
사운드 데이터를 이용한 실시간 3차원 지형 생성

Realtime 3D Terrain Generation with Sound Data

김원섭, Wonseop Kim*, 장규식, Kyusik Chang**, 김태용, Taeyong Kim***

요약 ~ 본 연구는 기존에 저장되어 있는 음원이나 입력되는 사운드 데이터를 사용하여 게임 플레이상에 지형을 생성함으로써 사운드의 적극적인 활용을 도모하고 렌더링되는 화면과 입출력되는 오디오 간의 동기화를 통해 사용자들에게 공감각적인 상호작용을 제공한다. 사용자가 직접 사운드를 발생시키거나 원하는 음원을 입력하면 그 입력받은 사운드 진폭 데이터를 푸리에 변환하여 주파수 데이터로 변환하고 취득한 데이터를 분석 및 재가공하여 시간에 따라 변화하는 3차원 지형을 만든다. 지형이 생성될 때는 여러 가지 보간 처리를 통하여 에이전트들과 오브젝트들이 활동 가능하도록 지형을 보정하며, 확장 역시 용이하도록 디자인한다.

Abstract ~ In this paper, the sound data either from the sampled or streamed source are utilized for generating a map in the video game play for the dynamic use of sound data and synesthesia to users. When users can generate sound in real-time or put the sampled source, it is analyzed and re-processed through Fourier transformation to show the 3D map in dynamic shape over time. We interpolate the generated data to enable the game agents and objects to move.

핵심어: 3차원 지형 생성(3d Terrain Generation), 사운드 시각화(Sound Visualization), DSP, 3D 게임(3D Game) 실시간(Realtime.), Frequency(주파수)

본 연구는 ITRC(Information Technology Research Center)와 서울시 산학연 협력사업의 지원으로 수행되었음.

*주저자 : 중앙대학교 첨단영상대학원 영상공학과 석사과정

**공동저자 : 중앙대학교 첨단영상대학원 영상공학과 박사과정

***교신저자 : 중앙대학교 첨단영상대학원 교수; e-mail: kimty@cau.ac.kr

1. 서론

최근 컴퓨터 하드웨어 발전에 힘입어 컴퓨터 게임은 진행적인 측면에서 더욱 복잡해지고, 3D 기반이 되면서 무대가 되는 지형 또한 방대함과 사실성을 요구하고 있다. 90년대의 둌(DOOM)과 같은 몇 명만 참여하는 실내형(indoor) 게임에서 인터넷의 보급으로 인한 리니지와 같은 대규모의 인원이 참여하는 실외형(outdoor) 게임이 최근 들어 계속 등장하고 있다. 이러한 추세에 따라 게임 개발에 있어 방대한 지형의 효과적인 표현에 대한 비중은 점점 높아져가고 있다.

또한 게임에서 음악에 대한 관심 역시 증가하고 있다. 하지만 게임에서의 음악은 시각적인 내용에 덧붙여지는 보조적인 역할을 수행하는데 그치고 있다. 최근 조사의 의하면 실제 게임 이용자들의 게임 시 음악에 대한 관심은 평균적으로 4.9%이며 이 수치는 그래픽이나 스토리에 비해 상대적으로 매우 미약한 것을 볼 수 있으며 그 내용은 표 1과 같다[1].

표 1. 게임이용자별 게임시 가장 관심 있게 보는 요소(1순위)

	온라인 게임	모바일 게임	PC게임	비디오 게임	아케이드 게임
스토리	38.6%	43.5%	41.5%	50.0%	29.2%
기획	14.4%	12.9%	17.6%	3.3%	16.7%
사운드	2.2%	1.6%	0.7%	0.0%	0.0%
그래픽	20.7%	19.4%	19.0%	26.7%	20.8%
동영상	4.4%	3.2%	7.0%	3.3%	12.5%
프로그래밍 (디버깅)	1.6%	4.8%	2.8%	0.0%	0.0%
연출력 (구성)	7.1%	6.5%	5.6%	13.3%	4.2%
홍보/마케팅	0.5%	3.2%	0.0%	0.0%	0.0%
가격 경쟁력	2.2%	0.0%	1.4%	0.0%	4.2%
서비스 수준	4.7%	3.0%	1.4%	3.3%	8.3%
기타	0.9%	0.0%	1.4%	0.0%	4.2%
없음	2.7%	1.6%	1.4%	0.0%	0.0%

따라서 본 연구는 기존에 저장되어 있는 음원이나 입력되는 사운드 데이터를 사용하여 게임 플레이상에 지형을 생성함으로써 사운드의 적극적인 활용을 도모하고 렌더링되는 화면과 입출력되는 오디오 간의 동기화를 통해 사용자들에게 공감각적인 상호작용을 제공하는데 목적이 있다. 이를 위해 우선 입력받은 사운드 진폭 데이터를 푸리에 변환하여 주파수 데이터로 변환한다. 이후 취득한 데이터를 분석 및 재가공하여 시간에 따라 변화하는 3차원 지형을 만든다. 지형이 생성될 때는 여러 가지 보간 처리를 통하여 에이전트들과 오브젝트들이 활동 가능하도록 지형을 보정하며, 확장 역시 용이하도록 디자인한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 지형 생성 방법들과 사운드 데이터를 활용하여 시각화한 연구

들에 대해 각각 분류한 뒤 연구의 내용과 문제점에 대하여 알아본다. 3장에서는 제안한 사운드 데이터를 활용한 게임 지형 생성 시스템에 대한 구현 방안에 대하여 기술한다. 4장에서는 제안한 시스템을 토대로 기존의 지형 생성기법에 의해 생성된 지형을 성능을 비교하며, 마지막으로 5장에서는 결론 및 향후 과제에 대해 논의한다.

2. 관련연구

연제혁 등[2]은 기존의 게임지형 생성 방법을 크게 세 가지로 나누어 응용프로그램을 이용한 방법, 실제 지형 데이터를 사용하는 방법, 마지막으로 지형 자동 생성 알고리즘을 이용하는 방법으로 나누었다.

3차원으로 구성된 실외 게임 지형은 주로 데이터 처리가 쉬운 하이트 맵(height map)을 사용한다. 하이트 맵이란 지형을 일정한 격자 간격으로 나누어 그 높이 정보만을 기억하고 있는 자료구조로써 규칙적인 격자 형태로 정렬된 높이 값을 2차원 배열이다. 이 격자에 있는 각 x, y의 위치에 따라 z의 값을 저장하며, 이 z값이 x, y의 좌표 값을 높이 값이 된다. 하이트 맵 기반의 지형은 자유로운 지형 표현에 한계가 있지만 빠른 시간 내에 지형의 높이 정보를 찾을 수 있으며, 쉽게 시각화할 수 있다는 장점으로 게임 지형에 많이 사용되고 있다[2][3].

하이트맵 생성 방법은 크게 3가지로 나누어 볼 수 있으며 첫 번째는 응용프로그램을 이용한 방법, 두 번째는 실제 지형데이터를 이용하는 방법, 세 번째는 지형 자동생성 알고리즘을 이용하는 방법이다.

응용프로그램을 이용하여 지형을 생성하는 방법은 게임 엔진에 포함되어 있는 맵 에디터 또는 일반 모델러 프로그램(3ds MAX, MAYA 등)이나 2D 그래픽 프로그램(Photoshop 등)을 이용하여 제작하는 것으로 제작자가 조작함으로써 변화하는 환경을 시각적으로 보면서 손쉽게 제작할 수 있도록 도와준다. 그러나 게임 디자이너에 의해 게임 컨셉에 맞는 지형을 만드는데 있어 많은 시간을 소비한다. 또한 다양한 맵에서 일회성 게임을 즐기는 온라인 전투 게임과 같이 실시간으로 지형의 변화를 요구하는 게임에서는 활용에 어렵다.

비행 시뮬레이션 게임과 같은 경우 사실적인 지형을 필요로 하기 때문에 실제 지형 데이터를 이용하기도 한다. 이러한 지형자료는 크게 등고선에 의한 방법과 수치고도 모델(DEM)에 의한 방법으로 나눌 수 있다. 등고선 방법은 추출 및 보간하는 과정에서 많은 오차가 발생하기 때문에 정확성이 떨어지는 단점이 있다. 그러나 수치고도모델은 지형의 고도데이터를 실측한 뒤 레스터 데이터 형태로 만든 것으로 지형의 높이를 나타내는 수치들의 연속된 값을

나타낸다. 그러나 실제 지형 데이터를 이용하는 방법은 비용이 많이 소요되고, 자료를 구하기가 쉽지 않다는 단점이 있으며, 무엇보다 게임 디자이너가 레벨 디자인을 함에 있어서 기존 데이터에 의존적으로 디자인할 수밖에 없다.

자동생성 알고리즘을 이용한 방법에는 난수에 의해 생성되는 퍼지 랜드스코핑, 스무싱, 파울트 라인 알고리즘 등이 있다. 이 방법들은 난수 시드(random seed)를 사용하여 지형을 생성하기 때문에 너무 비사실적인 지형을 생성한다. 다음으로 펠린 노이즈 기반 알고리즘 방법이 있다 [4]. 펠린 노이즈는 함수를 만들기 위해서는 노이즈 함수와 보간 함수가 필요하다. 노이즈 함수는 본질적으로 시드 값에 의한 난수 생성기이다. 전달 인자로 하나의 정수를 받아 전달인자를 기반으로 둔 난수를 생성하고 생성된 값을 부드럽게 보간함으로서 전달 인자를 정수로 받지 않은 연속적인 함수를 정의할 수 있다. 이렇게 생성된 값을 2차원 형태로 표현하여 자연스러운 형태의 하이트맵 이미지를 획득할 수 있다.

마지막으로 프랙탈 지오메트리 모델 기반 알고리즘은 프랙탈의 특성을 이용하여 좀 더 사실적인 지형을 생성하고자 하는 알고리즘이다. 프랙탈은 자기 복제 성질과 무한대 순환 반복의 두 가지 성질을 가지고 있다. 프랙탈의 기하학은 다음과 같은 간단한 절차에 의해 정량적으로 표현된다.

$$P_1 = F(P_0), P_2 = F(P_1), P_3 = F(P_2), \dots \quad (1)$$

※ $F = \text{TransformationFunction}$
(Regular or Random variations)

세부적으로는 단층 변환 기법, 중점 변위 기법, 입자 퇴적 기법이 제안되었으며 프랙탈 지형의 3D 렌더링 결과는 다음과 같다[3].



그림 1. 단층 변형(좌측), 중점 변위(중간), 입자 퇴적(우측)

사운드 데이터를 시각적으로 변화시키는 연구에서는 스펙트로그램을 이용하여 사운드 데이터들을 시각화하는 연구도 있었지만 이는 어디까지나 음악을 해석하기 위한 도구였을 뿐 다른 범위로 확장하는 데에는 무리가 있었다 [5].

비주얼라이저를 통하여 사운드데이터를 시각화하는 방법도 있었지만 그 패턴이 생성되는 근거가 개발자의 주관적인 견해였으며 그 근거조차 매우 미약하였다[6].

MIR(Music information Retrieval) 분야에서 악보의 동일한 패턴을 동일하게 표시해 시각화하는 방법이 연구되었지만 이는 음악을 기계적으로 분석하기 위한 수단으로 활용되고 있으며 사용자들이 이러한 기호를 통해 정보를 전달하는데 사용하고 있기 때문에 음악을 이해하는데 전문지식을 필요로 하였다[7].

또한 컴퓨터 게임에서의 오디오와 비디오의 동기화에 대한 연구도 진행되었으며, 동기화는 임계값으로 40ms(±20ms)에서 이루어져야 사용자 인터랙션을 보장할 수 있다고 보고하였다[8].

3. 사운드 데이터를 이용한 실시간 3차원 지형 생성 시스템 구현

구현하는 시스템은 사운드 데이터를 받아서 하이트맵 데이터로 재생성한 다음 렌더링하여 출력하기 때문에 사운드 처리와 그래픽 처리가 유기적으로 이루어져야 한다.

기본적으로 제안하는 시스템은 지형의 한 부분부터 순차적으로 변화하도록 설계되어 있다. 이 시스템의 전제는 다음과 같다.

- 전제 1. 사운드 데이터는 디바이스에 제한이 없어야 한다.
- 전제 2. 시스템은 모듈화되어 있고 확장성이 보장되어야 한다.
- 전제 3. 시스템은 다채널을 지원해야 하며, 사용자 환경을 반영해야 한다.

이 시스템의 전반적인 흐름은 그림 2에 나타나 있다.

3.1 사운드 취득

사운드 데이터는 두 가지 방식으로 취득한다. 마이크로폰 등의 디바이스를 통해 들어온 사운드를 수음하는 것과 wav, mp3 등의 디지털 포맷으로 저장되어 있는 음원 데이터를 그대로 취득하는 방식이다.

마이크로폰을 통해 사운드를 수음하는 경우 1차적으로 중요한 것은 사운드를 어떻게 선택해서 수음하는 것인가이다. 구현된 시스템은 2채널 이하의 사운드를 반영하도록 설계되어 있다. 따라서 수음에 있어서 2채널 이하로 수음을 하여야 한다. 또한 사용자가 원하는 음향이 있을 경우 그 음향을 받아들일 수 있도록 하여야 한다. 하지만 대부분의 사용

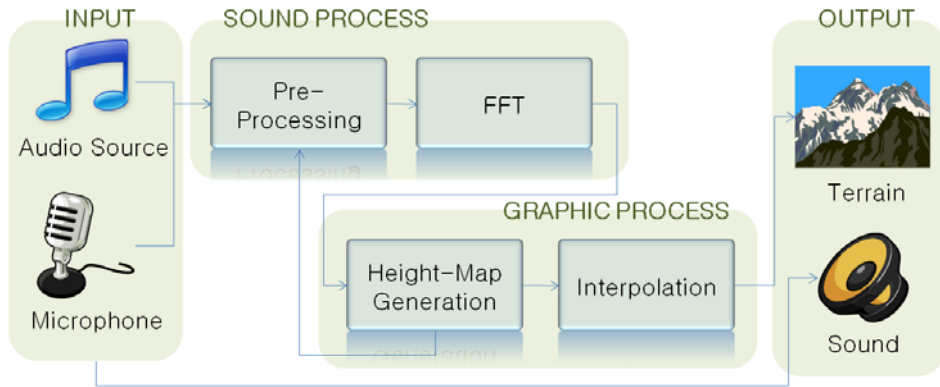


그림 2. 엔진 프로세싱

자들은 1채널 마이크를 사용하기 때문에 범용적인 시스템을 구축하기 위해서는 이 역시 고려해야한다.

3.2 사운드 데이터 처리

사운드 데이터는 전처리 과정을 거치게 된다. 실제 음원은 다양하다. 이리 저장되어 있는 음원이라면 샘플 레이트와 비트심도, 채널 수 등이 다양하고 그런 다양한 변수들을 단 일화해야 한다.

사운드 데이터 처리를 위해서는 그 데이터를 얼마만큼 나누어 높이 정보 변환 모듈에 보내느냐 고려해야 한다. 제한하는 시스템에서는 사운드 데이터를 받아들이는 버퍼의 크기를 1024byte의 크기만큼으로 제한하여 변환모듈로 전송한다. 또한 사운드 입력 모듈에서 너무 많은 정보가 들어올 시를 고려해 일정부분을 제거해주는 처리가 반드시 필요하다.

전처리가 끝난 사운드 데이터에서 어떤 부분을 지형의 높이 정보로 사용하는가에 있어서 여러 가지 견해가 있을 수 있다. 기존의 오디오에 포함된 정보를 시각적으로 표현하는 것은 음량, 웨이브 폼, 주파수 스펙트럼 정보를 활용한다.

음량 정보는 오디오의 전체적인 음량감을 말한다. 음악을 시각화하려는 시도의 전반적인 분야에서 사용되는 것으로 흔히 볼 수 있는 비주얼라이저들이 많이 채용하고 있다. 이는 가장 청중들이 쉽게 변화를 이해할 수 있기 때문이다.

웨이브폼은 흔히 시간 축에 사인곡선으로 표시되는 그래프로 분석이 가능하다. 순차적인 분석이 가능하기 때문에 대부분의 오디오 편집 도구에서는 웨이브폼으로 이루어진 그래프를 조작하여 음악을 조각내고 편집한다. 하지만 전문가가 아닌 일반 사용자가 이 데이터를 통해 음악을 이해하기 어렵고 1차원 정보이기 때문에 지형의 높이 정보에 적용하기에는 문제점이 많다.

주파수 스펙트럼은 사운드의 주파수 분포를 가장 잘 알

수 있는 방법으로 사운드의 배음구조 및 저주파, 고주파에 대한 판정을 명확하게 할 수 있다. 하지만 주파수 스펙트럼 역시 2차원 정보이기 때문에 스펙트로그램의 개념을 사용하여 분절된 스펙트럼을 시간에 따라 순차적으로 표현한다면 선형적인 표현이 가능하다.

따라서 이중에 주파수의 정보를 활용하여 하이트맵을 생성하도록 한다. 주파수 정보를 이용하면 시간에 따라 순차적으로 변하는 고주파에서부터 저주파까지로 이루어진 지형을 생성해 게임에 적용이 가능하기 때문이다.

버퍼 상의 사운드 진폭 데이터는 푸리에 변환을 적용하여 주파수 데이터로 변환한다. 이 때, 주의할 것은 푸리에 변환을 적용하기 위하여 버퍼 크기는 2의 자승배이어야 하며 기존의 언급하였던 1024byte 역시 2의 자승배라는 조건을 만족하고 있다.

3.3 지형 생성

지형 생성 단계에서는 주파수 정보의 데이터를 사운드처리 모듈에서 받아들여 하이트맵을 생성한다. 주파수데이터를 시간축으로 나열해 나열하기 때문에 한쪽에서는 계속 데이터가 생성되고 반대쪽에서는 데이터가 소멸한다. 하지만 메모리의 한계 및 연산량으로 인해 새로 생성하고 소멸하기 보다는 기존 하이트맵 정보에서 높이 정보만을 변경하여 실제로 생성되지 않지만 생성되는 것처럼 보이게 하는 방식으로 지형을 생성한다. 이에 대한 정리는 다음과 같다.

하이트 맵의 인덱스가 0이 아닌 정점 옆에 대해
 현재 인덱스의 정점 옆 = 이전 인덱스의 정점 옆

하이트 맵의 인덱스가 0인 정점 옆에 대해
 하이트 맵의 인덱스가 0인 옆
 = 현재 버퍼에 있는 주파수 데이터

3.4 보간

높이를 지닌 하이트맵이 생성되었지만 이를 실제 게임에 적용하기는 무척 어렵다. 캐릭터가 이동할 수 있는지 여부와 주파수에 의한 급격한 피크치 변화가 여전히 상존하기 때문이다. 기존의 지형을 보간하기 위해서는 3차 큐빅 스플라인 곡선을 사용하는 방법이 제안되었다 [2]. 하지만 이 방법을 본 시스템에 적용하기에는 연산량이 너무 많고 순차적으로 변화하는 지형에 적용할 경우 기존 지형까지 영향을 주기 때문에 적합하지 않았다. 따라서 제안하는 보간 방법은 간단하면서도 연산량을 줄일 수 있어 비교적 빠르게 보간할 수 있는 방법으로 수식은 다음과 같다.

$$v(a,b) = (v(a,b) + v(a-1,b) + v(a+1,b) + v(a,b-1)) / 4 * \log_c b * w \quad (2)$$

여기서 $v(a,b)$ 는 하이트맵 상에서 a열 b행에 위치 정점이다. 여기서 연산은 저주파에서 고주파로 일어나며, 로그값으로 필터를 적용함에 따라 원래 값에 비해 저주파의 높이값은 낮아지고 고주파의 높이 값은 올라가는 현상을 볼 수 있다. 이는 상대적으로 저주파의 정보가 많이 들어오는 자연계의 특성을 반영한 것이다. w 는 높이보정을 위한 가중치로서, 실제 가중치만을 아주 낮게 적용하여도 평탄화되는 효과를 볼 수 있다. 하지만 여전히 돌출된 부분이 많기 때문에 양 옆 정점의 높이와 바로 이전 정점 높이 정보 및 현재 높이 정보의 평균값을 내어 갑작스러운 돌출을 막는다.

4. 실험결과

개발은 AMD Athlon 64 3200+ 에 2G Memory, Geforce 6600gt가 설치된 데스크탑과 Fast Track Pro와 지향성 마이크인 Sennheizer ME 66으로 구성된 수음 설비를 갖추어 진행하였으며, directx 9.0 sdk와 C#.net 기반으로 하여 작업하였다. 설치 환경은 그림 3과 같다.



그림 3. 실험 환경

우선적으로 고려해야 될 것은 실시간으로 적용이 가능한지 여부이다. 실시간 적용의 기준은 15frame 이상에서 퍼포먼스를 보여야 한다고 정하였다[9]. 제안하는 시스템은 정점 갯수가 256X256으로 이루어진 40만개 이상 폴리곤의 지형에서 40frame이상의 퍼포먼스로 6ms 이하의 화면 출력 속도를 보였다.

캐릭터 이동성을 보장하는 경사도는 기존의 엔진을 파악하여 45°이하 일 경우라고 정의되었다[2]. 따라서 제안하는 시스템은 보간을 통하여 45도 이하의 경사를 생성하였으며 대부분의 지형에서 게임 캐릭터의 이동성을 보장 할 수 있었다. 하지만 캐릭터 이동성에 입각해 입력되는 사운드의 정보를 파악하는데 무리가 발생하는 것을 막기 위해 파라미터 수준에서의 조절이 필요하다. 그 결과 캐릭터나 오브젝트가 이동하기 적합한 환경이 만들어 졌고, 그 결과는 그림 4와 같다.



그림 4. 게임 상 적용화면

또한 대부분의 게임들이 시나리오 배경에 따라 파라미터를 변경하며 지형을 생성하고 있기 때문에 본 시스템도 재질의 종류에 따라 차등적인 하이트 맵을 생성하도록 하였다. 각각 빙설, 사막, 바다라는 재질적 설정을 놓고 하이트맵의 높이와 완만함을 결정하는 파라미터를 차등적으로 적용하여 게임 시나리오에 대응할 수 있는 기반을 마련하였다.



그림 5. 재질 종류에 따라 차등적으로 생성된 지형(왼쪽부터 시계방향으로 빙설, 사막, 바다)

5. 결론

본 연구를 통하여 사운드 데이터를 이용해 지형을 실시간으로 생성하는 시스템을 제안하였다. 또한 이를 이용하여 여러 게임 장르에 사운드를 통한 지형 디자인의 가능성을 보였다.

하지만 물리 엔진이 적용되지 않아 단순히 지형을 보여주는 수준에 머물렀고 디바이스가 어느 정도의 대역폭으로 사운드를 받아들일 수 있는가에 대한 고려가 제외되어 있기 때문에 일부 사용자들이 아주 낮은 음이나 아주 높은 음을 내어도 시스템 상에서는 처리가 되지 않을 수 있다. 따라서 사용자 측면에서의 수음 시 사용되는 장치 콘텍스트를 알아 이를 반영하는 것 역시 필요하며 이에 대하여 연구해 나갈 것이다.

참고문헌

- [1] (재)한국게임산업개발원 산업정책팀, 2006 대한민국 게임백서, (재)한국게임산업개발원, 서울, 한국, 2006, pp. 418~419.
- [2] 연제혁, 김성수, 임형준, 이원형, "게임 난이도를 고려한 게임지형 자동생성 기법에 관한 연구", 한국인터넷정보

학회 논문집, 제5권, 제2호, 한국인터넷정보학회, pp. 447~481, 2004.

- [3] Lecky-Thompson, Guy W, "Algorithms for An Infinited Universe", Gamasutra, 1999
- [4] Perlin Noise, http://freespace.virgin.net/hugo.elias/models/m_perlin.htm
- [5] Serra, X. Smith, J. 1990. "Spectral Modeling Synthesis:A Sound Analysis/Synthesis Based on a Deterministic plus Stochastic Decomposition," Computer Music Journal Vol.14 .4, pp. 12-24.
- [6] Misra, A., G. Wang, and P. R. Cook. "SndTools: Real-time Audio DSP and 3D Visualization" In Proceedings of the 2005 International Computer Music Conference (ICMC), Barcelona, Spain, September 2005. 예
- [7] Bryan Pardo and William P. Birmingham, "Algorithms for Chordal Analysis", Computer Music Journal, Vol. 26:2, pp. 27-49, 2002.
- [8] Peter Ward, Slawomir Zielinski, and Francis Rumsey, "Can playing a computer game affect perception of audio-visual synchrony?", in *Proceedings of 117th Audio Engineering Society Convention*, San Fransisco, CA, USA, October 2004, Paper 6224.
- [9] Tomas Akenine-Moller, Eric Haines, "Real-Time Rendering", AK Peters Ltd, 2002