

4장

영상 (IMAGE) – 개요

김성영교수
금오공과대학교
컴퓨터공학부

학습 목표

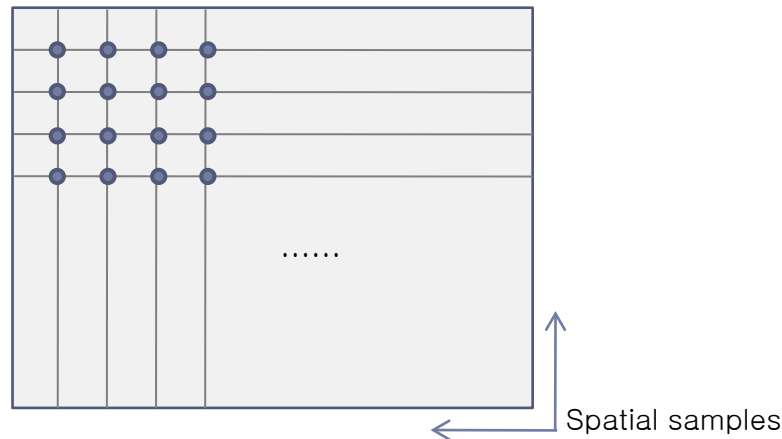
- 영상의 디지털화 과정을 설명할 수 있다.
- 디지털 영상의 표현 방식을 설명할 수 있다.
- Bitmap의 종류를 구분해서 설명할 수 있다.
- True Color 영상과 Indexed Color 영상을 구분해서 설명할 수 있다.

영상의 디지털화

- Sampling (표본화)

- 공간상의 2차원 함수를 규칙적이고 일정하게 특정한 좌표 x, y 에서 존재하는 값으로 표본(sample)을 취하는 작업

- picture element, pixel, pel



영상의 디지털화

- Sampling (표본화) (cont'd)

- 해상도 (resolution)

- 2차원 공간 영역에서 x, y 축 방향의 표본화된 픽셀들의 개수



a) 64 x 64 영상



(b) 128 x 128 영상



(c) 256 x 256 영상

표본화가 영상에 미치는 영향

영상의 디지털화

- Quantization (양자화)

- 표본화한 각 픽셀의 밝기 값을 정해진 단계의 밝기로 근사화 시키는 과정
 - 명암도 또는 그레이 레벨(gray level): 개개의 양자화된 값
 - 그레이 스케일(gray scale): 양자화된 값들의 범위

영상의 디지털화



(a) 2 레벨



(b) 4 레벨



(c) 16 레벨



(d) 256 레벨

양자화가 영상에
미치는 영향

디지털 영상의 표현 방식

- 비트맵(bitmap) 영상

- 작은 점을 사용하여 그림을 그리는 방식

- 'bit' + 'map'

- bit: the simplest element in which only two digits are used

- 'map': a two-dimensional matrix of these bits

- A bitmap is a data matrix describing the individual dots of an image

- 픽셀 (pixel, picture element)

- 영상을 구성하는 각각의 점을 나타내는 용어

- 서로 붙어있고 아주 작기 때문에 착시현상을 일으켜 이미지를 연속적으로 보이게 함

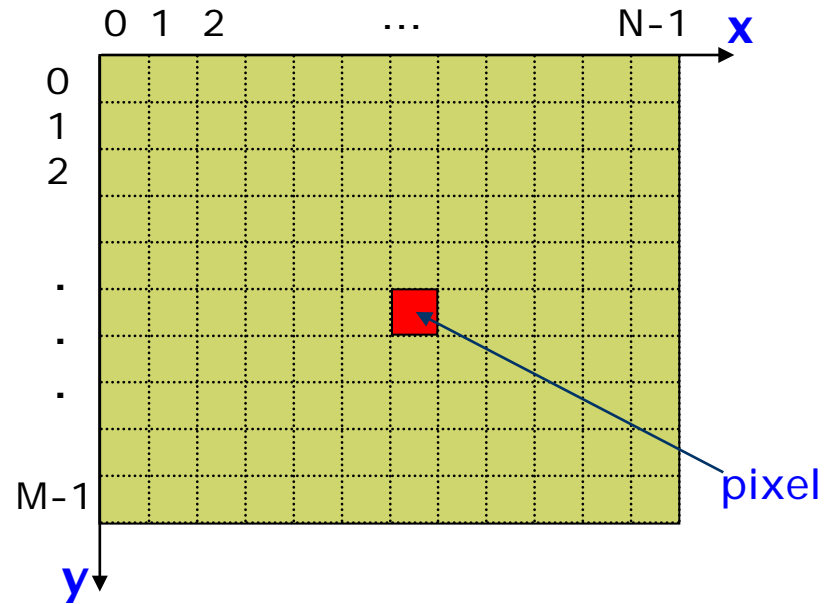
Bitmap (1)

$I(x, y)$

where

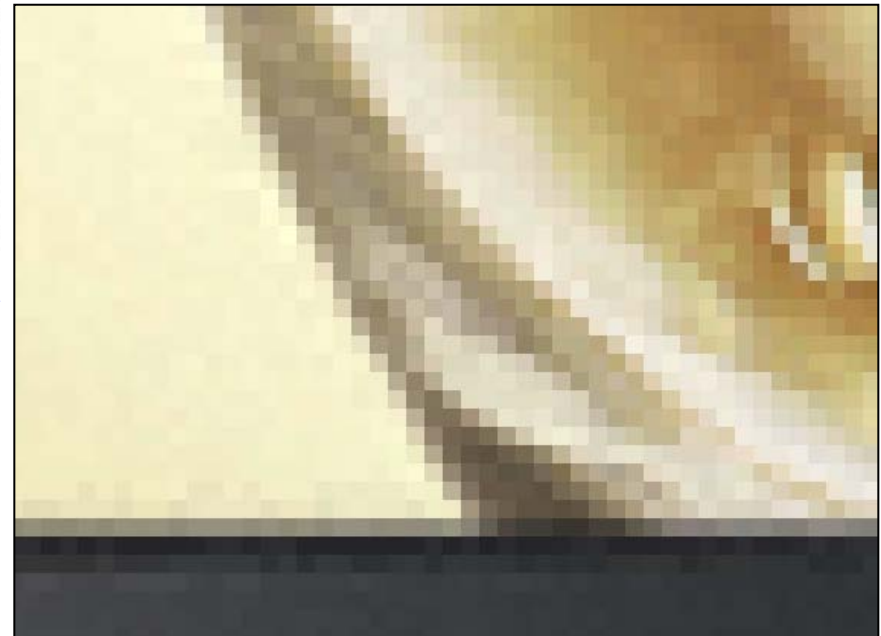
x, y : spatial coordinates

I : amplitude (intensity)



$$I(x, y) = \begin{bmatrix} I(0,0) & \dots & I(N-1,0) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ I(0,M-1) & \dots & I(N-1,M-1) \end{bmatrix}$$

Bitmap (2)



Bitmap의 종류 (1)

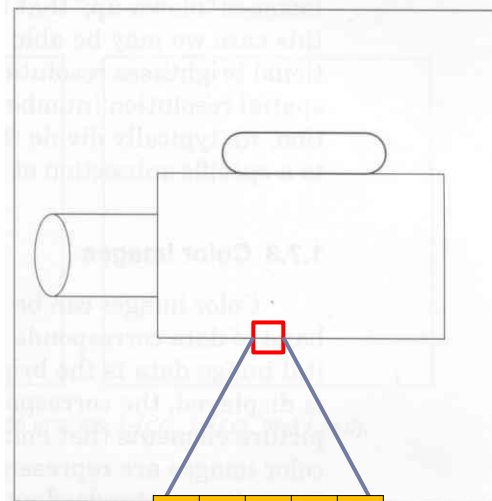
- 이진 영상 (Binary Images)

- 두 가지 값을 가짐 (black and white, or '0' and '1')

- 1 bit/pixel image

- 도형의 모양이나 외각선 정보를 표현하는데 유용함

```
wht2d(3)      C Library Functions      wht2d(3)
NAME
    wht2d - performs Walsh or Hadamard transform
SYNOPSIS
    #include <stdio.h>
    #include <stdlib.h>
    #include <math.h>
    #include "CvIPtools.h"
    #include "CvIPimage.h"
    #include "CvIPdef.h"
    IMAGE *wht2d(IMAGE *in_IMAGE, int ibit, int block_size)
    <in_IMAGE> - pointer to an IMAGE structure
    <ibit> - 0=inverse Walsh transform, 1=Walsh transform
           2=inverse Hadamard transform, 3=Hadamard transform
    <block_size> - block size (4,8,16,...largest_dimension/2)
PATH
    $CVIPHOME/TRANSFORMS/wht2d.c
DESCRIPTION
    This function performs a fast Hadamard-ordered Walsh-
    Hadamard Transform on an image. The result is then reor-
    dered for display in sequency order. The routine works on
    any image with dimensions that are powers of 2. Optional
    zero-padding may be performed if input image has different
    dimensions.
```



Bitmap의 종류 (2)

- 그레이스케일 영상 (Gray-Scale Images)

- 모노크롬(monochrome) 혹은 단일 색상 영상(one-color images)으로 지칭
- 칼라 정보는 없고 밝기 정보만을 포함하고 있음
- 픽셀 당 사용되는 비트 수에 따라 다른 레벨의 밝기를 표현 가능
 - Typically 8 bits/pixel
 - Sometimes 12 or 16 bits/pixel

10	11	155	167	170
15	17	151	158	162
9	13	161	157	155
25	14	159	160	159
19	13	164	155	161

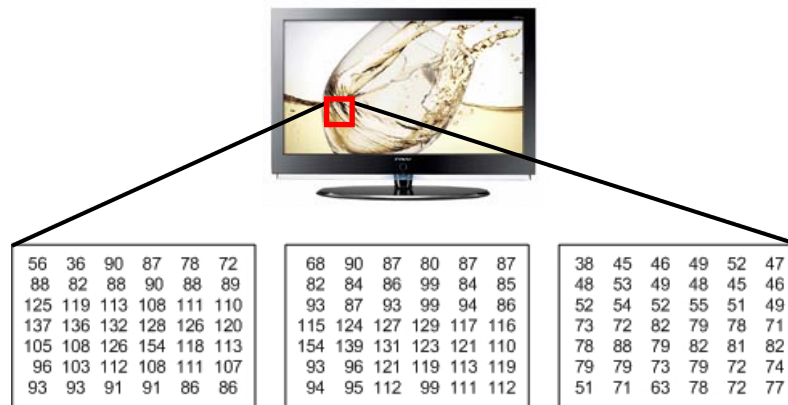


Bitmap의 종류 (3)

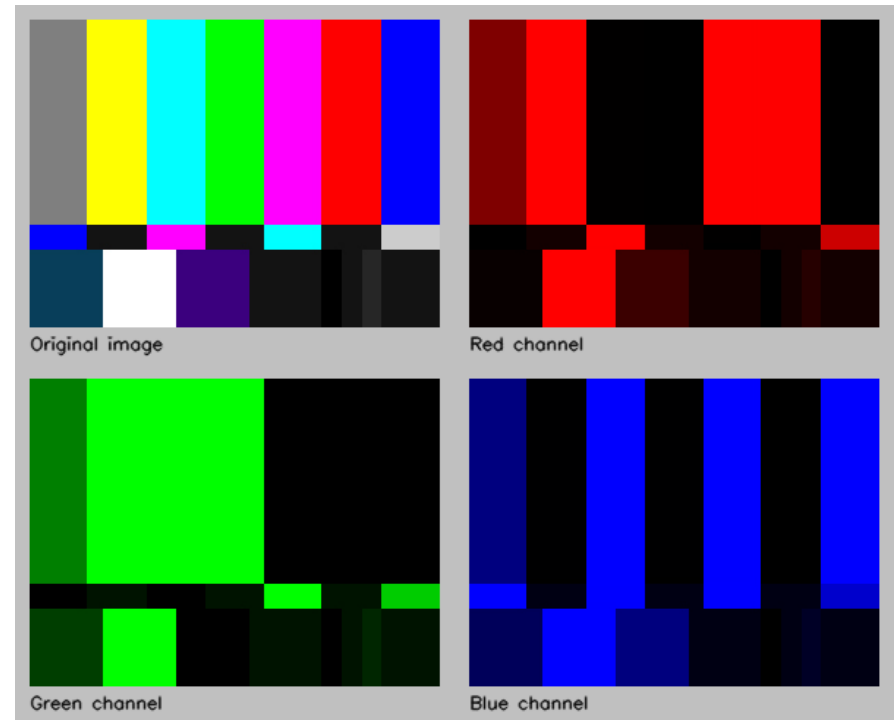
- 칼라 영상 (Color Images)

- 세 개의 대역(band 혹은 channel)를 갖는 영상 데이터
- 데이터의 각 밴드는 서로 다른 칼라(R, G, B)를 나타냄
- 24bit color images (**True color** images)
 - 1 pixel당 24 bit를 사용 (R,G,B 당 8 bit)
 - 16,777,216개의 색상을 지원

$$f_c(x, y) = \{f_{c_1}(x, y), f_{c_2}(x, y), f_{c_3}(x, y)\}$$



Bitmap의 종류 (4)

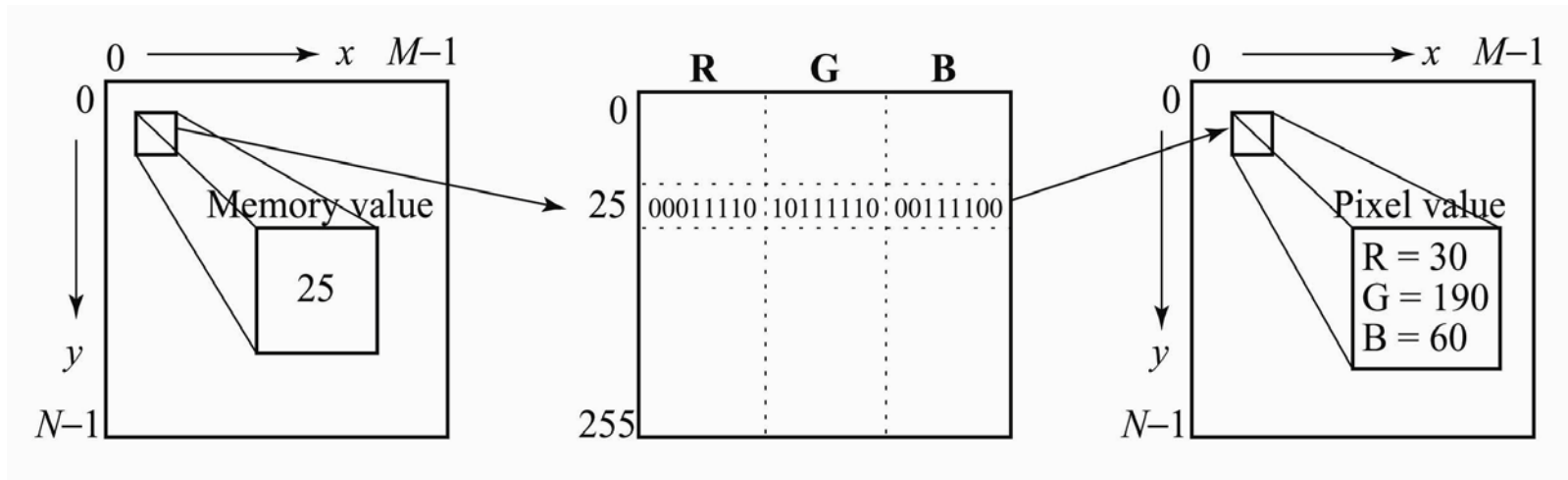


Bitmap의 종류 (5)

- 칼라 영상 (Color Images) (cont'd)

- 8 bit color images (**Indexed color** images)

- 1 pixel당 8 bit (256 color) 사용
- 컬러 정보를 저장하기 위해 **참조표(lookup table)**를 사용



Demonstration: Bitmap 종류 변환

- Using doll.jpg
- 포토샵: 이미지>모드 (image>mode) 메뉴
 - RGB 색상
 - 인덱스 색상
 - 회색 음영
 - 비트맵
 - 50% 한계값
 - 디더(패턴 디더, 확산 디더)
 - 해프토닝(하프톤 스크린)

학습정리

1. 영상의 디지털화

- 표본화: 일정한 간격으로 특정한 좌표에서 표본을 취하는 과정 (영상의 해상도를 결정)
- 양자화: 각 표본(픽셀)의 밝기에 대한 단계를 결정하는 과정 (픽셀의 그레이스케일을 결정)

2. 비트맵(bitmap)

- 작은 점(픽셀)을 사용하여 디지털 영상을 이차원 행렬 형식으로 표현하는 방식
- 이진 영상, 그레이스케일 영상, 칼라 영상(true color, indexed color)

1. 벡터 영상이란 무엇인가?

2. 벡터 영상의 장·단점 및 용도는 무엇인가?

Bitmaps vs. Vectors

- 비트맵(Bitmap) 영상
 - 표시 속도 빠름
 - 자연스러운 이미지 효과를 나타내거나 깊이 있는 색조와 부드러운 질감, 자연스러움에 적당하여 사진이나 회화 이미지의 표현에 적당
 - 많은 디스크 공간 소모
 - 파일의 크기는 해상도에 비례
 - 확대나 축소할 경우 그림의 모양이나 외곽선이 변형됨
 - 활용 S/W 계열: 영상처리 계열
 - 포토샵, 페인터, 코렐페인터 등

Bitmaps vs. Vectors

- 벡터(Vector) 영상

- 비트맵에 비해 적은 데이터 용량을 사용
- 화면 확대/축소 시 화질의 변화가 없음
- 파일크기의 큰 변화 없이 해상도에 대한 조절 가능
- 이미지의 선과 면이 깔끔하고 정갈하게 표현될 수 있음
- 사진과 같은 현실감 있는 영상을 표현하기 힘들
- 확대/축소할 때마다 벡터를 계산하여 매끄럽게 해주어야 하기 때문에 연산속도가 느림
- 활용 S/W 계열: 드로잉 및 애니메이션 계열
 - 일러스트레이터, 코렐드로우, 플래시 등

Bitmaps vs. Vectors

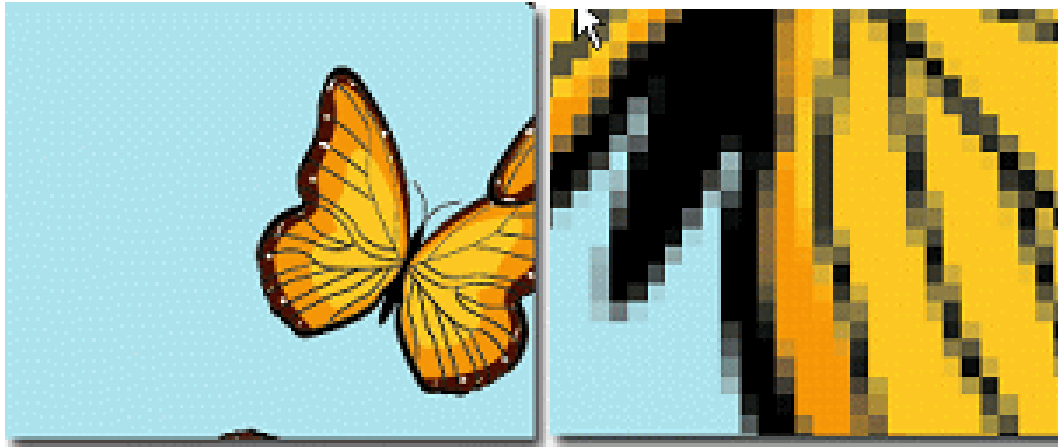


bitmap image

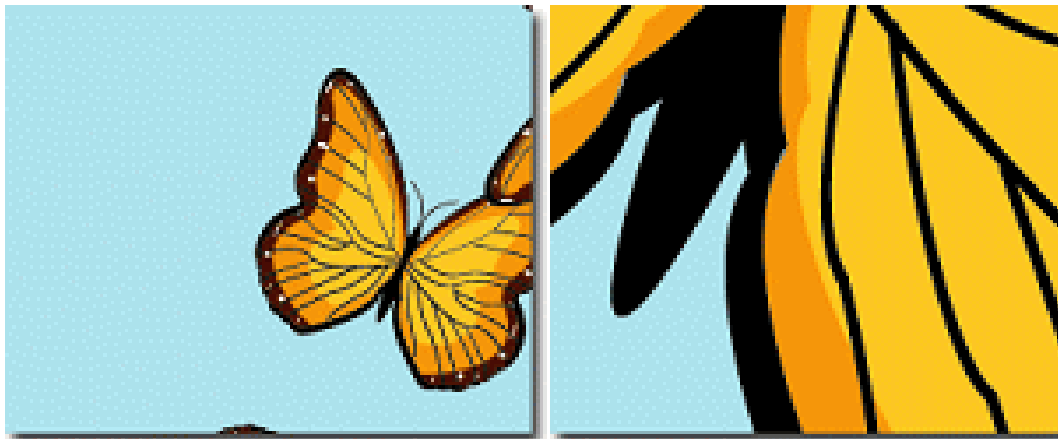


vector image

Bitmaps vs. Vectors



Bitmap image



Vector image