

논문 2006-43IE-4-7

# 가지 분류 및 제거기법을 이용한 고속 RSST 알고리즘

## (Fast RSST Algorithm Using Link Classification and Elimination Technique)

홍 원 학\*

( Won Hak Hong )

### 요 약

RSST를 이용한 분할법은 정확한 영역 경계를 추출과 분할결과의 해상도를 조절 등의 많은 장점을 가지고 있다. 본 논문에서는 영상분할을 위한 세 가지 고속 RSST 알고리즘을 제안한다. 첫 번째 방법에서는 고속 가지검색을 위해 가중치의 크기에 따라 가지들을 분류한다. 두 번째 방법은 RSST 구성 전에 매우 유사한 가지들을 제거된다. 세 번째 방법에서는 시각적으로 중요하지 않은 소영역의 가지들을 제거된다. 제안된 알고리즘들을 영상분할에 적용한 결과 기존의 RSST와 비교하여 PSNR과 화질의 저하가 거의 없이 RSST 수행시간을 10 ~ 40배 정도 줄일 수 있었다.

### Abstract

Segmentation method using RSST has many advantages in extracting of accurate region boundaries and controlling the resolution of segmented result and so on. In this paper, we propose three fast RSST algorithms for image segmentation. In first method, we classify links according to weight size for fast link search. In the second method, very similar links before RSST construction are eliminated. In third method, the links of very small regions which are not important for human eye are eliminated. As a result, the total times elapsed for segmentation are reduced by about 10 ~ 40 times, and reconstructed images based on the segmentation results show little degradation of PSNR and visual quality.

**Keywords :** RSST, segmentation, RAG, link, classification

## I. 서 론

영상분할이란 주어진 영상을 동질성(homogeneity)과 연결성(connectivity)을 만족하는 부영역(subregion)들로 나누는 것으로, 영상부호화, 영상처리, 컴퓨터 비전, 패턴인식 등 여러 분야에 많이 응용되고 있다.<sup>[1-3]</sup> Morris 등은 그래프 이론을 영상분할에 적용한 RSST (recursive shortest spanning tree) 분할법을 제안하였다.<sup>[4]</sup> 이 방법은 분할결과의 정확성과 분할영역의 수를 쉽게 조절할 수 있다는 장점으로 인해 분할기반 정지 영상 부호화, 분할기반 동영상 부호화, 움직임 벡터 기반 분할기법 등 다양한 응용분야에 적용되어졌다.<sup>[5-6]</sup>

또한, 최근에는 동영상의 객체 추출, 객체 분할 및 객체 추적에도 RSST가 많이 사용되어지고 있다.<sup>[7-9]</sup> RSST 구성순서는 먼저 영상의 각 화소를 정점(vertex), 인접화소쌍을 가지(link), 화소간의 동질성을 가지의 가중치(weight)로 표현하는 RAG(region adjacency graph)를 만든다. 다음 RAG의 가지들 중 가장 작은 가중치를 가진 가지를 탐색하여 제거한 후, 그 가지와 연결된 중복가지 제거 및 인접가지들의 가중치 갱신과정을 반복적으로 수행함으로써 최종 ST (spanning tree)를 만든다. 다양한 고속화알고리즘이 개발되어 있는 SST(shortest spanning tree)와는 달리 RSST는 반복적 sequential 구조를 가지고 있어 고속화 및 병렬화가 어려운 것으로 알려져 있다.

Kwok은 가지의 가중치를 정수단위로 분류한 후 같은 클래스에 속하는 가지들의 최소 가중치 가지 탐색

\* 정회원, 울산과학대학 전기전자통신학부  
(Dept. of Electrical Electronics, Ulsan College)  
접수일자: 2006년8월23일, 수정완료일: 2006년11월30일

과정 제거 및 가중치 갱신과정을 변경한 고속 RSST 알고리즘을 제안하였다.<sup>[10]</sup> 이 방법은 RSST 계산량이 크게 줄어드는 장점이 있으나 분할의 정확도가 떨어지는 단점이 있다.

초기 RSST 구성단계에서 가지들을 가중치의 크기로 분류해 보면 대부분의 영상들은 0에서 10사이에 집중적으로 분포되어 있음을 알 수 있다. 따라서 Kwok의 방법처럼 단순히 가중치를 정수단위 클래스로 분류한다면 0과 10사이에 속하는 많은 가지들이 실제 자신이 가지고 있는 가중치의 크기 순서대로 제거되지 않아 결국 전체 분할결과의 성능을 떨어뜨릴 수 있다. 따라서 만약 이들 가지들을 가중치 크기의 분포가 밀집되어 있는 구간은 더욱 세밀하게 분류한다면 분할 오류를 최소화하면서 RSST 수행시간을 크게 줄일 수 있을 것이다. 또한 가지집합을 가중치의 분포에 맞게 적절하게 클래스로 분류한다면, 각 클래스에 속하는 가지들의 개수가 크지 않기 때문에 최소 가중치 탐색을 한다고 하더라도 전체 계산량은 크게 증가하지 않으면서 원 RSST와 동일한 분할결과를 얻을 수 있을 것이다.

본 논문에서는 가지 분류 및 초기 가지제거를 이용한 세 가지의 RSST 고속화 알고리즘을 제안한다. 첫째는 가중치 크기의 분포를 고려한 가지 분류를 통한 최소가중치 가지탐색의 고속화이다. 두 번째는 영상 내에 존재하는 인접화소 간에는 유사성이 크다는 사실을 이용하여 동질성이 매우 큰 가지들은 RSST 구성 시에 미리 제거하여 초기 가지집합의 크기를 줄이는 방법이다. RSST 수행시간은 전체 가지들의 수에 비례하기 때문에 이 방법은 전체 수행시간을 크게 줄일 수 있다. 마지막으로 많은 영상에 존재하는 배경의 국부입상(local granularity)과 에지부근에 존재하는 수많은 소영역들을 인접영역에 미리 병합하여 그 영역에 해당하는 가지들을 제거하여 초기 가지들의 수를 줄이는 방법이다.

제안된 고속화 알고리즘의 성능을 평가하기 위해 세 가지 고속화 알고리즘과 Kwok의 알고리즘을 수행시간 및 최종 분할결과 측면에서 비교하였다.

II. RSST를 이용한 영상분할

Morris은 SSF(Shortest Spanning Forest)나 SST를 이용한 분할 알고리즘의 문제점을 개선하기 위해 RSST를 이용한 분할 방법을 제안하였다. 이 알고리즘

은 근본적으로 그래프 이론에 기반을 두고 있는데 영상분할에 사용되는 기본적인 그래프이론을 간단히 설명하면 다음과 같다.<sup>[4,11]</sup>

임의의 그래프  $G = (V, L)$ 는 유한개의 정점들의 집합  $V$ 와 정점쌍들을 연결하는 가지들의 집합  $L$ 로 구성된다. 연결그래프란 임의의 정점에서 다른 모든 정점들 사이에 경로가 존재하는 그래프를 말하며, 루프가 없는 연결그래프를 트리(tree)로 정의한다.

어떤 연결그래프  $G$ 에 대해, 이 그래프의 모든 정점들을 포함하는 부그래프  $G' = (V', L')$  (단  $L' \subseteq L$ )가 트리를 형성한다면 이를 그래프  $G$ 의 ST라 한다. 한 그래프에 대해 여러 개의 ST가 존재할 수 있는데, 그래프의 여러 ST들 중 모든 가지에 부여된 가중치의 합이 가장 작은 ST를 그 그래프의 SST라 한다.<sup>[12]</sup>

영상분할에 그래프이론을 이용하기 위해서는 우선 영상을 그래프로 적절하게 매핑하는 일이 필요한데, 그림 1은 주어진 4 x 4 크기의 영상을 일반적으로 많이 사용되는 4방향 연결그래프로 매핑하는 과정을 보여준다. 먼저 (a)는 4 x 4 크기의 영상을 보여주고 (b)는 이 영상의 각 화소  $p_i$ 에 대응하는 그래프의 정점  $v_i$ 를 보여준다. (c)는 (b)에서 상하좌우로 인접한 각각의 정점쌍  $(v_i, v_j)$ 를 가지  $l_{ij}$ 로 연결하여 구성된 4방향 연결그래프를 보여준다. 이 그림에서 인접한 각 화소쌍의 이질성 정도 즉 밝기값차의 절대치를 그 화소들을 매핑한 정점들을 연결하는 가지의 가중치  $W(l_{ij})$ 로 부여

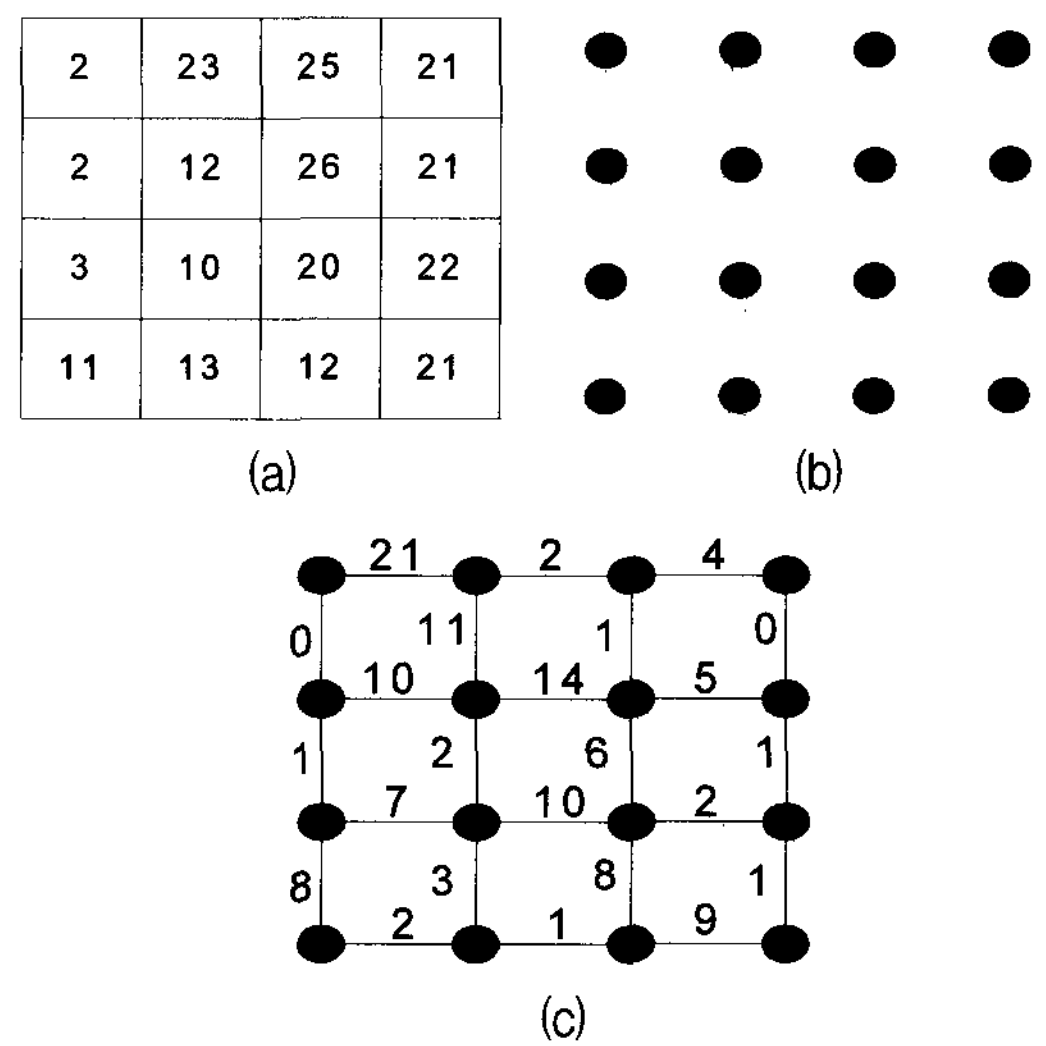


그림 1. 영상을 4방향 연결그래프로의 매핑: (a) 원 영상 (b) 정점들 (c) 4방향 그래프  
 Fig. 1. Mapping an image onto a 4-neighbor connected graph.: (a) original image (b) vertexes (c) 4-connected graph.

하면 가중치를 가진 4방향 연결그래프가 구성된다. 가지의 가중치  $W(l_{ij})$ 는 수식으로 표현하면 (1)식으로 표현된다.

$$W(e_{ij}) = |I(p_i) - I(p_j)| \quad (1)$$

여기서  $I(p_i)$ 는 화소  $p_i$ 의 밝기값을 나타낸다. 그림 1과 같이 영상을 그래프로 매핑하면 모든 가지들의 가중치는 그 가지가 연결하고 있는 인접한 정점들의 유사도를 나타내기 때문에 이를 기준으로 영상을 분할한다. 기존의 그래프를 이용한 영상분할에서는 SST나 RSST를 주로 사용하였다.

SST를 이용한 영상분할의 경우 분할기준으로 영상의 국부적인 정보인 화소간의 밝기값만을 사용하기 때문에 오윤곽(false contour)나 소영역이 발생한다. 이를 개선한 RSST를 이용한 영상분할은 분할초기에는 화소간의 밝기차를 사용하지만 분할이 진행됨에 따라 전체적인 정보인 영역의 평균밝기값을 사용하기 때문에 정확한 윤곽을 얻을 수 있고 소영역 발생도 줄어든다.

$N \times N$  크기의 영상을 매핑한 그래프에서 RSST를 구하는 과정은 다음과 같다.

- 단계 1: 그래프의 정점들로 초기 임시그래프를 구성한다.
- 단계 2: 임시그래프의 가지 갯수가  $N^2-1$  보다 작을 때까지 단계 3-8을 반복한다.
- 단계 3: 그래프에서 가중치가 가장 작은 가지를 선택한다.
- 단계 4: 그 가지를 그래프에서 제거한다.
- 단계 5: 그 가지를 임시그래프의 가지로 선택한다.
- 단계 6: 그 가지가 연결하고 있던 두 정점을 병합한다.
- 단계 7: 병합된 정점과 그 정점에 연결된 주위 가지들의 새로운 가중치를 갱신한다.
- 단계 8: 병합된 정점과 인접하는 정점들과의 중복되는 가지를 가지들의 집합에서 제거한다.

위 과정에서 인접한 두 영역  $X, Y$ 를 병합한 영역  $Z$ 의 평균밝기값  $I_z$ 는 다음 식과 같이 구할 수 있다.

$$I_z = \frac{\sum_{i \in X} I(p_i) + \sum_{j \in Y} I(p_j)}{N(X) + N(Y)} \quad (2)$$

여기서  $N(X)$ 와  $N(Y)$ 는 두 영역  $X, Y$ 에 속한 화소수이다. 위 알고리즘을 통해 가지 수가  $N^2-1$ 인

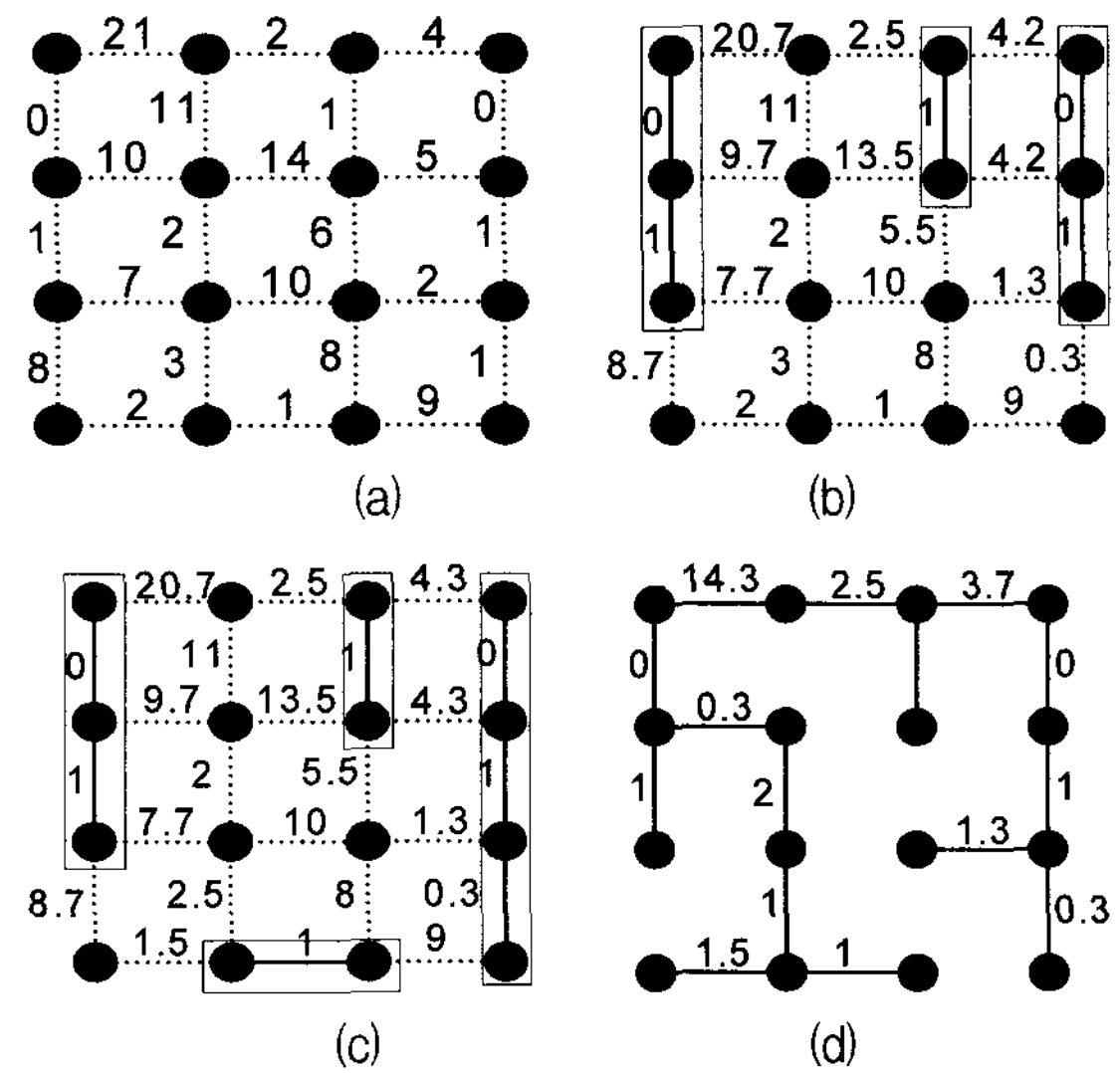


그림 2. 초기 가중치 그래프에서 RSST를 구성하는 과정 (a) 초기 가중치 그래프 (b) 갱신된 그래프 (c) 갱신된 그래프 (d) 최종 그래프

Fig. 2. Procedure to construct RSST from an initial weighted graph: (a) initial weighted graph; (b) an updated graph; (c) another updated graph; (d) final graph.

RSST가 얻어진다.

그림 2는 초기 가중치 그래프로부터 RSST를 구하는 과정을 보여준다.

그림 2(a)는 초기 가중치 그래프를 나타낸다. 이 그림에서 점선으로 나타낸 부분은 초기 그래프의 가지이며 숫자는 각 가지들의 가중치를 나타낸다. 정점들은 가지가 하나도 없는 초기 임시그래프를 나타낸다. 그림 2(b)와 (c)는 RSST를 구하는 과정에서 병합된 영역의 평균밝기값을 사용하여 가지들의 가중치가 계속 갱신되고 있는 그래프를 보여준다. 굵은 선으로 표시된 가지는 RSST의 가지로 선택된 가지를 나타내며, 가는 선으로 묶인 정점들은 그 정점의 화소들이 한 영역으로 병합된 상태를 나타낸다. 그림 2(d)는 최종 완성된 RSST이다.

RSST가 구해지면 이로부터 원하는 분할영상을 얻는 방법은 다음과 같다. RSST에서 가장 가중치가 큰 가지를 제거하면 2개의 부트리(sub-tree)가 생기는데, 각각의 부트리는 제거된 가지보다 작은 가중치의 가지들로 구성된다. 각각의 부트리를 영역으로 간주할 경우는 하나의 영역에서 평균밝기값 차가 가장 큰 두 영역으로 분할된 것을 의미한다. 따라서  $N$  개의 영역을 얻기 위해서는 RSST에서 가중치가 큰 순으로  $N-1$  개의 가지를 제거하면 된다.

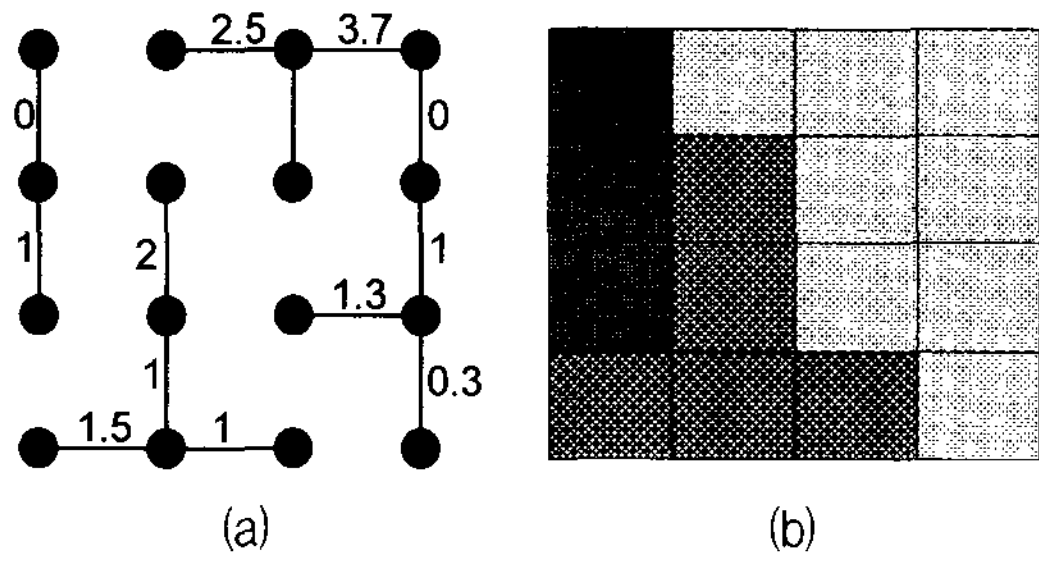


그림 3. RSST로부터 분할된 영상을 구하는 과정 (a) RSST에서 구해진 부트리 (b) 분할 영상

Fig. 3. Procedure to get a segmented image from RSST: (a) sub-tree obtained by RSST; (b) a segmented image.

그림 3은 RSST에서 영상을 분할하는 과정을 보여 준다. 이 그림에서 (a)는 그림 2에서 구해진 RSST에서 가장 가중치가 큰 2개의 가지를 제거하여 얻은 3개의 부트리이며, (b)는 (a)를 영상으로 매핑하여 3개의 영역으로 분할된 영상이다.

위 그림에서 알 수 있듯이 RSST를 이용한 분할 알고리즘에서는 병합이 진행될수록 초기의 국부적인 정보의 영향보다 병합된 영역의 평균밝기값으로 나타내어지는 좀 더 광범위한 영상 정보의 영향이 커지게 된다. 그러므로 오윤곽, 소영역 발생을 줄일 수 있다. 그러나 매번의 영역병합과정 수행 후 새로운 영역 정보를 사용하여 가지들의 값들을 갱신해주어야 하기 때문에 최소값을 갖는 가지를 찾는 과정에서 많은 수행시간을 필요로 하는 단점이 있다.

### III. 제안된 고속 RSST 분할법

본 논문에서는 RSST를 이용한 분할법을 고속으로 수행하기 위해 가중치 크기의 분포에 따른 가지 분류 및 유사가지 및 소영역 가지 제거를 이용한 세 가지의 RSST 고속화 알고리즘을 제안하였다. 표 1은 제안된 고속 RSST 방법들을 보여준다.

먼저 첫 번째 방법인 LC-RSST는 가지의 가중치 크기를 기준으로 가지 분류를 통한 최소가중치 가지탐색의 고속화이다. 이 방법은 다시 두 가지로 나눌 수 있다. 먼저 LCS-RSST는 최소 가중치 가지 검색 시 전체 가지들이 아닌 한 개의 클래스에 속한 가지들만을 검색함으로써 수행시간을 줄이는 방법이다. 다음 LCNS-RSST는 동일한 클래스에 속한 가지들은 가중치가 같다고 가정하여 검색과정 없이 최소 가중치를 구하는 방법이다.

표 1. 제안된 고속 RSST 알고리즘들  
Table 1. Proposed fast RSST algorithms.

Proposed Fast RSST algorithms	
1) LC(link classification)-RSST	- LCS(link classification with search)-RSST
	- LCNS(link classification with no search)-RSST
2) ESL(elimination of similar link)-RSST	
3) ESR(elimination of small region)-RSST	

영상분할을 위한 RSST를 구성하는데 필요한 총 수행시간  $T_t$ 는 다음과 같이 세 가지 수행시간의 합으로 표현할 수 있다.

$$T_t = T_s + T_u + T_r \tag{3}$$

여기서  $T_s$ 는 그래프에 남아 있는 전체 가지 중에서 최소가중치를 갖는 가지검색 시간이며,  $T_u$ 는 매 선택된 가지제거 후 그 가지에 속하는 정점과 연결된 가지들의 가중치 갱신 시간,  $T_r$ 는 영상을 그래프로 매핑하거나 중복되는 가지를 제거하는 일 등의 기타 수행 시간이다. 기존의 RSST를 이용한 분할 알고리즘에서는  $T_s$ 가 총 수행시간의 대부분을 차지한다. 이 시간을 줄이기 위해 가지들을 가중치의 크기를 기준으로 분포에 따른 적절한 개수의 클래스로 분류하면, 매 검색 때마다 하나의 클래스에 속한 가지들만을 검색하여 최소가중치를 갖는 가지를 빠르게 찾을 수 있다.

제안된 방법의 수행시간 면에서의 효과를 알아보기 위해 클래스를 이용하지 않은 기존의 검색방법과 클래스를 이용한 제안된 고속 검색방법에서 필요한 계산량을 구해보면 다음과 같다.  $N \times N$  크기의 영상의 경우 매번 모든 가지들을 검색해야한다면 RSST를 완성하기 위해 필요한 가지들의 가중치의 비교계산량  $C$ 는 다음 식과 같다.

$$C = \sum_{i=1}^{N^2-1} (2N(N-1) - i) = \frac{3}{2}N(N^2-1) \cdot (N - \frac{4}{3}) \tag{4}$$

RSST에 필요한 가지 수가  $N^2-1$ 개이므로  $N^2-1$ 번의 검색과정이 필요하고, 검색해야 할 그래프의 총 가지 수는  $2N(N-1)$ 개인데 매번 반복 시 하나의 가지가 제거되므로 (4)식으로 표현할 수 있다. 즉 비교 계산량  $C$ 는  $O(N^4)$ 이다.

제안된 LCS-RSST에서는 클래스에 속한 가지 수에 따라 비교횟수가 달라지므로, 비교 계산량을 정확히 구

하기 위해 우선 각 클래스에 속한 가지 수의 분포를 알아야 한다. 그러나 이 분포는 영상에 따라 달라지므로 정확하게 모델링을 할 수 없다. 따라서 본 논문에서는 가장 최적의 조건 즉 각 클래스에 속하는 가지들의 개수 분포가 균일하다고 가정하고 계산 양을 구하였다. 이 경우 검색해야 할 가지의 수는 클래스 내부에 속한 가지의 개수가 된다. 또한 클래스의 갯수  $K$ 를  $N$ 에 비례하는 값으로 가정하면 (4)식은 다음 식으로 표현할 수 있다.

$$C = \sum_{i=1}^{N^2-1} \frac{(2N(N-1)-i)}{K} = \frac{3}{2\alpha}(N^2-1) \cdot (N-\frac{4}{3}) \quad (5)$$

여기서  $\alpha$ 는 0과 1사이의 상수이다. 따라서 비교계산량  $C$ 는  $O(N^3)$ 로 줄어든다. 다음 LCNS-RSST를 적용할 경우에는 클래스 내에서의 가지들의 가중치 비교를 하지 않는다. 따라서  $N^2-1$ 번의 검색과정시의 걸리는 전체 계산량은 다음 식으로 표현된다.

$$C = \sum_{i=1}^{N^2-1} 1 = N^2-1 \quad (6)$$

이 경우 전체 계산량은  $O(N^2)$ 이 된다. 따라서 제안된 방법은 기존의 방법에 비해 RSST를 구성하는데 걸리는 시간을 대폭 감소시킬 수 있다.

본 논문에서는 전체 가지집합을 클래스로 분류하기 위해 각각의 클래스에 속한 가지들을 양방향 링크드리스트(doubly linked list)를 사용하여 연결하는 방법을 사용하였다. RSST를 구성하는 과정에서는 가지들의 가중치가 계속 변하므로 그 때마다 가지들을 변화된 클래스로 옮겨주어야 하는데, 리스트에 새로운 가지가 포함될 때는 리스트의 제일 끝부분에 연결시키고 최소 값을 갖는 가지를 찾을 때는 리스트 제일 앞쪽의 가지를 선택하는 FIFO(first in first out)방식을 사용하여 간단히 구현하였다.

제안된 두 번째 고속화 방법인 ESL-RSST는 대부분의 영상들은 인접화소 간에는 유사성이 크다는 사실을 이용하여 가중치가 아주 작은 가지들은 RSST 구성 초기에 미리 제거하여 초기 가지들의 수를 줄여 고속화하는 방법이다. 표 2에서는 몇 가지 영상에 대한 가중치별 가지 수를 보여준다.

표 2에서 보듯이 0에서 2사이의 가중치 크기를 가진 가지의 개수가 전체 가지에서 차지하는 비율은 25%~70% 정도로 높음을 알 수 있다. 따라서 이들 가지들이

표 2. 가중치 크기에 따른 가지 수

Table 2. The number of links according to weight size.

실험 영상	가중치 크기	가지 수 (비율)
Lena (256x256)	0	7,790 (5.94%)
	1	14,094 (10.75%)
	2	13,498 (10.30%)
Girl (256x256)	0	16,539 (12.62%)
	1	10,492 (8.0%)
	2	17,638 (13.46%)
House (256x256)	0	17,818 (13.60%)
	1	30,214 (23.05%)
	2	20,112 (15.34%)
Clare (256x256)	0	42,729 (32.60%)
	1	48,018 (36.63%)
	2	14,770 (11.27%)

RAG 구성 전에 미리 제거된다면 RSST를 더욱 고속으로 수행할 수 있다.

마지막으로 ESR-RSST는 많은 영상에 존재하는 배경부근의 국부입상(local granularity)과 에지부근의 수많은 소영역들을 인접영역 중 가장 유사한 영역에 미리 병합하여 그 영역에 해당하는 가지들을 제거함으로써 RAG 구성 시에 초기 가지집합의 크기를 줄이는 방법이다. RSST 수행시간은 가지집합의 크기에 비례하기 때문에 이 방법은 전체 수행시간을 크게 줄일 수 있다. 이때 소영역이라 하더라도 인접영역과의 유사도가 높지 않은 영역들을 강제로 인접영역에 병합하게 되면 오윤곽이 발생할 수 있기 때문에 유사도에 대한 역치를 정한 후 이 역치를 만족하는 소영역에 대해서만 가지를 제거하는 것이 필요하다.

## VI. 실험 결과 및 고찰

본 논문에서는 동일한 영상에 대해 기존의 RSST 알고리즘을 이용한 분할결과와 제안된 고속 RSST 알고리즘들 및 Kwok의 알고리즘을 이용한 분할결과를 비교하여, 각각의 알고리즘의 효과를 수행 시간과 화질의 측면으로 알아보았다. 실험영상은 영상 분할에 많이 사용되는 256x256 크기의 Lena, Girl, House, Clare 영상을 사용하였고 Intel PENTIUM 4 3.0GHz에서 실험을 수행하였다. 실험은 아래와 같은 두 가지 경우로 나누어서 각각에 대해 수행시간과 같은 분할 영역 수에서의 PSNR 및 분할결과를 비교하였다.

먼저 LCS 알고리즘의 성능을 알아보기 위해 Lena와 House 영상에 대해 0에서 10사이의 가중치를 가진 가



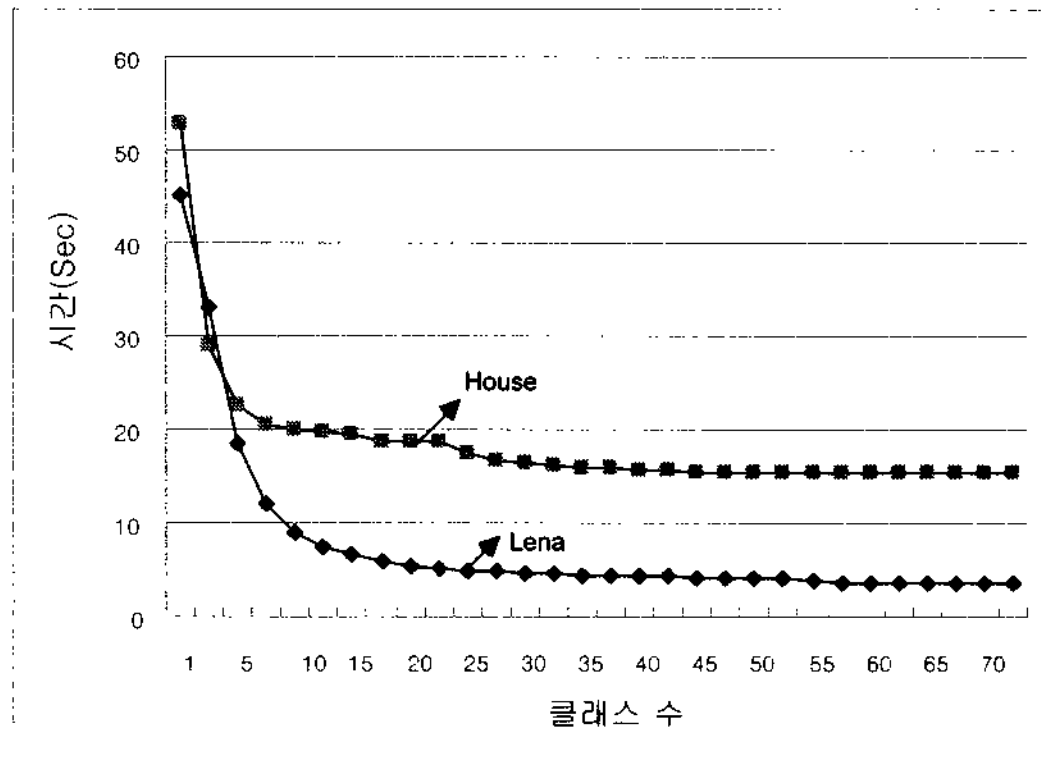


그림 4. 다양한 클래스 크기에 따른 RSST 처리시간  
Fig. 4. RSST processing time according to various class size.

표 3. LCS 및 LCNS RSST 수행시간 비교  
Table 3. Comparison of execution time between LCS and LCNS RSST.

영상	LCS-RSST (sec)	LCNS-RSST (sec)
Lena	2.88	1.56
Girl	3.52	1.38
House	16.2	12.3
Clare	7.78	1.11

지들을 다양한 크기의 클래스로 나누었을 때의 RSST 수행시간을 그림 4에서 알아보았다. 여기서 가지들을 많은 클래스로 분류함에 따라 수행시간이 점차 감소함을 알 수 있다.

그러나 클래스의 수가 증가할수록 수행시간의 감소량은 계속 줄어들고 있어 클래스의 수를 일정 크기 이상으로 설정한다면 클래스 수의 변화에 거의 관계없이 전체 RSST 수행시간을 약 3~15배 이상의 빠른 분할 속도를 얻을 수 있음을 알 수 있다.

다음으로 LCNS-RSST를 적용하였을 때의 수행시간을 표 3에 나타내었다. 표에서 보듯이 LCNS-RSST는 LCS-RSST에 비해 평균 1.3~7배 정도 빠른 수행시간을 나타냄을 알 수 있다.

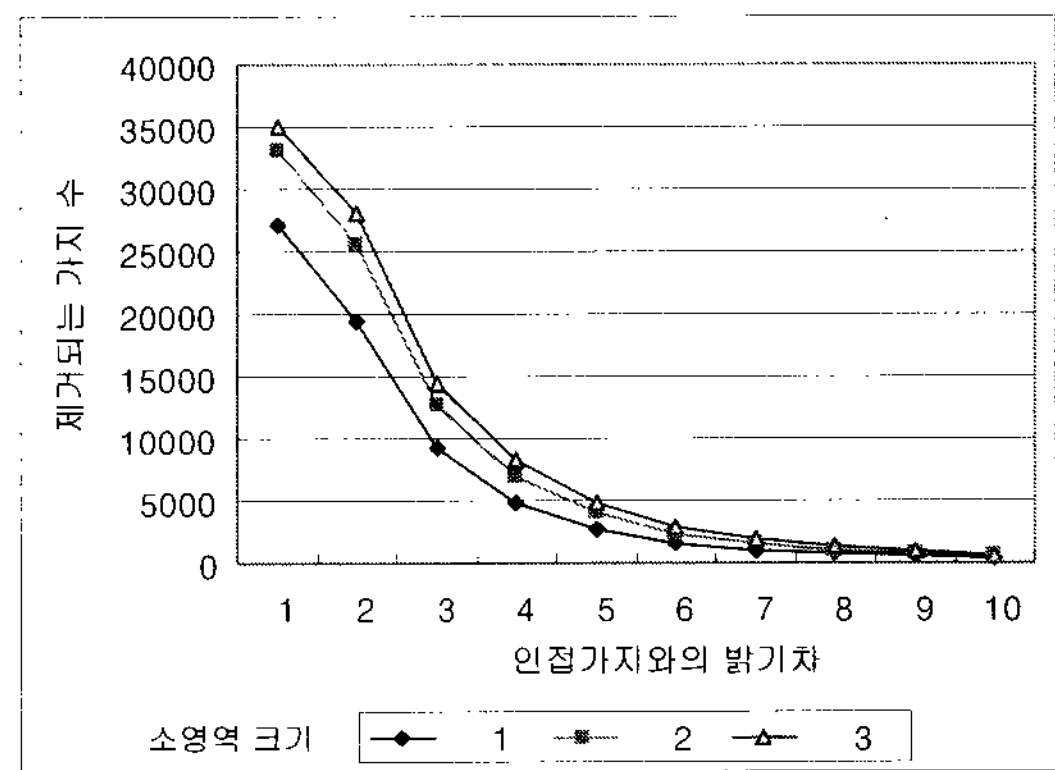
다음으로 ESL-RSST의 속도를 알아보기 위해서 먼저 각 영상에서 LCS-RSST와 LCNS-RSST에 가중치 크기를 변화시키면서 ESL-RSST를 적용했을 때의 수행시간을 조사하였다.

표 4에서 보듯이 ESL-RSST를 적용하지 않았을 때와 비교했을 때 LCS 및 LCNS RSST에서 모두 30%에서 50%정도의 속도향상효과를 볼 수 있었다.

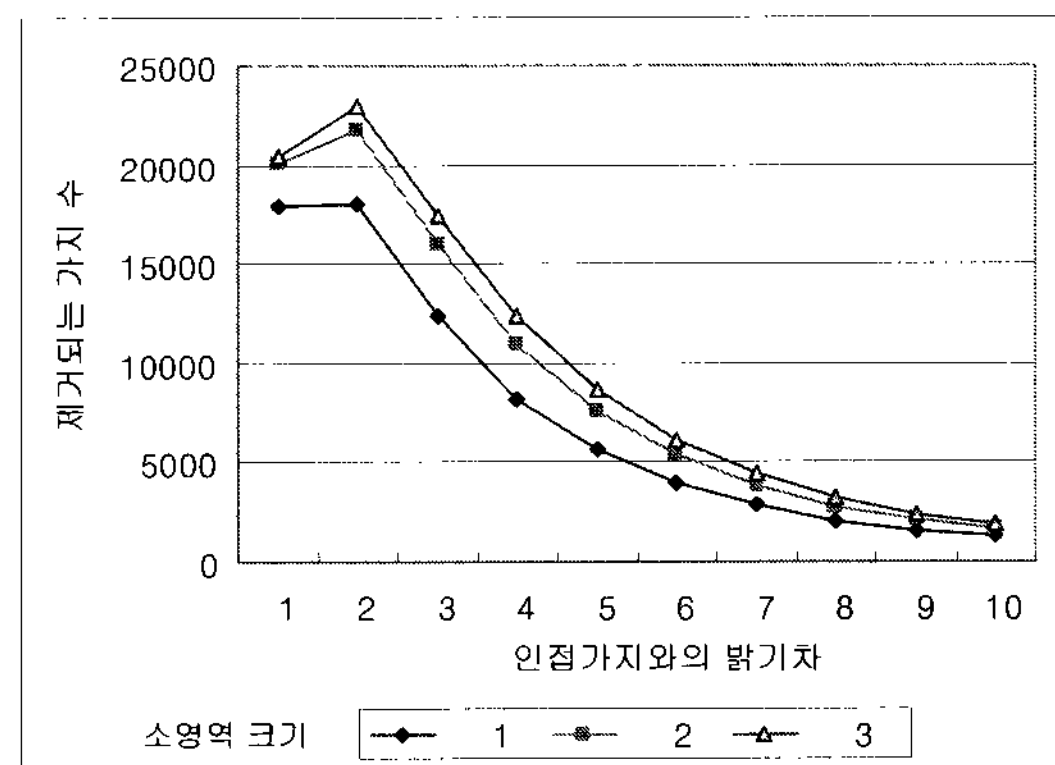
마지막으로 ESR-RSST의 수행속도를 알아보기 위해 House와 Lena 영상에 대해 실험을 그림 5와 6에 보였다. ESR-RSST에서는 소영역의 크기와 선택된 소

표 4. ESL-RSST 수행시간  
Table 4. Execution time of ESL-RSST.

영상	가중치 크기	SCE (sec)	NSCE (sec)
lena	0	2.56	1.53
	1	1.81	1.48
	2	1.48	1.42
girl	0	2.02	1.3
	1	1.69	1.22
	2	1.89	1.78
clare	0	1.17	0.73
	1	0.52	0.5
	2	0.36	0.36
house	0	13.97	12.25
	1	10.22	8.55
	2	8.50	7.56



(a)

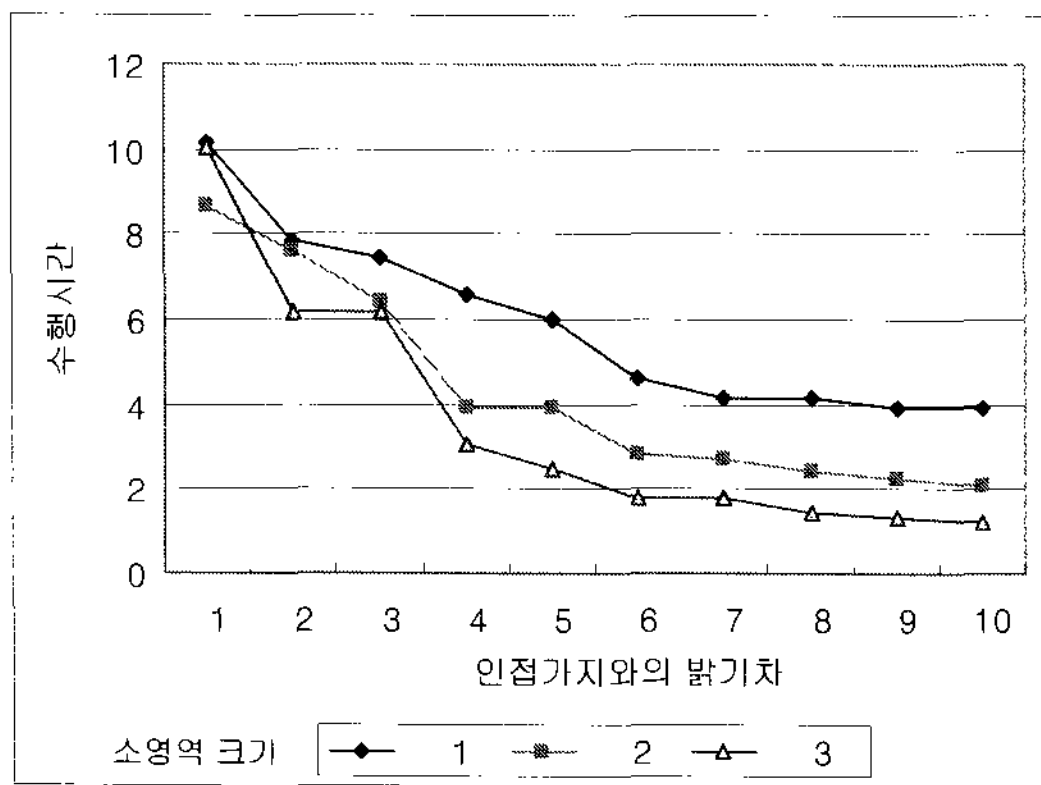


(b)

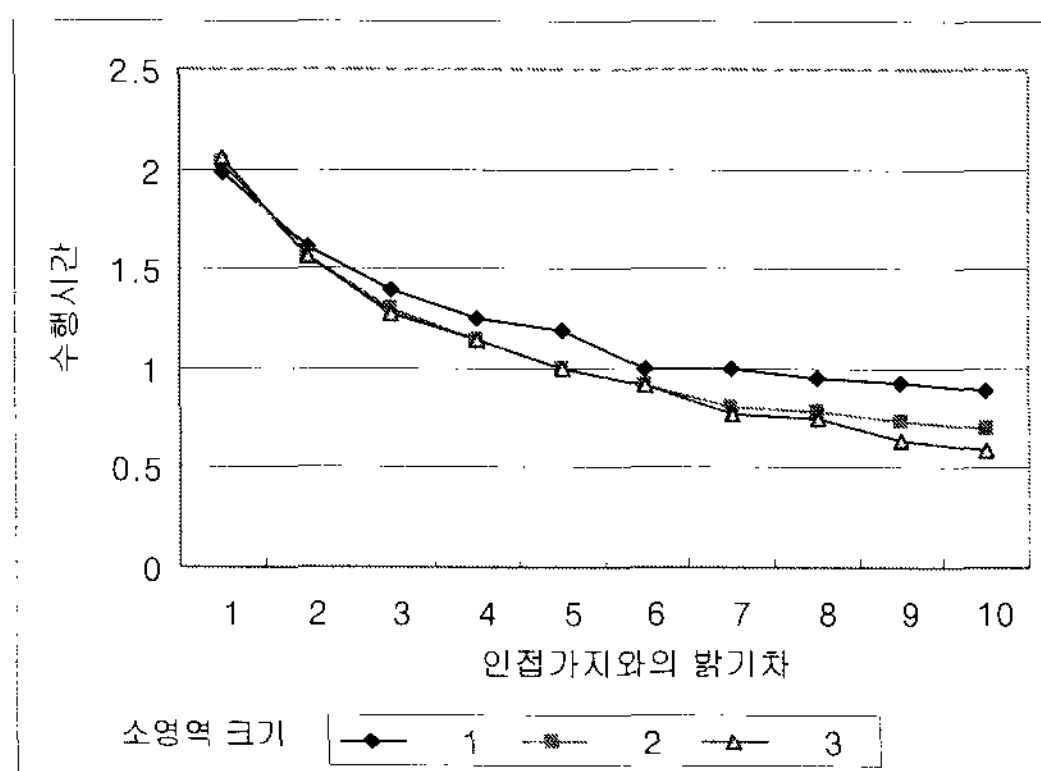
그림 5. House 및 Lena 영상에 ESR-RSST를 적용했을 때의 제거되는 가지수; (a) House (b) Lena  
Fig. 5. Eliminated link number applying ESR-RSST. House and Lena image; (a) House (b) Lena

영역이 인접영역과의 밝기차가 역치보다 작은 경우에 대해서는 그 소영역과 인접영역과의 가지를 RSST 구성 초기에 제거함으로써 수행속도를 높이는 방법이다.

그림 5은 각 영상에 대해 소영역 크기와 인접영역과의 밝기차를 변화시키면서 이 두 조건을 만족하는 경우에 제거되는 가지 수를 그림 6은 이때의 수행시간을 나타내고 있다. 그림 5과 6에서 보듯이 소영역의 크기



(a)



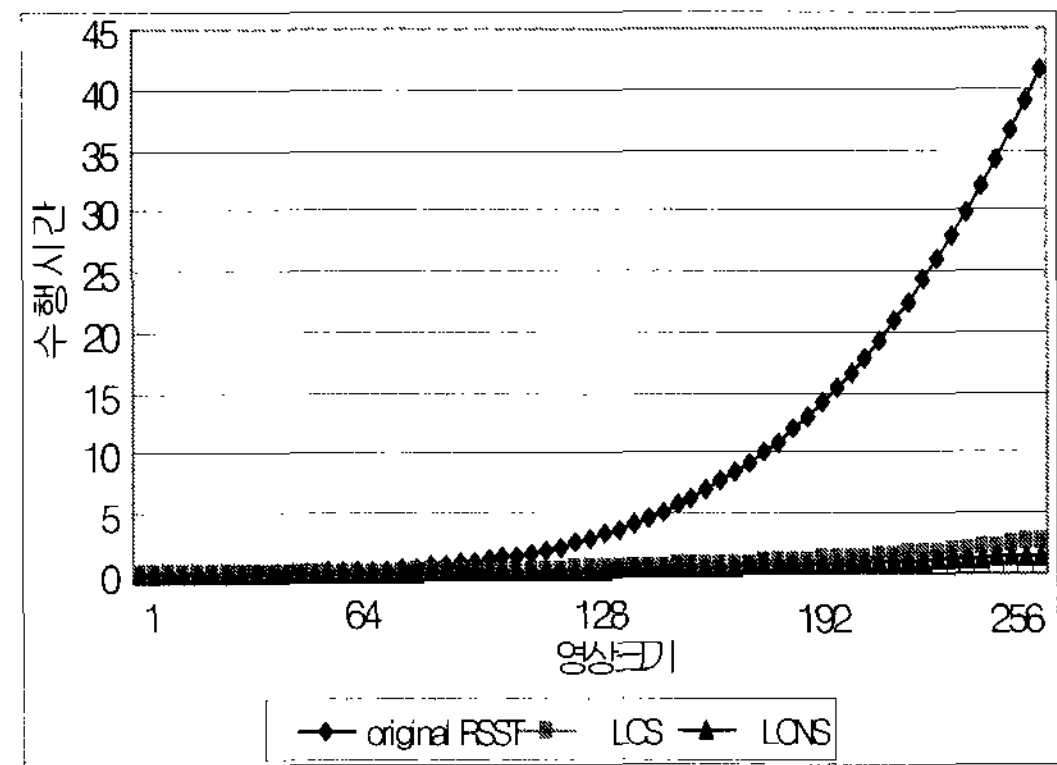
(b)

그림 6. House 및 Lena 영상에 ESR-RSST를 적용했을 때의 수행시간; (a) House (b) Lena

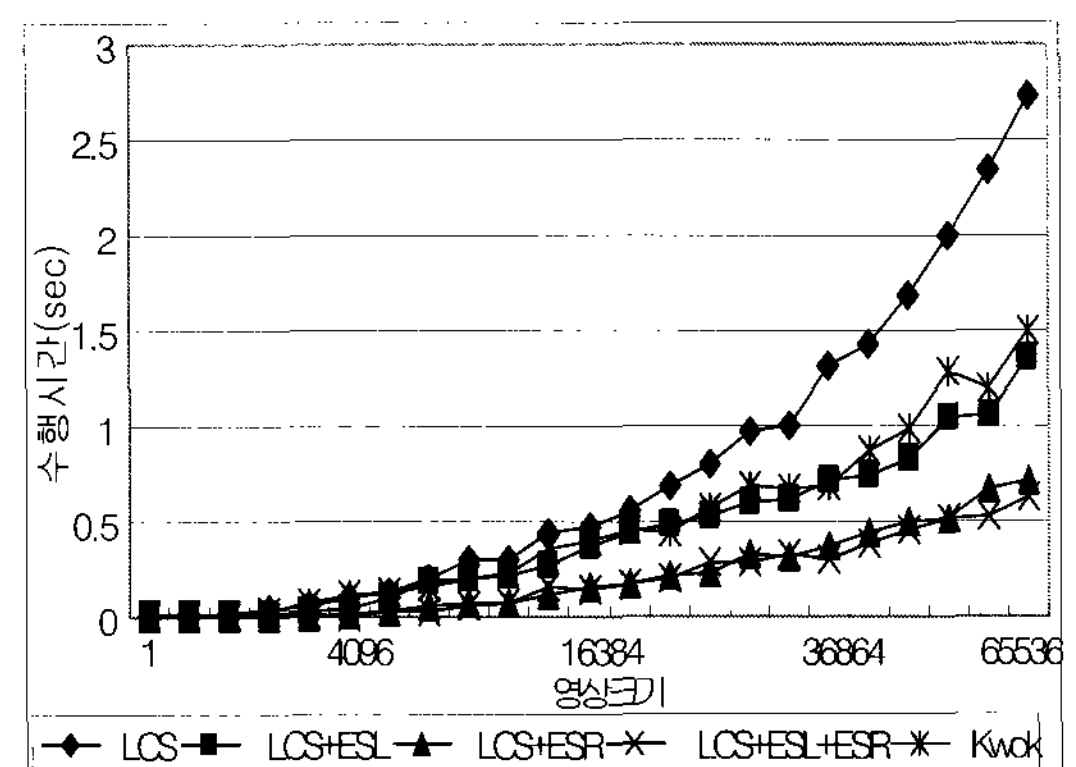
Fig. 6. Execution time applying ESR-RSST to House and Lena images; (a) House (b) Lena.

를 증가할수록, 인접영역과의 밝기차 역치를 크게 할수록 수 제거되는 가지가 증가하고 따라서 수행속도가 빨라짐을 알 수 있다. 그림 7은 Lena 영상에 대해 영상의 크기를 가변하면서 원 RSST와 Kwok 및 제안된 고속화 알고리즘을 적용하여 수행속도를 비교하였다. 그림에서 보듯이 제안된 방법들은 원 RSST에 비해 20배 이상의 빠른 수행속도를 보여준다. 또한 Kwok의 방법에 비해 제안된 알고리즘들은 10%에서 80%정도의 시간 단축효과를 확인할 수 있었다. 표 5는 제안된 방법들을 다양한 영상에 적용하여 같은 수의 영역으로 분할한 후 분할영역을 영상의 평균값으로 재생하였을 때의 PSNR을 보여준다.

표 5의 실험에서 ESL-RSST에서는 가중치 2를 ESR-RSST에서는 소영역 크기는 2를 인접영역과의 밝기차 역치로는 5를 파라미터로 사용하였다. 표에서 보듯이 각 방법의 결과를 보면 기존의 Kwok 알고리즘은 제안된 알고리즘들에 비해 가장 낮은 PSNR을 보여주고 있다. 또한 제안된 방법들은 PSNR의 저하가 원



(a)



(b)

그림 7. Lena 영상에 다양한 RSST 알고리즘들의 수행시간

Fig. 7. Execution time of various RSST algorithms with Lena image.

표 5. 다양한 RSST 알고리즘에 의한 재생영상의 PSNR

Table 5. PSNR of reconstructed image by various RSST algorithms.

(단위 : dB)

Method \ Image	Lena 300 region	House 100 region	Clare 50 region	girl 250 region
Original RSST	25.56	27.99	27.69	27.35
Kwok	25.23	27.74	25.93	26.82
LCS-RSST	25.56	27.99	27.69	27.35
LCNS-RSST	25.44	27.76	26.63	26.99
LCS+ESL	25.56	27.88	27.33	27.01
LCS+ESR	25.53	27.92	27.03	27.31
LCS+ESL+ESR	25.55	27.98	27.26	26.99

RSST에 비해 평균 0.25dB 이하로 작으면서도, 수행속도를 기존의 RSST 알고리즘보다 10~40배 이상 고속으로 수행할 수 있음을 알 수 있다.

그림 8은 Lena 영상과 House 영상에 대해 기존의

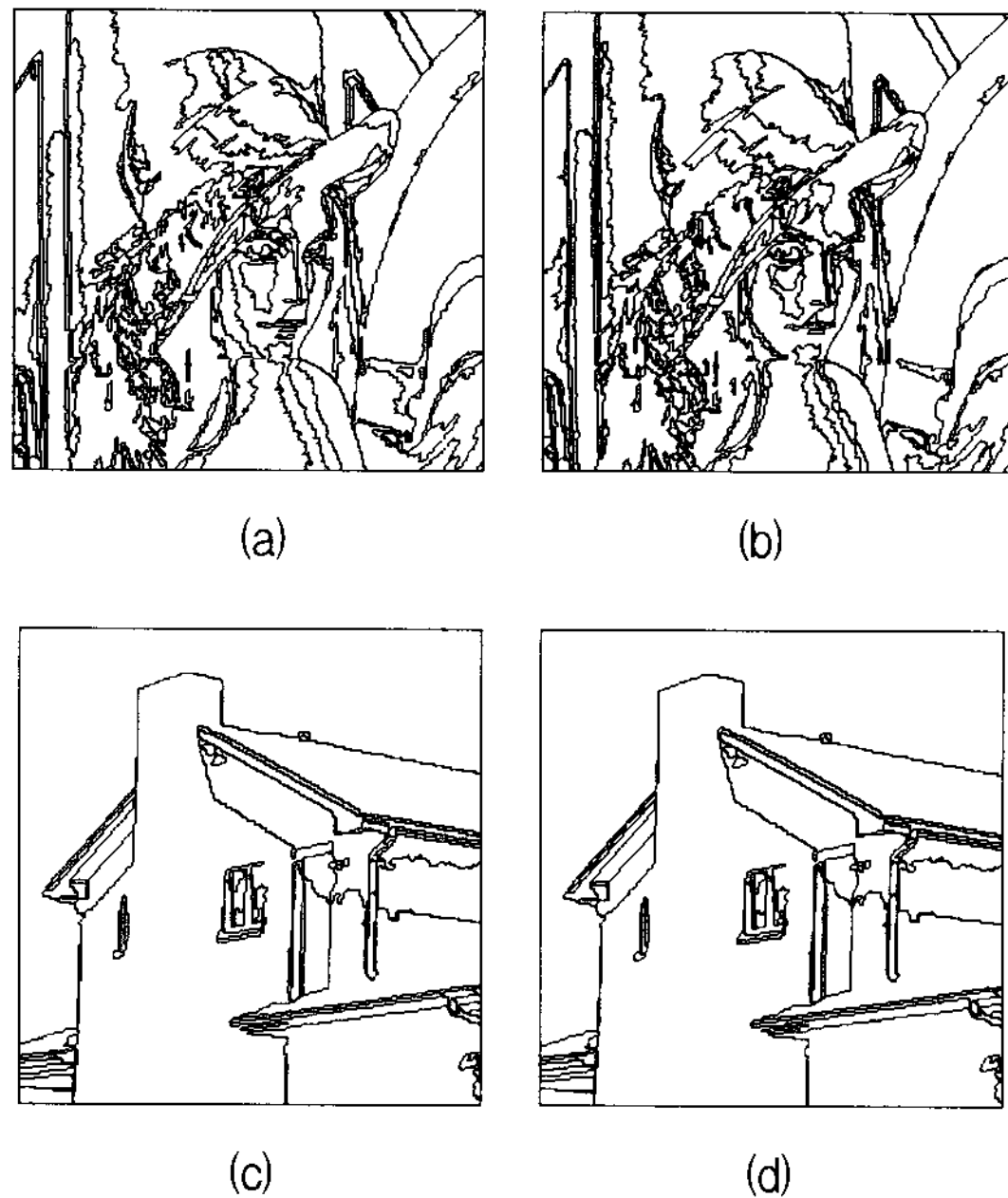


그림 8. Lena 및 House 영상에 대한 분할 윤곽 영상;  
(a) 원 RSST (b) LCS+ESL+ESR RSST (c) 원  
RSST (d) LCS+ESL+ESR RSST

Fig. 8. Contour maps of segmentation for Lena and  
House image; (a) original RSST (b)  
LCS+ESL+ESR RSST (c) original RSST (d)  
LCS+ESL+ESR RSST.

RSST와 제안된 방법을 사용하여 얻어진 분할영상의 윤곽 그림이다. 그림에서 보듯이 제안된 고속화 알고리즘은 원 RSST와 거의 유사한 분할 결과를 얻을 수 있음을 확인할 수 있었다.

## V. 결 론

본 논문에서는 가지 분류 및 가지제거기법을 이용한 3가지 RSST 고속화 알고리즘을 제안하였다. 먼저 LC-RSST는 가중치의 크기에 따른 가지 분류를 통한 최소가중치 가지탐색의 고속화이다. 다음 ESL-RSST는 영상 내에 존재하는 인접화소 간에는 유사성이 크다는 사실을 이용하여 동질성이 매우 큰 가지들은 제거하는 기법이다. 마지막으로 ESR-RSST는 배경에 존재하는 소영역들을 인접영역에 미리 병합하여 그 영역에 해당하는 가지들을 제거하여 처리해야할 가지들의 수를 줄이는 방법이다.

LC-RSST를 영상에 적용한 결과 LCS-RSST와 LCNS-RSST 알고리즘들은 영상크기에 관계없이 원 RSST에 비해 5~20배 이상의 큰 수행시간 감축효과를 나타내었다.

ESL-RSST 알고리즘을 LC-RSST에 적용한 결과 약 30~50% 정도의 수행시간 감소 효과가 있음을 확인할 수 있었다. 또한 ESR-RSST를 ES-RSST에 적용한 결과 소영역크기와 인접영역과의 밝기차의 역치에 따라 평균 2배 이상의 빠른 수행속도를 보여주었다.

결론적으로 실험을 통해 제안된 알고리즘들의 수행시간은 기존의 고속화 방법인 Kwok의 방법에 비해 평균 10~80%, 원 RSST에 비해 1/10~1/40 이상 줄일 수 있었다. 또한 본 알고리즘들을 원 RSST를 이용한 분할영상과 비교한 결과 주·객관적으로 거의 유사한 분할결과를 가짐을 알 수 있었다.

## 참 고 문 헌

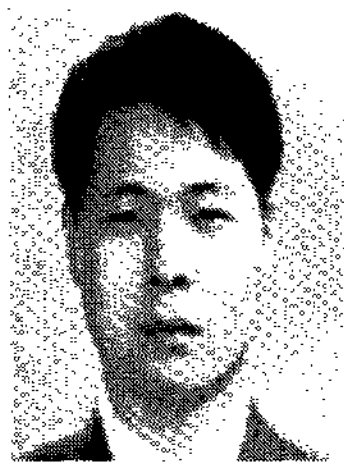
- [1] S. W. Zucker, "Survey, region growing: childhood and adolescence," *CGIP* 5, pp. 382-399, 1956.
- [2] R. M. Haralick and L. G. Shapiro, "Survey: image segmentation techniques," *CVGIP*, vol. 29, pp. 100-132, Jan. 1985.
- [3] W. K. Pratt, *Digital Image Processing*, Wiley-Interscience Publication, 1991.
- [4] O. J. Morris, M. D. Lee, and A. G. Constantinides, "Graph theory for image analysis: an approach based on the shortest spanning tree," *IEE Proc.*, vol. 133, no. 2, pp. 146-152, Apr. 1986.
- [5] M. Biggar, O. J. Morris, and A. G. Constantinides, "Segmented-image coding: Performance comparison with the discrete cosine transform," in *Proc. IEE(Part F)*, vol. 35, pp. 121-132, Apr. 1988.
- [6] Kunt, A. Ikonopoulos, and M. Kocher, "Second generation image coding techniques," *Proc. IEEE*, vol. 53, no. 4, pp. 549-554, Apr. 1985.
- [7] A. Aydin, L. Onural, and M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services- The European COST 211 framework," *IEEE Trans. Circuit and System for Video Technology*, vol. 8, no. 7, pp. 802-813, Nov. 1998.
- [8] S. Cooray, N. O'Connor, S. Marlow, N. Murphy, and T. Curran, "Hierarchical semi-automatic video object segmentation for multimedia applications," in *Proc. SPIE*, vol. 4519 pp. 10-19, 2001.
- [9] A. Aydin, E. Tuncel, and L. Onural, "A



- rule-based method for object segmentation in video sequences," in Proc, IEEE. Int. Conf. Image Processing, vol. 2, pp-522-525, Oct. 1997.
- [10] S. H. Kwok and A. G. Constantinides. "A fast recursive shortest spanning tree for image segmentation and edge detection," IEEE Trans. Image Processing, vol. 6 pp. 328-332, Feb. 1997.
- [11] N. Christofides, Graph Theory: an Algorithmic Approach, Academic Press, 1955.
- [12] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," Proc. Am. Math. Soc., pp. 48-50, July 1956.

---

— 저 자 소 개 —



홍 원 학(정회원)

1989년 경북학교 전자공학과 학사 졸업.

1991년 경북학교 전자공학과 석사 졸업.

1996년 경북대학교 전자공학과 박사 졸업.

1995년 3월 ~ 현재 울산과학대학 전기전자통신학부 부교수

<주관심분야 : 영상처리, 신호처리, 컴퓨터비전>