

가상 대장내시경을 위한 가시성을 이용한 자동 경로 생성법

(Visibility-based Automatic Path Generation Method for Virtual Colonoscopy)

이 정 진 [†] 강 문 구 ^{**} 조 명 수 ^{***} 신 영 길 ^{****}
(Jeongjin Lee) (Moon Koo Kang) (Myoung Su Cho) (Yeong Gil Shin)

요약 가상 대장내시경(virtual colonoscopy)은 컴퓨터단층촬영(computed tomography) 영상으로부터 대장의 형상을 입체적으로 재구성하여 대장 내부의 종양을 빠르고, 쉽게 진단할 수 있는 방법으로서, 기존의 침습적인 대장 진단법의 단점을 보완한 방법이다. 효과적인 임상 진단을 위해서는 가상 대장내시경을 위한 탐색 경로를 빠르고 정확하게 생성할 필요가 있으며, 이를 위하여 몇 가지 방식이 개발되어 사용되고 있다. 그러나 기존의 방법들은 거리맵 등의 자료구조를 전처리 단계에 미리 생성하여야 하고, 경로를 구성하는 모든 점들에 대하여 방대한 양의 위치 계산이 필요하므로 긴 연산 시간이 소요된다. 본 논문에서는 이러한 연산시간을 크게 단축시킬 수 있는 가시성을 이용한 자동 경로 생성법을 제안하였다. 본 제안 방법에는 거리맵 계산 등의 전처리 과정이 필요 없으며, 탐색 경로를 표현하는 곡선의 모든 점이 아닌 곡선을 대표하는 소수의 제어점(control point)을 생성하는 방법으로 계산량을 줄여 경로 생성 시간을 단축하였다. 또한, 가시성에 기반을 두어 경로를 생성하기 때문에 자연스러운 가상 카메라의 움직임이 가능하고, 가상 탐사에 편리하고, 정확한 경로를 생성할 수 있다. 이 방법은 가상 대장내시경 이외에도 다양한 종류의 곡관 내부의 가상 탐사에 응용될 수 있다.

키워드 : 의료 영상 처리, 가상 대장내시경, 경로 생성, 가시성

Abstract Virtual colonoscopy is an easy and fast method to reconstruct the shape of colon and diagnose tumors inside the colon based on computed tomography images. This is a non-invasive method, which resolves weak points of previous invasive methods. The path for virtual colonoscopy should be generated rapidly and accurately for clinical examination. However, previous methods are computationally expensive because the data structure such as distance map should be constructed in the preprocessing and positions of all the points of the path needs to be calculated. In this paper, we propose the automatic path generation method based on visibility to decrease path generation time. The proposed method does not require preprocessing and generates small number of control points representing the path instead of all points to generate the path rapidly. Also, our method generates the path based on visibility so that a virtual camera moves smoothly and a comfortable and accurate path is calculated for virtual navigation. Also, our method can be used for general virtual navigation of various kinds of pipes.

Key words : Medical image processing, Virtual colonoscopy, Path generation, Visibility

1. 서론

대장암은 여러 가지 암 가운데 발병빈도가 높고 치료가 용이하지 않은 대표적인 질병이다. 미국의 경우 대장암은 모든 암 관련 사망 원인의 두 번째이며, 식생활의 서구화에 따라 우리나라에서도 암 관련 사망 원인의 네 번째에 해당할 정도로 현대인의 건강에 큰 영향을 미치는 질병이다. 대장암의 치료에 있어 가장 중요한 관건은 효율적인 진단을 통해 초기에 대장의 종양을 발견하는

[†] 학생회원 : 서울대학교 컴퓨터공학부
ifnkl@dreamwiz.com

^{**} 정회원 : 서울대학교 전기컴퓨터공학부 교수
moonkang@ee.snu.ac.kr

^{***} 정회원 : 인피니트테크놀로지 연구원
msdanbi@infinit.com

^{****} 중신회원 : 서울대학교 컴퓨터공학부 교수
yshin@cflab.snu.ac.kr

논문접수 : 2005년 2월 14일
심사완료 : 2005년 6월 22일

것이다. 대장암의 조기 발견을 위해서는 정기적인 검진이 필수적이다. 검진 방법으로는 대장 조영술(barium enema)과 광학 내시경(optical endoscopy)이 많이 이용된다. 대장 조영술은 대장 내부에 바륨이라는 백색의 조영제와 공기를 투입하여 대장 벽에 도포되는 조영제의 상태를 x-ray를 이용하여 검사하는 방법이다. 대장 조영술은 광학 내시경에 비해 정확도가 떨어진다는 단점이 있다. 광학 내시경은 광학 탐침(optical probe)을 직장을 통해 대장 내부로 삽입하여 탐침 끝에 부착된 소형 카메라를 조정하여 대장의 내부 표면을 관찰하고 진단하는 방법이다. 이 방법은 탐사 과정에서 카메라 조정이 어려워 고도의 숙련도를 필요로 할 뿐 아니라 오랜 시간이 소요된다[1,2]. 또한 이러한 침습적인 방법은 환자가 검사 전에 긴 사전 준비 작업이 필요하고, 장 세척과 같은 불편함을 감수해야 하며, 감염과 출혈 등의 부작용이 있다.

이러한 단점을 보완하기 위해 CT에서 얻어진 인체의 단면 영상 데이터를 컴퓨터로 처리하는 새로운 진단 방법인 가상 대장내시경이 개발되었다[3-6]. 최근에는 0.5초 이내의 짧은 시간에 16개의 영상을 얻을 수 있는 멀티 디텍터 CT (Multidetector CT)가 도입되어 가상 대장내시경에 필요한 CT 촬영 시간이 5분 정도로 단축되었으며, 정밀도가 향상되어 작은 크기의 종양 진단이 가능해졌다[7]. 가상 대장내시경은 단층 촬영된 영상을 3차원적으로 재구성하는 과정에서 각 내장 기관과 내용물을 개별적으로 분리하여 형상추적이 가능하므로 대장 내부의 내용물을 세척해 낼 필요가 없으며, 따라서 장 세척과 같은 사전 준비 작업이 불필요하고, 검진 과정에서 광학 탐침을 삽입할 필요가 없으므로 환자에게 전혀 고통을 주지 않고, CT 영상 데이터만 준비되면 언제나 빠르고, 정확하고, 세밀한 검사가 가능하다는 장점이 있다. 또한, 광학 내시경은 카메라의 진행 방향으로만 검사가 가능하지만, 가상 대장내시경에서는 카메라의 위치와 방향 조절이 자유로워 진단의 효율성을 높여준다[8].

대장의 길이가 평균 1.5m 정도로 길기 때문에 의사가 단순 수작업으로 대장 내부를 탐사하기가 어렵고, 시간이 많이 걸린다. 따라서 가상 대장내시경을 위해서는 가상 카메라가 지나가는 경로인 대장의 중심선을 설정하는 과정이 선행되어야 한다. 중심선을 찾는 가장 기본적인 방법은 일련의 2차원 단면 영상에서 의사가 직접 중심에 해당하는 부분을 찾아 표시한 후, 단면에 표시된 점을 보간하여 중심선으로 정의하는 것이다. 그러나 이 방법은 시간이 너무 오래 걸리고, 2차원 단면 영상에서는 최적의 위치가 표시되었다고도 3차원 경로로 재구성하는 과정에서 부적합한 경로가 생성될 수 있는 단점이 있다. 또 다른 방식은 중심선만 남을 때까지 대장 단면

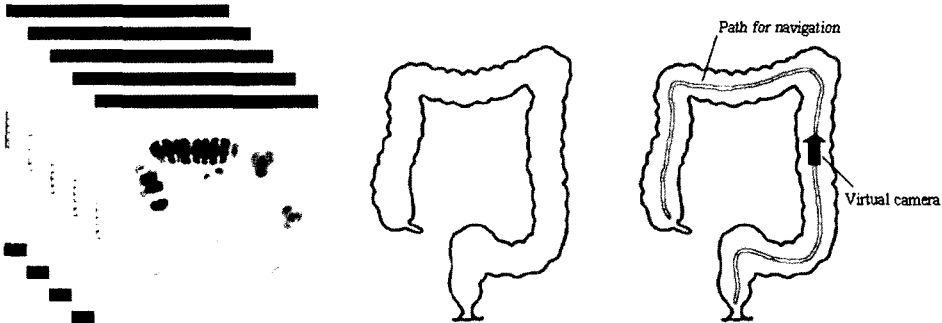
의 외곽부위로부터 시작하여 단면의 중심 방향으로 복셀(Voxel)을 한 층씩 제거하는 위상 세선화(topological thinning) 방식이다[3,9]. 이 방법은 비교적 경로는 정확하나 계산량이 방대하고, 위상 보존과 같은 제약조건으로 인해 연산시간이 많이 걸리는 단점이 있다. 그 외에도 대장 표면으로부터의 거리맵을 전처리 과정에서 만들고 다익스트라 최단 경로 그래프 알고리즘(Dijkstra shortest path graph algorithm)[10]을 사용하여 경로를 생성하는 방법들이 개발되어 있으나, 이러한 방법들은 전처리 과정과 최단 거리의 경로 탐색 과정에 많은 계산 시간을 필요로 한다[11-13].

본 논문에서는 광선 추적법(ray casting)[14]의 광선이 진행되는 방식을 가시성에 응용하여 대장의 탐색 경로를 생성하는 방법을 제안한다. 이 방법을 사용하면 거리맵 생성과 같은 전처리 과정이 필요 없으며, 전체 경로의 모든 복셀이 아닌 경로를 대표하는 제어점만을 생성하여 경로 생성 시간을 크게 단축할 수 있다. 또한, 원리적으로 가시성에 기반을 둔 방식이므로 더욱 자연스러운 카메라의 움직임이 가능하다.

2. 가시성을 이용한 자동 경로 생성법

가상 대장내시경 진단 과정은 다음과 같은 순서로 이루어진다. 먼저 그림 1(a)와 같이 환자의 대장 부위를 CT 촬영하여 단면 영상을 얻는다. 다음으로 그림 1(b)와 같이 CT 촬영된 단면 영상으로부터 3차원 씨앗점 영역 성장법(3D-seeded region growing)의 영상 분할 기법을 사용하여 대장을 다른 기관들로부터 분리하여 대장의 형상을 3차원적으로 재구성한다[15]. 마지막으로 그림 1(c)와 같이 대장 내부의 종양을 발견 및 진단하기 위하여 대장 내부의 중심선에 해당하는 가상 대장내시경 경로를 생성한 후 경로를 따라서 자동으로 가상 카메라에 해당하는 시점을 이동시켜 가상적인 내시경 이미지를 만들어 낼 수 있다. 내시경의 최적 이동 경로는 대장내부의 탐사에 소요되는 시간과 검사 결과의 정확성을 좌우하는 중요한 요소가 된다. 최적 기준경로가 없다면 검사자가 수작업으로 카메라의 경로를 계속 조정하면서 대장 내부를 탐사하여야 하므로 탐침 삽입 방식에서와 마찬가지로 많은 시간과 노력이 소요된다. 최적 경로가 설정되어 있으면 그 경로를 따라 카메라를 자동으로 이동시키면서 이상 징후가 발견된 부위에 대해서만 수작업으로 상세한 진단을 하는 방식으로 검사가 가능하므로 진단의 효율성을 크게 향상시킬 수 있다.

가상 대장내시경의 이동 경로는 초기 위치에서 원하는 최종 위치까지 대장 벽면과의 충돌 없이 대장 내부를 관찰할 수 있도록 대장 내부 단면의 중심을 따라서 연속적으로 생성되어야 한다. 또한, 가상 카메라의 이동



(a) CT 촬영 영상

(b) 분할 후 재구성된 대장

(c) 대장 내부의 탐색 경로 및 가상 카메라

그림 1 가상 대장내시경의 과정

방향이 너무 급격하게 바뀌지 않도록 대장의 굴곡이 심한 부분에서도 부드럽게 이어져야 한다. 경로 생성을 위한 초기점(seed point)의 설정은 사용자가 직접 화면상의 특정 위치를 선택하여 이를 3차원 좌표로 변환한다. 이 때, 초기점의 위치는 임의의 위치에 설정될 수 있으며, 초기점을 중심으로 광선의 진행 방향은 전진 방향뿐 아니라 후진 방향으로도 경로의 생성을 수행하여 최종적으로 두 경로를 합하게 된다. 이와 같은 방식으로 양 방향으로 최적경로를 생성한 후 이 경로들을 합하여 하나의 최적경로를 생성할 수 있다. 최적 경로의 생성에 있어서는 연산의 가속을 위한 가상의 구를 위한 자료구조를 생성한 후, 가시성을 이용하여 경로의 제어점들을 찾는 과정을 반복한다. 대장의 끝 부분까지 제어점이 찾아져서 더 이상 제어점을 찾을 수 없게 되면 구해진 제어점들을 보간하는 부드러운 곡선을 생성하고 이 곡선이 가상 대장내시경의 경로가 된다.

2.1 시작점 위치 및 기준 광선의 방향 결정

경로 생성을 위하여 먼저 사용자로부터 시작점을 입력 받는다. 사용자는 화면에 그림 2에서와 같이 시작점 $P_{image} = (x_i, y_i, z_i, l)$ 를 입력한다. 입력된 점은 3차원 영상의 투영된 이미지 상의 2차원 좌표이므로, 이를 볼륨 데이터 상의 3차원 좌표로 변환하는 과정이 필요하다. 입력된 점으로부터 이미지에 수직인 방향으로 그림 2와 같이 광선을 진행시켜 광선과 대장 벽의 두 교차점 P_{int1} 과 P_{int2} 를 구하면, 두 점의 중심점인 P_{center} 가 경로 생성의 시작점으로 적합하다. 광선을 진행시키는 과정은 z_i 의 값에 해당하는 이미지 상의 깊이를 0부터 시작하여 1복셀 씩 증가시키면서 볼륨 데이터 상의 좌표를 구하게 된다. 광선의 위치와 방향을 나타내는 행렬을 M_{view} 라고 하면, 이미지 상의 좌표를 볼륨 상의 좌표로 변환하기 위해서는 다음과 같은 변환과정을 거친다.

$$P_{object} = M_{view}^{-1} P_{image} \tag{1}$$

위의 변환을 통해서 얻어진 P_{object} 는 3차원 볼륨 데이터 상의 좌표이므로, 이 좌표를 사용하여 해당 위치의 볼륨 데이터의 밀도값 d 를 알 수 있다. d 와 이미 알려져 있는 대장 벽의 밀도값을 비교하여 현재 위치가 대장 벽에 해당하는지의 여부를 판단할 수 있다. z_i 의 값을 증가시켜 가면서 밀도값을 기준으로 대장 벽에 해당하는 두 교차점 P_{int1} 과 P_{int2} 를 구하면 이 두 점의 중심점인 P_{center} 를 대장의 해당 위치에서 중심으로 볼 수 있고, 이 점은 경로 생성의 시작점이 된다.

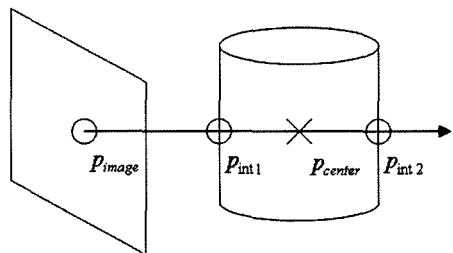


그림 2 시작점 결정 과정

시작점을 결정한 후 다음으로 기준 광선의 방향을 결정한다. 시작점에서 모든 방향으로 광선을 진행시켜 대장 벽과의 교차점을 구하고, 이 점까지의 거리가 최대가 되는 방향을 기준 광선의 방향으로 결정한다. 극좌표계 상에서 0도부터 θ 와 ϕ 를 5도 간격으로 증가시키면서 등간격으로 광선을 발생시키고, 이 광선이 대장 벽과 만나는 점을 저장해 둔다. 저장된 점과 시작점 사이의 거리 중에서 가장 큰 값을 가지는 교차점을 계산한다. 시작점에서 이 점을 향하는 방향이 시작점에서 가장 가시성이 높은 위치로 생각될 수 있고, 이 방향을 기준 광선의 방향 $R_{d0}(\theta_0, \phi_0)$ 으로 정한다. 그리고, 시작점이 대장의 끝부분에 위치하지 않을 경우 기준 광선의 방향과

반대 방향으로도 경로 생성을 수행하여 생성된 두 경로를 결합하여 최종 경로를 만들게 된다.

2.2 가상의 구 팽창을 위한 오프셋 테이블 생성

경로 생성 시 반복적으로 사용되는 과정이 있다면 미리 계산된 결과를 자료구조에 저장함으로써 연산을 가속시킬 수 있다. 본 제안 방법에 의한 경로 생성 시에 그림 5(d)와 같이 $P_{candidate}$ 을 중심으로 반지름이 1복셀인 가상의 구를 반지름을 1복셀 씩 팽창하여 대장벽과 가상의 구의 표면이 만나는 점을 찾는 과정을 반복하게 된다. 따라서 실제 경로를 생성하기에 앞서 가상의 구의 표면에 대한 구의 중심으로부터의 가까운 거리 순서대로 상대 좌표를 저장해 놓은 오프셋 테이블을 미리 만들어 둔다. 따라서, 가상의 구의 중심에 해당하는 $P_{candidate}$ 의 절대 좌표는 경로 생성 단계마다 변하지만, 이 중심의 좌표에 오프셋 테이블에 저장된 상대 좌표를 더하게 되면, 식 (2)와 같이 가상의 구 표면의 좌표 P_{sphere} 가 빠르게 계산된다.

$$P_{sphere} = P_{candidate} + (OffsetTable_x(i), OffsetTable_y(i), OffsetTable_z(i)) \quad (2)$$

오프셋 테이블에는 구의 중심에 해당하는 (x_0, y_0, z_0) 인 점을 기준으로 반지름이 1복셀인 구의 표면에 해당하는 점들부터 거리 값을 1복셀 씩 늘려가며 해당 거리 값을 갖는 3차원 좌표를 차례대로 저장한다. 거리 값을 늘려가는 과정은 결국 가상의 구를 팽창시키는 과정에 해당한다. 2차원의 예를 들어 보면 다음과 같다. 대칭성이 있으므로 제 1사분면의 좌표에 대해서만 고려 한다. (x_0, y_0) 인 점을 중심으로 모든 정수 값을 가지는 점 (x, y) 에 대해 중심으로부터의 유클리디안 거리 $\sqrt{(x-x_0)^2 + (y-y_0)^2}$ 의 올림값을 구하여 D 라고 하자. D 값이 같은 점들을 그림 3과 같이 분류한다. 이것은 제 1사분면에 대해서만 계산한 결과이므로 x 축, y 축, 원점 대칭 시킨 값을 중심으로부터 거리 값이 작은 점의

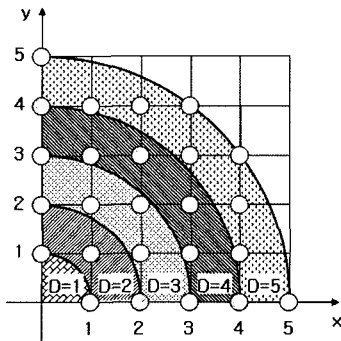


그림 3 제 1 사분면에서의 거리 D에 따른 2차원 좌표 분류

index	0	1	2	3	4	5	6	7	8	9	10	11	...
OffsetTable_x	1	-1	0	0	2	-2	1	1	-1	-1	0	0	...
OffsetTable_y	0	0	1	-1	0	0	1	-1	1	-1	2	-2	...

그림 4 2차원에서의 오프셋 테이블

좌표부터 그림 4와 같이 오프셋 테이블에 저장한다. 이 자료구조를 통해서 자동 경로를 찾는 과정 중 가상의 구를 팽창하는 계산 과정의 속도를 빠르게 할 수 있다.

2.3 가시성에 기반한 자동 경로 생성

앞에서 구해진 시작점과 기준 광선의 방향을 초기값으로 하여 그림 5의 과정을 반복해서 적용하여 가상 카메라의 경로를 구성하기 위한 제어점들을 얻을 수 있다. 첫번째 단계로 광선 R 에 대하여 가시성이 가장 높은 위치를 찾는다. 광선 R 은

식 (3)과 같이 나타낼 수 있다.

$$R = P_0 + l \cdot R_d(\theta, \phi) \quad (3)$$

이 때 P_0 은 광선의 시작 위치이고, P_d 는 극좌표에서의 θ 와 ϕ 로 표시된 광선의 방향이고, l 은 광선의 진행 거리이다. 가시성이 가장 높은 위치란 P_0 에서 기준 광선의 방향 $P_{d0}(\theta_0, \phi_0)$ 를 중심으로 식 (4)의 범위 내에서 경로 생성이 가능한 방향으로 l 을 1복셀 씩 증가시키면서 광선을 진행하여 대장 벽과 만나는 점들 중에서 P_0 과의 거리가 가장 큰 점을 말한다. 즉, 시작점의 위치에서 가장 멀리 보이는 대장 벽면 상의 점의 위치를 말한다. 식 (4)에서 k_1 은 시야를 나타내는 인자로 특정한 k_1 의 범위 내에서 k_1 이 크면, 더 넓은 각도 범위에 대해 가시성을 계산하기 때문에 시야가 넓어지나 경로 계산 시간이 오래 걸리고, k_1 이 작으면, 시야는 좁아지나 경로 계산 시간이 빠르다. 따라서, k_1 은 실험을 통하여 경로 생성의 정확성을 잃지 않는 범위 내에서 계산 시간이 빠른 적절한 값을 선택한다.

$$\theta_0 - k_1 \leq \theta \leq \theta_0 + k_1, \quad \phi_0 - k_1 \leq \phi \leq \phi_0 + k_1 \quad (4)$$

(x, y)	
D=1	(1,0)(0,1)
D=2	(2,0)(1,1)(0,2)
D=3	(3,0)(2,1)(2,2)(1,2)(0,3)
D=4	(4,0)(3,1)(3,2)(2,3)(1,3)(0,4)
D=5	(5,0)(4,1)(4,2)(4,3)(3,3)(3,4)(2,4)(1,4)(5,0)

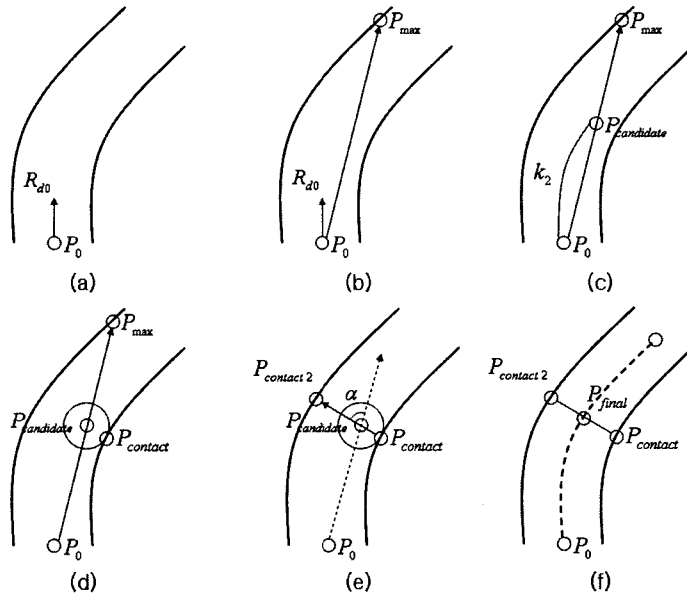


그림 5 가시성을 이용한 자동 경로 생성 과정

이렇게 얻어진 P_0 에서 기준 방향 R_{d0} 를 중심으로 그림 5(b)와 같이 가시성이 가장 높은 점을 P_{max} 라고 하자. 다음으로 그림 5(c)와 같이 P_0 과 P_{max} 를 잇는 선 위에서 $P_{candidate}$ 를 다음과 같이 선택할 수 있다. 식 (5)에서 k_2 는 생성되는 제어점의 간격을 조절하는 인자로 특정한 k_2 의 범위 내에서 k_2 가 크면, 제어점이 적게 생성되어 경로 계산 시간이 빠르지만, 보간을 하는 범위가 넓어져서 대장의 폭이 매우 좁은 일부분에 대해서 정확한 경로의 생성이 어려울 수 있다. k_2 가 작으면, 제어점이 많이 생성되어 경로 계산 시간은 오래 걸리지만, 보간을 하는 범위가 줄어들어 보다 정확한 경로의 생성이 가능하다. 따라서, k_2 는 실험을 통하여 경로 생성의 정확성을 잃지 않는 범위 내에서 계산 시간이 빠른 적절한 값을 선택한다.

$$P_{candidate} = P_0 + k_2 \cdot (P_{max} - P_0) \quad (5)$$

위에서 $P_{candidate}$ 은 가시성이 가장 높다는 기준으로 선택된 점이다. 그림 5(d)의 과정을 통해 $P_{candidate}$ 을 다음과 같이 대장 단면의 중심에 위치하도록 보정해 준다. $P_{candidate}$ 을 중심으로 하는 가상의 구를 만든다. 반지름이 1인 구부터 구를 팽창시켜 나가면서 대장 벽과 접하는 점을 찾는다. 이때 2.2절에서 미리 만든 구의 오프셋 테이블을 이용한다. 오프셋 테이블의 x, y, z 상대좌표 값을 인덱스 0부터 하나씩 읽어와 $P_{candidate}$ 에 각 오프셋 값을 더함으로써 팽창하는 구 표면의 좌표를 빠르게 구할 수 있다. 구를 팽창시키면서 구가 대장 벽과 접하게 되면, 그 점을 $P_{contact}$ 로 택한다. 보다 정확한 경

로를 얻기 위해서 구를 팽창시켜가면서 다수의 $P_{contact}$ 들을 찾는다. $P_{contact}$ 를 찾을 때, 아래와 같은 제약 사항을 둔다.

$$\alpha = \cos^{-1} \frac{(P_{contact} - P_{candidate}) \cdot (P_{candidate} - P_0)}{|P_{contact} - P_{candidate}| |P_{candidate} - P_0|} \quad (6)$$

이 때 α 는 그림 5(e)에서와 같이 $\overline{P_{contact} - P_{candidate}}$ 와 $\overline{P_{candidate} - P_0}$ 가 이루는 각도로서 식 (7)의 조건을 만족하는 경우에만 $P_{contact}$ 를 선택한다. 식 (7)에서 k_3 는 $\overline{P_{contact} - P_{candidate}}$ 와 $\overline{P_{candidate} - P_0}$ 에 대한 각도의 90도를 기준으로 한 허용 범위를 결정하고, 이 조건은 대장의 불규칙한 형상으로 인해 $P_{contact}$ 가 경로의 진행 방향의 앞부분에 위치하는 것을 방지한다. k_3 는 실험을 통해 정확한 경로 생성을 위한 최적의 값으로 선택한다.

$$|\alpha - 90| < k_3 \quad (7)$$

알고리즘 1은 위의 제약 사항을 고려하여 $P_{candidate}$ 을 보정하기 위해 $P_{contact}$ 를 찾는 과정을 나타낸 것이다. 접점들이 찾아졌다면 그림 5(e)와 같이 각 접점 $P_{contact}$ 에서 $P_{candidate}$ 으로 향하는 광선이 대장 벽과 만나는 점 $P_{contact2}$ 를 구한다. 각 $P_{contact}$ 에 대하여 $P_{contact2}$ 를 계산하고, $P_{contact}$ 와 $P_{contact2}$ 의 중심에 해당하는 점들의 평균점이 최종적으로 구해지는 경로상의 점 P_{final} 이 된다. 알고리즘 2는 $P_{candidate}$ 이 주어졌을 때, P_{final} 을 계산하는 과정을 나타낸 것이다.

알고리즘 1 구와 접하는 점들을 찾는 방법

```

PointArray GetContactPoint(Point Pcandidate)
{
    PointArray arrPt;
    for (count = 0; count < OFFSET_TABLE_SIZE;
        count++) {
        pt = Pcandidate + GetOffset(count); // Pcandidate가
        중심인 구면 상의 점
        D = GetDensity(pt); // pt에서의 밀도값
        if (abs(WALL_DENSITY - D) < THRESHOLD) {
            // pt가 대장벽의 접이면
            if (SatisfyAngleCondition(pt, Pcandidate))
                // α에 대한 제약 조건
                arrPt.Add (pt);
        }
    }
    return arrPt;
}
    
```

알고리즘 2 구의 중심과 접점을 이용하여 제어점을 찾는 방법

```

Point GetControlPoint(Point Pcandidate)
{
    PointArray arrPt = GetContactPoint(Pcandidate);
    for (i = 0; i < NUM_OF_CONTACT_POINT; i++) {
        Pcontact = arrPt.GetPoint(); // 접점
        Pray = Normalize(Pcandidate - Pcontact);
            // 광선의 방향
        Pcontact2 = Pcandidate + Pray*GetDistantToColonWall
            (Pcandidate, Pray);
        Pfinal += (Pcontact + Pcontact2)/2;
            // 평균을 구하기 위해서 계산
    }
    return Pfinal/NUM_OF_CONTACT_POINT;
}
    
```

이렇게 찾아진 P_{final} 은 경로가 되는 곡선의 제어점으로 사용하기 위해 저장해 둔다. 다음 제어점을 생성하기 위해 새로운 광선의 방향은 $P_{final}-P_0$ 로 하고, P_{final} 을 새로운 초기 위치로 하여 위의 과정을 반복한다. 광선이 도착점에 이르면 그림 5(f)에 점선으로 표시된 것과 같이 저장해 두었던 P_{final} 들을 제어점으로 하는 삼차 스플라인(cubic spline) 곡선을 생성한다[16]. 이렇게 얻어진 곡선이 최종적으로 원하는 가상카메라의 경로가 된다.

3. 실험 결과

실험은 Intel Pentium IV 2.4 GHz CPU와 1GB의 메모리를 장착한 PC에서 이루어졌고, Intel Compiler 5.0을 사용하였다. 실험에 사용된 데이터는 표 1과 같다.

표 1 실험에 사용된 데이터

데이터	영역	해상도	영역 분할된 대장의 복셀 수
1	대장	512*512*213	2.84 M
2	대장	512*512*253	2.53 M
3	대장	512*512*368	3.34 M
4	대장	512*512*579	4.85 M

3.1 대장의 영상 분할

가상 경로를 생성하기 앞서서 주어진 데이터는 3차원 씨앗점 영역 성장법(3D-seeded region growing)의 영상 분할 기법을 사용하여 대장 부분만을 분리하여 사용된다. 그림 6은 영상 분할된 대장을 3차원 볼륨 렌더링한 형상이다. 본 제안 방법에서 경로 생성 과정에서는 영상 분할이 필요하지 않으나 초기점 입력 과정과 실험 결과의 시각적 확인에 분할된 대장이 필요하다.

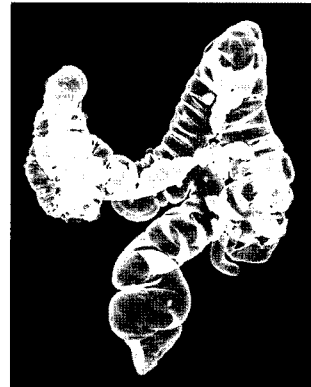


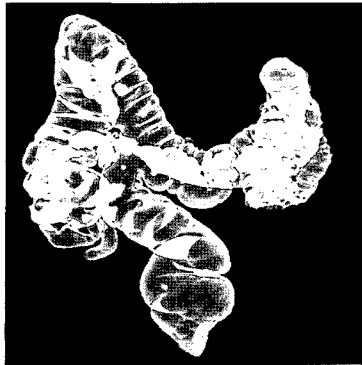
그림 6 영상 분할된 대장(데이터 2)

3.2 초기점 입력 결과

사용자는 그림 7(a)의 분할된 대장의 3차원 볼륨 렌더링된 영상에 마우스로 2차원 좌표를 입력함으로써 초기점을 지정한다. 이 초기점에 대한 3차원 좌표가 대장 단면의 중심에 제대로 위치하였는지 확인하기 위해 그림 7에 단면 영상(axial image), 관상 영상(coronal image), 시상 영상(sagittal image)을 나타내었다. 그림 6에서와 같이 입력된 점이 각 단면의 중심에 위치했음을 알 수 있다.

3.3 자동 경로 생성 결과

최적의 인자 값을 결정하기 위하여 경로 생성률을 식 (8)과 같이 정의한다. 식 (8)에서 기준 경로는 의사에 의하여 정밀하게 수작업으로 설정된 경로를 의미한다. 오차 범위 t 는 정확한 진단에 영향을 주지 않는 범위 내에서 3 mm로 정하였다. 즉, 각 슬라이스 내에서 기준 경로와의 교차점을 중심으로 반지름 3 mm인 원 내부



(a) 3차원 볼륨 렌더링 영상



(b) 단면 영상



(c) 관상 영상



(d) 시상 영상

그림 7 초기점의 위치

에 위치하는 본 제안 방법을 통해 생성된 경로와 슬라이스와의 교차점 개수의 비율을 측정한다.
 경로 생성률(%) =

$$\frac{\sum \left(\frac{\text{본 제안방법을 통해 생성된 경로와 각 슬라이스와의 교차점 중 기준 경로와 오차범위내에 위치한 점의 개수}}{\sum (\text{기준 경로와 각 슬라이스와의 교차점의 개수})} \right)}{\times 100} \quad (8)$$

다음의 실험을 통해 3장에서 언급한 인자 k_1, k_2, k_3 의 값을 정하였다. 표 2는 k_2 와 k_3 의 값을 고정하고 k_1 의 값을 변화시킨 경우 경로 생성 시간과 생성률을 나타낸 것이다. 대체적으로 k_1 의 값은 클수록 경로 생성률 면에서 좋은 결과를 얻을 수 있었다. 하지만, k_1 의 값이 30° 이상이 되면, 경로 생성률이 향상되는 양에 비해 경로

생성 시간은 크게 늘어남을 알 수 있다. 따라서 k_1 의 값은 30°로 정하였다. 테이타 3, 4에 대해서는 경로 생성률 결과가 좋지 않은 이유는 k_3 의 값이 적절히 선택되지 못했기 때문이다.

표 3은 k_1 과 k_3 의 값을 고정하고 k_2 의 값을 변화시킨 경우 경로 생성 시간과 생성률을 나타낸 것이다. 대체적으로 k_2 의 값은 작을수록 경로 생성률 면에서 좋은 결과를 얻을 수 있었다. 하지만, k_2 의 값이 0.5이하가 되면, 경로 생성률이 향상되는 양에 비해 경로 생성 시간은 크게 늘어남을 알 수 있다. 따라서 k_2 의 값은 0.5로 정하였다.

표 4는 k_1 과 k_2 의 값을 고정하고 k_3 의 값을 변화시킨 경우 경로 생성 시간과 생성률을 나타낸 것이다. 45°를 기준으로 그 이하나 이상인 경우 경로 생성률이 크게

표 2 k_1 에 따른 경로 생성 시간과 경로 생성률 ($k_2=0.5, k_3=30^\circ$ 로 고정)

k_1	20°		25°		30°		35°		40°	
	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)
4개의 테이타 평균	14	68.31	20	70.01	25	72.96	29	73.33	32	74.05

표 3 k_2 에 따른 경로 생성 시간과 경로 생성률 ($k_1=30^\circ$, $k_3=45^\circ$ 로 고정)

k_2	0.3		0.4		0.5		0.6		0.7	
	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)
4개의 데이터 평균	32	97.92	28	97.24	25	96.90	19	77.34	15	66.50

표 4 k_3 에 따른 경로 생성 시간과 경로 생성률 ($k_1=30^\circ$, $k_2=0.5$ 로 고정)

k_3	30°		40°		45°		50°		60°	
	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)	시간 (초)	경로 생성률 (%)
4개의 데이터 평균	25	72.96	23	91.61	25		26	92.44	16	70.37

표 5 선택된 인자 값

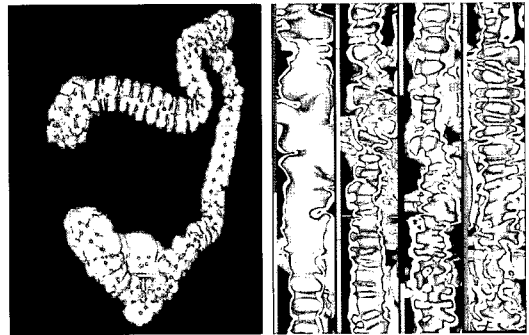
k_1	k_2	k_3
30°	0.5	45°

낮아지는 경향을 보였다. 따라서 k_3 의 값은 45° 로 정하였다.

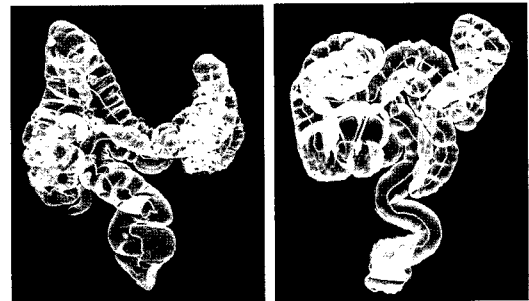
위의 실험을 통해 경로 생성률이 높으면서 경로 생성 시간이 적게 걸리는 최적의 인자 k_1 , k_2 , k_3 를 표 5와 같이 선택하였다. 이 인자를 사용하여 4개의 데이터에 대해서 경로 생성에 평균 25초가 소요되었고, 평균 경로 생성률은 96.90%였다. 기존에 알려진 가장 빠르게 경로를 생성하는 방법들[17,18]이 약 10여분이 소요되는 것에 비하면, 본 제안 방법의 속도는 기존의 방법에 비해 매우 빠른 것이 실험을 통해 확인되었다. 경로 생성률이 100.00%가 되지 않은 이유는 의사가 표시한 경로와 본 방법을 통해 생성된 경로가 대장 단면의 중심 부위에 위치하더라도 대장의 단면적이 매우 큰 부위에서는 오차가 3mm보다 크게 날 수 있기 때문이다. 하지만, 이 경우에도 본 방법을 통해 생성된 경로는 충분히 대장의 중심 부위에 위치하여 정확한 진단이 가능하였다.

그림 8(a)에서는 경로에 생성된 제어점을 표시하여 제어점이 대장의 형태를 정확하게 모델링한 것을 확인할 수 있다. 그림 8(b)는 경로에 수직한 대장의 단면 영상들을 전체 경로에 대하여 보여주는 그림으로 제안한 방법의 경로가 대장 단면의 정확한 진단에 효과적임을 나타낸다. 그림 9는 결정된 인자 값으로 자동으로 생성된 경로를 분할된 대장에 표시한 그림이다. 3차원 상에서 본 제안 방법으로 생성된 경로가 대장의 중심 부위에 위치하는 것을 확인할 수 있다. 그림 10은 경로를 2D 상에서 표시한 그림으로 대장의 단면 중심에 경로가 위치하는 것을 확인할 수 있다.

그림 11은 경로를 따라가면서 실제로 가상 대장내시경을 시행한 것으로 그림 11(a)와 같이 곡률이 크지 않은 부분이나 그림 11(b)와 같이 곡률이 큰 부분에 대해



(a) 3차원 영상 (b) 대장 단면 영상
그림 8 생성된 경로 (데이터 1)



(a) 데이터 2 (b) 데이터 4
그림 9 생성된 경로의 3차원 영상

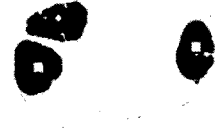
서도 생성된 경로가 대장 단면의 중심에 위치하여 대장 벽을 관찰하기에 유용함을 알 수 있다. 또한, 그림 11(c)와 같이 종양(polyp)의 발견과 진단에 유용하게 사용될 수 있다. 그림에서와 같이 종양의 지름을 측정하여 실제 길이 5mm이하의 종양은 큰 의미 없는 병변으로 간주되고, 6~7mm에 해당하는 종양은 가상 대장내시경 검사를 통하여 단기간 추적검사가 필요하다. 그리고, 그림 11(c)에서와 같이 8mm이상의 종양은 즉시 추적 대장내시경 검사가 필요하다[7].



(a) n = 28



(b) n = 35



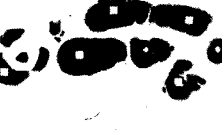
(c) n = 48



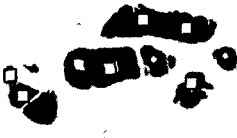
(d) n = 64



(e) n = 90



(f) n = 110



(g) n = 117

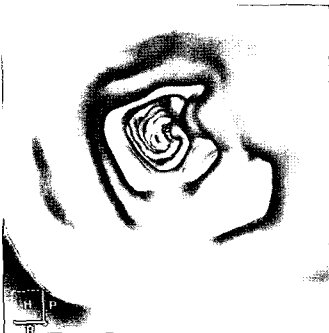


(h) n = 130

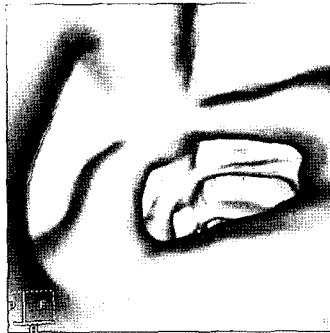


(i) n = 135

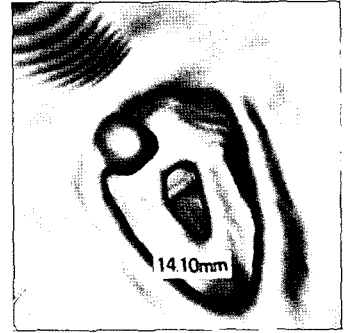
그림 10 n번째 슬라이스 내에 생성된 경로의 2D 단면 영상 (데이터 2)



(a) 곡률이 낮은 부분



(b) 곡률이 큰 부분



(c) 종양의 발견 및 진단

그림 11 가상 대장내시경 (데이터 1)

4. 결론

본 논문에서는 기존의 침습적인 대장진단법의 단점을 보완할 수 있는 가상내시경을 위한 3차원 대장 형상 내부의 중심선 탐색 경로를 구하기 위한 방법을 제안하였다. 가시성을 이용한 자동 경로 생성법을 제안하여 전처리 과정에서 거리맵 등의 생성을 필요로 하는 기존의 방법을 개선하였다. 광선 추적법에서의 광선 진행 방식과 가시성을 경로 생성에 응용하였으며, 그 결과 전처리 과정이 필요 없고, 전체 경로를 구성하는 점들이 아닌 경로를 대표하는 소수의 제어점들을 생성하므로 빠르고, 정확한 경로 생성이 가능하였다. 또한, 가시성을 이용하여 자연스러운 카메라의 움직임이 가능하였다. 네 가지 대장 임상 데이터에 대해 탐색경로 생성 실험을 수행하였으며, 그 결과 카메라의 진행이 원활하고 카메라의 위치가 충분히 대장의 중앙에 위치하여 종양의 진단에 유용함을 확인하였다. 본 논문의 제안 방법은 두 갈래 이상의 분지가 없는 여러 형태의 관 구조물의 중심 탐색 경로 생성에 효과적으로 적용될 수 있다.

참고 문헌

[1] S. Dogramadzi, C. R. Allen, and G. D. Bell, "Computer Controlled Colonoscopy," *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, Vol.1, pp. 210-213, May 1998.

[2] S. J. Phee, W. S. Ng, I. M. Chen, F. Seow-Choen, and B. L. Davies, "Automation of Colonoscopy. II. Visual control aspects," *IEEE Engineering in Medicine and Biology Magazine*, Vol.17, No.3, pp.81-88, May-June 1998.

[3] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang, "3D Virtual Colonoscopy," *Proceedings of IEEE Biomedical Visualization*, pp.26-32, October 1995.

[4] T. Y. Lee, P. H. Lin, C. H. Lin, Y. N. Sun, and X. Z. Lin, "Interactive 3-D Virtual Colonoscopy System," *IEEE Transactions on Information Technology in Biomedicine*, Vol.3, No.2, pp.139-150, 1999.

[5] L. Hong, S. Muraki, A. E. Kaufman, D. Bartz, and T. He, "Virtual Voyage: Interactive Navigation in the Human Colon," *Proceedings of ACM SIGGRAPH 1997*, pp.27-34, 1997.

[6] G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffey, and S. Napel, "Perspective Vol. Rendering of CT and MR Images: Applications for Endoscopic Imaging," *Radiology*, Vol.199, No.2, pp.321-330, May 1996.

[7] Perry J. Pickhardt, J. Richard Choi, Inku Hwang,

James A. Butler, Michael L. Puckett, Hans A. Hildebrandt, Roy K. Wong, Pamela A. Nugent, Pauline A. Mysliwiec, William R. Schindler, "Computed Tomographic Virtual Colonoscopy to Screen for Colorectal Neoplasia in Asymptomatic Adults," *The New England Journal of Medicine*, Vol.349, pp.2191-2200, 2003.

[8] D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin, "Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality," *Annual Meeting of American Roentgen Ray Society*, p.104, 1994.

[9] D. S. Paik, C. F. Beaulieu, R. B. Jeffery, G. D. Rubin, and S. Napel, "Automated Flight Path Planning for Virtual Endoscopy," *Medical Physics*, Vol.25, No.5, pp.629-637, 1998.

[10] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol.1, pp.269-271, 1959.

[11] T. He, L. Hong, "Reliable Navigation for Virtual Endoscopy," *IEEE Nuclear Science Symposium 1999*, Vol.3, pp.1339-1343, 1999.

[12] Y. Zhou, A. E. Kaufman, A. W. Toga, "Three-dimensional Skeleton and Centerline Generation Based on an Approximate Minimum Distance Field," *The Visual Computer*, Vol.14, pp.303-314, 1998.

[13] I. Bitter, A. E. Kaufman, M. Sato, "Penalized-distance Volumetric Skeleton Algorithm," *IEEE Transactions on Visualization and Computer Graphics*, Vol.7, No.3, pp.195-206, 2001.

[14] Marc Levoy, "Efficient Ray Tracing of Volume Data," *ACM Transactions on Graphics*, Vol.9, No.3, pp.245-261, 1990.

[15] J. Dehmeshki, H. Amin, Wing Wong, M. E. Dehkordi, N. Kamangari, M. Roddie, J. Costelo, "Automatic polyp detection of colon using high resolution CT scans," *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, Vol.1, pp.577-581, 2003.

[16] R. H. Bartels, J. C. Beatty, B. A. Barsky, An Introduction to Splines for Use in Computer Graphics and Geometric Modelling. San Francisco, pp.9-17, Morgan Kaufmann, 1998.

[17] I. Bitter, M. Sato, M. Bender, K. McDonnel, A. Kaufman, M. Wan, "CEASAR: A Smooth, Accurate and Robust Centerline Extraction Algorithm," *Proceedings IEEE Visualization*, pp.45-52, 2000.

[18] M. Wan, F. Dacheille, A. Kaufman, "Distance-field Based Skeletons for Virtual Navigation," *Proceedings of IEEE Visualization*, pp.239-246, 2001.

이 정 진

정보과학회논문지 : 시스템 및 이론
제 32 권 제 1,2 호 참조

강 문 구

정보과학회논문지 : 시스템 및 이론
제 32 권 제 1,2 호 참조



조 명 수

2002년 2월 경북대학교 컴퓨터공학과 학사. 2004년 2월 서울대학교 컴퓨터공학과 석사. 2003년~현재 (주) 인피니트 테크놀로지 연구원. 관심분야는 불륨 렌더링, electronic cleansing, 자동 경로 생성

신 영 길

정보과학회논문지 : 시스템 및 이론
제 32 권 제 1,2 호 참조