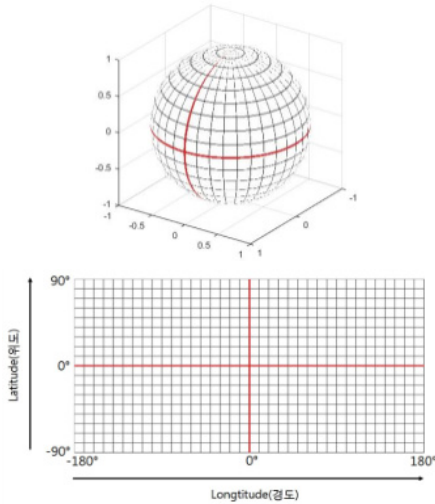
본 논문에서는 구면 파노라마 영상을 큐브맵으로 변환하여 패턴을 탐색하고 텍스처를 대치하는 방법을 제안한다. 패턴이 큐브맵의 경계에 걸치는 경우를 해결하기 위하여 원래의 구면 파노라마 영상을 회전 변환하여 회전된 구면 파노라마 영상을 생성하고 패턴 탐색을 중복 실시한다. 제안 방법은 큐브맵을 활용하여 평면 영상에서 사용되는 패턴 탐색 및 텍스처 대치 기법을 구면 영상에서도 사용될 수 있도록 하는 효과적인 방법이다.

2. 구면 파노라마 영상에서의 패턴 매칭

일반 평면 영상에서의 패턴 매칭은 여러 가지 방법이 있다. 대표적인 패턴 매칭 방법으로 패턴의 크기와 회전에 불변하는 특징들을 추출하여 패턴을 매칭하는 SIFT 방법이 있고 SIFT의 느린 속도를 보완한 SURF 방법이 있다. 본 논문에서의 패턴 매칭은 빠른 속도의 패턴 매칭 구현이 가능한 SURF 방법을 사용한다.

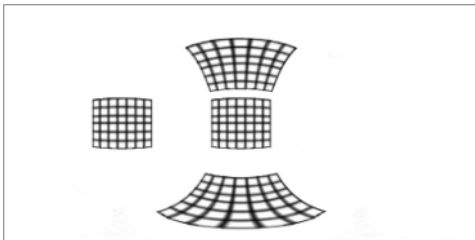
전방위 영상을 하나의 영상으로 표현한 구면 파노라마 영상은 특별한 좌표계로 표현될 수 있다. [Fig. 1]과 같이 구면 파노라마 영상은 구의 중심에 카메라가 있다고 생각했을 때 카메라가 촬영한 전방위 영상을 펼쳐서 하나의 평면 영상으로 표현한 영상이다. 이 때 구면 파노라마 영상에서의 가로 및 세로는 경도와 위도에 해당하며 경도의 범위는 $[-180^\circ, 180^\circ]$ 이며 위도의 범위는 $[-90^\circ, 90^\circ]$ 이다. 이처럼 구면 파노라마 영상은 수평방향으로

360°를 표현하고 수직방향으로 180°을 표현하기 때문에 영상의 크기는 너비와 높이의 비가 2:1로 나타난다.



[Fig. 1] A coordinate system for spherical image

평면 영상과 다른 특징을 가지고 있는 구면 파노라마 영상에서는 구면 왜곡이 존재한다. [Fig. 2]는 구면 파노라마 영상에서 패턴의 위치에 따른 왜곡을 나타낸 그림이다. 이와 같이 왜곡의 정도는 적도 부근에서는 심하지 않으며 위도의 변화가 심할수록 왜곡이 심해져서 패턴의 모양 변화가 눈에 띄게 달라진다.

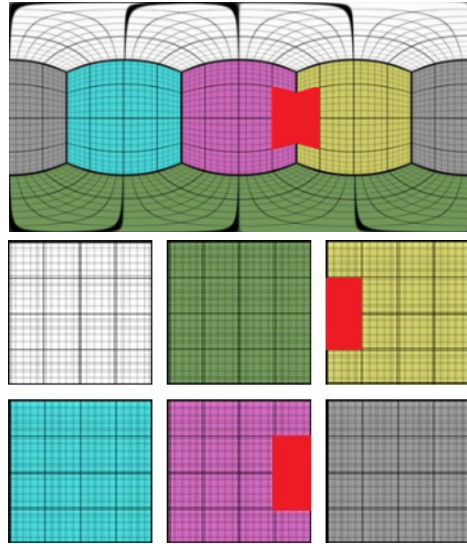


[Fig. 2] Pattern distortion in spherical image

구면 파노라마 영상에서 패턴이 왜곡이 심하지 않은 부분에 위치할 경우 SURF를 통한 패턴 매칭이 어느 정도 가능하지만 찾으려는 패턴의 모양은 왜곡이 존재하지 않는 평면 패턴이므로 위치에 따

라 왜곡의 형태가 다양한 구면 파노라마 영상에서는 SURF를 통한 패턴 매칭을 적용하기는 어렵다.

구면 파노라마 영상의 왜곡 특성으로 평면 영상에서 쓰이는 영상 처리 기술의 직접 사용은 어렵다. 한편 구면 파노라마 영상을 구면에 배치하고 구면의 중심에서 정면, 후면, 좌측면, 우측면, 상단, 하단으로 나누어 표현한 큐브맵 영상은 구면 왜곡이 존재하지 않는다. 본 논문에서는 구면 파노라마 영상에서의 패턴 매칭을 위하여 구면 파노라마 영상을 큐브맵 영상으로 변환하여 왜곡이 제거된 큐브맵 영상의 각 면마다 SURF를 실행하여 패턴을 탐색한다.



[Fig. 3] A case that makes searching difficult

왜곡이 제거된 큐브맵 영상으로 변환한 후에 각 면에서 패턴을 탐색하는 접근에도 문제점이 존재한다. 한 패턴이 큐브맵의 여러 면에 걸쳐서 나타나는 경우에는 패턴 탐색에 어려움이 생긴다. [Fig. 3]은 찾고자 하는 패턴이 큐브맵 영상에서 6개의 면 중에서 2개의 면에서 걸쳐서 나타날 경우를 보여준다. 이런 경우에는 각각의 면에서의 패턴 매칭으로 대상 패턴을 찾기가 어려워진다. 본 논문에서는 이러한 문제점을 구면 파노라마 영상의 회전 변환으로 해결하는 방법을 제안한다. 3절에서 제안

방법에 대해 자세히 설명한다.

3. 제안하는 텍스처 대치 방법

제안하는 구면 영상에서의 평면 패턴 탐색 및 텍스처 대치 방법을 설명한다. 먼저 평면 영상에서 쓰이는 패턴 매칭 알고리즘을 사용하기 위하여 구면 파노라마 영상을 큐브맵 영상으로 변환한다. 큐브맵 영상으로 변환한 후 패턴이 큐브맵의 두 개의 면에 걸쳐서 나타나는 경우를 해결하기 위해 제안 방법은 구면 파노라마 영상을 회전 변환하여 추가적인 구면 파노라마 영상을 생성한다. 찾고자 하는 평면 패턴의 크기가 큐브맵의 각 면의 영상의 크기의 1/4보다 작다는 가정을 한다면 3가지의 회전 변환으로 3개의 추가적인 구면 파노라마 영상들을 생성한다면 평면 패턴이 두 개의 큐브맵 면들에 걸쳐서 나타나는 경우를 해결할 수 있다. 회전 변환은 총 3번을 진행하며 경도 45°, 위도 45°, 경도 및 위도 45°만큼 회전 변환한 3개의 구면 파노라마 영상을 추가로 만들고, 원본 구면 파노라마 영상까지 총 4개의 구면 파노라마 영상을 큐브맵 영상으로 변환한다.

Algorithm ReplacePattern

```

Input : pano_in, patt_src, patt_dst
Output : pano_out
Begin
    faces ← getCubemapFaces(pano_in)
    faceIndex ← searchPattern(faces, patt_src)
    replacePattern(faces[faceIndex], patt_dst)
    cubemap_out ← mergeToCubemap(faceIndex)
    pano_out ← cubemapToSphpano(cubemap_out)
End
    
```

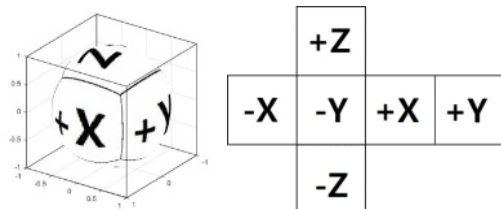
[Fig. 4] Pseudo-code of the proposed method

두 번째 단계로 각 큐브맵 영상에서 Top, Bottom, Front, Back, Right, Left의 6개 면의 영상으로 분할하여 총 24개의 평면 영상을 얻는다. 마지막으로 각각의 면 영상에서 SURF를 실행하여

총 24개의 패턴 매칭 결과 값을 얻는다. 가장 좋은 매칭 결과를 보이는 영상에서 텍스처 대치를 실행하고 대치가 완료된 큐브맵 영상을 구면 파노라마 영상으로 바꾸어서 최종적인 텍스처가 대치된 구면 파노라마 영상을 얻는다. 본 논문에서 제안하는 방법의 전체적인 알고리즘은 [Fig. 4]과 같다. 제안하는 알고리즘은 입력 영상에서 24개의 평면 영상을 얻고 각각의 24개의 영상에서 평면 패턴을 탐색 및 비교하여 대치시킨 후 다시 구면 파노라마 영상으로 바꾸는 과정으로 진행된다.

3.1 큐브맵 영상으로 변환

평면 영상에서 쓰이는 기존의 영상 처리 기술들은 데카르트 좌표계(Cartesian coordinate)를 기반으로 동작하기 때문에 평면 영상 처리 알고리즘을 구면 파노라마 영상에서 사용하려면 구면 파노라마 영상의 변환 과정이 필요하다.



[Fig. 5] Cubemap image description

Algorithm GetCubemapCoordinate

```

Input : x, y, face, edge
Output : pt
Begin
    a ← 2 · x/edge, b ← 2 · y/edge
    if face=0 then /*Back*/ pt ← (-1, 1-a, 3-b)
    else if face=1 then /*Left*/ pt ← (a-3, -1, 3-b)
    else if face=2 then /*Front*/ pt ← (1, a-5, 3-b)
    else if face=3 then /*Right*/ pt ← (7-a, 1, 3-b)
    else if face=4 then /*Top*/ pt ← (a-3, 1-b, 1)
    else if face=5 then /*Bottom*/ pt ← (a-3, b-5, -1)
End
    
```

[Fig. 6] Pseudo-code to get cubemap coordinates

큐브맵 영상은 [Fig. 5]의 왼쪽 그림과 같이 구

면 파노라마 영상을 구면에 배치한 후 정육면체 6개의 면에 투영한 영상이다. 큐브맵 영상의 형식은 여러 가지가 있으나 본 논문에서는 [Fig. 5]의 그림과 같은 형식을 사용한다. 큐브맵 영상에서의 좌표로부터 큐브맵 좌표(Cubemap coordinate)를 얻는 알고리즘은 [Fig. 6]와 같다.

Algorithm *SphpanoToCubemap*

```

Input : sphpano
Output : cubemap
Begin
    outWidth ← sphpano.width
    outHeight ← 3/4 · sphpano.width
    edge ← sphpano.width
    for (x←0; x<outWidth; x++)
        face ← x/edge
        if face=1 then /*Left*/
            startIndex ← 0, range ← 3 · edge
        else startIndex ← edge, range ← 2 · edge
        for (y←startIndex; y<range; y++)
            if y<edge then /*Top*/ face←-4
            else if y ≥ 2 · edge then /*Bottom*/ face←-5
            pt ← GetCubemapCoordinate(x,y,face,edge)
            θ ← atan2(pt.y,pt.x)
            φ ← atan2(pt.z, √(pt.x2+pt.y2))
            u ← 2×edge×(θ+π)/π
            v ← 2×edge×(π/2-φ)/π
            cubemap(x,y) ← sphpano(u,v)
        endfor
    endfor
End
    
```

[Fig. 7] Pseudo-code to convert spherical image to cubemap

구면 파노라마 영상을 큐브맵 영상으로 변환하는 과정은 세 단계를 거쳐서 진행된다. 첫 번째로 큐브맵 영상에서의 좌표가 큐브맵에서 Top, Bottom, Front, Back, Right, Left 중 어떤 면에 위치했는지 찾고 이에 따른 큐브맵 좌표를 얻는다. 두 번째로 (x,y,z)로 표현되는 큐브맵 좌표를 (r,θ,φ)로 표현되는 구면 좌표계로 변환한다. 마지막으로 구면 좌표계를 구면 파노라마 영상의 좌표계로 변환하고 입력된 구면 파노라마 영상에서 해당하는

좌표의 정보를 큐브맵 영상에서의 좌표에 저장한다. 구면 파노라마 영상과 큐브맵 영상 간의 변환 알고리즘은 [Fig. 7]과 같다.

3.2 구면 파노라마 영상의 회전 변환

구면 파노라마 영상을 회전 변환하는 과정에는 좌표계의 변환이 필요하고 그 변환 과정은 다음과 같다. 첫 번째, 구면 파노라마 영상 좌표계에서 구면 좌표계(Spherical coordinate)로 변환해준다. 구면 파노라마 영상의 너비가 W, 높이가 H일 때 구면 파노라마 영상을 구면 좌표계로 변환하는 식은 (1)과 같다.

$$\phi = \frac{x - W/2}{W} \times 2\pi, \theta = \frac{-1 \times (y - H/2)}{H} \times \pi \quad (1)$$

두 번째, 구면 좌표계를 3차원 직교 좌표계로 변환한다. 구면 좌표계에서 연산의 편의상 구의 반지름을 1이라 할 때, 직교 좌표계로 변환하는 식은 (2)와 같다.

$$x = \cos\theta \times \cos\phi, y = \cos\theta \times \sin\phi, z = \sin\theta \quad (2)$$

세 번째, 직교 좌표계에서 회전 변환을 한다. z축 중심 회전은 Yaw 회전이라고 하며, y축 중심 회전을 Pitch 회전이라고 한다. Yaw와 Pitch의 회전 변환 행렬은 회전 변환 인자를 각각 α와 β라 할 때, 식 (3)과 같다.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

네 번째, 회전 변환이 완료된 직교 좌표계를 구면 좌표계로 변환한다. 직교 좌표계와 구면 좌표계의 변환 식에서는 삼각함수 atan는 범위가 -π/2부터 π/2까지 이므로 -π부터 π까지인 atan2 함수를 이용해서 변환을 구현한다. 직교 좌표계에서 구면 좌표계로의 변환식은 (4)와 같다.

$$\phi = \text{atan2}(y,x), \theta = \text{atan2}(z, \sqrt{x^2+y^2}) \quad (4)$$

마지막으로, 구면 좌표계에서 구면 파노라마 영

상 좌표계로 변환한다. 구면 파노라마 영상의 너비를 W 라 하고 높이를 H 라 할 때, 변환식은 (5)와 같다.

$$x = \frac{\phi + \pi}{2\pi} \times W, y = \frac{-\theta + \pi/2}{\pi} \times H \quad (5)$$

Algorithm RotateSphpano

Input : P_{input}, θ, ϕ

Output : P_{output}

Begin

$P_{sphere} \leftarrow sphpanoToSphere(P_{input})$

$P_{cart} \leftarrow sphereToCart(P_{sphere})$

$P_{shift} \leftarrow rotateCart(P_{cart}, \theta, \phi)$

$P_{sphere} \leftarrow cartToSphere(P_{shift})$

$P_{output} \leftarrow sphereToSphpano(P_{sphere})$

End

[Fig. 8] Pseudo-code to rotate spherical panorama

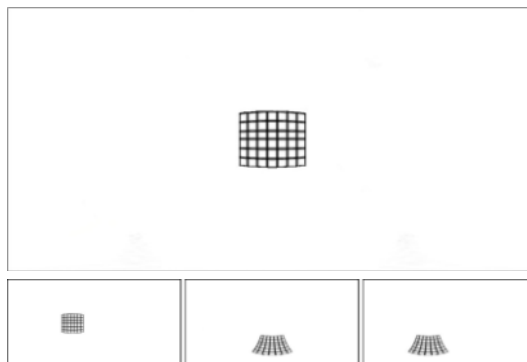
구면 파노라마 영상을 회전 변환하는 전체적인 알고리즘은 [Fig. 8]과 같다. 이러한 구면 파노라마 영상의 변환은 패턴이 두 개의 큐브맵 면에 걸쳐서 나타날 경우 탐색이 되지 않는 문제를 해결하기 위한 절차에서 사용된다.

3.3 구면 영상의 회전 영상 생성

제안 방법에서는 먼저 입력 영상을 회전 변환한 3개의 구면 파노라마 영상을 얻는다. 평면 패턴의 크기가 큐브맵 단일면 영상의 크기의 1/4보다 크면 45° 회전으로는 패턴이 큐브맵 영상에서 두 개의 면에서 나타나는 모든 경우에 대해 해결할 수 없다. 본 논문에서는 패턴의 크기가 큐브맵 영상에서 얻는 각 평면 영상의 크기의 1/4보다 작다고 가정한다. 회전 변환을 통해 큐브맵 영상을 추가로 생성하여 패턴이 두 개의 면에 걸쳐서 나타나는 경우를 해결한다.

패턴의 위치가 큐브맵 영상에서 두 개의 면에 걸쳐서 나타날 경우 패턴의 탐색이 실패할 수 있고 따라서 평면 패턴의 대치에도 문제가 생긴다.

이 문제의 해결을 위해서 경도와 위도 변환 인자를 각각 θ 와 ϕ 라고 할 때 원본 구면 파노라마 영상을 세 가지의 경우로 나누어 변환하여 세 개의 변환된 구면 파노라마 영상을 생성한다. 첫 번째 변환은 경도 θ 만 $\pi/4$ 만큼 이동시키는 변환으로, 이 경우에는 구면 파노라마 영상에서 패턴의 위치를 좌측으로 이동시키는 효과가 있다. 두 번째 변환은 위도 ϕ 만 $\pi/4$ 만큼 이동시키는 변환으로 구면 파노라마 영상에서 물체의 위치를 구면 좌표계로 표현했을 경우 좀 더 위쪽으로 옮기는 효과가 있다. 마지막으로 경도 θ 와 위도 ϕ 를 모두 $\pi/4$ 만큼 이동시키는 변환으로 패턴의 위치를 원래의 위치보다 왼쪽 위로 이동시켜서 평면 영상 두 개 사이에 걸칠 경우 대부분을 해결할 수 있다. 세 종류의 회전 변환의 결과는 [Fig. 9]와 같이 구면 파노라마 영상에서의 패턴의 위치 변화로 나타난다.



[Fig. 9] Rotation results of spherical panorama

3.4 패턴 탐색과 대치

세 가지 경우로 회전시킨 구면 파노라마 영상과 원본 구면 파노라마 영상을 합하여 총 4개의 구면 파노라마 영상을 모두 큐브맵 영상으로 변환한다. 이후 각 큐브맵 영상을 면마다 분리하여 총 24개의 평면 영상을 얻는다. 24개의 평면 영상을 얻는 과정은 [Fig. 10]에 있다. 알고리즘의 수행 과정은 먼저 *SphpanoToCubemap* 함수를 통하여 구면 파노라마 영상을 큐브맵 영상으로 바꾸어 저장한다. *getFacesFromCubemap*를 통해 큐브맵 영상에서 6

개의 면의 평면 영상을 모두 얻어 저장한다. 그리고 *RotateSphpano*를 통하여 입력된 구면 파노라마 영상을 회전시킨 3개의 새로운 구면 파노라마 영상을 저장한다. 또한 이 3개의 영상도 큐브맵으로 변환하고 평면 영상을 얻는다.

Algorithm GenerateCubemapFaces

```

Input :  $P_{input}$ 
Output :  $faces$ 
Begin
   $theta\_phi[3][2] \leftarrow \{\pi/4, 0, 0, \pi/4, \pi/4, \pi/4\}$ 
   $cubemap_{input} \leftarrow SphpanoToCubemap(Pano_{input})$ 
   $faces[0..5] \leftarrow getFacesFromCubemap(cubemap_{input})$ 
  for ( $i \leftarrow 0; i < 3; i++$ ) {
     $\theta \leftarrow theta\_phi[i][0], \phi \leftarrow theta\_phi[i][1]$ 
     $Pano_{shift}[i] \leftarrow RotateSphpano(Pano_{input}, \theta, \phi)$ 
     $cubemap[i] \leftarrow sphpanoToCubemap(Pano_{shift}[i])$ 
  }
endfor
  for ( $i \leftarrow 0; i < 3; i++$ )
    for ( $j \leftarrow 6(i+1); j < 12+6i; j++$ )
       $faces[j] \leftarrow getFacesFromCubemap(cubemap[i])$ 

```

End

[Fig. 10] Pseudo-code to get all 24 face images

모든 큐브맵 평면 영상들이 준비된 후에는 24개의 평면 영상에서 목표 패턴을 찾고 이를 수정할 텍스처로 대체시킨다. 24개의 평면 영상에서 패턴 탐색을 진행하고 패턴 매칭의 정확도를 나타내는 변수의 비교를 통해 가장 매칭이 잘 된 영상을 찾는다. 패턴 매칭 및 결과 비교 과정은 [Fig. 4]의 알고리즘에서 *searchPattern* 함수에 해당한다. 매칭 결과를 비교하는 방법은 OpenCV의 *DMatch* 구조체의 *distance*를 이용해 구현했다. *DMatch* 구조체는 특징점 기술자 매칭 결과를 저장하기 위한 구조체이며 *distance*는 기술자 매칭 방법에 의해 계산된 거리로 값이 적을수록 좋은 매칭 결과를 나타낸다.

평면 영상들을 비교하여 어떤 영상에 패턴이 있는지 파악이 완료되면 [Fig. 4]의 *replacePattern* 함수로 특정한 평면 영상에서 나타난 패턴을 원하는 평면 패턴으로 대체한다. 대체가 완료된 평면 영상

들은 [Fig. 4]의 알고리즘에서 *mergeToCubemap* 함수를 통하여 다시 큐브맵 영상으로 합친다. 마지막으로 *cubemapToSphpano* 함수는 큐브맵 영상을 다시 구면 파노라마 영상으로 변환한다. 만약 경도 또는 위도를 변환한 경우였다면 반대로 다시 변환하는 과정을 거치므로 원본 영상과 동일한 위치에 물체가 존재하게 된다.

4. 실험 결과

본 논문의 실험은 Intel Core i7 CPU, NVIDIA GeForce GTX 970, 16GB RAM, Windows 7 64bit OS의 시스템 환경에서 수행되었다. 실험을 위한 구면 파노라마 영상은 Ricoh Theta 카메라를 사용하여 촬영하였다. 다양한 환경에서의 구면 파노라마 영상에서 특정 사각 패턴을 찾아 지정된 텍스처를 다른 텍스처로 대체하는 실험을 진행하였다.

구면 파노라마 영상은 경도가 아닌 위도의 변화에 따라서 왜곡의 정도가 결정된다. 따라서 본 논문에서 실험에 쓰인 영상들은 패턴이 다양한 위도에 위치하도록 촬영하여 왜곡의 정도가 심한 경우에도 텍스처 대체가 가능하다는 것을 보이는 것을 목표로 진행되었다.

4.1 실내 영상에서의 텍스처 대체



[Fig. 11] Test set #1: the target search pattern and the replacement pattern

먼저 조명의 영향이 적은 실내 영상에 대해서

실험하였다. 영상에서 탐색 대상 패턴의 위치는 큐브맵의 두 면에 걸치는 경우의 영상을 선택하여 제안 알고리즘이 올바르게 동작하는지의 실험을 진행하였다. [Fig. 11]의 왼쪽은 구면 파노라마 영상에서 탐색할 대상 패턴이며 오른쪽은 대치할 패턴이다.



[Fig. 12] Test set #1: spherical images and matching results

[Table 1] Test set #1: matching result

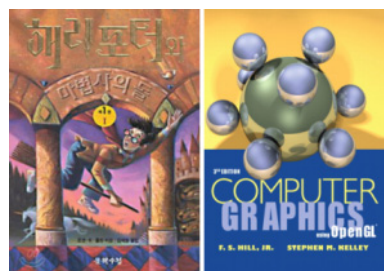
	θ	ϕ	Top	Bottom	Front	Back	Right	Left
P1	0	0	0.147627	0.174463	0.140414	0.163914	0.116581	0.143186
P2	$\pi/4$	0	0.154233	0.143575	0.119233	0.128316	0.139188	0.165246
P3	0	$\pi/4$	0.177769	0.156842	0.155913	0.138731	0.103035	0.158595
P4	$\pi/4$	$\pi/4$	0.184056	0.145101	0.104973	0.150316	0.144216	0.153029

[Fig. 12]에서 왼쪽 4개의 영상은 원본 구면 파노라마 영상(P1), 경도를 $\pi/4$ 만큼 회전한 영상(P2), 위도를 $\pi/4$ 만큼 회전한 영상(P3), 경도 및 위도를 각각 $\pi/4$ 만큼 회전한 영상(P4)들이다. [Fig. 12]의 오른쪽 영상들은 왼쪽의 구면 파노라마 영상들에서 얻은 평면 영상에서 패턴 매칭을

실시한 결과이다. 순서대로 Top, Bottom, Front, Back, Right, Left 면에 해당한다. P1과 P2 영상에서는 패턴 매칭이 실패하였지만 P3의 Right 면에서와 P4의 Front 면에서 패턴 매칭이 성공하였다. [Table 1]은 24개의 영상의 매칭 결과를 보여주며 결과 값은 낮을수록 더 정확한 결과를 의미한다. 가장 좋은 매칭 결과가 나타난 영상은 세 번째 영상의 Right 면이다. [Fig. 13]은 평면 영상에서 찾은 사각 패턴을 대치시키고 다시 구면 파노라마 영상으로 변환시킨 결과 영상이다.



[Fig. 13] Test set #1: result of replacement



[Fig. 14] Test set #2: the target search pattern and the replacement pattern

실내 환경에서의 두 번째 실험 세트는 패턴의 위치가 적도에서 더 멀어져서 구면 왜곡이 더 심하게 나타나도록 촬영되었다. [Fig. 14]는 실험에 사용된 패턴을 보여준다. [Fig. 15]는 탐색 대상 패턴을 큐브맵의 24개의 평면 영상에서 매칭한 결과

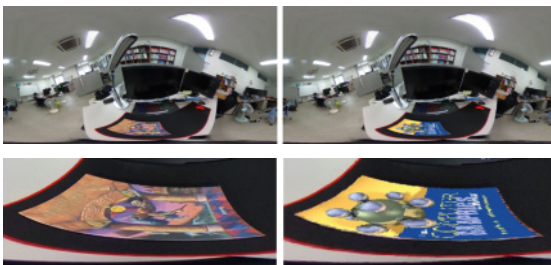
를 나타낸다. 패턴은 모든 파노라마 영상의 Bottom 면에서 검출된다. [Table 2]는 매칭 결과를 보여준다. 매칭 결과에 따라서 원본 구면 영상의 Bottom 면이 선택되어 패턴 대치에 사용된다. [Fig. 16]은 평면 영상에서 패턴을 대치시키고 구면 영상으로 바꾼 결과 영상이다.



[Fig. 15] Test set #2: spherical images and matching results

[Table 2] Test set #2: matching result

	θ	ϕ	Top	Bottom	Front	Back	Right	Left
P1	0	0	0.205665	0.074448	0.144937	0.156372	0.150086	0.150010
P2	$\pi/4$	0	0.206528	0.094375	0.157136	0.166960	0.180541	0.164490
P3	0	$\pi/4$	0.221346	0.093660	0.159961	0.157849	0.164241	0.160853
P4	$\pi/4$	$\pi/4$	0.205865	0.097719	0.188237	0.161884	0.141575	0.170822



[Fig. 16] Test set #2: result of replacement

4.2 실외 영상에서의 텍스처 대치

실외 환경에서는 촬영 시의 조명이나 주변 장애물 등의 가변 요소들로 인하여 매칭의 정확성이 현저하게 낮아진다. 세 번째 실험 세트에서는 탐색 패턴이 원본 구면 영상의 좌우 끝부분에 걸쳐서 나타나도록 촬영되었다. [Fig. 17]는 실험에 사용된 패턴을 보여준다.



[Fig. 17] Test set #3: the target search pattern and the replacement pattern



[Fig. 18] Test set #3: spherical images and matching results

[Fig. 18]은 탐색 대상 패턴을 큐브맵의 24개의 평면 영상에서 매칭한 결과를 나타낸다. [Table 3]

은 매칭 결과의 정확성을 보여준다. P3의 Back에서 가장 정확한 매칭이 이루어졌음을 알 수 있다. [Fig. 19]는 평면 패턴의 텍스처 대치가 완료된 영상이며 양 쪽에서 걸쳐서 나타나는 패턴도 텍스처 대치가 잘 동작함을 보여준다.

[Table 3] Test set #3: matching result

	θ	ϕ	Top	Bottom	Front	Back	Right	Left
P1	0	0	0.217170	0.155790	0.164087	0.131059	0.197597	0.166876
P2	$\pi/4$	0	0.180232	0.140729	0.182645	0.148994	0.140351	0.130322
P3	0	$\pi/4$	0.199534	0.195058	0.134328	0.100091	0.201524	0.192218
P4	$\pi/4$	$\pi/4$	0.147275	0.179171	0.180009	0.110326	0.120617	0.141255



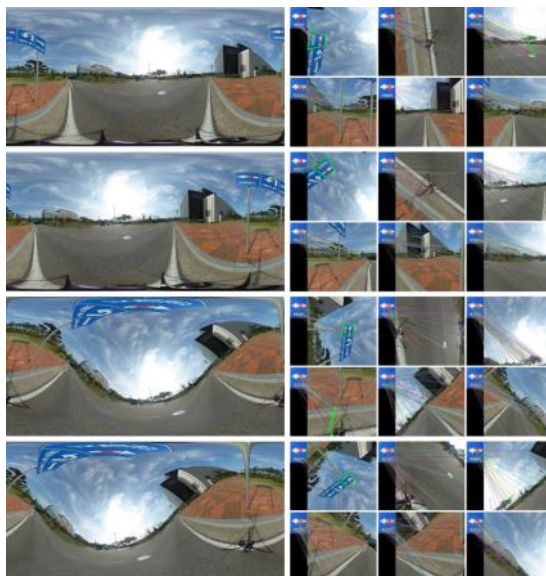
[Fig. 19] Test set #3: result of replacement



[Fig. 20] Test set #4: the target search pattern and the replacement pattern

네 번째 실험 세트는 도로에 있는 표지판을 포함한 구면 파노라마 영상을 사용하였다. [Fig. 20]은 실험에 사용된 패턴을 보여준다. 왼쪽 표지판을 오른쪽 표지판으로 바꾸는 텍스처 대치를 목적으로 한다. 원본 구면 영상에서 탐색 대상 패턴이 양 끝에 걸쳐서 보인다. 또한 고위도에 위치하여 구면 왜곡도 심한 편이다. [Fig. 21]은 탐색 대상 패턴을 큐브맵의 24개의 평면 영상에서 매칭한 결과를 나타낸다. [Table 4]은 매칭 결과의 정확성을 보여준다. P2의 Top 면에서 패턴 매칭이 가장 정확하게 되었다. 이 면에서 텍스처 대치를 실시하여 얻은 결과가 [Fig. 22]에 있다. 패턴의 왜곡이 심하고 양

끝에서 걸쳐서 보이는 경우에도 평면 패턴의 텍스처 대치가 가능함을 보여주고 있다.



[Fig. 21] Test set #4: spherical images and matching results

[Table 4] Test set #4: matching result

	θ	ϕ	Top	Bottom	Front	Back	Right	Left
P1	0	0	0.070233	0.174517	0.156441	0.161378	0.098133	0.167243
P2	$\pi/4$	0	0.061669	0.156901	0.159038	0.160108	0.147623	0.135001
P3	0	$\pi/4$	0.083970	0.173540	0.161813	0.189951	0.184774	0.160794
P4	$\pi/4$	$\pi/4$	0.082512	0.145607	0.118619	0.170815	0.162730	0.216616



[Fig. 22] Test set #4: result of replacement

5. 결 론

본 논문에서는 구면 파노라마 영상에서의 평면 패턴의 텍스처 대치 방법을 제시하였다. 구면 파노

라마 영상을 큐브맵 영상으로 변환하고 큐브맵의 각 면에서 평면 패턴 매칭을 실시하여 패턴을 찾고 이를 새로운 텍스처로 대체하였다. 큐브맵 영상으로 변환하여 패턴 매칭을 할 경우에 패턴이 큐브맵의 이웃 면들에 걸쳐서 나타나는 경우의 한계점을 극복하기 위해 구면 파노라마 영상의 회전 변환을 통해 추가적인 세 개의 구면 파노라마 영상을 생성하고 이들 모두에서 패턴 매칭을 실시하였다. 실내 및 실외 환경에서의 실험 결과를 통해 본 논문에서 제안하는 방법은 구면 파노라마 영상에서 평면 패턴을 거의 동일하게 매칭할 수 있었으며 이에 따라 평면 패턴의 텍스처 대체도 이루어질 수 있음을 확인하였다.

향후 연구과제로서 처리 속도의 고속화를 위하여 구면 파노라마 영상을 변환하지 않고 구면 파노라마 영상에서 직접 패턴 매칭을 실시하여 연산량을 줄이는 방법의 연구가 필요하다. 제안한 텍스처 대체 기술 방법은 증강현실 환경에서의 게시판이나 광고물 등의 평면 패턴의 변경을 위한 분야에서 활용될 수 있다.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. NRF-2015R1D1A1A01058809).

REFERENCES

- [1] G. Fangi, C. Nardinocchi, "Photogrammetric processing of spherical panoramas", *The photogrammetric record*, Vol. 28, No. 143, pp. 293-311, 2013.
- [2] D.G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, Vol. 60, No. 2, pp. 91-110, 2004.
- [3] B. Herbert, A. Ess, T. Tuytelaars, L.V. Gool, "Speeded-up robust features (SURF)", *Computer vision and image understanding*, Vol. 110, No. 3, pp. 346-359, 2008.
- [4] B.S. Kim, J.S. Park, "Estimating Geometric Transformation of Planar Pattern in Spherical Panoramic Image", *Journal of KIISE*, Vol. 42, No. 10, pp. 1185-1194, 2015.
- [5] B.S. Kim, J.S. Park, "Matching between Spherical Panorama and Planar Image", *Proc. Korea Computer Congress*, pp. 1322-1324, 2014.
- [6] D.R. Kim, J.S. Park, "Pattern Replacement in Spherical Panoramic Image Using Pattern Matching Technique", *Proc. Korea Game Society Fall Conference*, pp. 193-196, 2016.
- [7] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, J.-P. Thiran, "Scale invariant feature transform on the sphere: Theory and applications", *International journal of computer vision*, Vol. 98, No. 2, pp. 217-241, 2012.
- [8] S. Liu, L. Zhao, J. Li, Q. Cai, "The SIFT features matching for spherical panoramic images", *11th IEEE International Conference on Control & Automation*, pp. 914-917, 2014.
- [9] G. Tong, X. Chen, N. Ye, "A Spherical Model Based Keypoint Descriptor and Matching Algorithm for Omnidirectional Images", *Advances in Mechanical Engineering*, Vol. 6, 154376, 2014.
- [10] G. Tong, J. Gu, "Locating objects in spherical panoramic images", *International Conference on Robotics and Biomimetics*, pp. 818-823, 2011.
- [11] X. Guan, L.-K. Shark, G. Hall, W. Deng, "Distortion correction for immersive navigation in spherical image environment", *International Conference on CyberWorlds*, pp. 96-101, 2009.
- [12] C.M. Grimm, B. Niebruegge, "Continuous cube mapping", *Journal of Graphics Tools*, pp. 25-34, 2007.
- [13] F. Kangni, R. Laganieri, "Epipolar geometry for the rectification of cubic panoramas", *Canadian Conference on Computer and Robot Vision*, pp. 70-79, 2006.

- [14] Z. Zhou, B. Niu, C. Ke, W. Wu, "Static object tracking in road panoramic videos", International Symposium on Multimedia, pp. 57-64, 2010.



박정현 (Park, Jeong-Hyeon)

2012- 인천대학교 컴퓨터공학부 학사과정

관심분야 : 컴퓨터게임, 게임엔진, 영상분석



박종승 (Park, Jong-Seung)

1992 경북대학교 전자계산학과(이학사)

1994 포항공과대학교 컴퓨터공학과(공학석사)

1999 포항공과대학교 컴퓨터공학과(공학박사)

2004- 인천대학교 컴퓨터공학부 교수

관심분야 : 게임제작, 게임프로그래밍, 증강현실
