Journal of Digital Contents Society Vol. 15 No. 4 Aug. 2014(pp. 457-464)
http://dx.doi.org/10.9728/dcs.2014.15.4.457

3차원 공간자료의 저장 공간 최적화를 고려한 텍스쳐 생성기법 연구

진기호*, 하성룡**

요 약

국토교통부 공간정보 오픈플랫폼으로 인해 3차원 공간정보자료에 대한 관심이 증가하고 있다. 본 연구는 3차원 공간정보자료 중 텍스쳐 자료를 효율적으로 관리 및 저장에 관한 것으로, 다방향 항공 카메라 및 수직 항공 카메라에 의해 획득된 원시영상을 이용하여 항공삼각측량 및 3차원 저작을 통해 제작된 3차원 공간자료를 추출하고, 이 자료로부터 해당 텍스쳐 자료를 최적 저장 공간에 정렬 및 배치하여 단일 텍스쳐 자료를 생성하는 방안을 연구하고 관련 기법을 개발하였다. 실험 결과, 기존 방안인 단일 파일 저장 방식에 비해 최대 1/8배에 상응하는 용량 축소 효과가 있었으며, 다중 파일 저장 방식과 비교하였을 때 파일 관리 효율성을 증진시킬 수 있음을 최종 확인하였다. 본 연구 성과는 다수의 3차원 공간정보자료를 효율적으로 관리하기 위한 초석이 될 것으로 판단되며, 향후 알고리즘의 개선 및 추가 연구를 통해 활용 방안이 증대될 것으로 사료된다.

키워드 : 3차원 공간정보자료, 공간정보 텍스쳐링 기법

Texture Image Generation Technique Considering Storage Optimization of 3D-Spatial Data

Gi-Ho Jin*, Sung-Ryong Ha**

Abstract

Recently, interests in space information data are increasing due to the initiation of spatial information open platform service by the Ministry of Land, Infrastructure and Transport. The purpose of this study is optimizing management and storing of the texture data, one kinds of 3D-spatial data. First, extract 3D-spatial data through the aerial triangulation and 3D-writing using raw image taken with the Multi-directional aerial camera and the vertical aerial camera. And develop the method to create single texture data and related technique by align and place corresponding 3D-spatial data to optimal storage space. Through experiment, the results show effect of 8 times of storage capacity reduction compared to existing single-file storage method, additionally, new method can improve file management efficiency in comparison with multiple file storage method. The results of this study can be cornerstone of three-dimensional space information management when dealing with bulk data, and utilizations will be enhanced through the further studies and algorithm improvement.

Keywords : Texture of 3D Spatial data, Storage Efficiency of texture images

1. 서론

국토교통부에서 개시한 공간정보 오픈플랫폼은 미국 구글社에서 제공하는 “구글 어스” 서비스와 유사한 한국형 3차원 대민 지도 서비스이

※ 교신저자(Corresponding Author): Sung-Ryong Ha
접수일:2014년 07월 04일, 수정일:2014년 07월 30일
완료일:2014년 07월 31일
* 충북대학교 환경·도시공학과
Tel: 010-9246-2766 , Fax: 02-732-0999
email: ghjin124@naver.com
** 충북대학교 환경·도시공학과
▣ 이 논문은 2012학년 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음

다. 2012년 공식 서비스를 개시한 공간정보 오픈 플랫폼은 우리나라의 주요 도심지나 관광지와 같은 랜드마크성 지역을 3차원 공간정보자료로 제작한 뒤, 이를 인터넷을 통해 서비스함으로써 가상의 현실 세계를 일반인도 쉽게 활용할 수 있도록 제작한 지도 서비스이다[3, 10]. 서비스에 사용된 3차원 공간정보자료는 고정밀의 3차원 공간정보 제작을 지원하는 사진측량기법에 의해 구축된 것으로써, 고해상도의 항공사진과 지상에서 취득한 사진 및 3차원 좌표 정보를 사용하여 제작된다[6, 7].

사진측량기법에 의해 초기 제작된 3차원 공간정보자료는 현실에 존재하는 인공 지물과 형상만이 동일하고 색상은 부여되지 않는다. 따라서 대상물을 촬영한 항공사진 혹은 지상사진을 해당 공간정보자료에 입히는 텍스쳐 부여 작업을 수행하여야 현실과 동일한 3차원 공간정보자료 제작이 가능하다[1, 5, 7, 8, 9]. 한편, 각 인공 지물에 부여된 사진 즉, 텍스쳐는 해당 3차원 공간정보자료와 함께 전자 파일 형태로 저장되는데, 인공 지물의 개수가 많아질수록 전체 파일 용량이 증가하고 관리해야할 텍스쳐의 수가 급증한다. 즉, 공간정보 오픈플랫폼과 같이 공간정보자료의 관리 대상이 우리나라 전체가 될 경우, 각 공간정보자료에 대응되는 텍스쳐 이미지들은 그 개수를 헤아릴 수 없으며 전체 용량 또한 테라바이트 급에 상응한다. 따라서 텍스쳐 이미지들을 효율적으로 관리하고 용량 또한 축소시킬 수 있는 방안 연구가 필수적이다.

이에, 본 연구에서는 3차원 공간정보자료의 텍스쳐 저장 방안을 연구하고 새로운 방법을 도출함으로써 다수의 텍스쳐 파일을 효율적으로 관리하고 저장 공간을 최소화할 수 있는 최적의 텍스쳐 저장 방안을 도출하였다.

2. 텍스쳐 정보 저장 방식

현재 3차원 공간정보자료는 Autodesk社 3D Max 소프트웨어의 전용 파일 포맷인 3DS (*.3ds)를 준용하고 있는데[2, 4], 객체 정보를 3차원으로 관리하고 사용자가 원하는 텍스쳐를 원하는 위치에 부여하기가 용이하며 이를 지원하는 타 소프트웨어가 다양하기 때문이다. 3D Max의 경우, 각각의 3DS파일에 대해 사용자가 입력한 텍스쳐 이미지를 그대로 저장한다. 따라서 사용자가 한 3DS 파일에 대해 여러 장의 텍스쳐 이미지를 입력하게 되면, 하나의 3DS 파일이 여러 개의 이미지 파일을 가지게 된다. 만약 입력한 텍스쳐 이미지의 영역 중 일부만을 사용할 경우, 해당 위치에 대한 좌표 정보를 3DS 파일 내부에 별도로 저장 및 관리함으로써 이를 화면에 도시하게 된다.

3D Max 소프트웨어의 텍스쳐 저장 방식에 따라 공간정보자료를 제작하게 되면, 추후 입력된 텍스쳐 이미지 파일 개수만큼 관리해야하는 단점이 존재한다. 또한, 간단히 텍스쳐만을 수정하면 편집을 완료할 수 있는 상황에서도 해당 텍스쳐가 저장된 이미지 파일이나 위치가 명확치 않아 일일이 검색하거나 다시 3D Max 소프트웨어를 실행하여 편집해야하는 번거로움이 있다.

한편, 3D Max 소프트웨어의 방식과는 달리 사용자가 입력한 여러 개의 텍스쳐 이미지를 하나의 파일에 일렬로 복사함으로써 텍스쳐를 관리하는 방식이 존재한다. 이는 하나의 파일에 텍스쳐를 모두 저장하기 때문에 파일 관리에 매우 용이한 방법이다. 그러나 텍스쳐에 사용되는 이미지 파일은 서로 크기가 다른 경우가 대부분이므로 이로 인해 아무런 정보도 담기지 않은 빈 공간이 발생하게 된다. 이는 이미지 파일의 특성에 의한 것으로 직사각형 형태의 2차원 배열 저장만을 지원하기 때문인데, 이때 발생한 빈 공간만큼 불필요한 용량이 기존의 용량에 더 가산되는 단점이 있다.

본 연구에서는 앞서 언급한 텍스쳐 저장 방식을 통해 단일 파일 저장 방식을 선택하는 것이 관리 측면에서 가장 용이할 것으로 판단하였다. 단, 이로 인해 발생하는 용량 낭비 문제가 발생하므로, 이를 해결하기 위한 방안으로 파일 내 텍스쳐 이미지 배치 최적화를 모색하였다.

3. 최적 텍스쳐 관리 및 저장 방안

3.1 초기 저장 공간 설정

3차원 공간정보자료의 텍스쳐 처리는 면의 개수에 따라 적게는 10개 이하에서 많게는 수백

개로 처리된다. 이렇게 처리된 텍스쳐 자료를 최적 저장 공간에 저장하기 위해서는 초기 저장 공간을 설정하여야 한다.

초기 저장 공간의 가로크기는 처리된 텍스쳐 자료의 크기(면적)를 기준으로 결정하고, 세로크기는 텍스쳐 자료의 가장 큰 세로크기를 기준으로 설정한다. 여기서, 텍스쳐 저장 공간은 원활한 3차원 공간 자료의 도시를 위해 2^n으로 저장된다. 또한 최종 저장 공간은 각 텍스쳐 배치를 완료한 후에 결정된다. 식 (1)은 초기 저장 공간의 가로크기와 세로크기를 설정하는 것을 나타낸다.

$$X_0 = \sqrt{\sum_{i=1}^{count} (x_i \times y_i)} \qquad (1)$$

여기서, $count$: 텍스쳐 처리 개수,
x_i , y_i : 각 텍스쳐의 가로 및 세로크기

초기 저장공간의 가로 크기는 $X_o < 2^n$ 을 만족하는 최초 2^n 값이며 초기 저장 공간의 세로크기는 $Max(y_i)$ 값이다.

3.2 텍스쳐 정렬

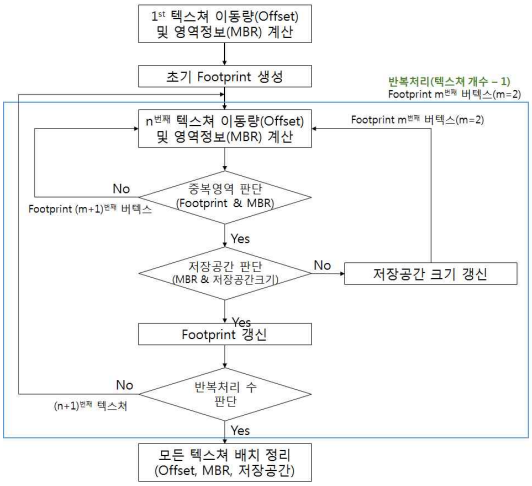
처리된 텍스쳐 자료는 임의의 순서에 의해 생성된 자료이므로, 최적 저장 공간을 고려하기 위해서는 텍스쳐의 크기(면적) 순서 즉, 내림차순으로 정렬이 필요하다.

여기에서는 가장 간단한 알고리즘 방법을 사용하였으며, 처리절차는 다음과 같다. 먼저, 텍스쳐 면적에서 가장 큰 값을 찾아서 그것을 첫 번째 위치에 있는 텍스쳐와 교환을 한다. 그 후 두 번째로 가장 큰 면적을 찾고, 두 번째 위치에 있는 텍스쳐와 교환을 하고 그리고 전체 텍스쳐 수가 끝날 때 까지 이 같은 방법을 반복하여 처리함으로써 해결한다(그림 1).

3.3 텍스쳐 배치

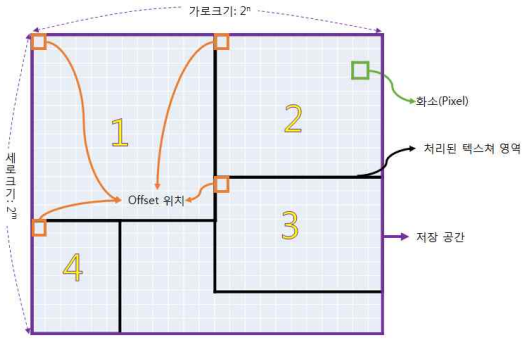
처리된 텍스쳐 자료의 저장 공간을 최소화하기 위해서는 정렬된 텍스쳐 자료를 초기 저장 공간상에 최적의 위치에 배치하여야 한다. 최적의 텍스쳐 배치는 다음과 같다. 우선, 초기 저장 공간상의 좌측 상단 즉 원점(0,0)에 첫 번째로 정렬된 텍스쳐를 배치하고 배치된 영역(Footprint)을 저장한다. 그 후 정렬된 두 번째 텍스쳐 자료는 배치된 영역의 버텍스를 기준으로 이동량(Offset)과 텍스쳐 경계를 설정한다. 설정된

(그림 1) 텍스쳐 정렬 흐름도



(Figure 1) Flow chart of Texture arrangement Process

(그림 2) 텍스쳐 배치 순서 샘플 1



(Figure 2) Example 1 of Ordering Textures

두 번째 텍스쳐 영역 정보는 앞서 계산된 텍스쳐 경계와 겹치지 않으면 두 번째 텍스쳐를 배치하고 텍스쳐 영상경계를 갱신한다. 만약, 앞서 설정된 텍스쳐 경계와 겹치면 다음 버텍스로 이동하여 이동량와 배치될 영역정보를 재설정하여 앞서 설정한 텍스쳐 경계와 겹치지 않을 때 까지 위 과정을 반복 수행한다. 이 후의 텍스쳐 자료의 배치는 이 같은 방법을 반복하여 처리한다. 특히, 배치될 텍스쳐 자료의 이동량 및 배치될

영역정보(MBR, Minimum Boundary Rectangle)는 다음과 같이 설정한다.

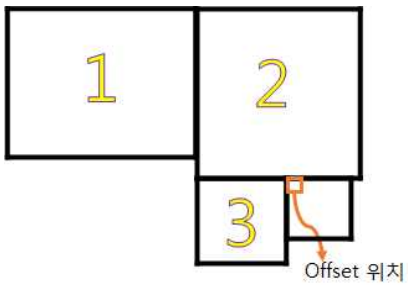
① 첫 번째 텍스쳐 자료 이동량
(그림 2)에서 텍스쳐 배치 순서의 “1” 항목을 나타내며, 이동량 값은 (0, 0)으로 저장된다.

② 두 번째 이후의 텍스쳐 자료 이동량
배치된 영역의 버텍스 값을 활용하여 이동량을 처리하며, 처리되는 유형은 다음과 같다.

i. 버텍스의 x 혹은 y좌표가 0 일 경우, 아래와 같이 이동량을 계산한다.
(a) x = 0 일 경우 즉, (그림 2)에서 “2” 항목에 해당하면 버텍스의 x좌표에 크기 1만큼을 증가시키고, y좌표는 버텍스의 y좌표를 그대로 사용
(b) y = 0 일 경우 즉, (그림 2)에서 “4” 항목에 해당하면 x좌표는 버텍스의 x좌표를 사용하고, y좌표는 버텍스의 y좌표에 크기 1을 증가시킴

ii. 현재의 버텍스와 다음 버텍스의 관계에 따라 이동량을 계산한다.
(a) 두 버텍스의 x좌표의 차이가 0 이고, y좌표의 차이가 1보다 클 경우(그림 3), 버텍스의 x좌표에 1을 증가시키고 버텍스의 y좌표를 그대로 사용함

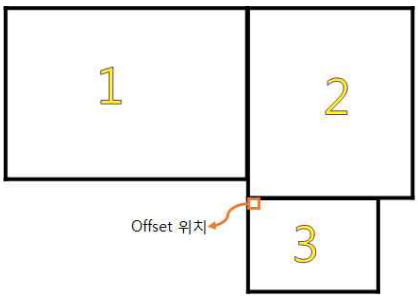
(그림 3) 텍스쳐 배치 순서 샘플 2



(Figure 3) Example 2 of Ordering Textures

(b) 두 버텍스의 x좌표의 차이가 0 이고, y좌표의 차이가 -1보다 작을 경우(그림 4), 버텍스의 x좌표에 1을 증가시키고 버텍스의 y좌표는 그대로 사용함
(c) 두 버텍스의 x좌표의 절대값 차이가 1이고, y좌표의 절대값 차이가 1보다 클 경우(그림 2의 “3” 항목), 버텍스의 x좌표에 1을 증가시키고, 버텍스의 y좌표를 그대로 사용함

(그림 4) 텍스쳐 배치 순서 샘플 3



(Figure 4) Example 3 of Ordering Textures

③ 배치될 텍스쳐 영역정보(MBR)
i. 좌측상단 :
(a) x 좌표: x 축 방향 이동량
(b) y 좌표: y 축 방향 이동량
ii. 우측상단 :
(a) x 좌표: x 이동량과 텍스쳐 가로크기
(b) y 좌표: y 축 방향 이동량
iii. 우측하단 :
(a) x 좌표: x 이동량과 텍스쳐 가로크기
(b) y 좌표: y 이동량과 텍스쳐 세로크기
iv. 좌측하단 :
(a) x 좌표: x 축 방향 이동량
(b) y 좌표: y 이동량과 텍스쳐 세로크기

④ 초기 저장 공간 갱신
배치될 텍스쳐의 영역정보의 우측 하단의 x, y좌표의 값과 초기 저장 공간의 가로크기 및 세로크기를 초과할 경우 초기 저장 공간을 갱신하여야 하며 갱신될 가로 및 세로크기는 초기 저장 공간의 가로 및 세로 크기를 비교하여 작은 것부터 갱신된다. 만약 가로 및 세로 크기가 동일할 경우, 가로 크기를

우선적으로 증가시킨다. 다음은 갱신될 저장 공간의 가로 및 세로크기를 결정하는 방법이다.

$Size \le 2^n$ 을 만족하는 최초 2^n 값이다. 여기서, $Size$는 초기 저장 공간의 가로 및 세로크기(X_o, Y_o)에 우측 하단의 x, y좌표를 각각 더한 값이다(식 (2)).

$$Size_x = X_0 + RD.x,\ Size_y = Y_0 + RD.y \qquad (2)$$

X_{size}는 $Size_x \le 2^n$ 수식을 만족하는 최초 2^n 값이 갱신될 이다. 이와 동일한 방법으로 Y_{size}도 결정된다.

3.4 최적 텍스쳐 제작

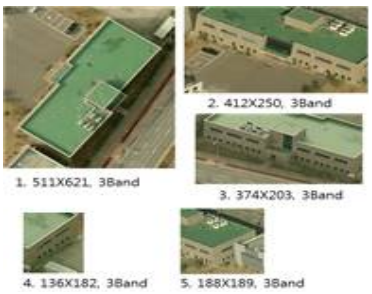
최적 텍스쳐 제작은 최종 저장 공간 크기와 텍스쳐 영상의 밴드 수를 이용하여 텍스쳐 저장 공간을 확보한 후에 각 텍스쳐 자료를 해당 위치, 즉 이동량(x, y좌표) 위치에 배치하여 제작한다. 아래는 최적 텍스쳐 자료를 생성하는 과정은 다음과 같다.

첫째, 텍스쳐 자료의 처리 및 크기를 산정한 다음(그림 5), 초기에 저장할 공간을 설정한다. (그림 5) 예제의 경우, 전체 텍스쳐에 의한 영상 면적은 556,537 pixel2인데, 이 면적 내에 영상이 가득 차 있다는 가정 하에 한 변의 길이를 구하면 746.01 pixel이 된다. 따라서 초기 저장 공간의 크기는 다음과 같이 계산된다.

i. $X_o = 746.01 < 2^n$ 을 만족하는 최소 2^n은 1024이므로, 초기 저장 공간의 가로 크기(X_0)는 1024가 된다.

ii. $Y_0 = \max(y_i) = 621 < 2^m$ 을 만족하는 최소 2^m은 1024이므로, 초기 저장 공간의 세로 크기(Y_0)는 1024가 된다.

(그림 5) 각 텍스쳐 크기



(Figure 5) Each size of Textures

최초 저장 공간의 크기가 산출되면 각각의 텍스쳐의 크기에 대해 내림차순 정렬을 수행한다(그림 6). 이후, 텍스쳐를 배치하기 위해 각 텍스쳐 자료의 이동량, 영상영역(MBR) 및 최종 텍스쳐 영상경계를 계산한다<표 1>. 연산에 필요한 각종 정보가 산출되면, 해당 정보를 이용하여 최종 텍스쳐 경계를 생성한다(그림 7).

(그림 6) 텍스쳐 면적에 따른 내림차순 정렬

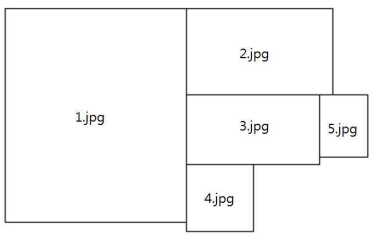
(Figure 6) Sort by area in descending order

<표 1> 각 텍스쳐 영역 정보

Name	MBR (x, y)			
	Upper Left	Upper Right	Lower Right	Lower Left
1.jpg	(0, 0)	(510, 0)	(510, 248)	(0, 248)
2.jpg	(511, 0)	(922, 0)	(922, 249)	(511, 249)
3.jpg	(511, 250)	(884, 250)	(884, 452)	(511, 452)
5.jpg	(511, 453)	(698, 453)	(698, 641)	(511, 641)
4.jpg	(885, 250)	(1020, 250)	(1020, 431)	(885, 431)

<Table 1> Calculate boundary information for each Texture

(그림 7) 최종 텍스쳐 배치 결과물



(Figure 7) Final result of Texture arrangement

최종 제작된 최적 텍스쳐 경계 영역 정보를 이용하여 해당 영상을 배치시키면 (그림 8)과 같이 최적 상태의 텍스쳐 이미지 배치가 완성된다.

(그림 8) 정렬된 텍스쳐 영상

(Figure 8) Arranged the Texture images

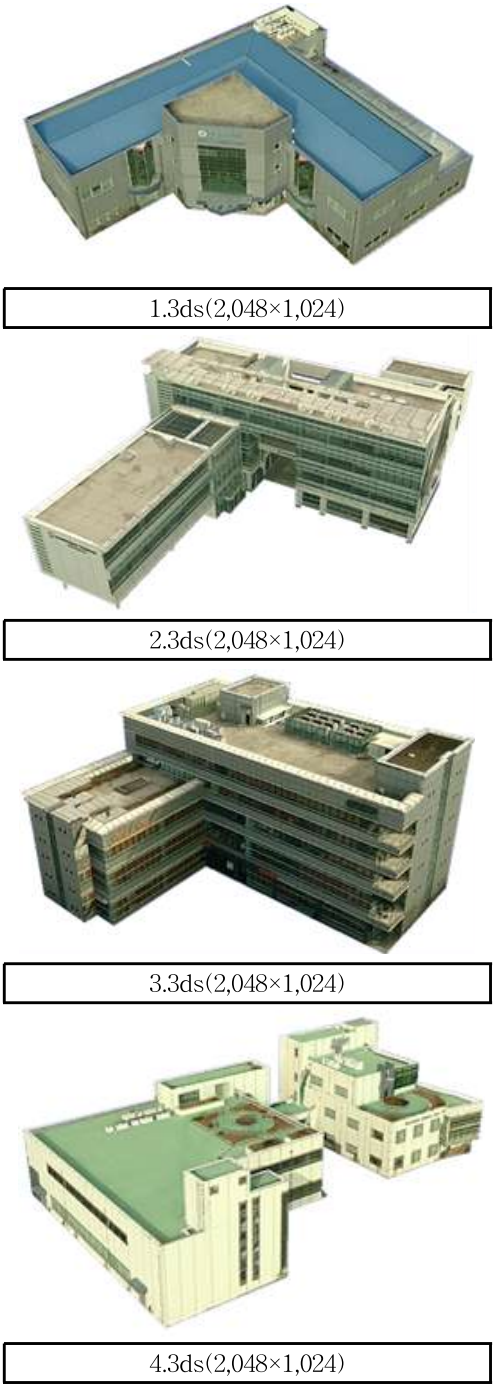
4. 최적 텍스쳐 배치 알고리즘 성능 비교

실험에 사용한 데이터는 상용 소프트웨어를 통해 제작한 3차원 건물로써 공간정보 오픈플랫폼에 사용되는 3차원 공간정보자료와 동일한 사양을 가진 파일이다. 비교적 모양이 간단하고 용량이 적은 파일과 복잡하고 텍스쳐의 개수가 많은 건물을 선별하여 실험을 진행하였다.

실험은 선별된 3차원 공간정보자료에 대한 텍스쳐 이미지를 다중 파일과 단일 파일로 각각 제작하였으며, 본 연구에서 도출한 최적 방안으로 제작하였다. <표 2>는 각 방안에 대해 제작된 텍스쳐 자료의 최종 용량 및 파일 개수를 정리한 것으로 본 연구에서 도출한 최적 텍스쳐 배치 알고리즘의 성능을 비교하기 위함이다.

실험 결과, 선별된 4가지 실험 대상에 대해 텍스쳐 이미지를 다중 파일로 저장할 경우에 대해 파일 용량이 가장 적은 것으로 나타났다. 비록 파일의 개수가 많으나, 타 방안과 달리 불필요한 공간 없이 사용자가 입력한 영상 정보만을 보유하고 있기 때문인 것으로 파악된다. 반면, 텍스쳐 이미지를 단일 파일로 저장할 경우, 불필요한 공간이 너무 많이 늘어나 다중 파일 저장 방식에 비해 최소 4.3배에서 최대 9.6배까지 용량이 배가되는 것을 확인할 수 있었다. 크기가 작은 텍스쳐 이미지가 많을수록 이로 인해 발생한 빈 공간을 보충하기 위해 추가된 불필요한 공간이 기하급수적으로 늘어났기 때문이다.

(그림 9) 실험 객체



(Figure 9) Experiment objects

<표 2> 텍스쳐 저장 기법에 따른 최종 결과물

The results of existing method				
Name	Multiple file storage method		Single file storage method	
	Storage (MB)	No. of file	Storage (MB)	No. of file
1.3ds	2.76	15	12.0	1
2.3ds	3.85	104	24.0	1
3.3ds	4.08	245	48.0	1
4.3ds	4.99	190	48.0	1
The result of Optimized Texture Storage Method				
Name	Storage(MB)		No. of file	
1.3ds	6.00		1	
2.3ds	6.00		1	
3.3ds	6.00		1	
4.3ds	6.00		1	

<Table 2> Final results of Texture Storage Method

한편, 본 연구에서 도출한 최적 텍스쳐 배치 알고리즘을 적용한 결과, 실험 대상에 관계없이 동일 용량을 가진 하나의 텍스쳐 이미지 파일이 생성되었다. 모든 실험 대상에 대해 동일한 용량으로 이미지 파일이 생성된 이유는 최종 제작된 텍스쳐 경계의 최대 영역이 모두 2,048×1,024 크기를 초과하지 않기 때문인 것으로 확인되었다. 비록 각 실험 대상이 서로 다른 텍스쳐의 크기와 개수를 가지지만 크기에 따라 내림차순으로 정리된 텍스쳐 이미지들이 내부에 존재하는 빈 공간에 모두 배치된 것이다. (그림 10)은 최종 제작된 텍스쳐 경계 영역에 대해 해당 이미지들을 배치하여 전자 파일로 저장한 최종 결과로써 모든 텍스쳐 이미지가 삽입된 것을 확인할 수 있다.

5. 결 론

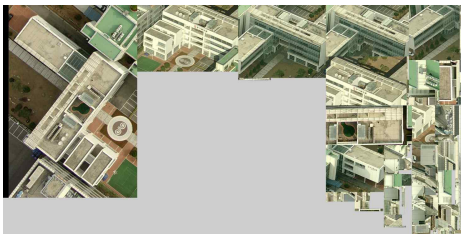
본 연구는 3차원 공간정보자료를 효율적으로 관리하고 각 자료의 용량을 최적화할 수 있는 방안을 도출하기 위해, 기존의 텍스쳐 이미지 파일 저장 방식을 분석함으로써 각 방안이 가진 장점을 취하고 단점을 보완할 수 있는 방안을 도출하여 최적의 방안을 도출하였다.

실험 결과, 연구를 통해 도출된 최적 텍스쳐 배치 알고리즘이 단일 파일 저장 방식에 비해 저장 공간을 최대 1/8배 축소시킬 수 있다는 것을 확인할 수 있었으며, 다수의 이미지 파일들을 관리해야하는 다중 파일 저장 방식과 달리 하나의 이미지 파일만을 관리하면 되어 관리의 효율성을 증진시켰다.

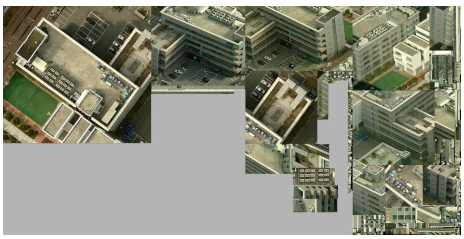
한편, 주어진 텍스쳐 이미지들을 이용하여 최

(그림 10) 최종 텍스쳐 영상

<1.bmp(2,048×1,024)>



<2.bmp(2,048×1,024)>



<3.bmp(2,048×1,024)>

<4.bmp(2,048×1,024)>

(Figure 10) Final Texture Images

적의 배치 방안을 수립하였음에도 불구하고, 최종 생성된 텍스쳐 이미지를 보면 비교적 넓은 공간들이 비어있는 것을 확인할 수 있다. 이는 기존의 텍스쳐 이미지를 변환시키지 않고 그대로 삽입하도록 명령하였기 때문이다. 만약, 각각의 텍스쳐 이미지에 대해 회전 요소를 고려하여 이미지를 배치하도록 구현한다면 최종 이미지의 왼편으로 텍스쳐 이미지들이 배치될 것으로 보이며, 경우에 따라 영역 확장 없이 최소 영역에 모든 이미지가 삽입되어 현재보다 더 용량이 축소될 가능성이 존재한다. 이에, 본 연구의 알고리즘을 개선하기 위한 추가 연구가 필요할 것으로 판단되나, 본 연구의 성과도 그 활용 가치가 매우 높을 것으로 판단된다.

References

[1] J. S. Go, I. H. Jeong, H. S. Shin, Y. S. Choi, S. K. Cho, “A Study on the Construction of Indoor Spatial Information using a Terrestrial LiDAR”, Journal of Korea Spatial Information Society, Vol.21, No.3, pp.89-101, June 2013.

[2] Ministry of Land, Transport and Maritime Affairs, Administrative guideline for 3D National Geographic Information Data Production Services, 2008.

[3] M. W. Oh, H. J. Kim, J. W. Koh, “A Study on the Development of the Tourism Information Service based on a Service Science -Focus on Using VWORLD-”, Journal of Korea Spatial Information Society, Vol.21, No.1, pp.23-36, February 2013.

[4] National Geographic Information Institute, Work regulation of 3D National Geographic Information Data, 2013.

[5] P. J. Neugebauer, K. Klein, “Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views”, Computer Graphics Forum, Vol.18, Issue3, pp.245 - 256, September 1999.

[6] S. B. Kim, The Study on 3D Model Construction Algorithm Using Digital Oblique Imagery, 2011.

[7] S. B. Kim, J. S. Go, H. G. Cho, J. W. Lee, “Construction of High-Precision 3D Spatial Information”, Korean Society of Civil Engineers Journal, Vol.59, No.10, pp.18-23, October 2011.

[8] S. J. Lee, (A) method for effective texture mapping using 3D graphics tools, 2010.

[9] Weinhaus, F. M., and V. Devarajan, “Texture mapping 3D models of real-world scenes”, Journal ACM Computing Surveys, Vol.29, Issue4, pp.325-365, 1997.

[10] Y. O. Kang, N. H. Cho, A. T. Kang, J. Y. Park, “Strategic Plan for Application of Geospatial Information Technology in the Field of Education”, Journal of Korea Spatial Information Society, Vol.22, No.2, pp.19-29, April 2014.

[11] S. C. Noh, S. C. Moon, “A Study on Analytical Methods of u-Healthcare Services Software Architecture Requirements for Quality Assurance”, Journal of Digital Contents Society, Vol.14, No.4, pp. 439-445, December 2013.

진 기 호

1983년 : 인하대학교 토목공학학사
2005년 : 연세대학교 대학원 (공학석사)
2014년 현재 : 충북대학교 대학원 (공학박사 수료-환경·도시공학과)
2003년 : 측량 및 지형공간정보 기술사
2004년～현 재: 중앙항업(주) 지리정보연구소장
관심분야 : GIS, 항공 및 위성영상, 3차원 공간정보, 빅데이터 등

하 성 룡

1980년 : 인하대학교 토목공학학사
1984년 : 인하대학교 대학원 (공학석사)
1989년 : 오사카대학교 대학원(공학박사-환경시스템공학)
1991년～현 재: 충북대학교 공과대학 환경·도시공학과 교수
관심분야 : 환경관리, 수질·수자원관리, GIS/RS, 환경모델링, 인공지능 등