이미지 분할 매칭 방법을 사용한 솔리드 텍스처 합성

서명국, 이관행
광주과학기술원
e-mail :{escmk, khlee}@gist.ac.kr

Solid texture synthesis using k-partition search

Myoung Kook Seo, Kwan H. Lee
Dept. of Mechatronics, Gwangju institute of science and technology

요 약

솔리드 텍스처 합성에서 최종 결과의 품질과 마찬가지로 텍스처 합성에 필요한 계산을 단축하는 것은 중요하다. 최근 연구에서는 데이터 차원 감소나 k-coherence search 같은 매칭 가속화 방법을 사용해서 솔리드 텍스처 합성 시간을 단축하였다. 본 논문에서는 빠르게 2D 이미지로부터 솔리드 텍스처 합성을 할 수 있는 새로운 방법을 제시한다. 제안하는 방법의 기본 아이디어는 주어진 이미지를 다수 영역으로 분할하여 형상 매칭(feature matching) 과정에서 사용되는 후보 수를 줄이는 것으로, 사물을 이루는 복셀과 관련한 픽셀이 포함된 분할 영역내의 후보들과 비교함으로써, 보다 빠르게 최적의 결과값을 제공한다. 실험적으로 본 방법은 k-coherence search 알고리즘에서 k 값을 1 로 사용했을 때보다 빠르거나 비슷한 시간을 갖는다.

1. 서론

솔리드 텍스처 합성(solid texture synthesis)은 2D 이미지 상의 특징적인 패턴을 3D 사물의 내. 외부에 표현하고자 할 때 사용하는 컴퓨터 그래픽스의 기술이다. 비슷한 기술로써 텍스처 맵핑(texture mapping)이 있는데, 이는 오래 전부터 주어진 텍스처를 사물 표면에 표현하고자 할 때 사용되어 왔다. 하지만 복잡한 형상을 지닌 사물에 대해서는 형상 왜곡이나 부자연스러운 결과를 주는 문제점을 가지고 있다. 솔리드 텍스처 합성은 텍스처 맵핑과 다르게 사물의 볼륨 전체에 이미지 패턴을 나타낼 수 가 있는데, 패턴의 왜곡이나 불연속 같은 문제를 최소화 할 수가 있다. 그리고 솔리드 텍스처의 볼륨 데이터는 사물의 변형이나 절단 시뮬레이션 같은 다양한 응용이 가능하다.

최근 솔리드 텍스처 합성은 2D 이미지를 기반으로 하는 경우가 많지만[1, 2] 초기에는 함수를 이용한 과정적 텍스처 합성이 많이 쓰였다. 이 방법은 파라미터 값 조절을 통해 빠르게 솔리드 텍스처를 구현할 수 있는 장점이 있지만 표현 할 수 있는 패턴이 제한적이고 목적으로 하는 패턴을 위한 파라미터 값을 컨트롤 하기 어려운 단점이 있다. 이미지 기반 솔리드 텍스처 합성은 다양한 종류의 이미지를 이용하여 고품질의 솔리드 텍스처를 구성할 수 있다. 한편, 솔리드 텍스처 합성은 사물의 내. 외부뿐만 아니라 다양한 방향에서 주어진 이미지에 가까운 형상을 나타내기 위해서는 다양한 관련 알고리즘 사용과 상당한 계산시간을 요구한다. 최근 많은 연구에서는 솔리드 텍스처의 결과 향상과 계산 시간 단축을 위해 여러 방법을 제안하고 있다.

본 논문에서는 솔리드 텍스처 합성의 계산 시간 단축에 관심을 가지고 있다. 비록 k-coherence search 알고리즘이나 PCA 투영(principal components analysis projection)를 통한 데이터 차원 감소를 이용하여 상당한 시간을 줄일 수가 있지만, 아직까지 128*128*128 볼륨을 기준으로 평균적으로 5 분 이상의 계산 시간이 필요하다. 합성 시간의 단축을 위해, 우리는 사물의 복셀 색을 결정하는 형상 매칭 과정을 분석하였다. 그리고 형상 매칭에 사용되는 이미지 상의 후보의 수를 줄임으로써 계산 시간을 단축시킬 수 있다는 점을 발견하였다. 매칭 과정에서의 후보 수를 줄인다는 면에서 k-coherence search 알고리즘과 유사하지만, 앞 방법이 사전에 이미지 상의 각 픽셀에 대해서 k 개의 후보를 지정하는 반면 우리의 방법은 각 픽셀을 포함하는 세부 영역으로 이미지를 나눈다. k-coherence search 방법이 상대적으로 후보 수를 유동적이고 작게 만들 수 있다는 면에서 유리할 수 있지만, 실제적으로 제안하는 방법이 다른 형상 매칭 가속화 방법과 같이 사용할 수 있기 때문에 텍스처 합성을 좀더 가속화 시킨다.

2. 솔리드 텍스처 합성

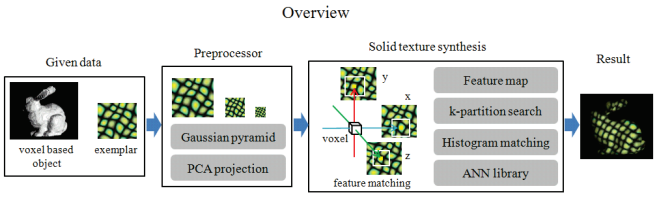
솔리드 텍스처 합성은 Markov Random Field (MRF)이라는 중요한 가정을 기반으로 하고 있다. 즉 솔리드 텍스처는 전 볼륨의 어느 단면에서도 주어진 이미지와 비슷한 형상 분포를 가져야 한다. 이를 위해 형상 매칭 과정에서는 사물을 이루고 있는 각 복셀의 색을 결정하기 위해 3 방향에 대해서 다음 에너지 함수(식 1)를 최소화 시킬 수 있는 후보를 찾는다. 그리고 최종 복셀의 색은 식 2 를 통해 계산된다.

$$E(s,\{e\}) = \sum_{v}\sum_{i\in\{x,y,z\}}\left\|s_{v,i}-e_{v,i}\right\|^{\gamma} \sum_{v}\sum_{i\in\{x,y,z\}}\sum_{u\in N_i(v)}\omega_{v,i,u}\left\|s_{v,i,u}-e_{v,i,u}\right\|^{2} \quad (1)$$

$$s_v = \frac{\sum_{i\in\{x,y,z\}}\sum_{u\in N_i(v)}\omega_{u,i,v}e_{u,i,v}}{\sum_{i\in\{x,y,z\}}\sum_{u\in N_i(v)}\omega_{u,i,v}} \quad (2)$$

여기서 e 는 이미지상의 후보를 가리키며, s 는 사물의 복셀을 나타낸다. 자세한 내용은 관련 논문[1, 2, 4]에서 찾아볼 수 있다.

2.1 전반적인 합성 과정



(그림 1) 합성의 전체적 흐름

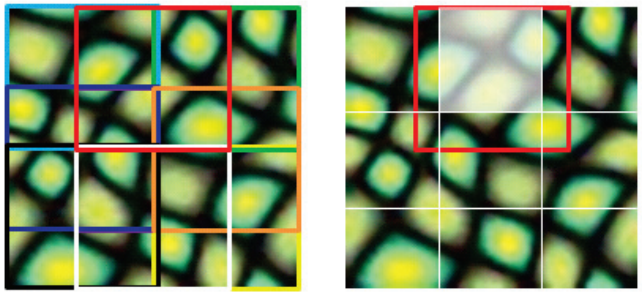
우리가 제안하는 솔리드 텍스처 합성의 전체적인 과정은 그림 1 에서 보여준다. 복셀로 이루어진 임의의 사물과 재질 이미지가 주어지면, 빠른 합성을 위해 Gaussian pyramid 와 PCA 투영을 적용한다. 다중 해상도(multi-resolution)라 불리기도 하는 Gaussian pyramid 는 이미지와 사물을 낮은 해상도에서 최종 해상도까지 단계적으로 합성하는 방법으로, 본 연구에서는 3 단계로 구성하고 있다. PCA 투영은 형상 매칭에 사용되는 데이터의 차원을 감소시키기 위해 사용한다. 예를 들어 사물과 이미지 상에서 9*9 이웃 데이터를 사용할 경우, RGB 색 기반이므로 243(3*9*9)차원의 벡터가 각 매칭과정에서 사용된다. 이러한 크기는 매칭 시간을 상당히 요구하기 때문에, 각 벡터에 PCA 투명을 적용, 10~30 차원의 변수 벡터로 전환한다. 감소하는 차원 수에 따라 정확도는 다르지만 보통 95%이상 유지하며, 본 연구에서는 30 차원을 사용한다.

솔리드 텍스처 합성은 3 개의 방향(x, y, z)에 대해서 식 1 을 만족 시키는 픽셀의 값을 찾아야 한다. 하지만 일반적으로 이미지의 RGB 색만을 이용할 경우, 솔리드 텍스처는 왜곡되거나 흐릿한 형상을 갖는 결과를 주는 경우가 많다. 이를 해결하기 위해서 주어진 이미지와 별도로 피처 맵(feature map)를 사용한다. 이것은 특정 형상과 배경을 의도적으로 분리한 것으로, 최근 연구에서 다양한 피처 맵 (이진, 컬러 그리고 사용자 지정 피처 맵)이 제안되었다[3, 5]. 이는 그림 3 에서 볼 수 있다. 한편, 최근 연구에서는 특정 색이나 형상이 과다하게 사용되는 것을 막기 위해 컬러 히스토그램[1]이나 위치 히스토그램[4]을 적용한다. 본 논문은 위치 히스토그램을 적용하였다.

빠른 형상 매칭을 위해 k-coherence search 와 ANN approximate nearest neighbor library[1]를 사용할 수 있다. k-coherence search 는 사전에 이미지 각 픽셀에 대해서 자신의 이웃 픽셀과 비슷한 k 개의 후보를 지정해 두어 각 매칭 과정에서 그 후보들만을 사용함으로써 매칭 시간을 줄이는 방법이다. 반면 ANN library 는 앞서 언급한 PCA 투명에 의해 감소된 벡터를 보다 빠르게 피처 매칭을 쓸 수 있도록 제안된 것으로 kd-tree 를 기반으로 하고 있다. 한편 k-coherence search 와 ANN library 를 동시에 사용하기는 어렵다. 그 이유는 ANN library 는 매칭 과정 이전에 후보 집단을 계산해 둘 때 빠른 매칭 결과를 제공하는데, k-coherence search 방법을 사용할 경우 각 매칭 과정에서 후보 집단을 갱신해야 한다. 그래서 두 기술을 동시에 사용하면 역으로 계산 시간이 상당히 늘어나게 된다. 앞 두 방법을 별도로 사용할 경우 128*128*128 해상도를 갖는 볼륨에 대해서 평균적으로 5 분 이상의 계산시간이 요구된다. 본 연구에서는 빠른 솔리드 텍스처 합성을 위해 k-partition search 라는 새 방법을 사용하기를 제안한다. 자세한 내용은 다음 문단에서 다루기로 한다.

3. k-partition search

제안하는 k-partition search 는 k-coherence search 와 마찬가지로 매칭 과정에서 사용하는 후보 수를 줄여 계산시간을 단축시킨다. k-coherence search 경우 만약 각 픽셀당 3 개의 후보(k=3)를 지정한다고 했을 때 9*9 사이즈의 이웃 데이터를 사용하면 243*3 개의 후보가 매칭 과정에서 사용된다. 하지만 별도의 매칭 과정을 가속 시킬 수 있는 방법을 사용하지 않으면, 후보 수(k)에 비례하게 연산시간도 증가한다. 우리 방법은 사물의 복셀과 연관된 픽셀의 주변 후보들과 형상 매칭을 한다. 여기서 연관된 픽셀이란 솔리드 텍스처 합성은 각 단계 별로 1~5 사이의 반복을 통해 완성되는데, 바로 전 단계 또는 앞서 찾은 복셀의 값으로 지정된 이미지 상의 특정 픽셀을 말한다. 우리는 다음 그림 2 에서 보는 것처럼 이미지를 k 개의 영역으로 분할하고(왼쪽 그림) 해당 영역내의 픽셀(오른쪽 그림의 흰색부분)이 대상 복셀 값으로 지정되어있을 때 그 분할 영역내의 후보만(빨간 사각 영역) 매칭 과정에 사용한다. 분할 영역 수 k 는 이미지의 크기와 이미지 내의 형상에 따라 달라지는데, 보통 64*64 이미지 기준으로 9, 16, 25 을 사용하기 권장한다.



(그림 2) K-partition search (k=9)

여기서 주의할 점은 각 분할 영역이 서로 겹치는 부분을 포함해야 하는데, 이렇게 하지 않을 경우 특정 영역만 전 볼륨에 대해서 중복되는 문제점이 발생한다. 제안하는 방법은 ANN library 를 동시에 사용할 수 있다. 각 분할 영역의 후보는 사전에 search tree 구성해, 필요 시 마다 빠르게 이용된다. 실험적으로 k-partition search 는 k-coherence search 을 기반으로 하는 합성에서 픽셀 당 후보를 하나만 사용하는 경우(k=1) 보다 더 빠르거나 비슷한 속도를 제공했는데, 이는 표 1 에서 확인할 수 있다.

<표 1> k-partition search 와 k-coherence search 시간 비교

방법		총 연산 시간(초)		
k-partition search	분할 수	9	16	25
	이미지 1	244s	237s	236s
	이미지 2	272s	256s	251s
k-coherence search	후보 수	1	3	5
	이미지 1	248s	379s	508s
	이미지 2	245s	376s	505s

4. 결과

그림 3 은 특정한 패턴을 지닌 이미지와 피처 맵을 사용하여 합성한 결과를 보여주고 있다. 얼룩말 무늬 같은 이미지는 상대적으로 분포 형상이 단순하여 피처 맵 필요 없이 이미지 만으로도 좋은 결과를 가져온다. 반면 색이 다른 형상을 지니거나(brick, green cell) 형상이 불분명한 이미지(wave)는 피처 맵을 같이 사용하였다.

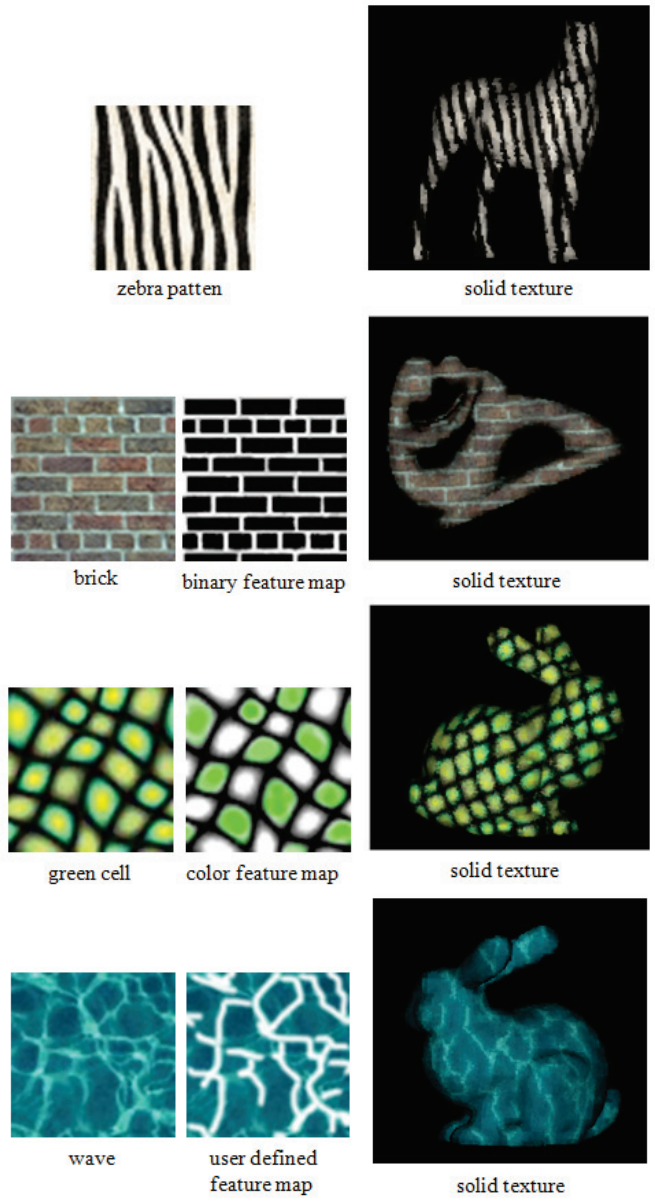
5. 분석 및 한계

본 논문에서 제안한 k-partition search 은 원래 분할된 영역의 크기에 비례하게 연산 속도를 단축시키기 위해 고안하였다. 즉 한 이미지를 1/3 크기의 다수 영역으로 나눌 경우 3 배 빠를 것으로 예측하였다. 비록 시간을 단축시키는 것은 분명하지만 예측한 것보다는 빠르지 않았다. 그 이유로 ANN library 를 같이 사용하기 때문에 후보 수 감소 효과가 서로 중첩된 것으로 보고 있다.

6. 결론

우리는 솔리드 텍스처 합성에서 매칭 속도를 가속화하기 위해서 k-partition search 을 제안하였다. 실험적으로 제안하는 방법은 다양한 이미지에 대해서 기존 방법보다 빠르게 결과를 제공하였다.

"이 논문은 2009 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 20090083091)."



(그림 3) 솔리드 텍스처 합성 결과

참고문헌

[1] J. Kopf, C. W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski and T.T. Wong, “Solid Texture Synthesis from D Exemplars”, ACM Transactions on Graphics, 26(3), 2:1--2:9, 2007.

[2] Y. dong, S. Lefebvre, X. Tong and G. Drettakis, “Lazy Solid Texture Synthesis”, Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering), 2008.

[3] Q. Wu and Y. Yu, “Feature matching and deformation for texture synthesis”, ACM Transactions on Graphics, 23(3), 364--367, 2004.

[4] J. Chen and B. Wang, “High quality solid texture synthesis using position and index histogram matching”, The visual computer, 2009.

[5] 서명국, 김강연, 김덕봉, 이관행, “사용자 지정 피처 맵을 사용한 솔리드 텍스처 합성”, 2010 한국 CAD/CAM 학회 학술 발표회, 91--94, 2010.