

논문 2011-48SP-1-10

내용기반 동영상 기하학적 변환을 위한 실시간 처리 기법

(A Real Time Processing Technique for Content-Aware Video Scaling)

이 강 희*, 유 재 욱*, 박 대 현*, 김 윤**

(Kang-Hee Lee, Jae-Wook Yoo, Dae-Hyun Park, and Yoon Kim)

요 약

본 논문에서는 동영상이 가지고 있는 콘텐츠를 보존하면서 영상의 크기를 조절할 수 있는 실시간 동영상 크기 변환 기법을 제안한다. 제안하는 방법은 동영상 내의 연속하는 두 프레임 사이에 존재하는 상관성(correlation)을 이용하여, 이전 프레임의 seam 정보로부터 현재 프레임의 seam을 결정한다. 따라서, 전체 프레임들을 분석하지 않으면서도 콘텐츠의 떨림 현상을 발생 시키지 않는다. 먼저, 전체 동영상 내에서 특징이 서로 비슷한 프레임들을 scene으로 구분하고, 각 scene 내의 첫번째 프레임은 정지영상의 seam carving을 사용하여 최대한 콘텐츠를 보존할 수 있도록 크기를 변환한다. 이 때, 영상의 크기를 변환하기 위해 추출한 seam에 대한 정보를 저장하고, 그 이후의 프레임들은 이전 프레임에서 저장된 seam 정보를 참조하여 프레임 단위로 영상의 크기를 조절한다. 실험 결과는 제안하는 방법이 처리 속도와 메모리 사용량 면에서 실시간 처리에 적합하고, 영상이 가지고 있는 콘텐츠를 보존하면서 영상의 크기를 조절할 수 있음을 보여준다.

Abstract

In this paper, a new real time video scaling technique which preserved the contents of a movie was proposed. Because in a movie a correlation exists between consecutive frames, in this paper by determining the seam of the current frame considering the seam of the previous frame, it was proposed the real time video scaling technique without the shaking phenomenon of the contents even though the entire video is not analyzed. For this purpose, frames which have similar features in a movie are classified into a scene, and the first frame of a scene is resized by the seam carving at the static images so that it can preserve the contents of the image to the utmost. At this time, the information about the seam extracted to convert the image size is saved, and the sizes of the next frames are controlled with reference to the seam information stored in the previous frame by each frame. The proposed algorithm has the fast processing speed of the extent of being similar to a bilinear method and preserves the main content of an image to the utmost at the same time. Also because the memory usage is remarkably small compared with the existing seam carving method, the proposed algorithm is usable in the mobile terminal in which there are many memory restrictions. Computer simulation results indicate that the proposed technique provides better objective performance and subjective image quality about the real time processing and shaking phenomenon removal and contents conservation than conventional algorithms.

Keywords: 내용기반 영상 기하학적 변환, 심 카빙, Content-Aware Image Resizing, Seam carving

I. 서 론

무선통신과 모바일 기기의 발전에 따라 영상에 대한 활용도가 증가하고 있고, 이에 따라 영상은 이미 중요

한 정보매체로 자리매김 하였다. 영상을 보다 효과적으로 이용하기 위해 영상의 크기를 변경하는 작업은 중요하고, 많은 수의 영상들을 보다 효과적으로 검색하기 위해 영상의 사이즈를 작게 만드는 썸네일(thumbnail) 기법 등은 없어서는 안 될 기술이 되었다. 그러나 기존의 단순한 크기 변경 기법은 영상이 가지는 콘텐츠에 대한 고려를 하지 않고 영상의 모든 부분을 동일하게 변경하기 때문에 콘텐츠가 변형되고 왜곡된다. 따라서

* 학생회원, ** 정회원, 강원대학교 컴퓨터정보통신공학과

(Dept. of Computer and Communications

Engineering, Kangwon National University)

접수일자: 2010년4월16일, 수정완료일: 2010년12월20일

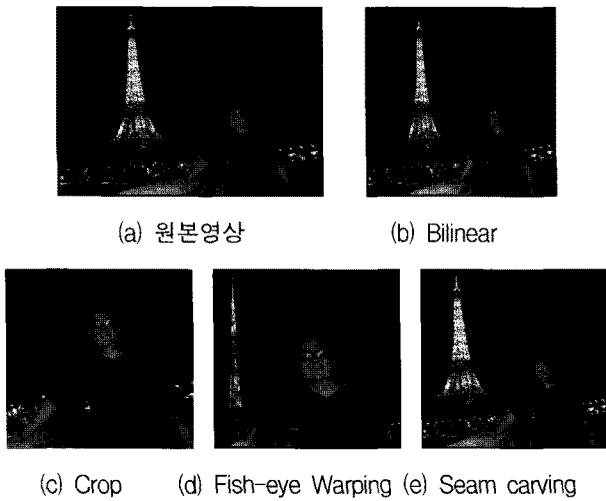


그림 1. 여러 가지 영상 크기 변환 방법 비교
 Fig. 1. Comparison of several methods to resize the image.

효과적으로 영상의 크기를 변경하기 위해서는 영상이 가지는 콘텐츠를 보존하면서 영상의 크기를 변경하는 내용 기반의 영상 크기 조절 방법이 필요하다.

대표적인 크기변환 방식인 보간법에서는 원본 영상의 비율과 다른 비율로 크기변환을 할 때에 영상이 가지는 콘텐츠의 변형이 발생한다(그림 1(b)). 이러한 현상을 방지하기 위해 연구된 것이 내용기반 영상 기하학적 변환이다. 가장 단순한 방법으로 영상내의 주요 콘텐츠를 찾아내고, 찾아낸 콘텐츠만 잘라 보여주는 crop 기법^[1]이 있다(그림 1(c)). 하지만 이 기법은 콘텐츠 이외의 영상 정보가 삭제된다는 단점이 있다. 이를 개선하여 주요 콘텐츠를 확대하고 이외의 정보를 축약하여 보여주는 Fish-eye Warping 기법^[2]이 있다(그림 1(d)). Fish-eye Warping은 영상의 주요 콘텐츠는 최대한 보존하지만 이외의 영상정보들은 왜곡이 심하다는 단점이 있다. 최근에 소개된 seam carving 기법^[3]은 영상 내에서 중요도가 낮고 영상을 왜곡시키는 영향력이 낮은 픽셀들을 삭제 혹은 추가하는 방법으로 영상 크기 변환에 높은 성능을 인정받고 있다(그림 1(e)). 이 밖에도 콘텐츠를 보존하면서 크기를 변경하는 방법에 대한 연구들이 진행되고 있다^[4-6, 13-16].

동영상에서의 내용 기반 영상 기하학적 변환^[7-8]은 정지 영상에서의 내용기반 영상 기하학적 변환과는 차이가 있다. 정지 영상에서는 영상 공간에 대해서만 처리하면 되지만, 동영상에서는 시간이라는 개념이 추가되기 때문에 인접한 프레임 간의 상호 고려가 필요하다. 상호 고려가 되지 않으면 각 프레임 영상이 가지고

있는 콘텐츠의 형태는 보존되지만, 시간축의 연결성을 잃기 때문에 영상 내에서 콘텐츠 위치의 불규칙적인 움직임이 발생한다. 이러한 현상은 콘텐츠의 떨림 현상(jitter)이라고 하며, 내용기반 영상 기하학적 변환 알고리즘이 동영상에 적용되기 위해선 콘텐츠의 떨림 현상의 방지는 필수적이다.

Seam carving을 동영상에 적용한 video carving^[9-10]은 시간축으로 프레임들을 연결한 3차원 큐브를 이용한다. 동영상에서는 영상의 콘텐츠가 시간의 흐름에 따라 위치적, 기하학적으로 변화하기 때문에, 효과적인 영상 크기 변환을 위해 공간축과 시간축의 energy를 모두 고려하여 3차원 큐브 전체를 탐색해야 한다. 이를 정지영상의 seam carving에서 사용하는 다이내믹 프로그래밍 기법^[17-18]으로 처리하면 많은 양의 메모리와 연산량을 필요로 하기 때문에 video carving에서는 graph cut 기법^[19-20]을 사용한다.

3차원 큐브는 시간축으로 프레임들을 연결한 프레임들의 집합이고, 큐브 내부에서 붉은 색으로 나타난 면이 큐브 전체를 탐색하여 얻은 하나의 최적 seam set이다(그림 2). 하나의 seam set을 추가 또는 삭제하는 것을 통해 동영상의 크기를 1픽셀씩 변환하는 것이 가능하다. 따라서 다양한 크기의 변환을 위해서는 원하는 seam set의 개수만큼 3차원의 큐브 탐색을 거쳐야 한다. Video carving은 3차원 큐브를 생성해야 하기 때문에 많은 메모리를 사용하며, 3차원 큐브를 탐색하고 분석하는 데는 많은 연산량을 필요로 하기 때문에 모바일 단말과 같은 한정된 자원을 가진 시스템에서 실시간으로 처리하는 것은 불가능하다.

본 논문에서는 실시간 처리를 위한 내용기반의 영상 기하학적 변환 기법을 제안한다. 동영상에서 연속하는 두 프레임 사이엔 상관성(correlation)이 존재하고, 연속

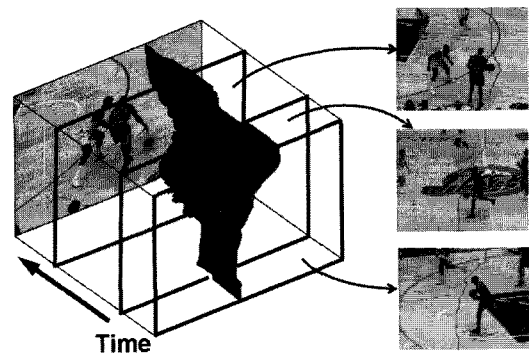


그림 2. Video carving
 Fig. 2. Video carving.

하는 두 프레임의 seam들은 비슷하게 나타날 가능성이 높다. 본 논문에서는 상관성을 이용하여 이전 프레임의 seam을 고려하여 현재 프레임의 seam을 결정함으로써, 전체 영상을 분석하지 않으면서 콘텐츠의 떨림 현상이 없는 실시간 동영상 크기 변환 기법을 제안한다.

제안하는 기법에서는 카메라가 연속적으로 촬영한 영상인 scene 단위로 연산하며, scene내의 모든 프레임은 비슷한 영상특징을 가진다. 먼저, 동영상내에서 효과적으로 scene을 분리하기 위하여, 움직임에 민감한 명도 값의 차와 색변화에 민감한 히스토그램 값의 차를 이용하여 장면전환(scene change)을 검출한다^[11~12]. 장면전환이 발생하여 새로운 scene이 시작되면, scene의 첫 프레임에서는 정지 영상의 seam carving 기법을 통하여 영상의 크기변환을 한다. 이 때, seam carving 기법에서 추출한 seam과 그 좌표에 따른 energy 정보를 저장한다. 이 저장된 정보들을 통하여 다음 프레임의 새로운 seam을 계산함으로써, 3차원 큐브를 구성하는 것과 달리 제안하는 알고리즘은 모든 영상 프레임의 정보 없이 프레임 단위로 실시간 처리된다.

본 논문의 구성은 다음과 같다. II장에서 제안한 알고리즘을 기술하고, III장에서는 제안한 알고리즘과 기존의 알고리즘들을 비교하는 실험을 통하여 본 알고리즘의 우수성을 입증한다. 최종적으로 IV장에서 결론을 맺는다.

II. 제안하는 동영상에서의 크기 변환 알고리즘

본 논문에서 제안하는 실시간 내용기반 영상 기하학적 변환 시스템은 그림 3과 같이 크게 scene change detection, seam carving, seam determination using previous frame의 3부분으로 구성된다.

입력영상은 프레임 단위로 획득되며, RGB 값을 갖는

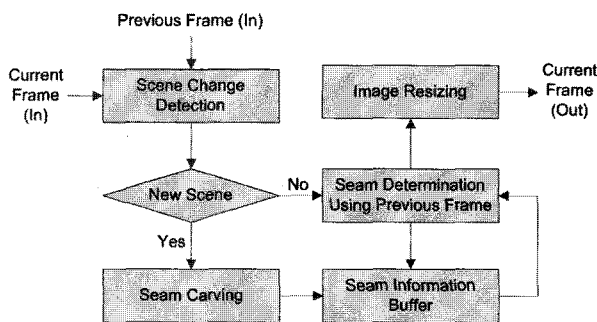


그림 3. 전체 시스템 블록 다이어그램
Fig. 3. Overall system block diagram.

다. 제안하는 알고리즘은 장면 전환 검출을 통하여 새로운 scene이 시작되었는지 여부를 판단한다. 새로운 scene이 시작되었을 경우에는 저장된 정보를 무시하고, 정지 영상에서의 seam carving 기법을 통하여 seam을 탐색한다. 이 때 탐색된 seam에 대한 정보를 저장한 뒤, 영상의 크기 변환을 수행한다. 반면에 장면 전환이 발생하지 않은 경우, 이전 프레임의 저장된 seam 정보를 이용하여 현재 프레임에서 seam을 계산하고 이를 통해 영상의 크기를 조절한다.

2.1. 장면 전환 검출^[11~12]

일반적으로 PC 환경과 모바일 환경에서 사용되는 동영상은 1초에 10 프레임 이상의 프레임들을 가지고 있다는 성질을 이용하여 장면 전환 검출을 10 프레임마다 수행한다. 사용한 장면 전환 검출 방법은 이전 10 프레임 내에서의 영상서의 특징 값을 나타낸다.

$$f_{hn} = \sum_{k=0}^{255} |h_n(k) - h_{n-1}(k)|,$$

$$f_{in} = \sum_{j=0}^{height-1} \sum_{i=0}^{width-1} |i_n(i,j) - i_{n-1}(i,j)| \quad (1)$$

여기서, $h_n(k)$ 는 n 번째 프레임에서 k 의 픽셀값을 가진 픽셀의 개수를 나타내며, 이웃하는 프레임 간의 $h(k)$ 의 차를 색상에 민감한 히스토그램의 특징 값 f_h 로 정의한다. 또한, $i_n(i,j)$ 는 n 번째 프레임의 (i,j) 번째 위치에서의 픽셀값이며, f_i 는 움직임에 민감한 명도의 특징 값을 나타낸다. Height와 width는 영상의 해상도를 나타낸다.

10개의 f_i 와 f_h 를 확보 후, 아래의 식 (2)를 통하여 이웃하는 두 프레임 사이의 장면 전환을 검출한다.

$$\begin{cases} \text{scenchange,} & \text{if } m_i > 3s_i \text{ and } m_h > 3s_h \\ \text{nonchange,} & \text{otherwise} \end{cases}$$

$$m_i = \max_{\alpha \in F_i}(\alpha), \quad s_i = \max_{\alpha \in F_i, \alpha \neq m_i}(\alpha),$$

$$m_h = \max_{\alpha \in F_h}(\alpha), \quad s_h = \max_{\alpha \in F_h, \alpha \neq m_h}(\alpha),$$

$$F_i = \{f_{i(n-9)}, f_{i(n-8)}, \dots, f_{i(n-1)}, f_{in}\},$$

$$F_h = \{f_{h(n-9)}, f_{h(n-8)}, \dots, f_{h(n-1)}, f_{hn}\} \quad (2)$$

F_i 와 F_h 는 이전 10 프레임에서 계산한 특징값들이다. m_i 은 F_i 내의 가장 큰 값이고, s_i 는 두번째 큰 값이며,

마찬가지로 m_h 는 F_h 내의 가장 큰 값, s_h 는 두번째 큰 값이다. 즉, 식 (2)는 가장 큰 값이 두번째로 큰 값의 3 배 이상일 때 장면 전환이라고 판단한다.

2.2. 첫 번째 프레임에서의 seam carving

장면 전환이 발생한 후, 첫 번째 프레임에 대하여 기존의 정지영상 seam carving을 사용한다. 그리고 다음 프레임의 seam을 구할 때 참조하기 위하여, 첫 번째 프레임에서 구한 seam들의 정보를 저장한다. 식 (3)은 $W \times H$ 크기의 프레임에서 저장되는 seam의 정보들을 보여준다.

$$S_f = [S_1, S_2, S_3, \dots], S = (C, E, e_t),$$

$$C = [x_0, x_1, \dots, x_{W-1}] \text{ or } [y_0, y_1, \dots, y_{H-1}] \quad (3)$$

$$E = [e_0, e_1, \dots, e_{W-1}] \text{ or } [e_0, e_1, \dots, e_{H-1}]$$

S_f 는 하나의 프레임에서 구한 seam들의 배열이며, 타겟 영상의 크기에 의해 개수가 정해진다. 집합 S 는 하나의 seam에 대한 정보를 나타내며, seam의 좌표 배열 C 와 각 좌표에서의 에너지를 나타내는 배열 E , 그리고 E 값을 모두 누적한 값 e_t 로 구성되어 있다. 이 때, C 는 seam의 x, y 좌표를 모두 저장하는 것이 아니라 vertical 방향의 seam일 경우는 x 좌표만을, horizontal 방향의 seam일 경우는 y 좌표만을 각각 저장한다. H 와 W 는 각각 height와 width를 나타낸다.

2.3. 이전 프레임의 seam을 이용한 현재 프레임의 seam 결정

장면 전환이 일어나지 않고 현재 프레임이 이전 프레임과 같은 scene에 속할 경우, 이전 프레임에서 저장된 seam의 정보를 이용하여 현재 프레임의 seam을 추출한다. 이 과정에서 같은 scene내의 연속하는 두 프레임들은 서로 유사하다는 상관성을 이용한다. 영상이 유사하다는 것은 상관성에 의해 energy 분포도 또한 유사한 것을 의미하며, 두 영상의 seam도 유사한 형태로 나온다. 이를 이용하여 이전 프레임에서 추출한 seam의 좌표의 일정 범위 내에서 현재 프레임의 seam을 추출한다. 즉, seam의 추출 시 시간축의 연속성이 고려되어야 하며, 이런 고려 없이 매 프레임마다 독립적으로 seam을 추출할 경우에 주요 콘텐츠가 흔들리는 떨림 현상(jitter)이 발생한다. 이런 떨림 현상이 발생하는 이유는 콘텐츠를 중심으로 주변에 결정되는 seam의 개수가 프



그림 4. 프레임 별 독립적 seam carving 결과
Fig. 4. The result of applying seam carving in each frame.

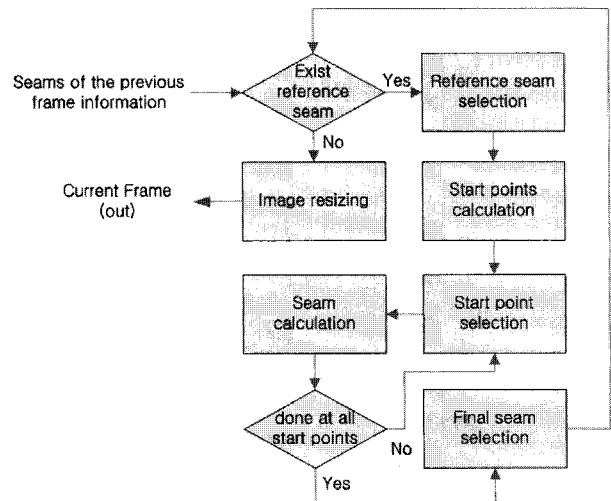


그림 5. “이전 프레임의 seam을 이용하여 현재 프레임의 seam을 결정”의 세부 과정
Fig. 5. The details of “Determining seams of the current frame using seams of the previous frame” process.

레이프 별로 다르기 때문이다. 그림 4는 seam carving에 의해 연속된 프레임의 크기를 독립적으로 확대하여 떨림 현상이 나타난 결과를 보여주고 있다.

빨간 원으로 표시한 문자를 중점적으로 본다면, 첫 번째 프레임은 문자의 왼쪽에 12개의 seam들이 존재하고 두 번째 프레임은 문자의 왼쪽에 11개의 seam들, 오른쪽에 1개의 seam이 존재한다. 원본 동영상에서의 문자는 고정된 위치에 존재하지만, 프레임 별로 독립적으로 수행된 seam carving에 의해 확대된 영상에서의 문자는 왼쪽으로 1픽셀만큼 움직인 것처럼 보인다. 이러한 과정들이 반복된다면 빨간 원으로 표시된 문자는 심

하게 흔들리는 것처럼 보일 것이다.

동영상에서는 최적의 seam을 찾는 것도 중요하지만, 더 중요한 것은 떨림 현상이 없도록 하는 것이다. 2.3. 절에서 전개하는 과정은 떨림 현상을 방지하면서 콘텐츠의 형태를 보존하는 seam을 추출하는 것에 목적이 있다. 그림 5는 이전 프레임의 seam을 이용하여 현재 프레임의 seam을 결정하는 전체과정을 나타낸다.

2.3.1. 참조 seam 선택

Seam carving을 할 때, seam들 간의 좌표가 겹쳐지는 것을 피하기 위해 추출한 seam의 좌표를 제거한 후 다음 seam을 추출한다. 그림 6에서는 seam들 간의 좌표가 겹쳐진 경우를 나타낸다.

2번째 seam과 3번째 seam이 만나게 될 경우 좌표의 중복성이 발생한다. 겹쳐진 부분의 동일한 좌표를 사용하지 않게 된다면, 추출된 seam들에 의해 영상의 크기를 조절할 때에 겹쳐진 부분에서 1픽셀이 부족하게 되고, 결국 영상의 왜곡이 생기게 된다. 이러한 왜곡을 막기 위하여 seam 간에 순서가 존재하고, 상관성에 의하여 이전 프레임에서 추출된 seam들의 순서대로 현재 프레임의 seam들을 추출해야 한다. 식 (4)는 현재 프레임의 seam을 생성하기 위하여 이전 프레임으로부터 참조할 seam정보를 가져오는 것이다.

$$S_{ref} = S_{f(n-1)}(i) \tag{4}$$

n 은 현재 프레임 번호이고, i 는 현재 생성하는 seam의 번호이다. S_{ref} 은 새로운 seam을 생성하기 위하여 이전 프레임에서 참조하는 seam이며, 식 (3)에서 S 와 같은 구조를 가지고 있다.

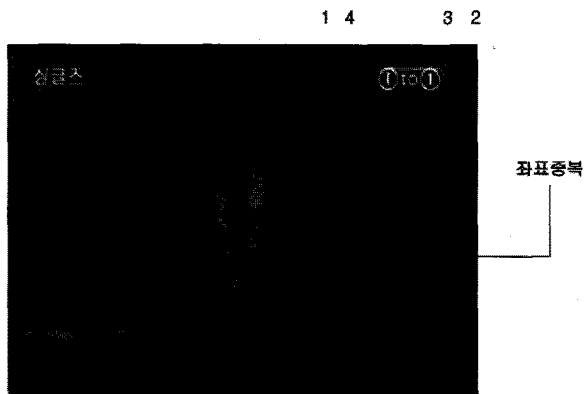


그림 6. Seam의 순서 및 좌표 중복성의 예시
Fig. 6. Examples of the order of seam and the redundant coordinates.

2.3.2. 시작 좌표 계산

추출될 seam의 시작 좌표를 정하는데 있어, 단순히 S_{ref} 의 출발 좌표와 동일한 좌표를 사용하는 것이 아니라 좌우로 연결된 k 개의 좌표들을 모두 고려한다. 연결된 좌표들을 모두 고려하는 이유는 seam의 출발 좌표에 따라 seam의 양상이 달라질 수 있기 때문이다. 식 (5)는 S_{ref} 로부터 시작 좌표를 계산하는 과정이다.

$$C \subset S_{ref} \rightarrow c = C(0),$$

$$SP = [c - k, c - (k - 1), \dots, c + k], \tag{5}$$

시작 좌표의 배열 SP 는 $C(\subset S_{ref})$ 에서의 첫 좌표와 거리 k 이하의 픽셀들이고, 각 SP 의 요소로 시작하는 seam들은 최종 seam이 될 후보들이다. 그림 7은 $k = 1$ 일 때, 각 시작 좌표에 따른 seam들의 예를 보여준다.

빨간색으로 표시된 것이 참조 seam S_{ref} 의 좌표이며, 녹색 박스 영역은 후보 seam 좌표의 예시이다.

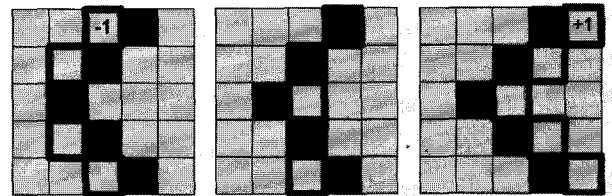


그림 7. 3가지 시작좌표 예시
Fig. 7. Examples of three kinds of start points.

2.3.3. seam 계산

후보 seam을 구하기 위하여 시작 좌표 $sp(\in SP)$ 로부터 픽셀 단위로 탐색한다. 그림 8은 sp 로부터 seam을 구하는 과정을 블록다이어그램으로 나타내고 있다.

후보 seam의 첫 좌표는 sp 이고, 이것은 초기 기준

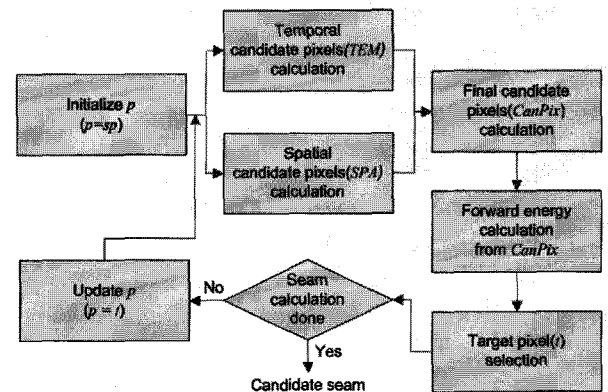


그림 8. seam 계산 과정
Fig. 8. Processing of seam calculation.

좌표 p 가 된다. 기준 좌표 p 와 참조 seam S_{ref} 를 이용하여 p 의 다음 좌표인 t 를 구한다. t 를 구하기 위하여 아래와 같은 조건을 고려한다.

1. p 와 t 는 공간적으로 연결되어 있어야 함 (공간적 연결).
2. t 와 $C(\subset S_{ref})$ 는 시간축으로 연결되어 있어야 함 (시간적 연결).

식 (6)는 위의 조건을 만족하는 후보 픽셀 집합인 $CanPix$ 를 구하는 과정이다.

$$CanPix = SPA \cap TEM,$$

$$SPA = \{spa | p-1 \leq spa \leq p+1\}, \quad (6)$$

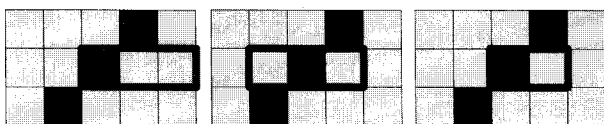
$$TEM = \{tem | C \subset S_{ref},$$

$$C(n+1)-1 \leq tem \leq C(n+1)+1\}$$

n 은 t 의 x (가로 방향의 seam) 또는 y (세로 방향의 seam)좌표이며, SPA 는 공간적 연결을 만족하는 픽셀 집합이고 TEM 은 시간적 연결을 만족하는 픽셀 집합이다. 즉, SPA 는 기준 픽셀 p 와 인접한 픽셀들이며, TEM 은 참조 seam 좌표 $C(\subset S_{ref})$ 와 인접한 픽셀들이다. 그림 9는 공간적, 시간적 연결 조건과 두 조건을 만족하는 후보 픽셀 집합 $CanPix$ 의 예시이다.

빨간색으로 표시된 것이 $C(\subset S_{ref})$ 이며, 녹색으로 표시된 것은 기준 픽셀 p 이다. 최종적으로 그림 9의 (c)의 녹색 박스 영역이 $CanPix$ 이 된다. 그림 9의 (b)와 같이 이전 프레임의 seam 좌표를 고려함으로써 떨림 현상을 방지할 수 있다.

이전 프레임과 현재 프레임간의 energy 분포도가 유사함을 이용하여, 앞서 구한 $CanPix$ 의 energy 값과 $E(C \subset S_{ref})$ 값을 비교하여 가장 유사한 하나의 픽셀이 다음 좌표 t 로 결정한다. 본 논문에서 사용한 energy 공식은 정지 영상의 seam carving에서 사용한 forward energy^[3]이며, 식 (7)을 통하여 t 를 구한다.



(a) 공간적 연결 (b) 시간적 연결 (c) 최종 후보

그림 9. 좌표 후보 예시.

Fig. 9. Example of coordinate candidates.

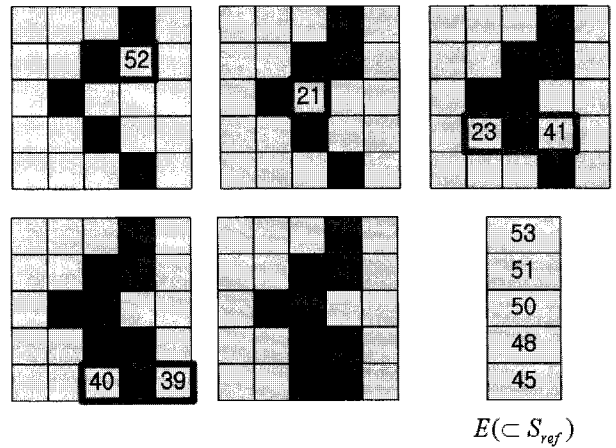


그림 10. Seam 좌표 선택 및 저장 예시

Fig. 10. Example of the selection and storage seam coordinates.

$$e_{min} = \min_{\alpha \in CanPix, E \subset S_{ref}} (|FE(\alpha) - E(n)|),$$

$$t_n = FE^{-1}(e_{min} + E(n)) \quad (7)$$

$FE(\cdot)$ 은 해당좌표로부터 forward energy를 구하는 함수이며, n 은 t 의 x (가로 방향의 seam)또는 y (세로 방향의 seam)좌표이다.

그림 10은 후보 좌표들의 energy 값들과 이전 프레임에서 구한 seam의 좌표별 energy 값을 비교하여 seam을 결정하는 과정을 보여준다.

빨간색으로 표시된 픽셀이 $C(\subset S_{ref})$ 이며, 녹색으로 표시된 픽셀은 확정된 후보 seam의 좌표이다. 그리고 녹색 박스 영역은 각 단계에서의 $CanPix$ 이다. $CanPix$ 의 energy 값과 같은 행에 있는 $E(C \subset S_{ref})$ 값과 비교하여 가장 유사한 픽셀 좌표를 t 로 선택 한다. 이와 같은 과정을 모든 SP 에 대하여 반복하여 후보 seam들을 구한다.

2.3.4. 최종 seam 결정

2.3.3.절에서 계산한 모든 후보 seam들 중의 하나가 최종 seam 좌표가 된다. 이 과정에서 $e_t(C \subset S_{ref})$ 와 각 후보 seam들의 좌표 별 energy 값의 총합들을 비교한다. 2.3.3절에서의 마찬가지로 후보 seam의 energy 총합이 $e_t(C \subset S_{ref})$ 와 가장 비슷한 것을 최종 seam으로 선택한다. 그림 11은 최종 seam을 선택하는 과정을 보여 준다.

빨간색으로 표시된 픽셀이 $C(\subset S_{ref})$ 이며, 녹색은

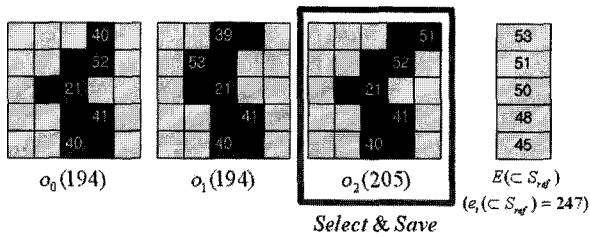


그림 11. 최종 seam 결정 예시
Fig. 11. Example of the final seam determination.

후보 seam을 나타낸다. 최종 결정된 seam은 2.2.결과 동일한 방법으로 정보들을 저장하고, 다음 프레임의 seam을 구하는데 이용한다.

2.3.5. 영상 크기 조절

현재 프레임에서 최종적으로 결정된 모든 seam의 좌표를 이용하여 영상의 크기를 조절한다. 영상의 크기를 줄일 때는 seam의 순서대로 한번에 하나씩 원하는 만

큼의 seam을 삭제한다. 반면에 영상의 크기를 늘릴 경우에는 seam의 순서대로 seam의 좌표들에 픽셀 값들을 넣어준다. 이 때, 픽셀 값들을 생성하기 위해 기존의 영상 보간 기법들을 사용한다.

그림 12는 영상의 크기를 조절하는 과정의 예시를 보여주고 있다.

먼저 그림 12 (a)와 같이 저장된 seam 정보를 이용하여 seam map을 생성한다. Seam map의 크기는 원본 영상의 크기와 동일하고, seam의 좌표에 해당 seam 번호가 저장된다. 이렇게 생성한 seam map을 이용하여 영상의 크기를 조절한다. 그림 12 (b)와 같이 영상의 크기를 축소할 때는 seam map을 탐색하여 첫번째 seam의 좌표들과 동일한 위치의 픽셀들을 제거한다. 제거하기를 원하는 seam의 개수만큼 이러한 과정을 반복함으로써 영상의 크기를 축소한다. 원본영상의 크기를 하나의 seam만큼 축소 후, 좌표의 동기화를 맞추기 위하여 seam map에서도 참조한 seam을 제거한다. 반면에 (c)와 같이 영상의 크기를 확대할 때에는 seam의 좌표들과 동일한 영상의 좌표들에 빈 공간을 삽입한다. 그리고 빈 공간에 보간법에 생성된 픽셀 값을 채워 넣음으로써 크기를 확대한다. 영상의 크기를 축소할 때처럼, 하나의 seam만큼 영상을 확대한 뒤에는 좌표의 동기화를 맞추기 위하여 seam map에서도 참조한 seam위치에 빈 공간을 삽입한다. 이러한 과정을 반복함으로써 최종 영상을 구하게 된다.

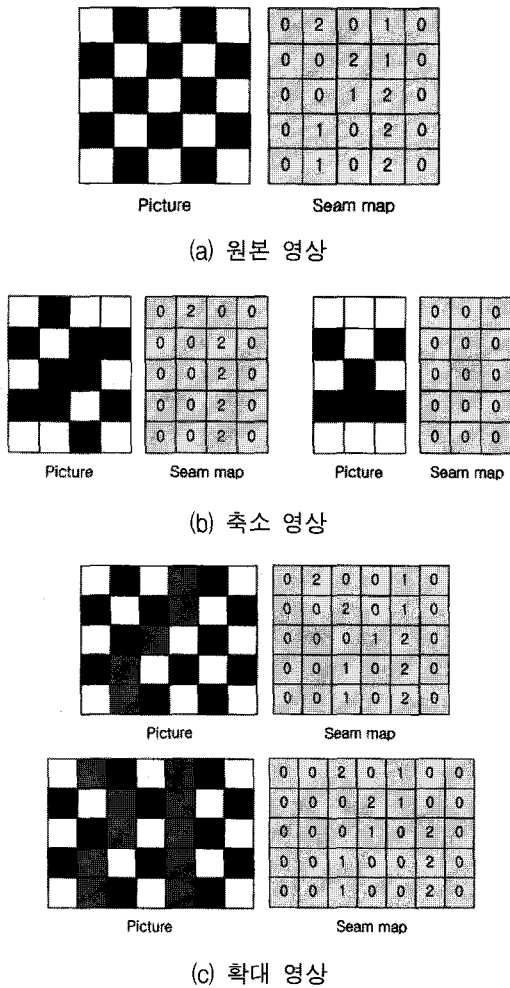


그림 12. 영상 크기 변환 과정 예시
Fig. 12. Examples of image resizing process.

III. 실험

본 절에서는 bilinear 방법과 Avidan의 기법[3]을 동영상에 적용한 방법, 그리고 제안하는 방법의 성능을 평가한다. 실험 영상으로는 320×240의 동영상을 사용하였고, 가로방향으로 10%와 30%를 확대하였다. 먼저 실시간 처리에서 가장 중요한 요소인 수행 시간과 사용되는 평균 메모리의 사용량을 통하여 평가하였다. 실험에 사용된 컴퓨터의 사양은 아래와 같다.

- CPU : Intel Core2 6320 1.86GHz
- RAM : 2 GB
- OS : Windows XP Professional Service Pack 2

측정치에 신뢰도를 높이기 위해서 동일한 과정을 10 번 반복하고, 결과 값들의 평균값을 비교하였다. 표 1은

표 1. 기존 알고리즘과 제안하는 알고리즘의 실행 시간 비교

Table 1. Comparing run-time of the proposed algorithm with those of other algorithms.

	352×240(10%)	416×240(30%)
Bilinear algorithm	6"234	6"875
Avidan's algorithm	1'31"890	4'12"218
Proposed	8"296	17"437

표 2. 기존 알고리즘과 제안하는 알고리즘의 메모리 사용량 비교

Table 2. Comparing memory usage of the propose algorithm with those of other algorithms.

	352×240(10%)	416×240(30%)
Bilinear algorithm	472.5 KB	517.5 KB
Avidan's algorithm	1973.4 KB	2018.4 KB
Proposed	565.2 KB	790.4 KB

각 알고리즘들의 실행 시간을 측정 한 결과를 보여준다.

결과에서 보듯이 제안한 방법이 단순한 bilinear 방법 보다는 느리지만, 정지영상의 seam carving 한 결과보다 월등히 빠른 처리 속도를 가지고 있음을 알 수 있다.

표 2는 각 알고리즘들의 평균 메모리 사용량을 측정 한 것이다.

제안한 방법에서는 이전 프레임의 seam의 정보들을 참조하여 현재 프레임의 새로운 seam을 연산하기 때문에 bilinear 방식과 비슷한 정도의 메모리만 사용하여 영상의 크기를 조절함을 알 수 있다.

둘째로, 주요 콘텐츠 유지 여부와 떨림 현상(jitter)에

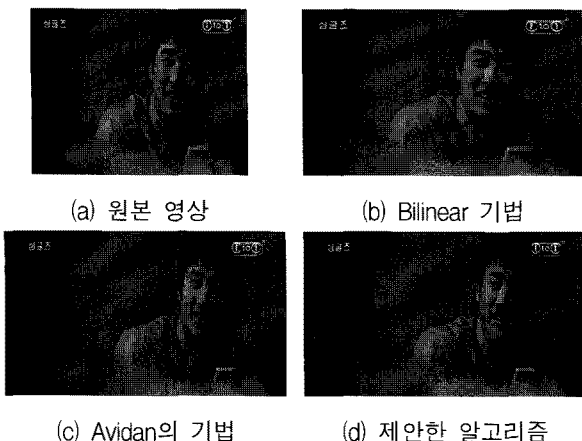


그림 13. 가로 방향으로 30% 확대 영상
Fig. 13. Results expanded to the cross direction as 30%.

대해 각각의 결과 영상을 통하여 비교하였다. 그림 13은 각 알고리즘들의 결과 영상에서 6번째 프레임들을 보여준다.

그림 13 (a)의 원본 영상과 비교해보면, 그림 13 (b)의 bilinear 기법의 결과에서는 사람 얼굴의 형태가 가로로 늘어난 것을 확인할 수 있다. 하지만, Avidan의 기법과 제안한 알고리즘의 결과 영상에서는 얼굴의 형태가 원본영상이 가지고 있는 얼굴의 형태를 유지하고 있다. 이것으로 제안하는 알고리즘이 영상의 주요 콘텐츠를 유지하는 것을 알 수 있다.

마지막으로 콘텐츠 보존에 대한 실험 결과와 원본 영상의 차이를 에러율을 통하여 나타내고 있다. 식 (8)은 본 논문에서 사용한 에러율에 대한 공식이다.

$$D_n = \sum_{j=0}^{height-1} \sum_{i=0}^{width-1} |f_n(i,j) - f_{n+1}(i,j)|,$$

$$\xi = c \sum_{k=1}^K D_k, \quad Error\ Rate = |1 - \frac{\xi}{\xi_0}| \times 100 \quad (8)$$

f_n 은 n 번째 프레임의 R, G, B 를 나타내며, D_n 은 n 번째와 $n+1$ 번째 프레임 간의 픽셀당 에러를 나타낸다. K 는 (총 프레임 수 - 1)값을 가지며, 동영상에서 얻을 수 있는 차영상의 개수와 동일하다. 그리고 ξ 은 결과 동영상에서 프레임 간의 에러이며, c 는 정규화 상수로서 영상 크기의 역수값을 갖는다. ξ_0 은 원본 동영상에서 프레임 간의 에러이고, $Error\ Rate$ 는 원본 동영상과 결과 동영상 간의 에러율이다.

표 3은 식 (8)을 이용하여 Avidan방법과 제안하는 방법이 원본 영상의 콘텐츠가 나타나는 양상과 얼마나 차이가 나타나는지 수치적으로 보여준다.

결과에서 확인할 수 있듯이 제안하는 기법이 훨씬 작은 에러율로서 원본 영상과 유사함을 나타내고 있다.

그림 14는 콘텐츠의 양상을 차영상을 통하여 보여주고 있다. 그림 14 (a)는 원본영상의 이웃프레임 간의 차영상이고, 그림 14 (b)와 (c)는 Avidan의 기법과 제안하는 방법의 이웃프레임 간의 차영상이다.

표 3. 원본 영상과의 에러율
Table 3. Error rate between the original image.

	352×240(10%)	416×240(30%)
Avidan's algorithm	76.5%	65.03%
Proposed	3.78%	5.57%

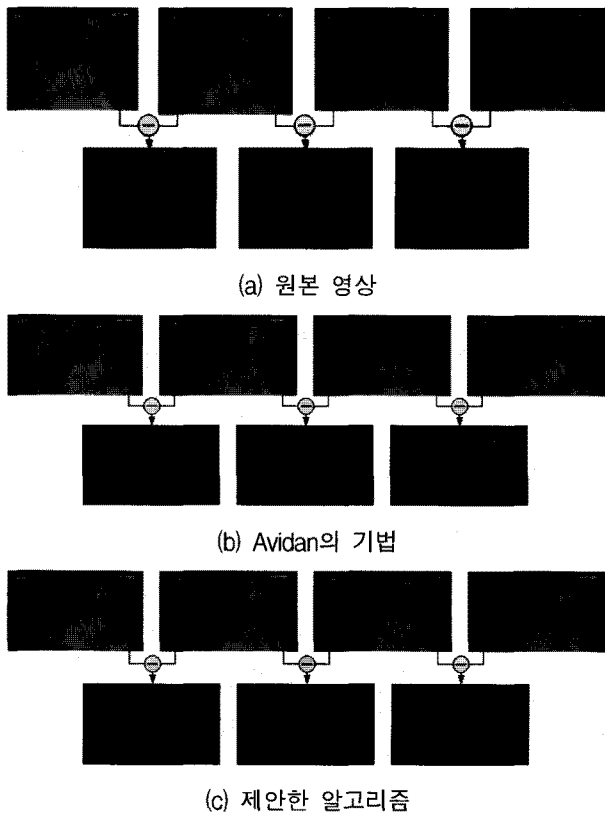


그림 14. 각 알고리즘의 결과 영상과 인접 프레임 간의 차 영상

Fig. 14. The difference image between result image of each algorithm and the adjacent frame.

그림 14 (b)에서 볼 수 있는 것처럼 Avidan의 기법을 동영상에 적용한 방식은 인접한 프레임간의 상관관계를 고려하지 않기 때문에 떨림 현상이 존재하고, 떨림 현상으로 인하여 이웃한 프레임간의 차이가 크게 발생한다. 반면에 제안한 알고리즘은 인접 프레임간의 상관관계를 고려하기 때문에 떨림 현상이 없으며, 이웃한 프레임 간의 차이가 원본 영상에서와 유사하게 나타나는 것을 그림 14 (c)에서 알 수 있다.

IV. 결 론

본 논문에서는 영상이 가지고 있는 콘텐츠를 보존하면서 영상의 크기를 조절할 수 있는 실시간 동영상 크기 변환 기법을 제안하였다. 동영상에서 연속하는 두 프레임 사이의 상관성을 존재한다. 제안하는 방법에서는 인접 프레임의 상관성을 이용하여 이전 프레임 seam의 정보를 활용하여 현재 프레임 seam을 결정함으로써, 전체 영상을 분석하지 않으면서 콘텐츠의 떨림 현상을 없앨 수 있었다. 개발된 알고리즘은 영상의 주

요 콘텐츠를 최대한 보존하면서도 기존의 Avidan의 기법에 비하여 빠른 처리 속도를 가졌으며, 메모리 사용량도 기존의 seam carving 방법에 비해 현저히 작기 때문에 메모리 제약이 많은 단말에서도 사용이 가능하다. 실험 결과에서 보였듯이 제안하는 기법이 실시간 처리와 떨림 현상 제거, 콘텐츠 보존에 대해 객관적인 성능과 주관적 화질 비교에서 기존 기법들보다 우수함을 확인할 수 있다.

참 고 문 헌

- [1] A. Santella and M. Agrawala and D. Decarlo and D. Salesin and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping," ACM Human Factors in Computing Systems, pp. 771-780, 2006.
- [2] F. Liu and M. Gleicher, "Automatic Image retargeting with fisheye-view warping," ACM Multimedia, pp. 153-164, Oct. 2005.
- [3] S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," ACM Trans. on Graphics, Vol. 26, Issue 3, Jul. 2007.
- [4] V. Setlur and S. Takagi and R. Raskar and M. Gleicher and B. Gooch, "Automatic Image retargeting," Proc. ACM, Vol. 154, pp. 59-68, 2005.
- [5] C. Tao and J. Jia and H. Sun, "Active window oriented dynamic video retargeting," ICCV Proc. Workshop on Dynamical Vision, 2007.
- [6] S. Cho and Y. Matsushita and S. Lee, "Image Retargeting with Importance Diffusion," KIISE, vol. 35, pp. 236-239, Jun. 2008.
- [7] V. Setlur and T. Lechner and M. Nienhaus and B. Gooch, "Retargeting Images and Video for Preserving Information Saliency," IEEE Computer Graphics and Applications, vol. 27, no. 5, pp. 80-88, Oct. 1998.
- [8] F. Liu and M. Gleicher, "Video Retargeting: Automating Pan and Scan," ACM Multimedia, pp. 241-250, Oct. 2006.
- [9] M. Rubinstein and A. Shamir and S. Avidan, "Improved seam carving for video retargeting," ACM Transactions on Graphics, Vol. 27, Issue 3, Aug. 2008.
- [10] B. Chen and P. Sem, "Video Carving," Proc. Eurographics, Short Papers, Apr. 2008.
- [11] U. Gargi and R. Kasturi, Sh. Strayer, "Performance Characterization of Video-Shot-Change Detection Methods," IEEE Trans. On

Circuits and Systems for Video Technology, vol. 10, no. 1, pp. 1-13, 2000.

[12] Y. Gong, "An accurate and robust method for detecting video shot boundaries," IEEE Multimedia Computing and Systems, vol. 1, pp. 850-854, 1999.

[13] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang, and H. Zhou, "A visual attention model for adapting images on small displays," Multimedia Systems 9, 4, pp. 353 - 364, 2003.

[14] H. Liu, X. Xie, W. Ma, and H. Zhang "Automatic browsing of large pictures on mobile devices," Proceedings of the eleventh ACM international conference on Multimedia, pp. 148 - 155, 2003.

[15] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, "Automatic thumbnail cropping and its effectiveness," In UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology, ACM Press, New York, NY, USA, pp. 95 - 104, 2003.

[16] X. Fan and X. Xie and H. Q. Zhou and W. Y. Ma, "Looking into video frames on small displays," ACM on Multimedia, pp. 247-250, 2003.

[17] Richard Bellman, "Some Problems in The Theory of Dynamic Programming," Econometrica, Vol. 22, No. 1, pp. 37-48, 1954.

[18] D. P. Bertsekas, "Dynamic Programming and Optimal Control," Athena Scientific, 2000.

[19] P. Kohli and P. H. S. Torr, "Dynamic graph cuts for efficient inference in markov random fields," IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 29, 12, pp.2079 - 2088, 2007.

[20] V. Kwatra, A. Schödl, I. Essa, G. Turk and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," ACM Trans. Graph. 22, 3, pp. 277 - 286, 2003.

저 자 소 개



이 강 희(학생회원)
2009년 강원대학교
컴퓨터정보통신공학과
학사 졸업.
2009년~현재 강원대학교
컴퓨터정보통신공학과
석사 과정.

<주관심분야 : 영상 신호처리, 컴퓨터 비전>



유 재 욱(학생회원)
2009년 강원대학교
컴퓨터정보통신공학과
학사 졸업.
2009년~현재 강원대학교
컴퓨터정보통신공학과
석사 과정.

<주관심분야 : 영상 신호처리, 코덱>



박 대 현(학생회원)
2007년 강원대학교
컴퓨터정보통신공학과
학사 졸업.
2009년 강원대학교
컴퓨터정보통신공학과
석사 졸업.

2009년~현재 강원대학교 컴퓨터정보통신공학과
박사 과정.

<주관심분야 : 영상 신호처리, 영상 압축, 멀티미
디어통신>



김 윤(정회원)-교신저자
1993년 고려대학교 전자공학과
학사 졸업.
1995년 고려대학교 전자공학과
석사 졸업.
2003년 고려대학교 전자공학과
박사 졸업.

1995년~1999년 LG-Philips LCD 연구소
주임연구원

2002년~2003년 (주)달리텍 선임연구원

2004년~현재 강원대학교 컴퓨터정보통신공학과
부교수.

<주관심분야 : 영상 신호처리, 멀티미디어 통신
및 신호처리>