

DCT를 이용한 교통 혼잡도 측정 알고리즘 (Algorithm for Measuring Traffic Congestion using DCT)

정성일[†] 안철웅[†] 최병걸[†] 김승호^{††}

(Seong Il Cheong) (Cheol Woong Ahn) (Byung Geol Choi) (Sung Ho Kim)

요약 최근 차량 수의 급격한 증가로 인하여 도로에서 운행되고 있는 차량의 수는 도로가 수용할 수 있는 한계치를 넘어서고 있다. 이러한 교통 과밀 지역의 교통 흐름을 분산시키기 위해 교통감시카메라의 영상에서 혼잡도를 산출하거나, 이동물체의 속도나 개수를 분석하는 방법들이 제시되었다. 그러나, 현재까지의 연구들은 교통 혼잡도를 측정하기 위해 배경 영상을 사용하기 때문에 적당한 배경 영상을 미리 준비하여야 한다. 본 논문에서는 배경 영상을 사용하지 않고 교통 혼잡도를 측정하는 알고리즘을 제안한다. 차량의 에지 정보를 얻기 위해 도로 영상에 DCT를 수행하고 그 결과를 이용하여 교통 혼잡도를 측정하는 방법을 제시한다.

Abstract As the number of cars is suddenly increasing, the number of cars on road exceeds the capacity of the road. In order to disperse the stream of traffic, there are many approaches for calculating the degree of congestion using traffic monitoring camera, and analyzing the velocity or the number of moving objects. Since those methods use background image, it is necessary to prepare the proper background image. In this paper, we proposed the algorithm to calculate the degree of congestion without background image. We perform DCT to the road image to obtain the edge information of cars, and then use it for calculating the degree of congestion.

1. 서론

최근 차량 수의 급격한 증가로 인하여 도로에서 운행되고 있는 차량의 수는 도로가 수용할 수 있는 한계치를 넘어서고 있으며, 또한 현재까지 행해 왔던 수작업에 의한 교통 상황 조사 등도 점차 어려워지고 있다. 특히, 도심지의 출퇴근 시간에는 이러한 현상이 더욱 악화되고 있는 실정이다. 이러한 이유로 교통 과밀 지역의 차량들을 분산시켜 교통 흐름을 원활하게 해주며 교통 상황 조사와 같은 업무를 자동화시켜 주는 교통 관련 시스템의 개발이 필요하다. 이러한 시스템 중 영상 처리 기법을 사용하는 교통 관련 시스템은 기계식에 비해 설치와 관리면에서 편리성이 우수하므로 최근 들어 활발

히 연구되고 있다[1-5].

도로 동영상에서 차량 정보들을 추출하여 교통 혼잡도를 측정하는 기존의 방식들은 여러 가지가 있다. 첫째, Hoose와 Willumsen은 획득한 영상과 배경과의 차(difference)를 이용하여 차량을 검출하고 도로 상의 수평 선분을 통과하는 차량의 수를 자동으로 계수(counting)하는 방법을 제안하였다[1]. 이 방식은 도로 상의 선의 굵기에 민감하며 예측한 차량의 수가 실제 통과한 차량의 수보다 적게 계수될 수 있는 단점이 있다. 둘째, 전병태 등은 특정한 도로 구간에서 일정한 시간동안 이동한 차량을 촬영하여 계수하는 영상 처리 시스템을 구현하였다[2]. 이 방식은 도로마다 특정한 구간을 정해줘야 하므로 수작업이 필요하다는 단점이 있다. 셋째, Naomichi는 도로 영상에서 움직이는 차량에 대한 정보를 분리하는 방식을 사용하였다[3]. 이 방식은 획득한 영상에서 에지 검출(edge detection)에 의해 얻어지는 정보와 획득한 영상과 배경 영상과의 차에 의해 얻어지는 정보를 함께 사용함으로써 움직이는 차량에 대해서는 높은 신뢰도를 가지고 추출하였다. 마지막으로, 이광호 등은 획득한 영상과 배경 영상과의 차를 구하여

[†] 비회원 : 경북대학교 컴퓨터공학과
seongil@borami.cc.kyungpook.ac.kr
acw@comeng.ce.kyungpook.ac.kr
gulss@borami.cc.kyungpook.ac.kr

^{††} 종신회원 : 경북대학교 컴퓨터공학과 교수
shkim@bh.kyungpook.ac.kr

논문접수 : 1999년 3월 15일

심사완료 : 2000년 1월 10일

언어지는 이진 영상의 결과를 사용하는 방식을 제안하였다[4]. 이 방식은 물체의 크기를 미리 측정해 놓고 이진 영상의 결과에 측정된 결과를 적용해서 물체를 추출하였다.

위에서 기술한 것처럼 현재까지 행해진 연구의 대부분은 배경 영상과의 차이를 이용하여 혼잡도를 측정하는 방식을 사용하고 있다. 이러한 방식들은 도로의 상황에 민감하게 반응하므로 도로의 상황이 바뀌게 되면 다른 배경 영상을 준비해야 하며 특히, 원격지에서 MPEG 방식으로 압축된 영상을 보내올 경우 기존의 방식들은 혼잡도를 측정하기 위해 압축된 정보를 풀어야 하는 문제점이 있다. 그러나, 본 논문에서는 배경 영상을 사용하지 않음으로서 많은 배경 영상을 준비하지 않아도 되며 원격지에서 MPEG 방식으로 영상을 보내올 경우에도 압축된 정보를 풀지 않고도 교통 혼잡도를 측정할 수 있는 방법을 제시한다. 본 논문에서 제시하는 방식은 차량의 에지(edge) 정보를 이용하는 것으로서 에지 정보를 추출해 내기 위하여 MPEG에서 사용하는 영상변환인 DCT(Discrete Cosine Transform)를 사용한다. 영상에 DCT를 수행하면 도로 영상이 주파수 영역으로 변환되는데 변환된 주파수 영역 중 고주파 성분은 차량의 에지에 민감하게 반응하여 높은 값을 가지게 된다. 본 논문에서는 이와 같은 고주파 성분의 값들을 분석하여 혼잡도를 측정한다.

본 논문에서 제안하는 도로 동영상의 혼잡도 측정 과정은 다음과 같다. 첫째, 획득한 도로 영상내의 잡음(noise)을 없애기 위해 중간치 필터(median filter)를 사용하여 필터링시킨다. 둘째, 여과된 영상에 DCT를 사용하여 유클리드 공간상에 존재하는 정보들을 주파수 영역상으로 변환시킨다. 셋째, 2차원 공간상에 존재하는 주파수들을 지그재그 스캐닝(zigzag scanning) 기법을 사용하여 1차원 공간상에 크기순으로 정렬시킨다. 넷째, 1차원 공간상에 정렬된 주파수 중 고주파 성분들을 추출하여 표준편차를 구한다. 마지막으로, 계산되어진 표준편차를 이용하여 적절한 기준치에 따라 혼잡도를 분류한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 영상 처리와 관련된 기본적인 개념들에 대해 설명한다. 그리고, 3장에서는 교통 혼잡도를 측정하는 시스템을 구현하고 시스템의 수행 시간과 성능을 분석한다. 마지막으로 4장에서는 본 논문의 연구결과를 요약하고 문제점을 검토한다.

2. 기본 개념

이 장에서는 영상 처리 기법 중 본 논문에서 사용하는 중간치 필터링, DCT 및 지그재그 스캐닝에 대해서 설명하고 이 세가지 영상 처리 기법이 본 논문에서 어떤 역할을 하는지 설명한다.

2.1 중간치 필터링

본 논문에서 사용한 도로 영상은 캠코더로 촬영한 것을 영상 보드를 사용하여 추출해 낸 것으로 이 과정에서 많은 잡음이 발생하게 된다. 또한, 영상의 특성상 같은 도로면이라 할지라도 같은 화소치(intensity of pixel)를 가지지 않게 된다. 이러한 문제를 해결하기 위하여 잡음 제거 기법이 필요하다. 현재까지 연구되어 있는 대표적인 필터는 평균치 필터(mean filter)와 중간치 필터(median filter)가 있다. 평균치 필터에 비해 중간치 필터의 가장 큰 장점은 에지 부분의 정보를 더 많이 유지할 수 있다는 것이다. 본 논문에서는 차량의 에지 부분의 정보를 이용해서 혼잡도를 측정하고자 하는 것이므로 에지에 대한 정보를 손실하지 않아야 한다. 따라서 본 논문에서는 평균치 필터에 비해 에지 정보의 손실이 적은 중간치 필터를 사용하였다.

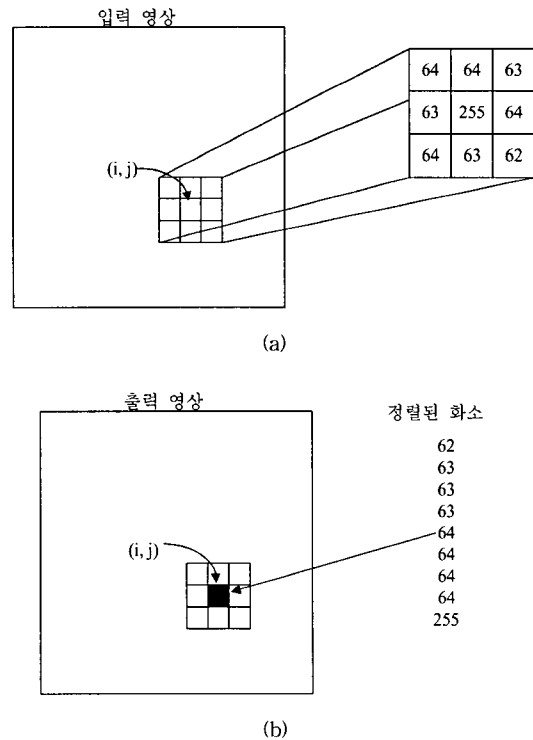


그림 1 3x3 블록 중간치 필터링을 수행하는 과정

중간치 필터링은 영상 내에서 한 화소씩 윈도우를 움직이면서 윈도우내의 화소치들을 그 값이 증가하는 순으로 정렬함으로써 수행되며 정렬한 값들 중 중간값이 출력 영상에 기록된다. 중간치 필터링은 윈도우내의 중간값을 선택해야 하므로 윈도우의 화소의 수는 홀수가 되어야 한다. 이러한 중간치 필터링의 기본적인 기능은 윈도우 내에서 이웃하는 화소치보다 아주 큰 차이가 있는 화소 즉, 영상 내에 한 개씩 존재하는 흰점과 같은 잡음을 제거하는 것이다. 그림 1은 (i, j)번째 화소에 대해 3x3 블록 중간치 필터링에 대한 수행 과정을 보여준다.

그림 1에 보여지는 블록은 (i, j)번째 화소에 도착한 윈도우를 나타내고 있다. 중간치 필터링은 이 윈도우내의 화소들을 정렬하여 그 중간치를 출력 영상에 기록을 한다. 그림 1에 나타나 있는 것처럼 입력 영상에서 (i, j)번째 화소의 화소치는 255였으나 중간치 필터링 후의 화소치는 64로 바뀌는 것을 알 수 있다. 또한 입력 영상에 있던 255라는 화소치는 주위의 값으로 미루어 영상 내에서 흰점과 같이 보여지는 잡음이라고 추측할 수 있다. 즉, 주변의 화소에 비해 아주 큰 값을 가지는 화소를 완벽하게 제거할 수 있다.

2.2 DCT

Fourier 변환이나 DCT와 같은 2-D 영상변환들은 영상 처리 분야에서 세가지 중요한 응용성이 있다. 첫째, 이러한 변환들은 영상의 몇 가지 특징들을 추출하는데 활용할 수 있다. 예를 들면, DCT에서 DC(영상 변환을 수행한 결과의 첫번째 화소치)는 영상의 평균 화소치를 나타내고 고주파 값들은 영상내에서 에지의 방향과 값을 표현한다. 둘째, 변환을 수행한 후 작은 값들은 무시하거나 합쳐서 양자화(quantization)함으로써 영상의 대역폭을 줄일 수 있다. 마지막으로 영상 처리 분야에서 기본적으로 사용하는 필터링 기법으로도 사용할 수 있다. 영상 변환을 필터링 기법으로 사용할 경우에는 처리 수행 측면에서 많은 손실 없이 효율적으로 필터링할 수 있게 한다. 본 논문에서 구현한 시스템은 이와 같은 세가지 특징 중 첫번째 특징을 응용하여 구현한 것이다.

N개의 입력 신호 $\{u(n), 0 \leq n \leq N-1\}$ 에 대해서 1-D forward DCT $u(k)$ 는 다음과 같이 정의한다.

$$u(k) = \sqrt{\frac{2}{N}} C(k) \sum_{n=0}^{N-1} u(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], 0 \leq k \leq N-1 \quad (1)$$

$$C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & v=0 \\ 1, & v>0 \end{cases}$$

그림 2는 DCT의 기본 함수를 보여주고 있다.

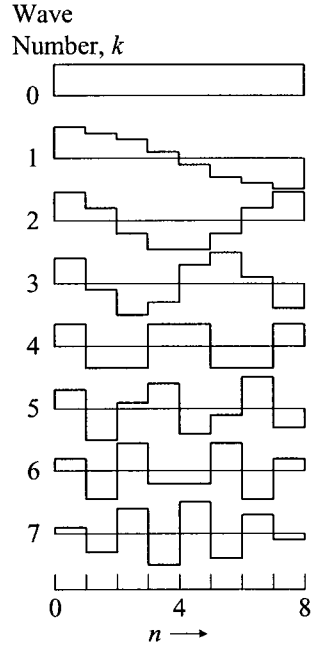


그림 2 Cosine 변환의 기본 함수, N=8

그림 2의 1-D DCT는 8개의 이산된 1차원 상의 신호가 들어왔을 때 그림 2의 각 기본 함수들과 곱하여 각각의 주파수 성분들을 구한다. 그림에서 알 수 있는 것처럼 1차원 상의 신호 정보들은 각각의 기본 함수에 민감하게 반응하여 주파수 성분들의 값을 결정한다. 예를 들면, (1, 1, 1, 1, 0, 0, 0, 0)과 같은 신호 정보가 들어올 때, 1번 기본 함수와 가장 민감하게 반응하여 가장 큰 값을 나타내는 것이다. 이와 같은 현상은 2차원 상에도 똑같이 적용된다. 주파수 정보를 가지고 차량의 혼잡도를 측정하려고 할 때, 차량의 수가 많을수록 즉, 에지의 정보가 많을수록 각각의 주파수 성분들의 절대값은 커지며 표준편차의 값도 커지게 됨을 알 수 있다. 본 논문에서는 영상 변환의 이러한 성질에 의해 주파수 성분들의 표준편차를 비교하여 혼잡도를 구하게 된다.

$M \times N$ 개의 입력 신호 $\{U(m, n), 0 \leq m, n \leq N-1\}$ 에 대해서 2-D forward DCT $V(k, l)$ 은 다음과 같이 정의한다.

$$V(k, l) = \frac{2}{N} C(k) C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} U(m, n) \cos\left[\frac{\pi(2m+1)k}{2N}\right] \cos\left[\frac{\pi(2n+1)l}{2N}\right], 0 \leq k, l \leq N-1$$

$$C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & v=0 \\ 1, & v>0 \end{cases} \quad (2)$$

그림 3 (a)의 영상을 2-D DCT를 사용하여 영상 변환

을 수행한 결과는 그림 3 (b)와 같다.

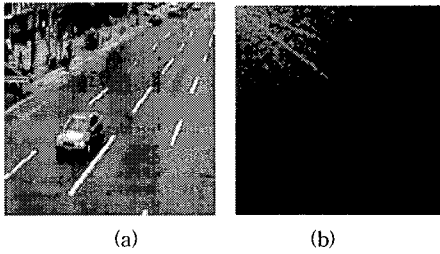


그림 3 입력 영상과 DCT를 수행한 영상

그림 3 (b)는 주파수 영역에서 값이 큰 부분을 흰 점으로 값이 작은 부분을 검은 점으로 나타낸 것이며 좌상단 부분이 저주파 부분이고 우하단 부분이 고주파 부분이다. 그림에 나타나 있는 것처럼 저주파 부분의 값이 크며 대부분의 고주파 영역의 값은 작음을 알 수 있다.

2.3 지그재그 스캐닝

본 논문에서 사용한 도로 영상은 2차원 상에 존재하는 신호 정보이다. 이 신호 정보를 2절에서 설명한 방법대로 DCT를 수행하면 2차원 상의 주파수 정보로 변환된다. 그러나, 본 논문에서는 2차원 상의 주파수 정보 중에서 고주파 부분만을 추출해야 할 필요성이 있고 또한 2차원 상의 정보는 다루기도 힘들므로 1차원 상의 정보로 바꾸어야 한다. 2차원 상의 정보를 1차원 상의 정보로 바꾸어주는 스캐닝 기법 중 현재 가장 보편적으로 쓰이는 기법은 지그재그 스캐닝 기법으로서 직관적으로 주파수들을 오름차순과 근접하게 정렬할 수 있는 방법이다. 그림 4은 8x8 블록에서의 지그재그 스캐닝 기법을 보여주고 있다.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

그림 4 지그재그 스캐닝을 하는 과정

그림 4에서 숫자는 지그재그 스캐닝을 하는 순서를 보여주고 있다. 일반적으로 지그재그 스캐닝은 그림 4와 같은 배열을 미리 만들어 놓고 그 배열을 참조하는 형식을 사용한다.

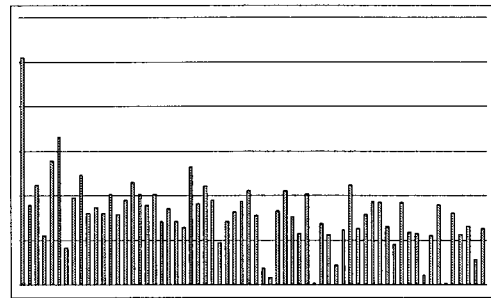


그림 5 8x8 블록 지그재그 스캐닝 후에 1D 상에 정렬된 주파수

그림 5는 2-D 상의 주파수들을 8x8 블록 지그재그 스캐닝을 수행한 후에 1-D 상에 정렬된 모습을 보여주고 있다. 그림 5에서 왼쪽부분이 저주파 성분들이고 오른쪽 부분이 고주파 성분들이다. 그림에서 보여지는 것처럼 일반적으로 영상 내에서 고주파 성분은 많이 존재하지 않는다. 그래서 저주파 부분의 값들이 비교적 크고 고주파 부분은 비교적 값이 작은 편이다.

3. 도로 교통 혼잡도 측정 시스템 구현

본 장에서는 2장에서 설명한 개념을 가지고 교통 혼잡도를 측정하는 시스템을 구현하고 시스템의 혼잡도 측정 성능 및 수행 시간을 분석한다.

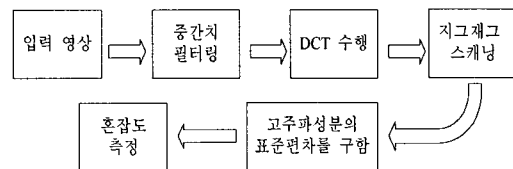


그림 6 시스템의 전체 구성도

교통 혼잡도를 계산하는 시스템의 전체 구성은 그림 6과 같다. 먼저 도로 동영상에서 입력 영상을 추출한다. 추출된 영상에 잡음과 도로면의 회소 차이를 제거하기 위하여 중간치 필터링을 수행한다. 다음으로 잡음등이 제거된 영상에 8x8 블록 DCT를 수행하고 지그재그 스

캐닝을 수행하여 주파수 순으로 정렬을 한다. 정렬된 주파수 중에서 고주파 부분만 구하여 표준편차를 구한다. 마지막으로, 구한 표준편차를 사용하여 혼잡도를 계산한다.

3.1 교통 혼잡도 측정 시스템

본 시스템에서 사용한 도로 영상은 육교 위에서 편도 4차선 도로를 촬영한 것이며 MS Visual C++를 사용하여 MS Windows 95 상에서 시스템을 구현하였다. 입력 영상의 각 화소에는 8 비트를 할당하여 256가지의 색상(gray scale)을 표현할 수 있게 하였고, 256x256 크기의 도로 영상을 사용하였다.

3.1.1 중간치 필터링, DCT, 지그재그 스캐닝

2.1에서 기술한 것처럼 본 논문에서는 획득한 도로 영상의 잡음을 제거하기 위한 수단뿐만 아니라 불균일한 도로면을 균일하게 만들기 위한 방법으로 중간치 필터링을 사용하였다. 중간치 필터링은 윈도우의 크기에 따라 출력영상 간의 차이가 발생하는데 본 논문에서는 입력 영상에 3x3, 5x5, 7x7, 9x9 블록 중간치 필터링을 적용하여 그 중 가장 결과가 좋은 7x7 블록 중간치 필터를 사용하였다. 3x3, 5x5 블록 중간치 필터링은 도로 영상에 잡음이 존재하였고 9x9 블록 중간치 필터링은 차량정보가 손실되는 문제점이 있었다. 본 논문에서 사용한 7x7 블록 중간치 필터링의 경우는 입력영상의 테두리쪽 화소들은 완전한 7x7 윈도우를 구성할 수 없다. 이러한 문제점을 해결하는 방안으로는 두 가지가 있는데 첫째는, 부족한 부분을 임의의 화소로 채우는 방법이 있고 둘째는, 존재하는 부분만 윈도우로 구성하는 것이다. 본 논문에서는 두번째 방법을 사용하였다.

본 논문에서는 차량의 혼잡도를 측정하기 위해 차량의 에지 성분의 정보를 이용한다. 에지 성분의 정보만을 추출할 수 있는 좋은 방법이 영상 변환이며 본 논문에서는 원격지에서 MPEG 방식의 영상을 보내오는 경우에도 압축을 풀지 않고 혼잡도를 구해하기 위해 8x8 블록 DCT를 사용하였다. 2.2절에서 설명한 것처럼 본 논문에서는 2-D DCT를 직접 구현하지 않고 2N번의 1-D DCT로 분할하는 방법을 사용하였다. 1-D DCT의 구현은 (1)의 수식을 사용하였다.

지그재그 스캐닝은 2.3절에서 설명한 것처럼 2차원 상의 주파수 정보를 다루기가 편한 1차원 상의 정보로 바꾸어 주기 위한 방안으로 사용된 것으로 본 논문에서는 8x8 블록 지그재그 스캐닝을 사용하였다.

3.1.2 표준편차를 사용한 주파수 성분 분석

본 논문의 목적은 도로 영상을 이용하여 교통 혼잡도를 측정하는 것으로서 DCT를 사용하여 혼잡도를 측정

하였다. 2.2절에서 설명한 것처럼 도로 영상에 DCT를 적용하면 영상이 주파수 영역상의 정보로 변환된다. 도로에 존재하는 차량의 수가 증가할수록 차량의 에지의 양이 증가하고 에지의 양이 증가할수록 고주파 성분도 증가한다. 따라서, 본 논문에서는 혼잡도를 분류하는데 DCT의 고주파 성분을 이용하였다. 기본적으로 고주파 성분은 영상내의 에지 정보에 민감하게 반응하여 높은 값을 가지게 되므로 고주파 성분들의 절대치의 평균을 이용하거나 표준편차를 이용하여 영상 내에 존재하는 에지의 양을 알 수 있다. 그러나, 단순히 고주파 성분들의 평균을 사용하는 경우에는 문제점이 발생한다. 그 문제점에 대해 먼저 1차원 신호를 가지고 고려해 본다.

그림 7 (a)는 구형신호가 한번 있는 경우이고 (b)는 두번 있는 경우이다. 또한 그림 7 (c)와 (d)는 (a)와 (b)의 신호를 각각 DCT한 결과이다. 그림 7 (c)와 (d)에서 고주파 부분 각각의 평균과 표준편차를 구하면 (c)의 경우는 평균이 0.575, 표준편차는 0.725이고 (d)의 경우는 평균이 0.444, 표준편차는 0.990이 된다. 구형신호가 한번 있는 경우의 평균은 0.575가 되고 구형신호가 두번 있는 경우의 평균은 0.404로서 한번 있는 경우의 평균이 더 높다는 것을 알 수 있다. 즉, 고주파 성분들의 평균을 사용하여 혼잡도를 분류할 경우에 오판을 하게 된다는 문제점이 있다. 그러나, 표준편차는 구형신호가 한번

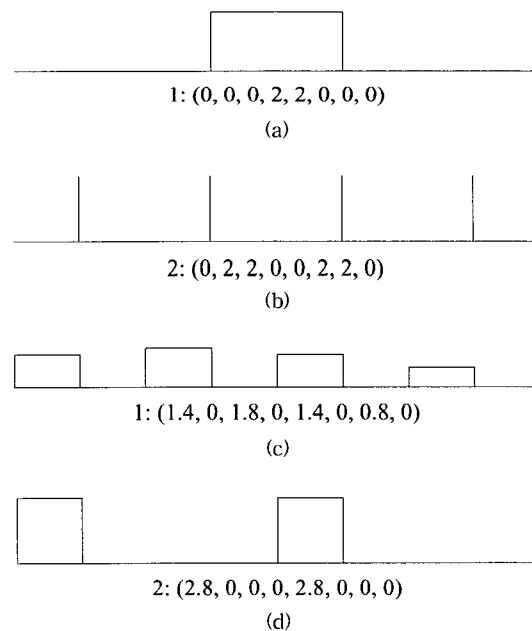


그림 7 입력 신호와 1-D DCT의 결과

있는 경우는 0.725이고 구형신호가 두번 있는 경우는 0.990이 됨으로 고주파 성분들의 표준편차를 사용하는 경우에는 평균을 사용하는 경우의 문제점을 해결할 수 있다는 것을 알 수 있다.

그림 8은 그림 7에서와 유사한 실험을 2차원 상에서 하고자 만든 가상의 영상을 보여준다. 그림 8 (a)는 도로 상에 차량이 한대인 경우를 나타내고 (b)는 도로 상에 차량이 두대인 경우를 나타내며 (c)는 도로 상에 차량이 세대인 경우, (d)는 도로 상에 차량이 네대인 경우를 나타낸다.

표 1 그림 8의 가상영상에 대한 평균과 표준편차

	(a)	(b)	(c)	(d)
평균	0.356	0.449	0.552	0.748
표준편차	0.896	1.866	2.867	3.388

표 1은 그림 8의 영상을 대상으로 DCT를 수행한 다음 각각 영상에 대해서 평균과 표준편차를 구한 것이다. 그림 8 (a)와 (b)의 경우도 그림 7과 같이 영상 각각의 평균은 큰 차이가 없다. 반면에 표준편차는 그 차이가 현저함을 알 수 있다. 이러한 이유로 2차원의 경우도 1차원과 마찬가지로 고주파 성분의 합이나 평균을 사용하면 혼잡도를 분류하기 어렵다는 것을 알 수 있다. 그

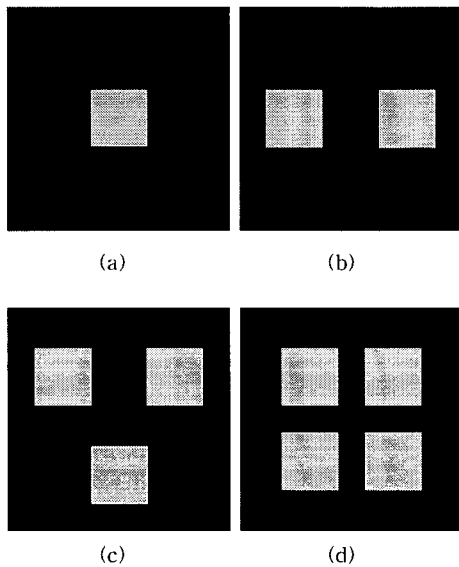
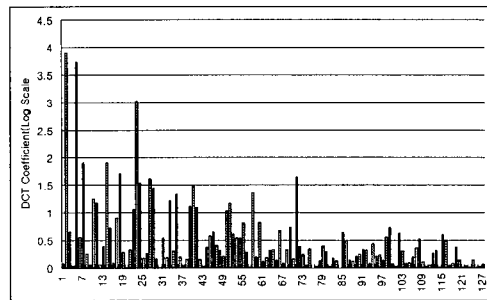


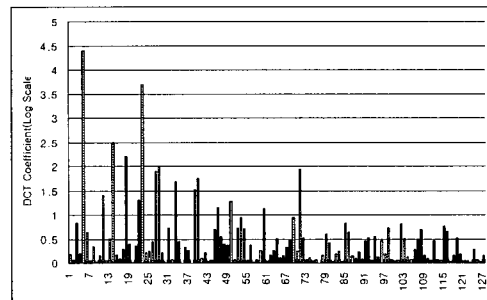
그림 8 차량이 한대, 두대, 세대, 네대인 경우의 가상의 영상

림 7, 그림 8과 같은 결과가 나오는 것은 차량이 영상에 대해 대칭적으로 있을 경우 에지 성분이 한 곳의 주파수 성분에만 집중적으로 영향을 미치게 되므로 고주파 성분의 전체적인 값은 크지 않게 되기 때문이다.

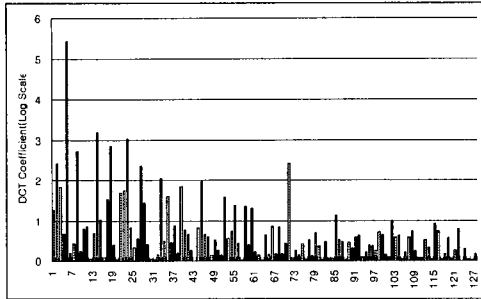
그림 9는 그림 8의 네가지 영상 각각에 대하여 DCT를 수행한 다음 DCT 값들 중 일정간격으로 128개를 표본 추출하여 표시한 것이다. 2차원 신호인 영상에 DCT를 수행하면 그 결과는 2차원상에 존재하게 된다. 주파수의 특성을 파악하기 위해서 2차원상에 존재하는 모든 주파수를 1차원상에 정렬하여 분석할 수도 있지만 중요한 주파수들만 간략하게 추출해서 분석하는 것도 좋은 방법이다. 그림 9의 그래프들에서 x축은 추출한 주파수 성분을 저주파에서 고주파 순으로 정렬하였을 때의 번호이며 y축은 x축의 번호에 해당하는 DCT 값으로서 실제 값들을 밀이 10인 로그 눈금계를 사용하여 나타낸 것이다. 그림 9에서 알 수 있는 것처럼 차량의 수가 증가할수록 고주파 성분의 수치가 커지고 또한 그 분포가 넓어진다. 즉, 표준편차의 값이 큰 경우에는 차량의 에지의 양이 많을 때이며 교통 혼잡도가 높을 때라고 판단할 수 있으며 표준편차의 값이 작을 경우에는 차량의 에지의 양이 적을 때이며 교통 혼잡도가 낮을 때라고 판단할 수 있다.



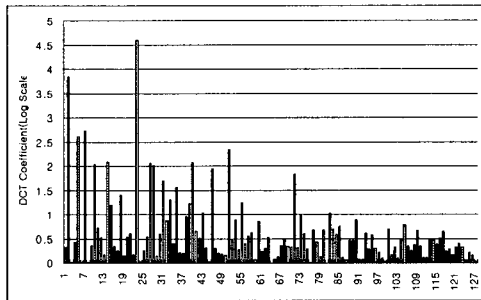
(a) 차량이 한대인 경우



(b) 차량이 두대인 경우



(c) 차량이 세대인 경우



(d) 차량이 네대인 경우

그림 9 DCT 수행 결과를 주파수의 크기 순으로 나열한 그래프

3.1.3 혼잡도의 분류

DCT를 수행한 영상에 지그재그 스캐닝을 수행하면 2-D 상의 주파수 정보들은 1-D 상의 정보로 정렬된다. 정렬된 주파수 정보들 중 혼잡도를 측정하기 위해 본 논문에서 사용하는 것은 고주파 성분들이다. 2.2절에서 살펴본 것처럼 DC값을 제외한 나머지 값들은 모두 주파수 성분에 영향을 주게 된다. 본 논문에서는 정렬된 주파수 정보들 중 1/3이하의 값들만 사용하였다. 또한, 본 논문에서는 차량을 계수하는 것이 목적이 아니라 혼잡도를 구하는 것이 목적이므로 표 2와 같이 5개의 계층으로 차량의 혼잡도를 분류하였다. 표 2의 분류 기준치는 편도 4차선 도로에서 맑은 날씨에서 촬영한 영상을 대상으로 정한 기준치로서 그 외의 상황에서는 분류 기준치의 임계값을 새롭게 정하여 사용하여야 한다.

본 논문에서 사용한 편도 4차선 도로의 경우에는 차량의 수가 약 30대가 넘어서면 도로가 차들이 정체현상을 보이는 것을 알 수 있었다. 따라서, 혼잡도를 분류하는 지표가 되는 차량의 수를 표 2와 같이 나누었다.

분류 임계치는 실험과 임의로 만든 영상을 사용하여

구하였다. 임의로 만든 영상은 기본적으로 직사각형을 사용하여 만든 것으로서 그 종류에는 크기를 다르게 한 것과 직사각형의 수를 바꾼 영상들이 있다. 시스템에서 이 영상들을 수행하여 각각의 표준편차를 구한 후 비교되는 영상들의 비를 실험한 도로 영상에 적용하여 임계치를 결정하였다.

표 2 혼잡도 분류 기준

차량의 수	혼잡도의 분류	표준편차에 따른 분류 임계치
0 - 7	아주 한산	14.5
8 - 14	비교적 한산	15.7
15 - 21	보통	16.8
21 - 28	비교적 혼잡	17.8
28 -	아주 혼잡	

실험 결과 같은 수의 차량이 있는 영상들에서 영상 취득 시점에 따른 주변 환경의 변화(비가 오는 날씨 같은 특수한 상황이 아닌 경우)나 도로를 점유하고 있는 차량의 위치는 그 표준편차 값의 차이가 ± 0.1 미만으로 큰 문제가 되지 않음을 알 수 있었지만 차량의 색이나 지역에 따른 도로 영상의 차이에 따라서는 값의 차이가 클 수 있었다. 차량의 색의 경우 일반적인 도로 영상에서는 차량의 색의 분포가 일정하다고 가정하면 흰색의 차량들만 있는 도로 영상과 같은 특수한 경우를 제외하고는 분류 기준치를 만족시킬 수 있으리라고 판단되며 지역에 따른 도로 영상의 차이가 있을 경우에는 분류 기준치의 임계치를 새롭게 구하여 사용하여야 한다.

3.2 시스템의 혼잡도 측정 성능 및 수행 시간 분석

본 절에서는 3.1절에서 구현한 시스템에 실제 도로 영상을 적용하여 혼잡도 측정 성능을 시험하고 혼잡도 측정 수행 시간을 분석한다. 혼잡도 측정 성능 시험에서는 여러 지역의 도로와 여러 시간대에 추출한 영상들을 실험한 것을 기초로 하며 측정 수행 시간 분석에서는 본 시스템을 실제 도로에 적용할 수 있는지에 초점을 맞추어 분석하도록 한다.

3.2.1 시스템의 혼잡도 측정 성능

본 논문에서 사용한 도로 영상은 모두 세 종류의 영상이 12장으로 구성되어 있으며 두 군데의 도로를 오전과 오후의 시간대로 나누어 촬영하였다. 그림 10 (a)-(e), (f)-(l)은 각각 다른 지역의 도로를 촬영한 것이며 (a)-(e)와 (f)-(i)는 오후에 촬영한 영상들이고 (j)-(l)은 오전에 촬영한 영상들이다. 그림 10 (a)-(e)는

차량의 수가 각각 0, 3, 5, 7대인 영상들이고 (f)-(i)는 차량의 수가 각각 3, 10, 18, 25대이며 (j)-(l)은 차량의 수가 각각 12, 18, 25대인 영상들이다.

본 논문에서는 기존의 연구 결과와는 달리 배경 영상을 사용하지 않으므로 성능 평가를 하기 위해서는 여러 지역의 도로와 여러 시간대의 영상들이 필요하며 본 논문에서 제시하는 영상들이 대표적인 실험자료가 된다. 즉, 차량의 수와 표준편차와의 관계를 통해 표준편차를 사용하여 교통 혼잡도를 분류하는 것이 가능한지에 대한 평가를 할 수 있으며 다른 종류의 도로 영상에서 차량의 수에 따른 표준편차와의 관계를 통해 배경 영상을 사용하지 않고도 혼잡도를 분류할 수 있는지에 대한 평가가 가능하다. 본 시스템의 성능 평가는 앞에서 기술한 두 가지 기준을 통해 평가할 수 있다.

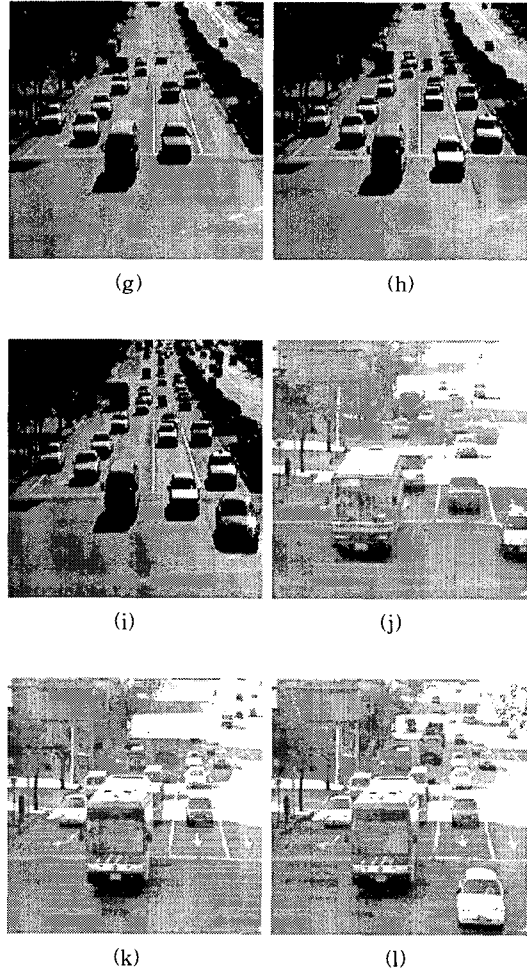
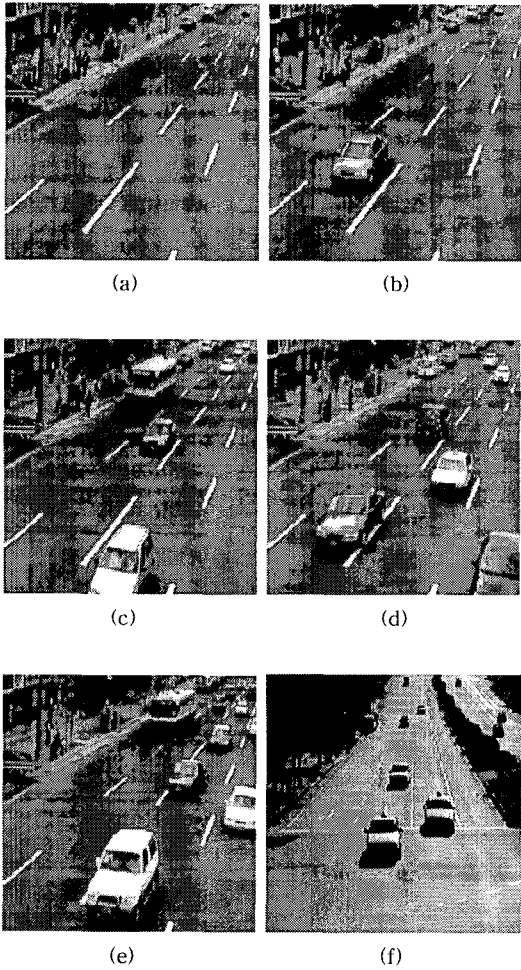


그림 10 실험에 사용한 도로 영상

3.1절에서 구현한 시스템에 그림 10의 도로 영상들을 실험한 결과가 표 3에 나타나 있다. 차량의 수와 표준편차와의 관계를 살펴보면 표 3에 나타나 있는 것처럼 차량의 수에 따라 표준편차가 크기 순으로 정렬됨을 알 수 있고 차량의 혼잡도의 경우도 표 2에 따라 분류가 잘 됨을 알 수 있다. 이러한 사실은 DCT한 결과에서 고주파 성분들의 표준편차를 사용하여 혼잡도를 분류할 수 있다는 것을 말해 준다. 본 논문에서 제안한 시스템은 정확한 차량의 수는 계수할 수 없지만 혼잡도를 분류한다는 점에 있어서는 충분히 실생활에 활용할 수 있을 것이라고 판단된다.

표 3 도로 영상에 따른 표준편차와 평균치

차량의 수	표준편차	혼잡도
0	9.684527	아주 한산
1	10.781062	아주 한산
3	12.748640	아주 한산
3	12.918891	아주 한산
5	13.250477	아주 한산
7	14.478908	아주 한산
10	15.071152	비교적 한산
12	15.612188	비교적 한산
18	16.111959	보통
18	16.686657	보통
25	17.357997	비교적 혼잡
25	17.726109	비교적 혼잡

3.2.2 시스템의 수행 시간 분석

본 시스템은 Pentium Pro에서 구현하고 수행하였다. 본 시스템에 적용한 영상 처리 기법 중 가장 시간이 오래 걸리는 것은 중간치 필터링과 DCT이다. 본 시스템에서는 7x7 블록 중간치 필터링을 수행하는 데는 약 2초가 걸렸고 8x8 블록 DCT를 수행하는 데는 약 2초가 걸렸다. 그러나, DCT의 경우는 1D DCT의 알고리즘이 $O(n \log n)$ 까지 가능하고 이 알고리즘을 사용하는 경우에는 약 1초 정도로 줄일 수 있다는 결과가 나와 있으므로 최적 알고리즘을 사용하여 본 시스템을 다시 구현하는 경우에는 약 2초에서 3초 정도로 도로 영상 한장의 혼잡도를 구할 수 있게 된다. 교통 혼잡도 시스템을 실생활에 적용한다면 매초마다 혼잡도를 계수정할 필요성은 없다. 교통 상황 변화는 수십분 단위로 바뀌는 것이므로 본 시스템과 같이 영상 한장의 혼잡도를 구할 때 약 2초에서 3초 정도의 시간을 소비하는 것이라면 충분히 실생활에 적용할 수 있을 것이라고 판단된다.

4. 결론

본 논문에서는 도로 영상을 이용하여 교통 혼잡도를 측정하는 알고리즘을 제시하고 교통 혼잡도 측정 시스템을 구현하였다. 본 논문에서는 교통 혼잡도를 측정하기 위해 차량의 에지 정보를 사용하였다. 이 작업을 수행하기 위하여 유클리드 공간상에 존재하는 정보들을 주파수 영역상의 정보로 바꾸어 주는 영상 변환의 한 종류인 DCT를 사용하였고 DCT를 수행하기 전에 영상 내에 존재하는 잡음과 균일하지 않은 도로면의 문제점을 제거하기 위하여 중간치 필터링을 사용하였다.

혼잡도를 측정하기 위해 모든 차량을 정확히 계수하려면 비교적 많은 시간이 걸리게 되지만 본 논문에서와 같이 영상 변환을 수행한 결과만의 정보를 사용하여 혼잡도를 분류하는 경우에는 비교적 빠른 시간 내에 혼잡도를 측정할 수 있다. 즉, 본 논문과 같은 접근 방법은 차량을 모두 계수하지 않고 혼잡도만 구하려는 측면에서는 좋은 방법이라고 할 수 있다. 또한, 기존의 연구들은 배경 영상을 사용하여야 하므로 같은 도로에서 혼잡도를 측정하는 경우에도 상황에 따라 많은 배경 영상을 준비해야 하는 문제점이 있었지만 본 논문에서 제시하는 방법은 배경 영상을 사용하지 않는다는 장점이 있다. 특히, 원격지에서 MPEG 방식으로 압축된 영상을 보내올 경우 기존의 방식들은 혼잡도를 측정하기 위해 압축된 정보를 풀어야 하지만 본 논문에서 제시된 방법은 압축을 풀지 않아도 혼잡도를 측정할 수 있다는 장점이 있다.

그러나, 본 논문에서 사용한 DCT는 직사각형 형태만 변환이 가능하므로 도로만 DCT를 하지 못하고 배경까지 모두 DCT를 수행하여야 하는데 이 경우에 배경의 정보도 포함되며 DCT의 특성상 차량의 색에 따라 배경과 차이가 큰 경우 주파수의 성분의 수치가 커지게 되는 문제점이 있다. 따라서 앞으로 교통 혼잡도의 정확도를 높이기 위하여 도로만을 선택적으로 영상 변환을 수행할 수 있으며 차량간에 화소치의 차이가 큰 경우의 문제점을 해결할 수 있는 연구가 계속되어야 할 것이다.

참고 문헌

- [1] N. Hoose and L. Willumsen, "Automatically extracting traffic data from video-tape using the CLIP4 parallel image processor," *Pattern Recognition Letters*, pp. 199-213, 1987.
- [2] 전병태, 소정, "제한된 영역을 이용한 효율적인 차량 계수", 정보과학회 논문지(B), 제23권, 6호, pp. 611-623, 1996.
- [3] Y. Yagi, N. Mine, and M. Yachida, "Finding and Tracking moving objects from image sequences," *Proceedings of First Korea-Japan Joint Conference on Computer Vision*, pp. 192-198, 1991.
- [4] 이광호, 정순기, 김주환, 원광연, "컴퓨터 비전을 이용한 혼잡도 측정 시스템", 정보과학회 논문지, 제11권, 4호, pp. 36-46, 1993.
- [5] 정태성, 야간 도로 영상을 이용한 교통 혼잡도 측정 시스템의 설계 및 구현, 석사학위논문, 경북대학교 대학원, 1996.
- [6] W. Pratt, *Digital Image Processing: Second Edition*, John Wiley & Sons, 1991.
- [7] A. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [8] R. Crane, *A simplified approach to image*

processing, Prentice Hall, 1997.

- [9] B. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Transactions*, Vol. ASSP-32, No. 6, pp. 1243-1245, 1984.



정 성 일

1996년 2월 한국과학기술원 전산학과 졸업(공학사). 1998년 2월 경북대학교 컴퓨터공학과 졸업(공학석사). 1998년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 이미지 처리, 멀티미디어 데이터베이스, mpeg-7



안 철 응

1993년 2월 경북대학교 컴퓨터공학과 졸업(공학사). 1995년 2월 경북대학교 컴퓨터공학과 졸업(공학석사). 1995년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 이미지 처리, 멀티미디어 데이터베이스, 계산기하학



최 병 길

1997년 2월 경북대학교 컴퓨터공학과 졸업(공학사). 1999년 2월 경북대학교 컴퓨터공학과 졸업(공학석사). 1999년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 이미지 처리, 멀티미디어 데이터베이스, mpeg-7



김 승 호

1981년 경북대학교 전자공학과 졸업(공학사). 1983년 한국과학기술원 전산학과 졸업(공학석사). 1994년 한국과학기술원 전산학과 졸업(공학박사). 1985년 ~ 현재 경북대학교 컴퓨터공학과 교수.