특집논문-10-15-2-02

단안 영상의 입체 자유시점 Tour-Into-Picture

김 제 동^{a)}, 이 광 훈^{a)}, 김 만 배^{a)†}

Stereoscopic Free-viewpoint Tour-Into-Picture Generation from a Single Image

Jedong Kim^{a)}, Kwanghoon Lee^{a)}, and Manbae Kim^{a)†}

요 약

자유시점 비디오는 원하는 시점을 자유로이 선택하여 보는 능동형 비디오이다. 이 기술은 박물관 투어, 엔터테인먼트 등의 다양한 분야에서 활용된다. 본 논문에서는 자유시점 비디오의 새로운 분야로 가상 카메라와 깊이맵을 이용하여 한 장의 영상 내부를 향해하는 입체 자유시점 Tour-Into-Picture (TIP)을 제안한다. 오래전부터 TIP가 연구되어 왔는데, 이 분야는 한 장의 단안 사진 내부를 향해하면서 애니메이션으로 볼 수 있게 하는 기술이다. 제안 방법은 전경 마스크, 배경영상, 및 깊이맵을 반자동 방법으로 구한다. 다음에는 영상 내부를 향해하면서 입체 원근투영 영상들을 획득한다. 배경영상과 전경객체의 3D 데이터를 기반으로 가상 카메라의 3차원 공간 이동, 요/피치/롤링 등의 회전, 룩어라운드, 줌 등의 다양한 카메라 기능을 활용하여 입체 자유시점 비디오를 구현한다. 원근투영은 직교투영보다 우수한 입체감을 전달하며, 기존 방법과 비교하여 텍스처의 3D 데이터를 직접 원근투영하여 처리속도를 향상시켰다. 소프트웨어는 MFC Visual C++ 및 OpenGL 기반으로 구축되었으며, 실험영상으로 신윤복의 단오풍정을 사용하여 고전화의 입체 자유시점 비디오를 시청이 가능하다.

Abstract

The free viewpoint video delivers an active contents where users can see the images rendered from the viewpoints chosen by them. Its applications are found in broad areas, especially museum tour, entertainment and so forth. As a new free-viewpoint application, this paper presents a stereoscopic free-viewpoint TIP (Tour Into Picture) where users can navigate the inside of a single image controlling a virtual camera and utilizing depth data. Unlike conventional TIP methods providing 2D image or video, our proposed method can provide users with 3D stereoscopic and free-viewpoint contents. Navigating a picture with stereoscopic viewing can deliver more realistic and immersive perception. The method uses semi-automatic processing to make foreground mask, background image, and depth map. The second step is to navigate the single picture and to obtain rendered images by perspective projection. For the free-viewpoint viewing, a virtual camera whose operations include translation, rotation, look-around, and zooming is operated. In experiments, the proposed method was tested eth'Danopungjun' that is one of famous paintings made in Chosun Dynasty. The free-viewpoint software is developed based on MFC Visual C++ and OpenGL libraries.

Keywords : Free viewpoint video, TIP (Tour-Into-Picture), stereoscopic image, depth map

a) 강원대학교 컴퓨터정보통신공학과

Kangwon National University, Dept. of Computer and Oommunications

† 교신저자 : 김만배 (manbae@kangwon.ac.kr)

※본 연구는 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과이며 또한, 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음 (NIPA-2010-(C1090-1011-0003))

· 접수일(2009년12월28일), 수정일(2010년2월16일), 게재확정일(2010년2월16일)

I. 서론

자유시점 비디오는 단순히 수동적으로 비디오를 보는 것이 아니라 이용자가 자신이 원하는 시점을 자유로이 선택하여 보거나, 또는 자신이 원하는 비디오 장면을 구성할 수 있는 이용자 중심의 능동형 비디오이다^{[1][2][3]}. 장면 네비게이션, 사용자의 자유시점, 실감형 비디오 등의 특성을 가지고 있으며, 카메라의 팬닝(panning), 틸팅(tilting), 줌인(zoom-in) 및 줌아웃(zoom-out)의 동작과 같이 상하, 좌우, 앞뒤로 시점을 이동시켜 볼 수 있도록 함으로써 비디오 장면의 실감성을 제공한다. 응용 분야는 교육/의료, 박물관 가상 투어, 엔터테인먼트 등의 다양한 분야로 지속적으로 확장되고 있다. 또한 향후 자유시점 비디오의 확산을 위해서 새로운 콘텐츠의 개발 또한 중요하다 할 수 있다.

본 논문에서는 꾸준히 연구되어 왔던 TIP(Tour-Into-Picture) 기술에 자유시점 및 3D 실감을 위한 입체영상을 접목하는 입체 자유시점 TIP를 제안한다. 제안 방법으로 교육 및 엔터테인먼트 분야에 응용이 가능한 새로운 자유시점 콘텐츠의 제작이 가능하다. TIP는 한 장의 단안 영상으로부터 애니메이션 동영상을 제작하는 것으로, 종종 놀라운 시각 효과를 전달한다. 기존 TIP 연구는 다음과 같다. Horry 등은 장면을 5 개의 사각형으로 구성되는 박스 형태의 모델링을 이용하는 TIP를 구현하였고^[4], Boulanger 등은 자동카메라 캘리브레이션을 이용하여 신속하게 TIP을 제작할 수 있는 방법을 제안하였다^[5]. 또한, Chu 등은 중국 풍경화 영상에 두 종류의 구도가 존재한다고 가정하고, 다투영(multi-perspective) 모델링을 통해 애니메이션을 만들었다^[6]. 또한 기존의 2D TIP를 입체 TIP로 확장한 연구가 [7]에서 제안되었다. 이 방법은 입력 영상의 깊이맵을 생성하고, 직교투형으로 투영영상을 만든 후에 입체 애니메이션을 제작하였다.

제안 방법은 단안영상을 자유시점 입체영상으로 제작되어 실감적인 영상을 제공한다. 제안 방법은 [7]의 직교투형 방식을 장면의 특성에 맞게 원근투형으로 구현하였으며, 자유시점 비디오를 위해 가상 카메라로 영상 내부부를 향해 하면서 투영 영상들을 획득한다. 이 투영된 영상들은 가상 카메라의 3차원 공간 이동, 요(yaw), 피치(pitch), 롤링

(rolling) 등의 회전, 룩아라운드(look-around), 줌(zoom) 등의 다양한 카메라 기능을 활용하여 자유시점 비디오를 구현한다. 또한 가상 스테레오 카메라를 조절하여 입체영상으로 디스플레이하며, 원근투형과 결합되어 입체영상의 입체감을 향상시킨다. 원근투영 방식은 직교투형 방식과 비교하여 입체감의 개선이 뚜렷하다. 또한 기존 연구에서는 직교투영된 깊이맵을 매 프레임마다 생성하기 때문에 처리 시간이 증가하였다. 개선 방법에서는 텍스처의 3D 데이터를 직접 원근투영하여 처리속도를 향상시켰다.

본 논문의 구성은 다음과 같다. II장에서는 입체영상의 제작에 필요한 깊이맵을 생성하는 방법을 소개하고, 입체 자유시점 영상을 제공해주는 렌더링 기법은 III장에서 제안된다. IV장의 실험에서는 카메라의 다양한 기능에 따른 생성된 자유시점 렌더링 영상을 보여준다. 마지막으로 V장에서는 결론을 맺는다.

II. 깊이맵 생성

입체영상을 생성하기 위해서는 기본적으로 단안 영상의 깊이맵을 제작해야 한다. 실험영상인 신윤복의 단오풍정의 깊이맵 제작방법은 [7]에서 제안되었고, 본 장에서는 기본 방법을 설명한다. 또한 입체 자유시점 동영상 생성과정에서 고려해야 하는 사항을 정리한다. 먼저 입력영상으로부터 전경 마스크, 배경영상, 및 깊이맵을 생성한다. 특히 자유시점 비디오를 구현하기 위해서 렌더링에서는 이동과 회전 등의 다양한 기능을 구현할 수 있는 가상 카메라로부터 투영된 텍스처 영상 및 깊이맵을 생성한다. 또한 입체 전경 객체들을 위해 보다 향상된 깊이맵의 제작이 필요하다.

영상에는 전경객체와 배경영상의 2 종류의 정보가 존재한다. 전경객체는 이진 마스크로 정의된다. 영상에서 마스크 영역을 제거하면, 한 개의 배경 영상이 얻어진다. 일반적인 고전화에서 영상 객체의 주변경계는 추상적인 구조 때문에 추출이 어렵다. 따라서 기존 객체 세그멘테이션 알고리즘의 사용으로는 원하는 객체를 얻을 수 없어서 포토샵을 이용하여 추출하였다. 전경 객체를 제거한 후에 발생하는 홀(hole)은 이미지 인페인팅 기법[8]과 포토샵을 사용하

여 채운다. 그림 1은 입력영상 및 추출된 전경객체와 홀이 채워진 배경영상을 보여준다.

배경영상의 깊이맵을 생성하기 위해 수작업으로 배경영상을 N 개의 폴리곤으로 분할한다. 폴리곤의 개수는 클수록 좀더 정확한 깊이를 얻을 수 있다. 각 폴리곤은 M 개의 정점으로 구성되며, 수작업으로 폴리곤의 각 정점에 $[0, 255]$ 의 범위에서 깊이값을 할당한다. 깊이값 설정은 영상의 구도에 따라 수작업으로 이루어지는데 생성된 각각의 폴리곤은 [9]의 알고리즘을 이용하여 3개의 정점으로 구성된 삼각형 메쉬로 분할한다. 다음에 셰이딩 기법을 모든 삼

각형에 적용하여 최종 배경 영상의 깊이맵을 생성한다. 그림 2는 깊이맵을 생성하는 과정을 보여준다. 그림 2(a)는 $N=6$ 인 실험영상의 폴리곤들을 보여주고, (b)는 폴리곤 분할에서 얻어진 삼각형 메쉬 구조이다. 삼각형에 구로 셰이딩(Gourad shading)을 적용한 후에 얻은 최종 깊이맵은 (c)에서 보여준다.

전경객체의 깊이맵은 전경객체 마스크와 배경영상의 깊이맵을 비교하여 만든다. 전경객체와 배경이 만나는 곳의 깊이 값은 유사하다는 가정하에 전경객체의 초기 깊이값을 할당한다. 전경객체의 위치를 알기 위해 연결요소기법(con-

그림 1. 전경객체 마스크와 배경 영상. (a) 입력영상, (b) 전경객체 마스크, (c) 홀이 채워진 배경영상

Fig. 1. Foreground object mask and background image. (a) input image, (b) foreground mask and (c) hole-filled background image

그림 2. 배경영상의 깊이맵. (a) 폴리곤 분할, (b) 삼각형 메쉬 (c) 깊이맵

Fig. 2. Background depth map. (a) polygon, (b) triangle mesh obtained from polygon decomposition, and (c) depth map of background images

그림 3. 전경객체의 깊이맵 생성 (a) 전경 객체의 바운딩 박스, (b) 배경 깊이맵에서 바운딩 박스의 위치, 및 (c) 초기 전경 객체의 깊이맵

Fig. 3. Depth map of foreground objects. (a) bounding box of foreground, (b) location of bounding boxes at the background depth map, and (c) initial depth map of foreground objects

nected component analysis)으로 객체의 레이블을 구한다. 동일 레이블 값을 가지는 전경객체 마스크의 바운딩 박스(bounding box)를 구한 후에, 박스의 아래 경계선에 있는 전경 마스크 픽셀과 인접한 배경 픽셀들의 평균 깊이값을 전경객체에 할당한다. 따라서 전경객체는 상수 깊이 값 DC을 가지게 된다. 그림 3은 실험영상에서 얻어진 인물 전경객체의 마스크 및 깊이맵을 보여준다.

입체영상에서 배경보다 전경객체에 인간의 입체시는 민감하다. 따라서 전경객체의 입체감 향상을 위한 작업이 필요하다. 그러나 영상내부에 존재하는 다양한 형태의 객체들의 3차원 데이터를 자동으로 생성하는 것은 어려운 작업이다. 기존 TIP에서는 단순한 방법으로 전경객체의 3D 모델링을 연구하였다. Horry 등은 계층적 폴리곤 모델 기법을 제안하였는데, 전경 인물의 모델을 네모꼴(quadrangle)로 가정한 후에 원근투영으로 2차원 평면에 투영하였다^[4]. 이 기법은 2D 영상에서는 약간의 원근감을 줄 수 있으나, 입체영상에서는 적용이 어려운 기법이다. Anjyo 등은 전경 객체를 배경에 포함하여 동일하게 처리하였다^[7]. 상기 방법들은 2D TIP에서는 적합한 방법이나, 입체 TIP에서는 전경객체를 3D로 지각할 때에 시각적 불편을 제공할 수 있다. 따라서 입체영상에 적합한 깊이 생성 방법이 필요하며, 수작업이 아닌 자동 깊이 추출이 요구된다.

전경객체들을 입체로 시청할 때에 발생하는 카드보드 효과(cardboard effect)를 감소하기 위해^{[10][11]}, 전경객체 깊이에 불륨을 주기 위한 필터링 기법을 사용한다. 전처리 단계에서 구해진 전경객체의 깊이는 픽셀의 위치에 관계없이 상수 깊이값인데, 이 깊이를 이용하여 생성된 입체영상을 시청할 때에 카드보드효과가 발생한다. 카드보드 효과는 입체로 시청할 때에 영상에 존재하는 객체가 계층화되어있는 것처럼 지각되는 현상이다. 즉, 납작하게 보이는 현상으로 눈의 피로도를 증가시킨다. 카드보드 효과를 감소하기 위해서는 불륨있는 깊이가 필요하고, 이를 위해 로스 텍스처 필터(Laws texture filter) 기반 깊이 생성 방법을 이용한다.

인간은 3D 깊이감을 얻기 위해 단안 및 스테레오 정보로 구성된 다양한 종류의 시각 정보를 이용한다^[12]. 단안 정보 중 에지(edge) 방향의 분포를 알려주는 텍스처 그라디언트(texture gradient)는 깊이 인식의 중요한 역할을 담당한다.

로스 텍스처 필터는 단안 비전에서 3D 깊이를 얻기 위한 방법으로 종종 사용된다^[13]. 실제 2D 영상에서 깊이를 예측하기 위해 일반적으로 텍스처 편차, 텍스처 그라디언트, 색상 등을 활용하는데, 이의 구현을 위해서 로스 텍스처 필터가 활용되어 왔다^[14].

로스 텍스처 에너지는 텍스처의 평균 명암, 에지, 스포트(spot), 리플(ripple) 및 웨이브(wave)의 특성을 가지고 있다. 에너지는 다음 벡터로부터 얻어진다.

$$\begin{aligned} L5 \text{ (Level)} &= \{1 \ 4 \ 6 \ 4 \ 1\} - \text{center weighted local average} \\ E5 \text{ (Edge)} &= \{-1 \ -2 \ 0 \ 2 \ 1\} - \text{row or column step edges} \\ S5 \text{ (Spot)} &= \{-1 \ 0 \ 2 \ 0 \ -1\} - \text{detects spots} \\ R5 \text{ (Ripple)} &= \{1 \ -4 \ 6 \ -4 \ 1\} - \text{detects ripples} \end{aligned}$$

2D 콘볼루션 마스크 M은 상기 1x5 벡터의 곱으로 만들어지는데, L5E5는 다음과 같이 구해진다.

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} [1 \ 4 \ 6 \ 4 \ 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (1)$$

이외에도 두 개의 1x5 벡터에 콘볼루션을 적용하면 다양한 5x5 텍스처 필터가 얻어진다.

카드보드 현상이 주로 발생하는 곳은 상수 깊이를 갖는 전경객체이다. 전경객체는 상대적으로 배경보다 앞에 보이게 되며, 또한 인간의 시각에 민감하다. K 개의 2D 로스 필터를 적용하는데, 식 (2)의 $F_k(i, j)$ 는 영상 I를 k번째 필터 M_k 로 콘볼루션하여 얻은 결과이다.

$$F_k(i, j) = I(i, j) \otimes M_k \quad (2)$$

여기서 $\otimes$ 는 콘볼루션 연산자이다. $k \in \{1, \dots, K\}$.

따라서, k번째 텍스처 에너지는 식 (3)과 같이 계산된다. 픽셀 주변의 15x15 블록의 에너지의 합이 현재 픽셀의 에너지가 된다.

$$E_k(r, c) = \sum_{i=r-7}^{r+7} \sum_{j=c-7}^{c+7} |F_k(i, j)| \quad (3)$$

각 텍스처 에너지는 입력영상에 k번째 필터를 적용하여 얻는다. 얻어진 에너지들의 합은 최종적으로 선형 또는 비선형 변환으로 로스 깊이값으로 변환한다.

최종적으로 객체의 깊이값 D는 상수 깊이값 DC에 가중치가 곱해진 DL을 더해 구해진다.

$$D = D_C + \lambda \cdot D_L \quad (4)$$

여기서 가중치 λ 는 사용자가 지정한 값이다.

그림 4는 최종적으로 얻어진 전경객체의 깊이맵을 보여준다.

그림 4. 전경객체의 향상된 깊이맵. (a) 상수 깊이맵, (b) 상수깊이맵에 로스 깊이를 더한 최종 깊이맵

Fig. 4. Enhanced depth map of objects. (a) constant depth map, (b) final depth map

III. 입체 자유시점 비디오 렌더링

자유시점은 기본적으로 이용자가 카메라의 팬닝, 틸팅, 줌인 및 줌아웃 등의 기능처럼, 상하, 좌우, 앞뒤로 시점을 이동시켜 볼 수 있도록 함으로써 실감 비디오를 제공한다. 그림 5는 입체 자유시점 비디오 생성과정을 보여주는데, 먼저 입력 데이터인 전경객체와 배경영상의 텍스처 및 깊이맵 데이터를 3차원 공간으로 매핑한다. 입체 가상 카메라는 평행 또는 아크 형태로 구성할 수 있다. 평행 카메라는 3D 객체가 모니터 앞으로 튀어나오게 보이는 음시차를 구현하며, 반대로 아크 형태는 두 카메라 광축의 수렴점을 위치에 따라 자유롭게 3D 장면의 양시차 및 음시차의 조절이 가능하다. 마지막으로 촬영된 좌우 입체영상이 생성된다.

그림 5. 입체 자유시점 비디오 생성의 블록도

Fig. 5. The block diagram of generating free-viewpoint stereoscopic video

3차원 공간에서의 가상 카메라의 위치와 방향을 변경하면서 투영영상을 생성한다. 렌더링에서는 카메라 시점을 이동하면서 촬영된 텍스처 및 깊이맵을 저장하는데, 깊이맵은 입체영상 생성에 필요하다. 제안 시스템은 단일 소프트웨어로 통합하여 사용자 입력을 받아 임의의 자유 시점에서 입체영상을 시청할 수 있게 한다.

3D 공간에서의 렌더링 입체영상을 얻기 위해서 배경의 깊이맵을 가지는 3D 모델링 데이터에 배경영상을 텍스처 매핑하여 배경영상을 얻는다. 얻어진 3차원 배경 영상에 전경객체는 깊이맵 데이터에 따라 한 점씩 배경위에 렌더링되므로, 다수의 점들이 모여 형상이 만들어진다.

3D 장면의 애니메이션을 제작하기 위하여 가상 카메라의 위치 및 방향을 제어하는 것이 필요하다^[15]. 가상 카메라는 카메라 위치인 뷰평면의 기준점(VRP, View Reference Point), 뷰평면의 법선벡터(VPN, View Plane Normal), 및 뷰앵글(FOV, Field Of View)의 조절을 활용한다. 3D 공간 세계좌표계는 입력 영상의 중심을 (0, 0, 0)으로 하고, (X, Y, Z) 축으로 정의한다. 그리고 카메라 좌표계는 (u, v, n)으로 정의된다. 카메라의 기능은 회전, 이동, 줌, 뷰앵글, 룯어라운드(look-around)로 구성되어있다. 그중 회전은 VPN 벡터를 조절하면서 틸팅 및 패닝하며, 롤링(rolling),

요(yaw), 피치(pitch)의 효과를 얻는다. 각 기능에 대한 구현은 다음과 같다.

A) 회전 : (u, v, n) 축에 대한 카메라의 회전은 세 가지 방식으로 제공된다. 롤링은 카메라의 $\vec{n}$ 벡터를 축으로 회전하는데, u_0, v_0 에서 α° 만큼 변환된 u, v는 다음식에서 계산된다.

$$\begin{aligned} u &= \cos\alpha \times u_0 - \sin\alpha \times v_0 \\ v &= \sin\alpha \times u_0 + \cos\alpha \times v_0 \end{aligned} \quad (5)$$

피치는 카메라의 $\vec{u}$ 벡터를 축으로 β° 만큼 회전하는데 n, v는 다음식에서 구해진다.

$$\begin{aligned} n &= \cos\beta \times n_0 + \sin\beta \times v_0 \\ v &= -\sin\beta \times n_0 + \cos\beta \times v_0 \end{aligned} \quad (6)$$

요는 카메라의 $\vec{v}$ 벡터를 축으로 γ° 회전하는데 이것은 식 (7)과 같다.

$$\begin{aligned} u &= \cos\gamma \times u_0 + \sin\gamma \times n_0 \\ n &= -\sin\gamma \times u_0 + \cos\gamma \times n_0 \end{aligned} \quad (7)$$

B) 이동: VPN_z 을 고정시킨 상태에서, 카메라 위치를 상하 또는 좌우로 이동한다. 즉, (u, v) 축으로 ($\Delta u, \Delta v$) 만큼 이동하는데 식 (8)에서 구해진다.

$$\overrightarrow{VPN} = \overrightarrow{VPN_0} + \vec{u} \times \Delta u + \vec{v} \times \Delta v \quad (8)$$

C) Zoom: 이동과 같은 방식으로 카메라 위치를 VPN_z 방향으로 식 (11)과 같이 이동한다.

$$\overrightarrow{VPN} = \overrightarrow{VPN_0} + \vec{n} \times \Delta n \quad (9)$$

D) View-angle: 카메라 위치와 VPN은 고정된 상태에서 뷰 앵글을 수정한다.

E) Look-Around: 카메라는 VPN의 방향을 투시점을 향하게 하고, 원의 궤적을 따라 위치점을 이동시킨다. 따라서, (u, v, n)을 동시에 변경하게 된다.

투영방식에는 일반적으로 원근투영과 직교투영이 있다. 본 실험에서는 원근투영을 기반으로 영상을 제작하였다. [7]에서는 직교투영 방식을 사용하였는데, 직교투영에 비해 원근투영은 입체감이 많이 개선되지만, 단점은 빈 공간이 보이는 단점이 있다. 그림 6은 배경 영상의 삼각형 매쉬를 원근투영으로 투영한 영상을 보여준다. 입체 가상 카메라는 평행식인 경우 카메라의 u 축으로 카메라 위치를 좌, 우로 이동하여 각각 영상을 생성하게 된다.

IV. 실험 결과

입체 자유시점 TIP의 소프트웨어는 MFC Visual C++ 및 OpenGL을 기반으로 구축하였다. 단오풍정의 텍스처 영상과 깊이맵을 사용하여, 제안하는 자유시점 기능을 적용하여 입체영상을 제작 및 시청하도록 하였다. 영상 해상도는 600×477 이다. 단오풍정에서는 모든 전경객체에 로스 값

그림 6. 깊이맵과 원근투영된 영상들. (a) 배경영상, (b) 깊이맵, (c) 및 (d) 시점 이동 영상

Fig. 6. Depth map and projected images. (a) background image, (b) depth map, (c) and (d) other projected images

이를 적용하여 깊이맵을 사용하였다. 그림 7의 GUI 우측에 보여지는 자유시점 메뉴는 다음과 같다: Translation (Left, Right, Up, Down), Zoom (In, Out), Pitch (Up, Down), Roll (Left, Right), Yaw (Left, Right), View Angle (Wide, Narrow), Look Around (Left, Right) 및 디스플레이 모드는 2D, Interlaced, 및 Anaglyph의 3가지 입체 시청을 지원한다. 각 메뉴의 구현 방법은 이전 장에서 설명하였다. 입체시청 메뉴를 사용하면 그림 8과 같이 화면의 우측에 좌우입체영상이 디스플레이된다.

그림 7. 입체 자유시점 TIP의 GUI
Fig. 7. GUI of free-viewpoint video software

그림 8. 우측에 입체영상을 보여주는 GUI
Fig. 8. The stereoscopic image is shown at the left of the screen.

그림 9는 이동에서 left/right/up/down중 좌우 이동에서 얻어진 연속 영상을 보여준다. 이동 명령에 의해 카메라는 영상의 중심 (0, 0, 0)에서 좌우로 움직여서 얻어진 영상이다. 영상은 카메라의 움직임과 반대로 아래와 왼쪽으로 움직인 것 같은 모습이다. 물체의 움직임에 따른 줌인 및 줌아웃 효과는 그림 10에서 보여진다. 이동은 3차원 공간에서 (u, v)축으로의 카메라 움직임이고, 줌은 n축으로의 이동이다. 이 기능을 통하여 사용자는 자신이 원하는 부분에 대하여 집중적인 감상이 가능하다.

그림 9. 이동에 의해 얻어진 연속 영상
Fig. 9. Image frames rendered from camera translation

그림 10. 줌에 의해 얻어진 연속 영상
Fig. 10. Image frames rendered from zooming

그림 11. roll/pitch/yaw에 의해 얻어진 연속 영상. (a) pitch, (b) roll, 및 (c) yaw
Fig. 11. Image frames rendered from roll, pitch, and yaw

사용자는 roll/pitch/yaw의 회전 명령을 통하여 현실상에서 보기 힘들었던 각도에서의 감상이 가능하다. 그림 11은 단순한 회전에서 얻은 영상이 아닌 그림 9와 10에서 사용한 이동과 줌을 병행하여 사용하였다. 따라서 사용자는 마치

작품이 여러 방향으로 틀어진 듯한 영상을 감상할 수 있다. n축으로의 이동을 통한 확대 축소가 줄이었다면, 부앵글은 카메라의 확대 축소의 구현이다. 사용자는 카메라 렌즈의 각도 조절을 통하여 그림 12처럼 3차원 영상의 확대 축

그림 12. 뷰앵글과 얻어진 연속영상
Fig. 12. Image frames rendered from varying view angles

그림 13. 룩어라운드에 의해 얻어진 연속영상
Fig. 13. Image frames rendered from look-around operation

그림 14. 자유시점에서 얻어진 영상들

Fig. 14. Images from free-viewpoint navigation

소를 경험할 수 있다.

3차원영상에서 각기 다른 위치 정보를 가지는 배경과 전경은 록어라운드 기능을 통하여 원하는 위치의 측면에서 영상을 감상할 수 있게 한다. 이 명령은 이동과 회전 명령의 복합적인 활용과 같다. 단순한 카메라의 좌우 움직임이 아닌 원의 궤적을 따라 이동하는 록어라운드는 전경과 배경사이의 원근을 더욱 사실적으로 느낄 수 있다. 있다. 그림 13은 록어라운드 메뉴를 활용하여 얻어진 영상들이다.

그림 14는 자유시점에서 렌더링된 다양한 투영영상들을 보여준다. 단순히 한 장의 영상을 보는 것보다, 새로운 영상물을 보는 듯한 느낌을 전달할 수 있는 것이 자유시점 TIP의 장점이다.

V. 결 론

본 논문에서는 단안영상으로부터 입체 및 자유시점 비디오 시청이 가능한 입체 자유시점 TIP를 제안하였다. 제안 기법에서는 다양한 카메라 기능과 입체영상 제작을 통합하여, 사용자가 원하는 위치와 시점에서 한 장의 영상 내부를 향해할 수 있다. 이를 위해 다양한 카메라 기능을 정의하고, 가상 카메라에 따라 입체영상을 렌더링하는 알고리즘을 포함하는 소프트웨어를 제작하였다. 기존 방법과 달리 입체영상과 원근투형을 결합되어 입체영상의 입체감을 향상하였으며, 텍스처의 3D 데이터를 직접 원근투영하여 처리속도를 향상시켰다. 향후 단안영상의 자유시점 비디오 기술은 교육, 엔터테인먼트의 새로운 분야에 활용이 가능하다.

참 고 문 헌

- [1] A. Smolic, and P. Kauff, "Interactive 3D Video Representation and Coding Technologies", Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery, vol. 93, no. 1, Jan. 2005.
- [2] Masayuki Tanimoto, "Free Viewpoint Television - FTV", Proc. PCS 2004, Picture Coding Symposium, San Francisco, CA, USA, December 15.-17. 2004.
- [3] A. Smolic, and D. McCutchen, "3DAV Exploration of Video-Based Rendering Technology in MPEG", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 14, No. 3, pp. 348-356, March 2004.
- [4] Y. Horry, K. Anjyo, and K. Arai, "Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image," Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 225-232, 1997.
- [5] K. Boulanger, K. Bouatouch, S. Pattanaik, "ATIP: A Tool for 3D Navigation inside a Single Image with Automatic Camera Calibration," EG UK Theory and Practice of Computer Graphics 2006.
- [6] N. S. Chu and C. Tai, "Animating Chinese landscape paintings and panorama using multi-perspective modeling," Proceedings of International Conference on Computer Graphics, IEEE, pp. 107-113, 2001.
- [7] 조철용, 김만배, "3DTIP: 한국 고전화의 3차원 입체 Tour-Into- Picture," 방송공학회논문지, 제14권 제5호, pp. 616-624, 2009년 9월.
- [8] A. Criminisi, P. Perez and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Trans. Image Processing, Vol. 13, No. 9, pp. 1200-1212, 2004.
- [9] R. Seidel, "A simple and fast randomized algorithm for computing trapezoidal decompositions and for triangulating polygons," Computational Geometry Theory & Application, Vol. 1, pp. 51-64, 1991.
- [10] H. Yamanoue, M. Okui and F. Okano, "Geometrical analysis of puppet-theater and cardboard effects in stereoscopic HDTV Images," IEEE Trans. On Circuits and Systems for Video Technology, Vol. 16, No. 6, June 2006.
- [11] H. Yamanoue, M. Okui and I. Yuyama, "A study on the relationship between shooting conditions and cardboard effect of stereoscopic Images," IEEE Trans. On Circuits and Systems for Video Technology, Vol. 10, No. 3, Apr. 2000.

- [12] D. F. McAllister (editor), Stereo computer graphics and other true 3D technologies, Princeton, NJ:Princeton University Press, 1993.
- [13] E. R. Davies, Laws' texture energy in Texture. In Machine Vision: Theory, Algorithms, Practicalities, 2nd Ed., Academic Press, San Diego, 1997.
- [14] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," Proc. of International Conference on Machine Learning, ACM, 2005.
- [15] F. S. Hill, Jr, Computer graphics using OpenGL, 2nd Ed. Prentice-Hall, 1990.

저 자 소 개

김 제 동

- 2009년 : 강원대학교 컴퓨터정보통신공학과 학사
- 2009년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 석사과정
- 주관심분야 : 입체영상처리, 3D변환

이 광 훈

- 2010년 : 강원대학교 컴퓨터정보통신공학과 학사
- 2010년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 석사과정
- 주관심분야 : 입체영상처리, 3D비디오처리

김 만 배

- 1983년 : 한양대학교 전자공학과 학사
- 1986년 : University of Washington 전기공학과 공학석사
- 1992년 : University of Washington 전기공학과 공학박사
- 1992년 ~ 1998년 : 삼성종합기술원 수석연구원
- 1998년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 교수
- 주관심분야 : 3D비디오처리, 입체변환, 다시점영상처리, 3DTV시스템, 증강현실