

OpenGL ES 를 이용한 Android Platform에서의 TIP 기술

이준호, 장민석, 이연식
군산대학교 컴퓨터정보공학과

e-mail :yslee{msjang}@kunsan.ac.kr

TIP Technique using the OpenGL ES for android platform

Junho Lee, Minseok Jang, Yonsik Lee
Dept. of Computer Information Engineering, Kunsan National University

요 약

TIP 기술은 2D 그림 또는 한 장의 사진으로부터 기하정보를 추출하여 3 차원 입체 효과를 만들어 영상 내부를 네비게이션할 수 있는 기술로써, 게임, 엔터테인먼트, 교육, 홍보 등 다양한 분야에서 요구되는 주요기술이다. 본 논문에서는 최근 대두되고 있는 스마트 device 의 platform 가운데 하나인 android platform 상에서의 OpenGL ES Library 를 이용한 TIP 기술 적용 및 구현 기술을 제안한다. 제안 방법은 전경객체의 추출이 어려운 상황을 감안하여 보다 사실적 장면 구성이 용이하도록 사용자의 선택에 의한 소실점을 이용하고, OpenGL ES Library 를 이용하여 3D 배경 모델을 획득하고, 이미지를 텍스처 매핑하여 3D 가상공간을 완성한 후 카메라의 시점 변환을 통해 이미지 내부를 네비게이션할 수 있도록 한다. 실험영상은 android platform 상의 device 에서 촬영한 이미지를 사용하고, android 2.1 및 OpenGL ES 1.0 기반으로 구축함으로써, 제안 기술을 다양한 android platform smart device 에서 적은 비용과 시간으로 응용 개발에 효과적으로 적용 가능하도록 구현하였다.

1. 서 론

3차원 입체영상 기술은 기존의 2차원 평면 영상과는 달리 사람이 보고 느끼는 실제 영상과 유사하여 시각정보의 질적 수준을 몇 차원 높여 주는 새로운 개념의 영상미디어로서 차세대 디지털 영상문화를 주도하게 될 것으로 전망되고 있다[1]. 일반적으로 한 장의 영상으로부터 깊이 정보를 알아내기가 쉽지 않기 때문에, 사용자로부터 수작업을 통해 영상내부의 기하정보를 이용하였으며, Horry[2]등은 하나의 영상으로부터 원근변환 정보만을 이용하여 네비게이션이 가능한 3차원 환경을 구성하여 애니메이션 효과를 보여주는 TIP(Tour Into the Picture) 기술을 제안하였다. Boulanger[3]등은 자동 카메라 캘리브레이션을 이용하여 TIP를 신속하게 제작하는 방법을 제안하였고, Zhang[4]은 깊이 영상 기반으로 한 렌더링 시스템을 이용하여 3차원 입체 영상을 생성하는 방법을 제안하였다. 이외에도 TIP기술과 관련된 다양한 연구가 활발히 진행되고 있다[5,6,7,8,9].

본 논문에서는 최근 대두되고 있는 스마트 device의 platform중 하나인 android platform상에서의 카메라를 이용한 촬영과 유저의 선택에 의한 기하정보 추출 및 과정을 통해 OpenGL ES Library를 이용하여 TIP기술을

적용 하는 기법에 대해 제안하여, 향후 스마트 device에서의 게임 및 엔터테인먼트 분야에서 TIP기술 응용을 지원하고자 한다.

2. TIP 제안 모델

본 장에서는 기존 TIP 배경모델과 본 논문에서 제안된 배경 모델에 대해 설명하고, 기존 연구들에서 3D 배경 구축을 위해 사용되었던 OpenGL 과 더불어 본 논문에 사용된 OpenGL ES Library에 대해 설명한다.

2.1 TIP 배경 모델

Horry 등은 2D 이미지의 소실점 위치로부터 배경의 3D 정보를 복원하여, 한 장의 이미지만으로 그림이나 사진 속으로의 전진이나 비행 같은 시점 이동의 애니메이션을 구현 가능하도록 하였다[2]. 또한 Kang[11] 등은 소실선을 이용하여 배경을 구성하도록 확장된 기법을 제안하였으며, 이는 배경 평면을 5 개로 구성한 [2]의 모델에 비하여 2개로 간단하게 구성할 수 있지만 소실선의 유무가 확실하지 않으며 템플릿을 이용한 전경 객체의 추출이 어렵다. 본 논문에서는 Android platform상의 device의 카메라를 이용하여 촬영한 배경을 이용하기 때문에, 두 방법 중 전경객체 추출을 적용하지 않았을 때 좀더 사실적인 장면 구성이 용이한 소실점을 이용하는 방법을 사용한다.

2.2 소실점을 이용한 TIP

소실점을 이용하는 방법은 전경객체의 추출 없이도 스마트 device에서 3D 배경 구축 시 입체감 있는 배경을 구축할 때 소실선을 이용한 방법 보다 효과적이다. 또한, 제한된 이미지가 아닌 스마트 device에서 촬영한 이미지의 경우 소실점의 유무 및 개수 파악이 힘들다는 점과 스마트 device의 터치 스크린 방식의 편리함을 이용하여 소실점을 대체할 3D 배경의 벽면이 될 한 점을 사용자로부터 입력 받는 방식으로 3D 배경을 구축할 수 있어 배경 구축에 용이하다.

논문에서 제안한 소실점을 이용한 TIP 구현 과정은 먼저, 최초의 입력 영상을 저장하고 사용자로부터 소실점을 대체할 3D 배경의 벽면이 될 사각형을 입력 받은 후 이를 기준으로 벽면으로 사용될 영상을 위해 5개의 벽면으로 자른 후 텍스처 크기 제한에 알맞게 크기를 재조정 한다. OpenGL ES Library 를 이용하여 3D 배경 모델을 획득한 후 카메라의 위치를 초기화하고, 생성된 3D 배경에 준비된 5개의 벽면을 텍스처 매핑하여 모델을 완성한다. 본 논문에서는 android platform 상의 스마트 device 에서의 OpenGL ES Library를 이용하여 TIP 가상공간을 구축함으로써, PC상에서의 OpenGL을 이용한 기존 TIP 기술과 windows mobile OS 상의 mobile device에서의 TIP 기술을 동시에 다양한 android device 응용 개발에 적용 가능하도록 한다.

2.3 구현 환경

PC 상에서는 3D 모델링을 위해 일반적으로 DirectX와 OpenGL[12] 등의 그래픽 라이브러리를 이용한다. 본 논문에서는 PC 환경이 아닌 android platform 상의 스마트 device에서의 TIP 구현을 목적으로 기존 OpenGL의 mobile 버전인 OpenGL ES Library를 사용한다. OpenGL이 워크스테이션이나 고성능 PC와 같은 환경에서 3D 이미지를 렌더링 하는 것을 목표로 하는 것과 달리 OpenGL ES는 부족한 메모리와 낮은 속도의 CPU를 가진 임베디드 환경에 최적화 되었으며, 이는 OS에 비 종속적이며 OpenGL과 유사한 형태의 API와 사용법을 가지고 있다. OpenGL ES는 사각형을 그리는 함수가 제외됨으로써 모든 물체를 삼각형으로 구현하여야 하며, 본 논문에서도 3D 배경을 구성할 때 삼각형으로 배경을 구성하였다. 이와 같이 OpenGL ES는 mobile 환경에서 다양한 3D 작업을 할 수 있는 환경을 제공하며, 이는 플랫폼이나 OS에 의존적이지 않기 때문에 다양한 환경에 쉽게 적용 할 수 있다.

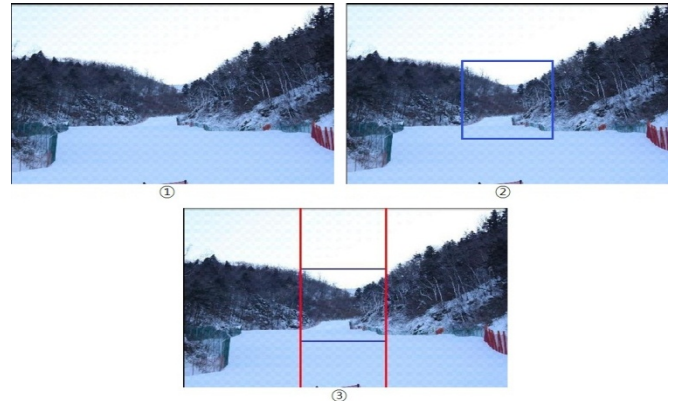
3. Android Platform 에서의 TIP 구현

제안 방법인 android platform상의 스마트 device에서 TIP 기술 구현의 세부과정은 다음과 같다.

3.1 Select vanishing point

일반적으로 소실점을 이용할 경우 여러 개의 소실점이 등장하거나 소실점이 존재 하지 않을 경우 적용에 어려움이 있다. 다음 [그림 1]의 ①은 입력 이미지로써 스마트 device에서 촬영한 사진이다. 스마트 Device에서의 단순한 배경촬영이 아닌 일반적인 사진 촬영 시 소실점이 존재 하지 않거나 여러 개일 경우가 많기 때문에 이를 극복하기 위해 사용자로부터 배경의 모델의 벽면이 될 한 점을 입력 받으며, 이는 다음 [그림 1]의 ②와 같다. 이 때, 선택된 점 주위에 사각형이 생성되며, 사용

자가 사각형의 내부를 터치하여 크기조절 및 드래그를 통해 위치를 변경할 수 있다. 이 사각형 영역은 3D 배경 모델에서 벽면이 될 사각형이다.



[그림 1] Android Platform 에서의 TIP 구현 과정

3.2 Wall texture cutting and resizing

[그림 1]의 ③은 Horry의 소실점 모델을 사용하여 3D 배경에 사용될 텍스처 이미지를 얻기 위해 사용자가 선택한 사각형을 중심으로 5개의 벽면으로 분할한 이미지를 보인다. 5개의 면은 각각 정면, 위, 아래, 좌측, 우측으로 나뉘게 되며, 생성된 3D 모델에 각각 텍스처 매핑된다. 이때 분할된 5개의 텍스처 이미지들이 OpenGL 1.0/1.1 버전에서는 텍스처 크기에 제한을 받으므로, 이를 해결 하기 위해 각 텍스처에 사용될 이미지를 선택한 후 각 이미지의 크기를 조사하여 가장 가까운 2의 승수 크기가 되도록 크기를 조정한다. 즉, 최초 잘라낸 텍스처 이미지의 크기가 30x30일 경우 가장 가까운 2의 승수인 32x32의 크기로 이미지를 조절하여 저장한다.

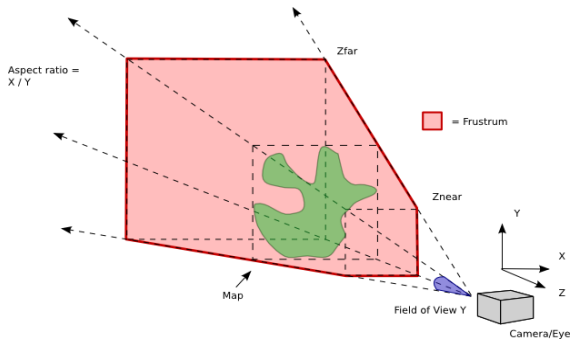
3.3 Background Modeling

배경 모델 구축과정에서 준비된 텍스처 이미지를 매핑하기 위하여 3D 배경을 구축한다. Android Platform 상에서의 그리기 시스템의 핵심은 입력과 출력을 담당하는 View이다. 기본 입출력 시스템인 View와 OpenGL ES를 연결하는 고리는 3D 그래픽 표면을 제공하는 GLSurfaceView 클래스이다. 이 클래스는 View 클래스의 자식 클래스인 SurfaceView의 자식 클래스로 형성된다. 자바 객체의 최상위 클래스인 Object 클래스에서 파생된 View 클래스는 android의 기본적인 입출력을 담당한다. View는 GDI Thread를 통해 화면을 그리기 때문에, 카메라 프리뷰나 동영상 같이 그려지는 양이 많거나 빠른 화면 전환을 위해서는 GDI Thread를 통해서 그리지 않고 SurfaceView를 사용하여 백그라운드를 통해서 그린다. GLSurfaceView는 View 시스템에 OpenGL ES를 연결할 수 있는 연결 코드를 제공하고, 적절한 프레임 버퍼 픽셀 형식을 쉽게 선택할 수 있도록 해주며, 분리된 렌더링 Thread를 생성하고 관리할 수 있는 기능들을 제공해 어플리케이션에서 OpenGL ES 렌더링을 쉽게 사용할 수 있도록 해준다.

3.4 Initial Camera Position

OpenGL ES 의 GLSurfaceview 클래스에서 제공되는 메소드를 이용하여 3D 배경을 구축 한 후, 카메라의 포지션을 초기화 하

기 위해 OpenGL ES에서 제공하는 투영변환 메소드를 사용한다. 객체의 좌표계를 변환하기 위해 투영변환을 하는데, 원근투영은 모든 점들을 투영 중심점(COP:center of projection) 지점과 연결되는 선에 따라 투영하는 방법으로써 직교투영에 비해 입체감이 우수하여 거리에 비례하는 원근감 표현이 가능하다. 다음 [그림 2]는 OpenGL ES에서 제공하는 Frustum과 Perspective 2가지 원근투영 방식을 나타내는 그림이다. 본 논문에서는 시선 각도인 fovy(Field Of View)와 종횡비를 사용하여 시야각을 조절하는 Perspective 방식이 스마트 device에서의 3D 배경 네비게이션에 더 적합하므로 Perspective 방식의 gluPerspective 메소드를 사용하여 카메라 시점을 조정한다. glFrustum 메소드를 사용하면 near와 far 절단면 사이의 거리를 변경할 경우 투영 점의 시야의 각이 아주 크게 변하기 때문에 배경을 네비게이션 하기에는 적당하지 않다. 또한, 어떠한 객체를 보기 위해서 이 거리를 변경하게 되면 시야의 각의 급변으로 인한 절단면 조절이 어렵기 때문에 본 논문에서는 gluPerspective를 이용하여 원근투영을 수행한다.



[그림 2] OpenGL ES의 원근투영

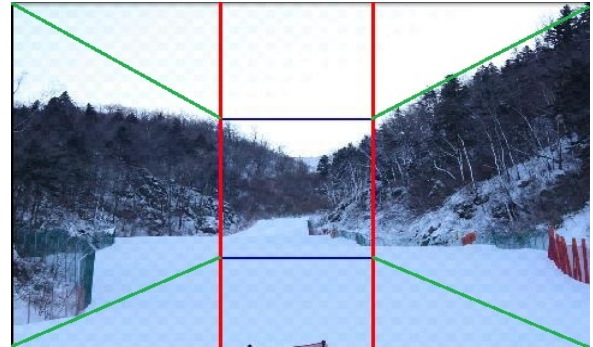
[그림 2]의 Znear 와 Zfar 값은 모두 양수 값으로 설정해야 하며, z축을 기준으로 관측자 눈에서부터의 거리를 의미한다. 그리고, 이 두면은 모두 z=0인 평면에 평행이며 깊이 버퍼에서 렌더링할 z의 최소값과 최대값으로 설정된다. 즉 near평면에서 좌측 끝은 x최소값이고, 우측 끝은 x최대값이며, 하단 끝은 y최소값이고, 상단 끝은 y최대값이 된다.

gluPerspective는 glFrustum과 비슷하지만 상하좌우의 값을 설정하는 대신에 y방향의 시선각도인 fovy와 종횡비인 aspect를 이용하는 점이 다르다. 이때 회전각도는 관측각도, 시야각도로 간주 된다. 시야가 넓어져서 한정된 공간에 그 모든 것을 표현하려면 각 객체의 크기를 줄여야 하는 원리에 의하여 fovy값이 커지면 객체의 크기가 줄어든다. 또한 aspect값이 1이 아닌 값이라면 객체가 좌우나 상하로 늘어나서 표현된다. 시야의 각을 설정한 후 이 메소드를 호출하여 fovy를 조절하여 원활한 네비게이션을 가능하도록 한다. 또한, 영상을 네비게이션할 때 사람이 걷는 듯한 느낌을 주기 위해 카메라 시점의 z 값을 변경하는 것이 아니라, x 값과 y 값도 동시에 변경하여 영상의 확대/축소 시 움직임을 보다 명확히 느낄 수 있도록 구현한다.

3.5 Wall Texture mapping

텍스처 매핑은 보다 사실감 있는 이미지를 생성하기 위하여 미리 생성된 이미지나 텍스처를 3차원 공간의 물체 표면에 매핑하는 과정이다.

배경모델을 구축한 후 텍스처를 매핑할 때 있어서 양쪽 벽면과 천장, 바닥 사이의 경계지점의 정합이 제대로 되지 않는 문제를 해결하기 위하여, 다음 [그림 3]과 같이 좌우 벽면과 천장, 바닥의 텍스처 매핑 과정에서 소실점 벽면의 크기에 비례하여 대각선 방향으로 매핑한다.

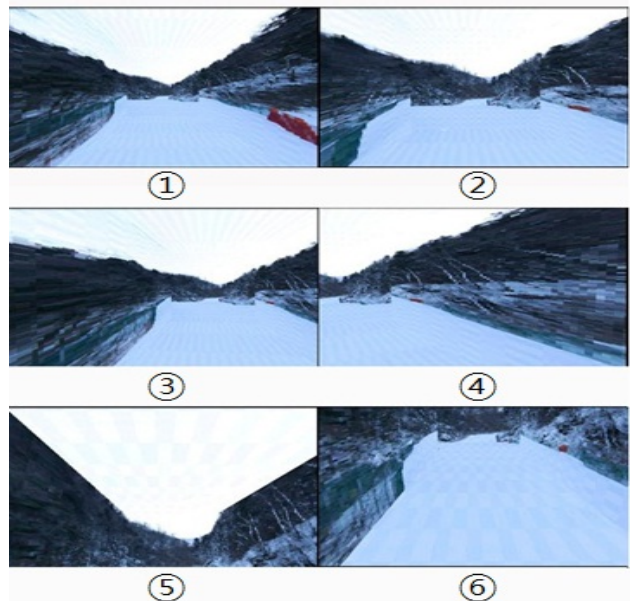


[그림 3] Texture mapping 벽면 처리

텍스처 매핑 구현 시 OpenGL ES Library의 GL10 클래스의 glGenTextures 메소드를 사용하여 텍스처 정점의 배열을 넘겨주고, glBindTexture 메소드를 이용하여 사용할 텍스처 이미지를 배열에 저장한 후 3D 배경의 각 정점과 텍스처의 정점을 1대 1 연결하면 텍스처 매핑이 완성된다. 이 때 OpenGL ES 1.0 버전에서는 반드시 텍스처 이미지의 크기가 2의 승수가 되어야 하며, 매핑 좌표를 지정할 때에는 매핑 대상의 크기에 무관한 S-T 좌표계를 사용한다. 이 때, 카메라의 시점은 기본 정면 벽면을 볼 수 있도록 gluPerspective 메소드를 이용하여 초기화하며, 본 논문에서는 OpenGL ES의 디폴트 값을 사용하였다.

4. 실험 결과

다음 [그림 4]는 완성된 3D 배경 안을 터치 이벤트를 이용하여 이미지를 네비게이션 할 수 있는 구현 결과를 보여준다.



[그림 4] 3D Navigation

[그림 4]의 ①은 3D 배경 구축이 완료된 후 카메라 시점이 초기화 된 화면이며, ②는 카메라 시점을 전진 시켜 영상 속으로 들어 갔을 때의 화면이다(확대). [그림 4]의 ③과 ④는 영상의 시점을 좌, 우로 변경 하였을 때이며, ⑤와 ⑥은 시점을 상, 하로 변경 하였을 때의 화면이다. 완성된 3D 배경은 깊이 값을 가지며 위와 같이 전, 후, 좌, 우, 상, 하의 시점 변경이 가능하며, 입력된 2D 이미지를 최종적으로 3D 배경화 하여 이미지의 다양한 네비게이션이 가능해진다.

5. 결론 및 향후 과제

본 논문에서는 TIP 기법을 이용하여 android platform 에서 한 장의 2D 이미지를 3D 배경 모델로 구축하였다. PC가 아닌 android platform을 채택한 device 에서의 OpenGL ES Library 를 이용한 3D 배경 구현에 있어서, 기존 OpenGL에서 지원되던 기능이 지원되지 않거나 변경된 내용으로 인하여 발생하는 애로사항을 해결하였다. 여러 가지 TIP 관련 기법 중 하나의 소실점 혹은 여러 개의 소실점 존재로 인한 문제를 해결하기 위하여, 본 논문에서는 사용자로부터 소실점 역할을 하는 카메라 시점의 정면 배경이 되는 사각형을 입력 받아 TIP효과를 구현하였다. 또한, android platform상의 device에서 촬영한 이미지를 사용하고 android 2.1 및 OpenGL ES 1.0기반으로 구축함으로써, 스마트 device에서 적은 비용과 시간으로 3D 배경의 TIP 기술을 요구하는 다양한 응용(게임, 엔터테인먼트, 교육, 홍보 등)에 적용 가능하도록 하였다.

향후 연구과제로는 단일 영상에서 3D 배경 구축을 함에 있어서 화질 저하현상의 처리가 요구되며, 스마트 device에서 촬영된 복잡한 사진을 대상으로 적용이 가능한 소실점 검출 알고리즘과 전경분리 및 홀을 채우는 알고리즘에 대한 적절한 알고리즘의 연구가 필요하다. 또한 소실점이 검출 되었을 때 3D배경의 크기를 소실점 검출 위치와 상황에 따라 적절히 조절하는 알고리즘의 연구도 필요하다.

Acknowledgement

이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No. 2009-0074891)과 군산대학교 공학연구소의 연구지원에 의하여 수행된 결과임.

참고문헌

- [1] 윤창옥, 윤태수, 이동훈, “영상기반 모델링 기법을 이용한 대화식 3차원 입체 영상 저작 시스템”, The Korean Society For Industrial And Applied Mathematics, IT series Vol.10, No.2, pp.53~66, 2006.
- [2] Y.Horry, K.Anjyo and K.Arai, “Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image”, ACM PAPH, 1997.
- [3] 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.
- [4] L.Zhang, W James Tam, “Stereoscopic Image Generation Based on Depth Images for 3D TV”, IEEE Transactions on

Broadcasting, pp.2993~2996, 2004.

- [5] 조철용, 김만배, 3DTIP: 한국 고전화의 3차원 입체 Tour-Into-Picture”, 방송공학회논문지, 제14권, 제5호, pp.616~624, 2009.
- [6] Alexey Turitsyn, “Algorithm and 3D application of tour into the picture technique for mobile devices”, 고려대학교 대학원, 2008.
- [7] 김제동, 이광훈, 김만배, “단안 영상의 입체 자유시점 Tour-Into-Picture”, 방송공학회논문지 제15권, 제2호, pp.163~172, 2010.
- [8] 노창현, 정광호, “TIP 기술을 이용한 3차원 배경 영상 구축에 관한 연구”, 한국게임학회 논문지, 제2권, 제1호, 2002.
- [9] 노창현, 이완복, 류대현, 강정진, “넓은 가상환경 구축을 위한 다수의 TIP(Tour into the Picture) 영상 합성”, 전자공학회논문지, 제42권, TE 제 1호, pp.61~68, 2005.
- [10] 조항신, 김창현 “Spidery 메쉬 인터페이스와 뷰 모핑을 이용한 두 이미지로부터의 효율적인 3차원 애니메이션”, 정보과학회논문지:컴퓨팅의 실제, 제7권, 제2호, pp.132~140, 2001.
- [11] H. W. kang et al, “Tour Into the Picture using a Vanishing Line and its Extension to Panoramic Images”, EUROGRAPHICS, Vol.20, No.3, 2001.
- [12] Edward Angel, “OpenGL을 이용한 Computer Graphics”, Addison Wesley, pp. 178~194, 2000.
- [13] android developers, “<http://developer.android.com/reference/android/opengl/GLSurfaceView.html>” .