

# A Survey on Object Detection and Tracking Methods

2022.05.15

발표자 | 김진영

# Abstract

: object tracking을 위한 다양한 방법에 대한 분석 및 비교 연구를 포함하여 다양한 알고리즘 간략 조사

## 1. The goal of object tracking

: 비디오 장면에서 관심 영역을 분할하고, 움직임/위치/**폐색(Occlusion)**을 tracking 하는 것이다.



### \* what is occlusion?

: 보고 싶은 것이 있으나 센서 설의 속성 및 외부의 상황으로 볼 수 없는 것을 의미



### \* example for occlusion situation:

- 물체를 추적하는 시스템의 occlusion : 물체가 다른 물체에 의해 가려지는 경우

ex) 두 사람이 서로를 스쳐 지나감, 다리 아래를 달리는 자동차.



- 범위 카메라의 occlusion: 정보가 없는 영역

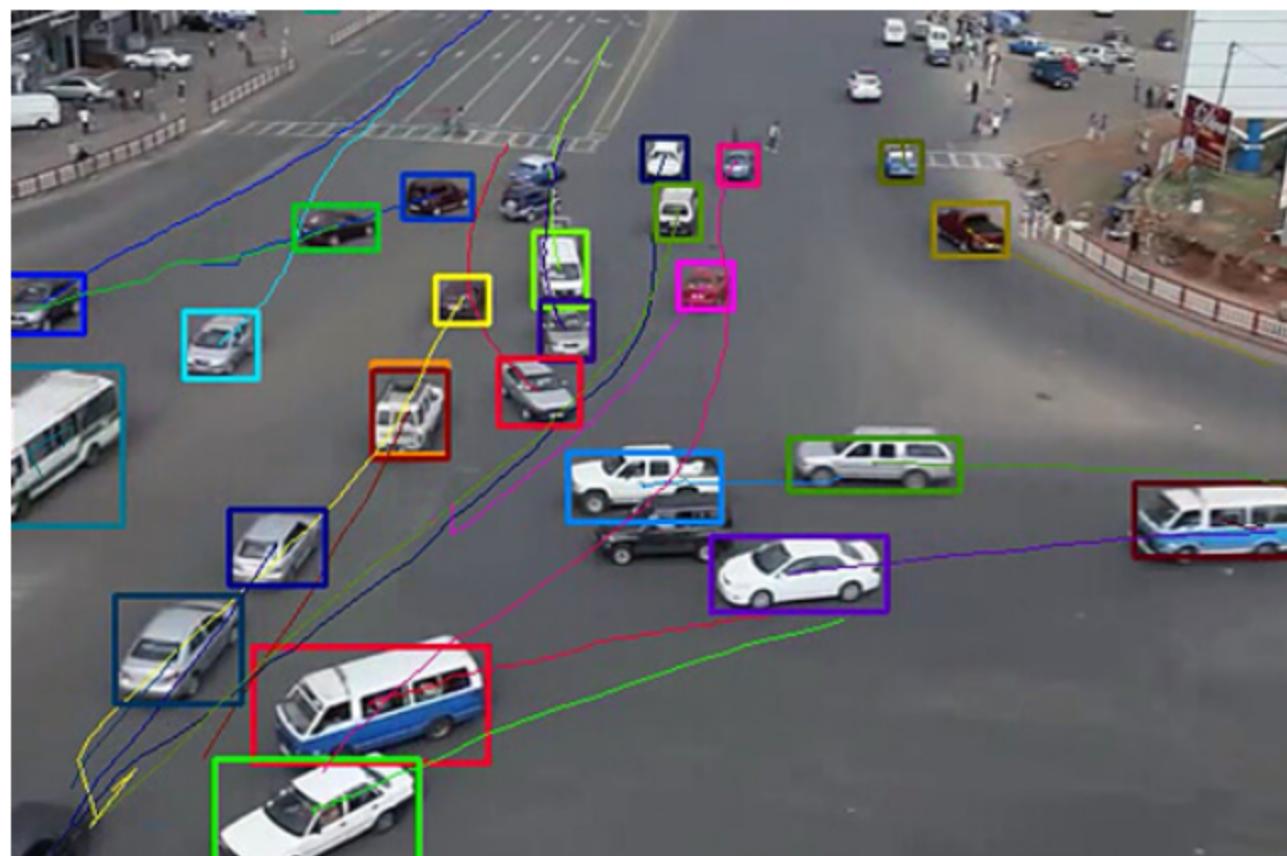
ex) 레이저 범위 카메라에서, 레이저가 충돌해 식별이 가능한 지점에서만 정보 획득이 가능

# Abstract

: object tracking을 위한 다양한 방법에 대한 분석 및 비교 연구를 포함하여 다양한 알고리즘 간략 조사

## 1. The goal of object tracking

: 비디오 장면에서 관심 영역을 분할하고, 움직임/위치/폐색(occlusion) 을 tracking 하는 것이다.



multi object traking 의 example

### \* object traking

비디오 안에서 움직이는 물체의 위치 및 관련 다른 정보들을 추정/ 예측

#### step1) Object detection

: 알고리즘을 사용하여, 개체 주위 경계상자 만들어 객체 분류 및 탐지

#### step2) Assigning unique identification for each ID

#### step3) Tracking the detected object as it moves through frames

: 관련정보 저장하는 동안, 프레임에서 이동하는 개체 추적

# Abstract

: object tracking을 위한 다양한 방법에 대한 분석 및 비교 연구를 포함하여 다양한 알고리즘 간략 조사

## 1. The goal of object tracking

: 비디오 장면에서 관심 영역을 분할하고, 움직임/위치/폐색(occlusion) 을 tracking 하는 것이다.

\* object tracking

비디오 안에서 움직이는 물체의 위치 및 관련 다른 정보들을 추정/ 예측

step1) Object detection & Object classification

: 알고리즘을 사용하여, 개체 주위 경계상자 만들어 객체 분류 및 탐지

step2) Assigning unique identification for each ID

step3) Tracking the detected object as it moves through frames

: 관련정보 저장하는 동안, 프레임에서 이동하는 개체 추적

관련 영상

# Introduction

Basic steps for tracking an object

: Object Detection → Object Classification → Object Tracking

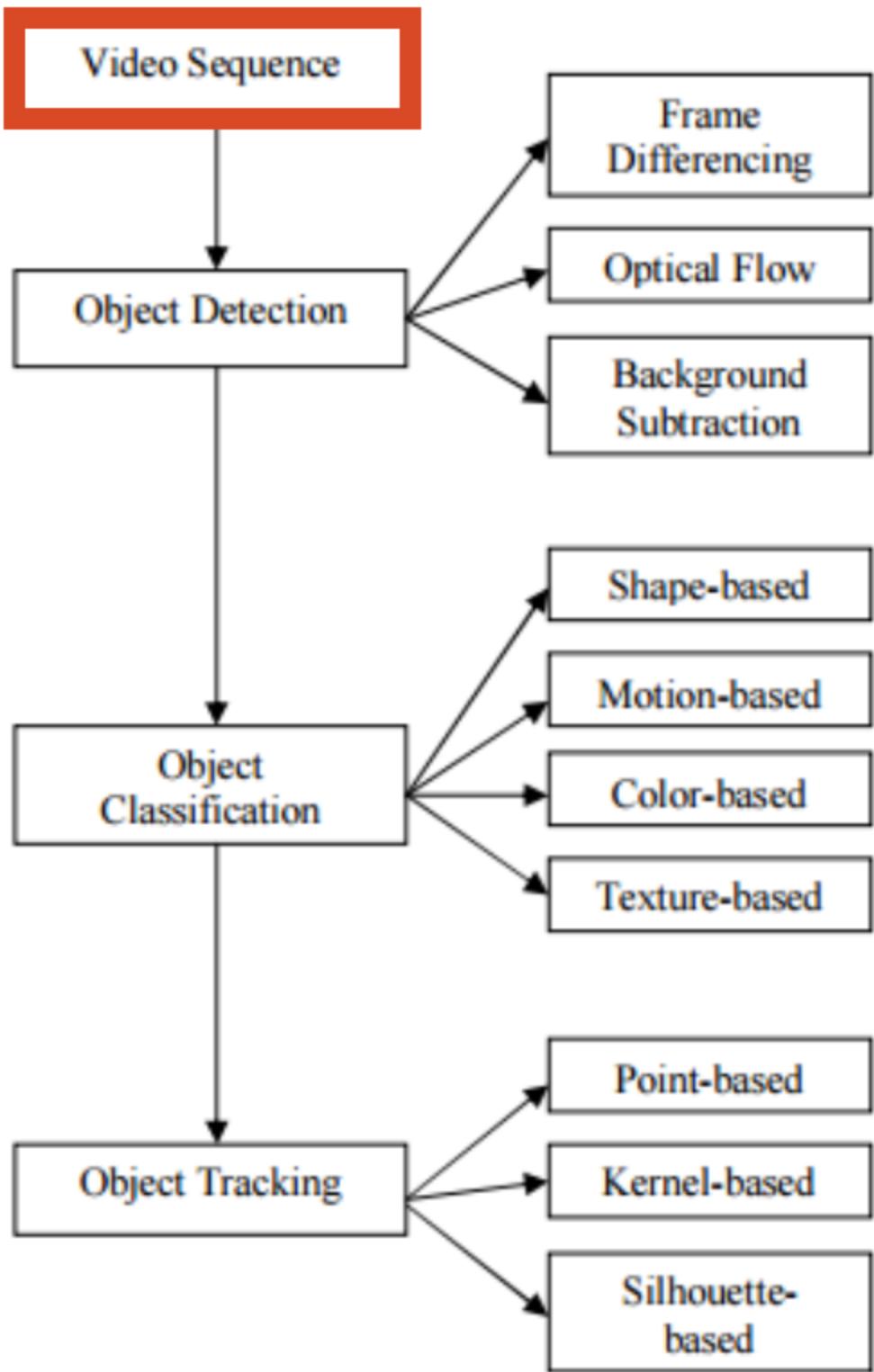
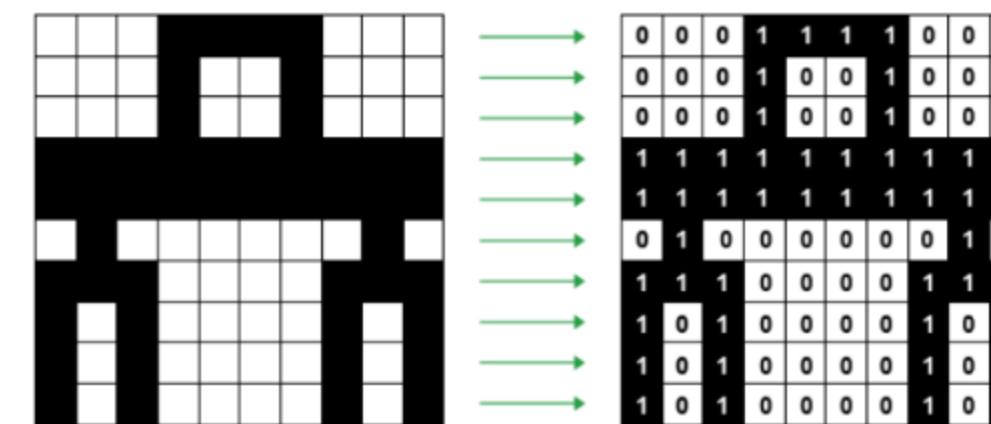


Fig 1: Basic steps for tracking an object [8]

\* 비디오로부터 입력받은 이미지는 두 가지 complementary sets of pixels로 나뉜다.  
처리된 이미지 결과는 **Binary Image**나 **Mask** 형태로 나타나게 된다.

1) Binary Image : 각 픽셀에 가능한 값이 2개인 이미지



2) Mask

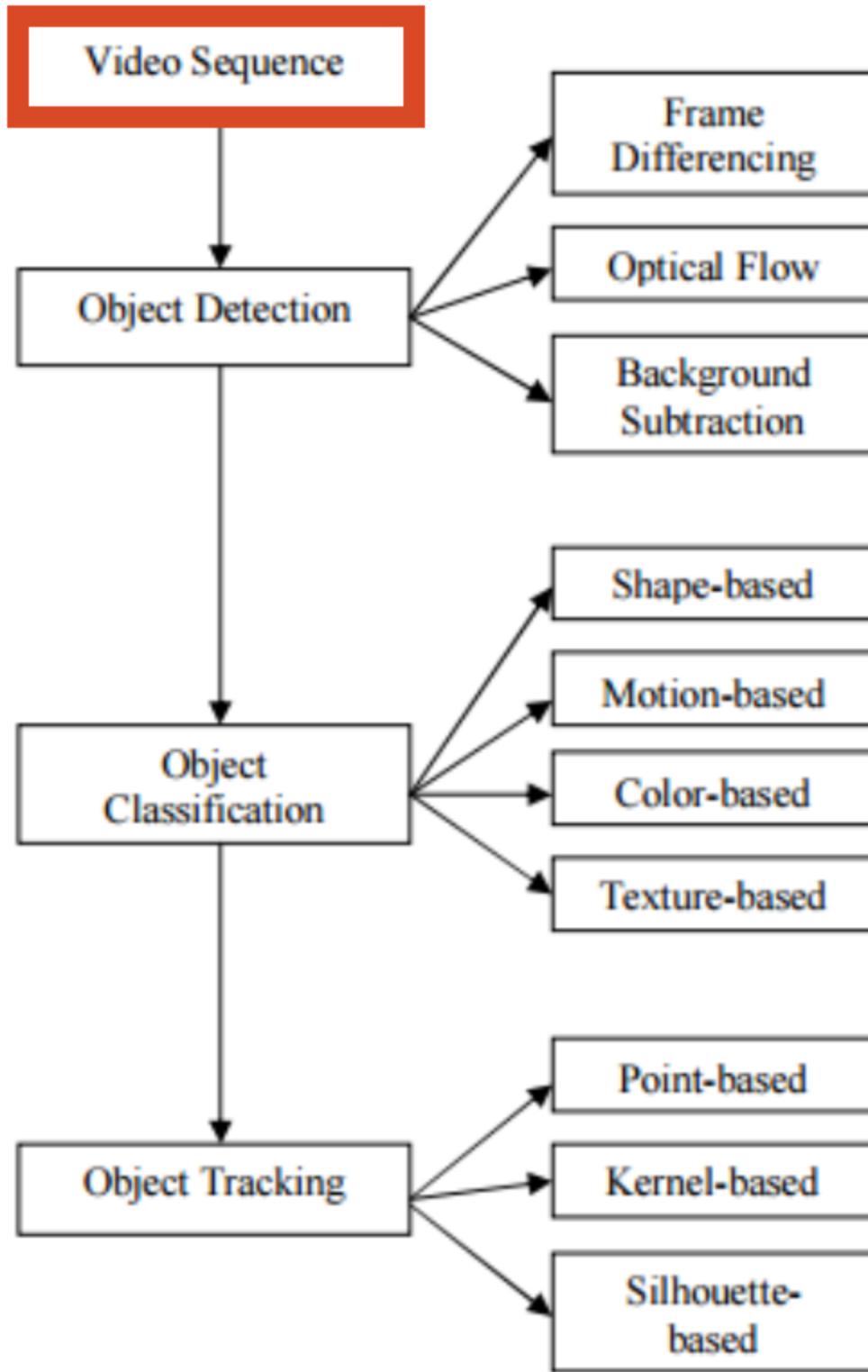
- **Binary Mask** : 사용자가 관심있는 영역의 픽셀값 255로 채우고 나머진 0으로!
- **RGB Mask** : 0이 아닌 픽셀값을 채워준다. 이 경우 사용자 관심영역이 multi class로 취급!



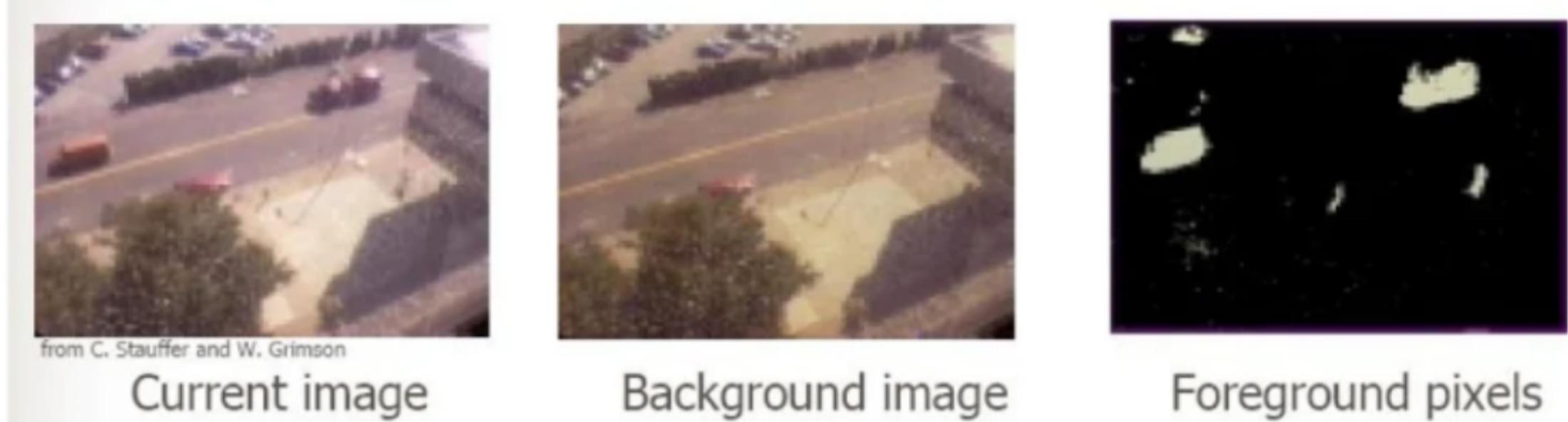
# Introduction

Basic steps for tracking an object

: Object Detection → Object Classification → Object Tracking



\* 비디오로부터 입력받은 이미지는 두가지 complementary sets of pixels로 나뉜다.  
처리된 이미지 결과는 Binary Image나 Mask 형태로 나타나게 된다.



1) **Foreground object:** 전경 객체에 대한 픽셀 정보를 담는다.

→ moving objects

2) **Background pixel:** 배경 객체에 대한 픽셀 정보를 담는다.

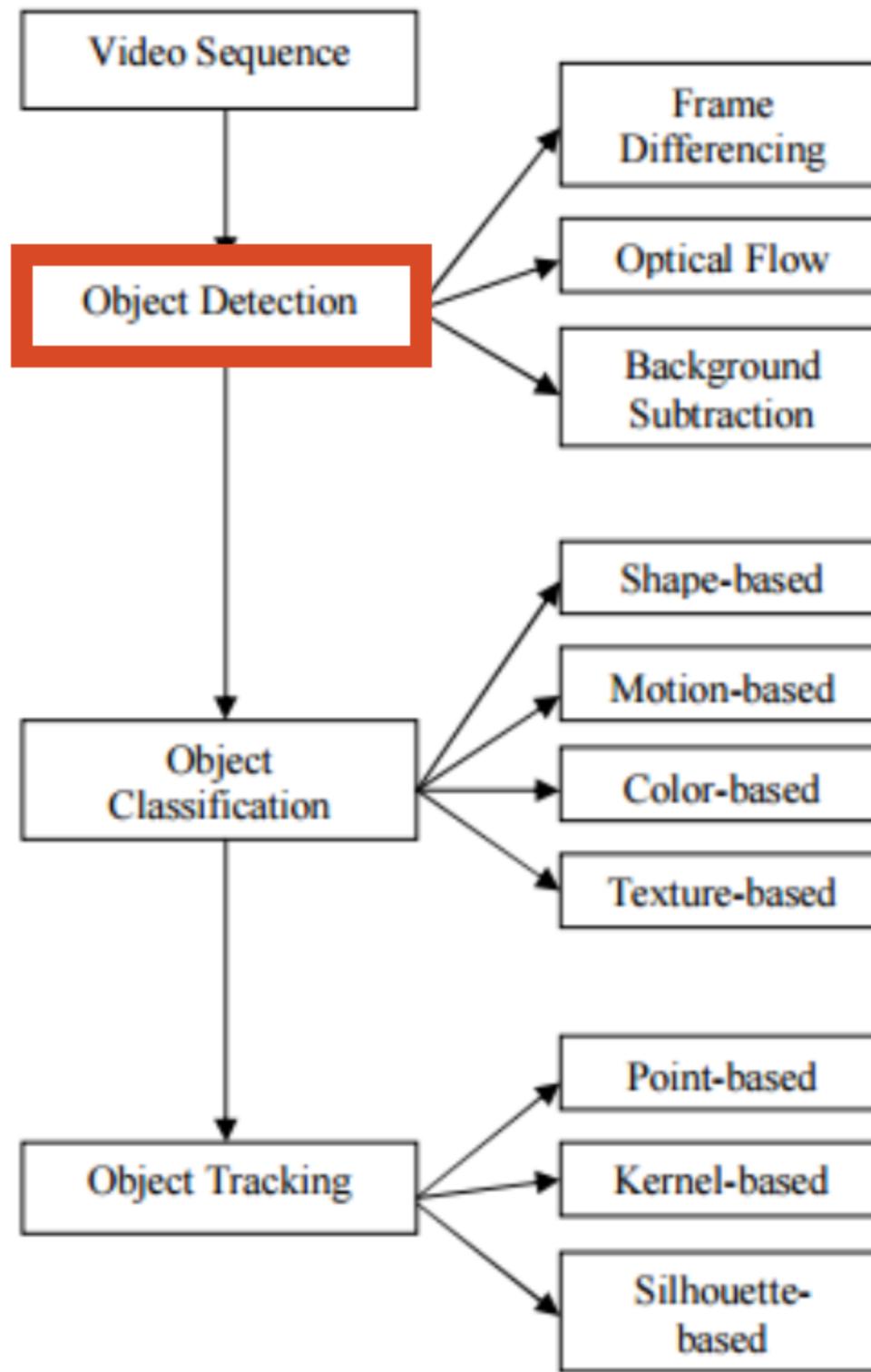
→ static scene

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking



## \* What is Object Detection?

: classification + localization이라고 생각하면 된다.

즉, 어떤 이미지가 입력되었을 때, 그 결과가 해당 객체의 bounding box를 그려주고, 해당 box 안의 객체가 무엇인지 분류해주는 것을 의미

## Classification



## Object Detection

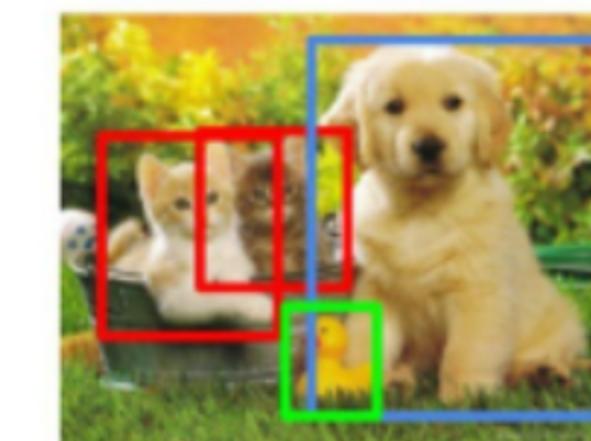
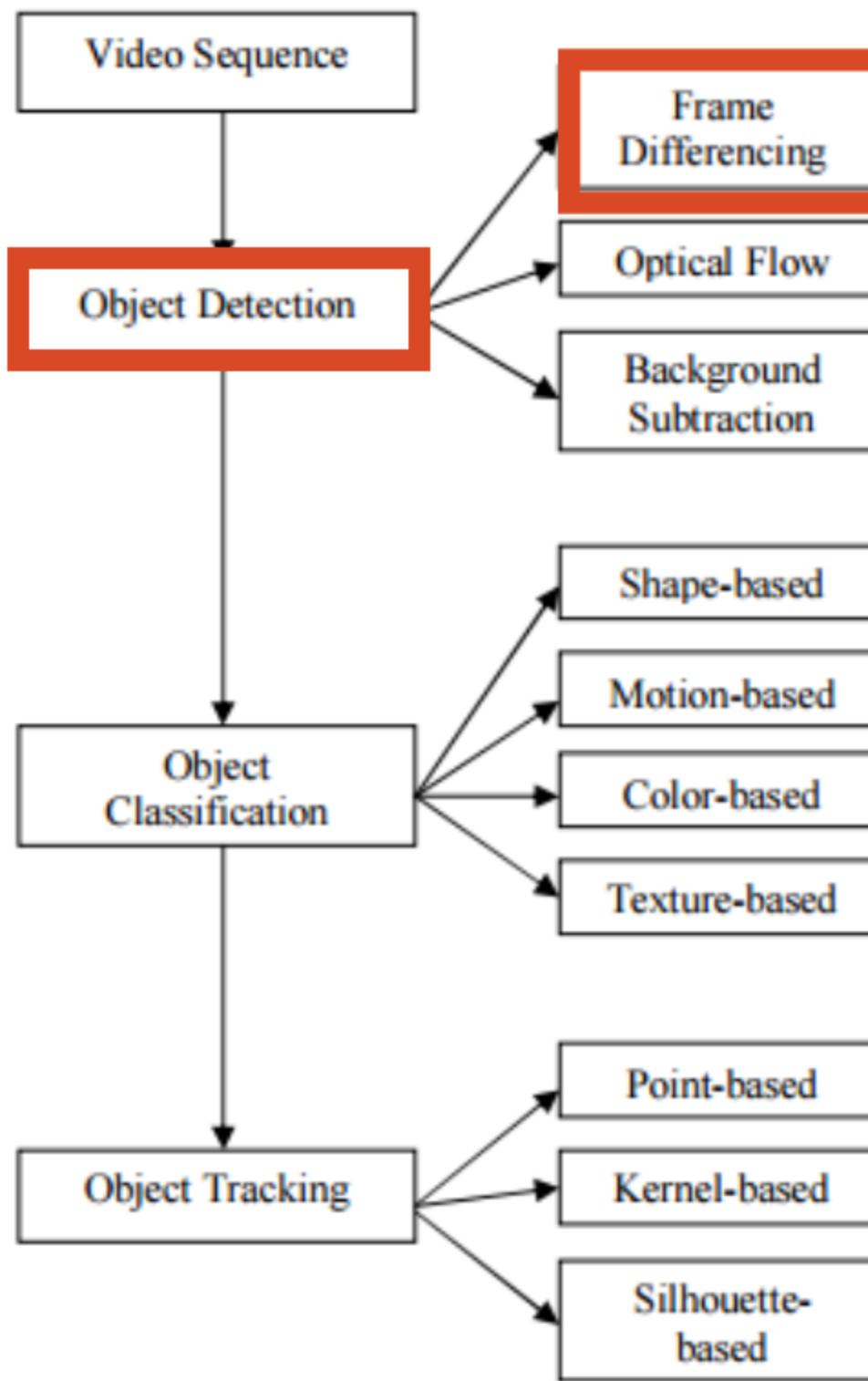


Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking



## \* Frame Differencing

두 개의 연속되는 이미지 사이의 차이를 계산함으로써 움직이는 객체가 존재한다고 판단하는 방법이다. 이미지 사이의 차이란, 결국 시간에 따라 변하기 때문에 시간변수  $t$ 를 사용하여, background를 제거하고, foreground pixels을 threshold하여 이미지를 개선한다. 이는 제일 쉽고 간단한 구현방법이고 empty phenomenon이 있을 때 적용하기 좋은 알고리즘이다.

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking

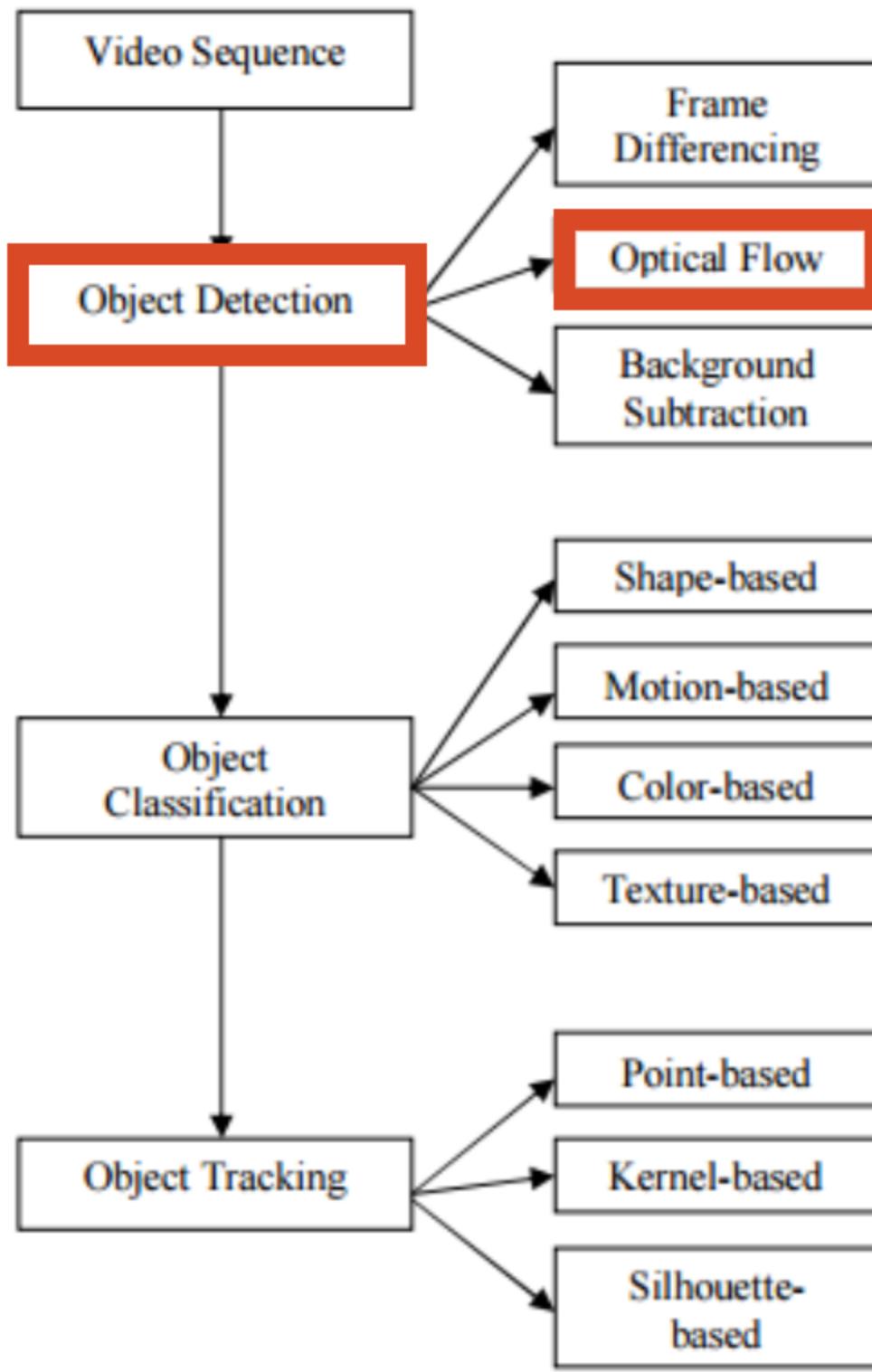
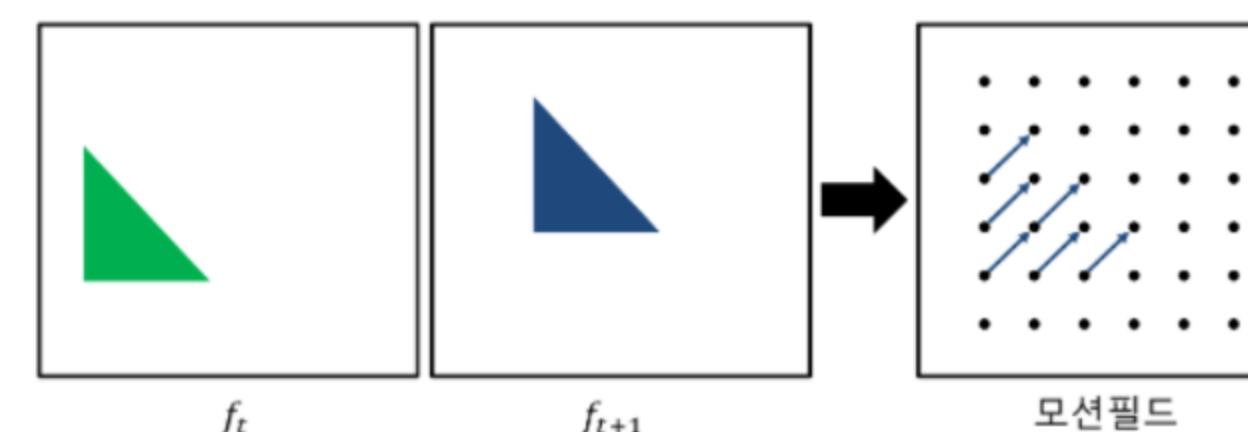


Fig 1: Basic steps for tracking an object [8]



## \* Optical Flow

motion field을 사용하여, 이전 프레임과 현재 프레임의 차이를 나타내는 관계, 각 픽셀의 이동을 계산하여 추출하고(=모션 벡터) 이를 통해 움직임을 구별하여 움직임 객체를 감지하는 원리



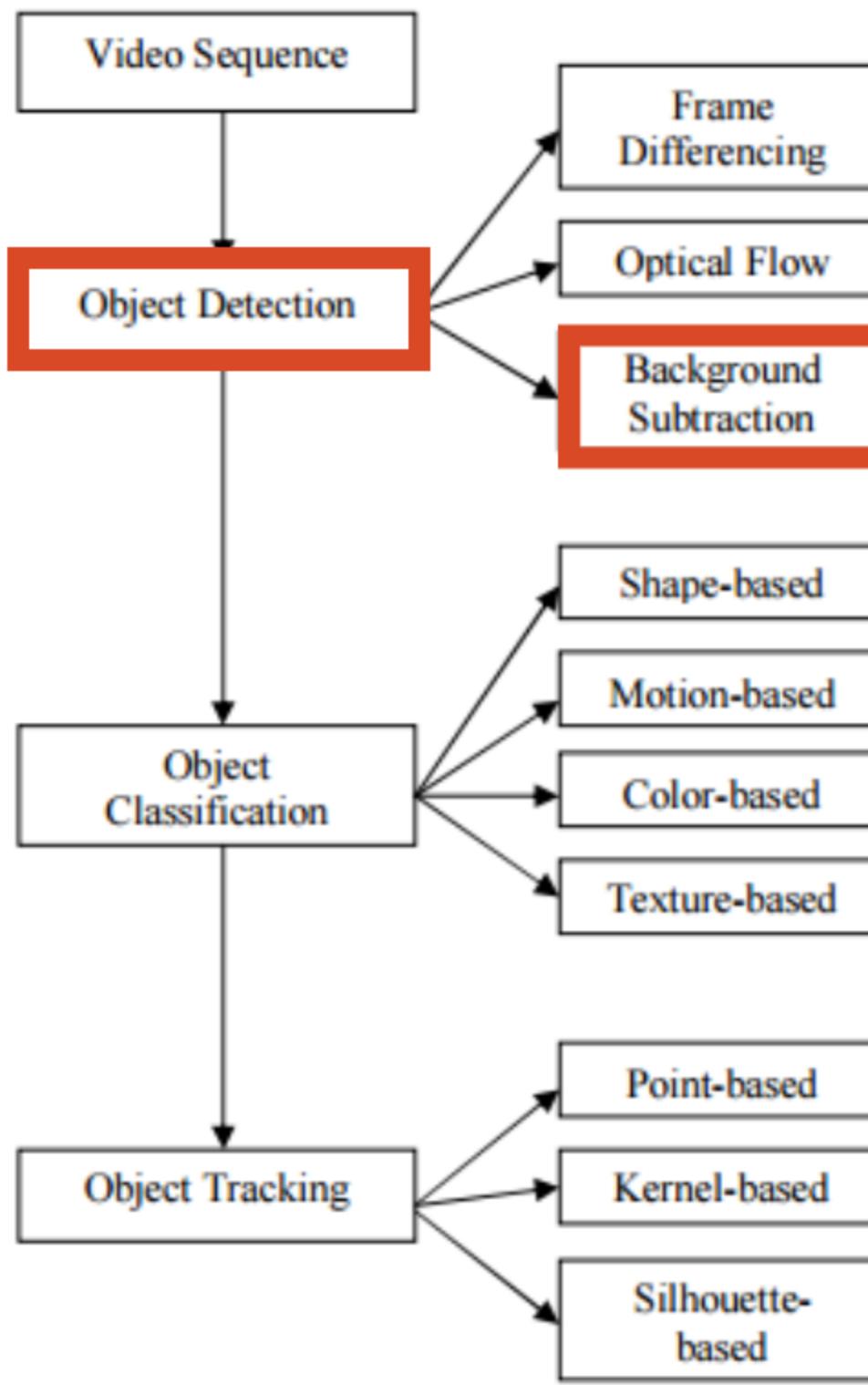
$f(y, x, t)$	$f(y, x, t + 1)$
11111111111111	11111111111111
11111111111111	11111111111111
11111111111111	11115111111111
11151111111111	11116511111111
11165111111111	11117651111111
11176511111111	11118765111111
11187651111111	11111111111111
11111111111111	11111111111111

$v = (v, u)$

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking



\* **Background Subtraction** Background Subtraction Algorithm Based Human Motion Detection

Background Subtraction을 하기 위해서는 먼저 Background modelling을 해야한다.  
background modelling must sensitive enough to recognize moving object!!!



Noisy image

Filtered by Mean Filter

Filtered by Median Filter

\* **Background modelling에 주로 사용되는 방법**

1) mean filter

2) median filter

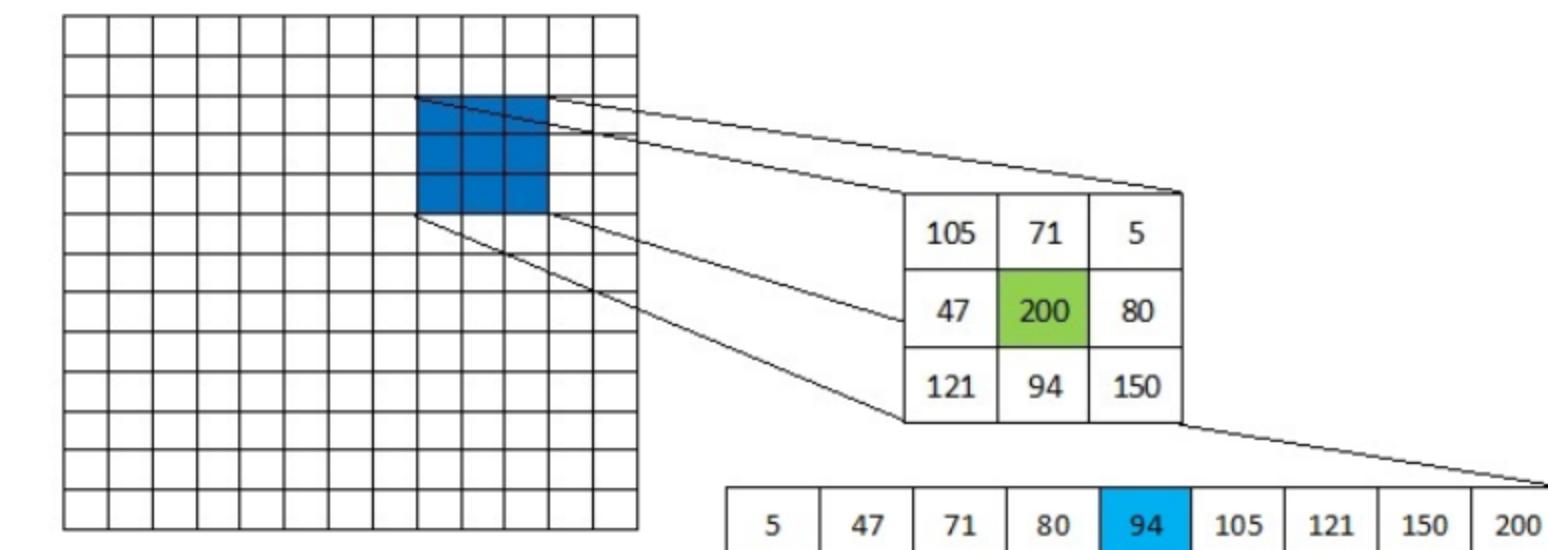


Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object  
: Object Detection -> Object Classification -> Object Tracking

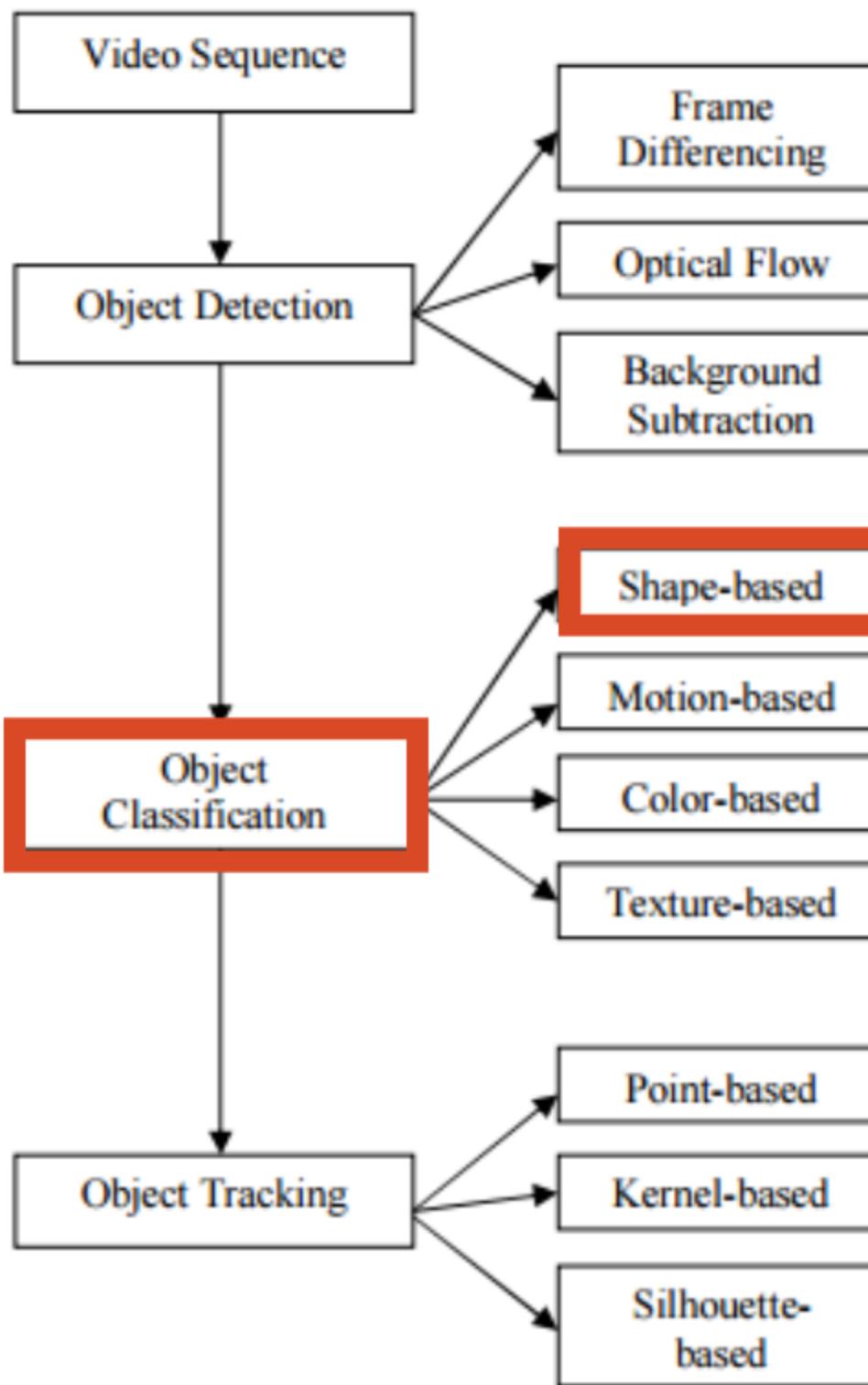
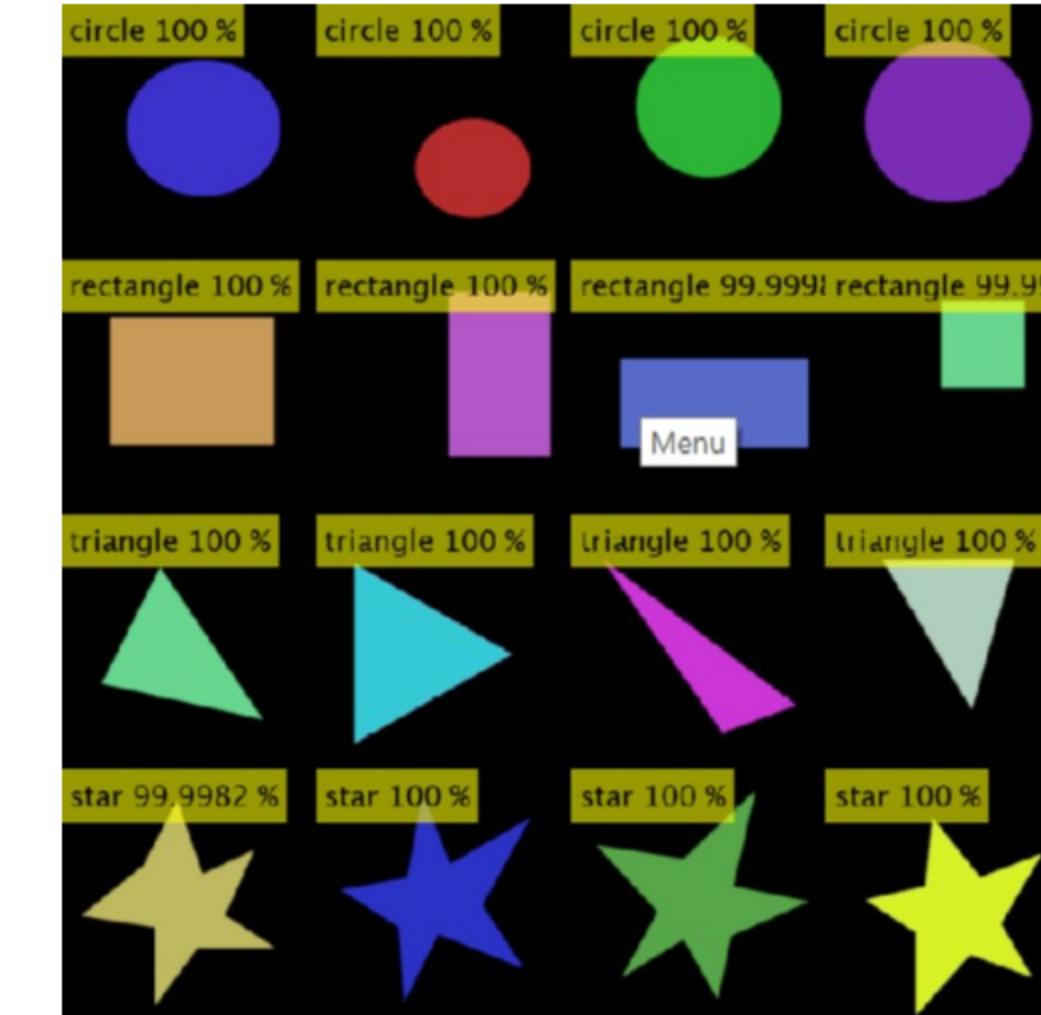


Fig 1: Basic steps for tracking an object [8]

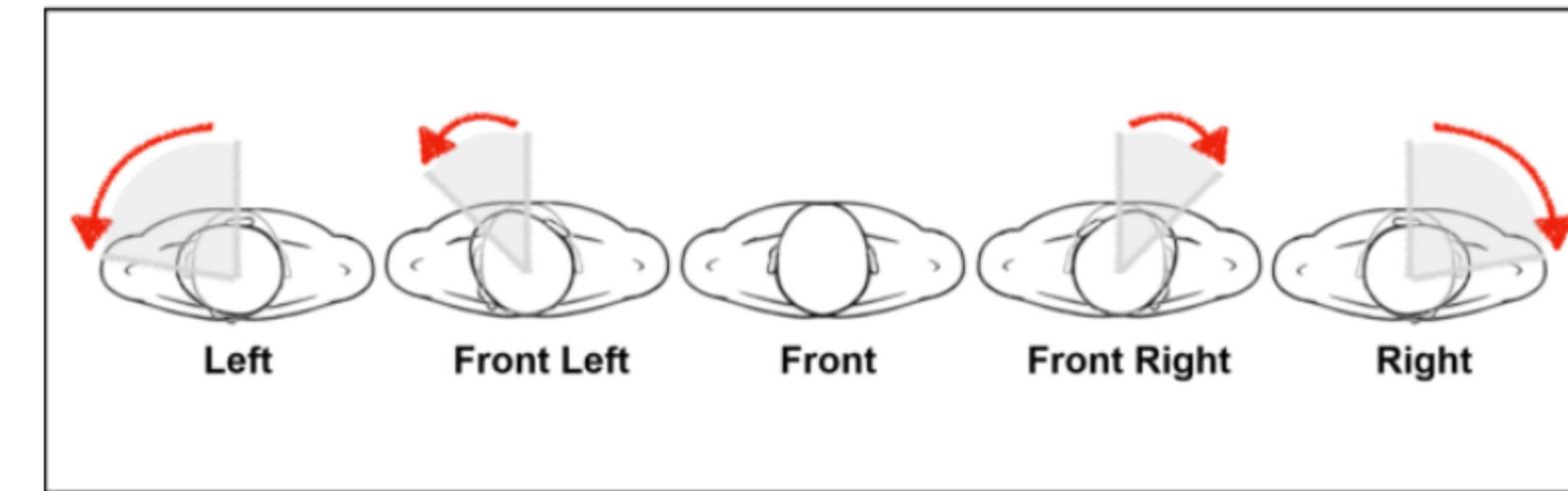
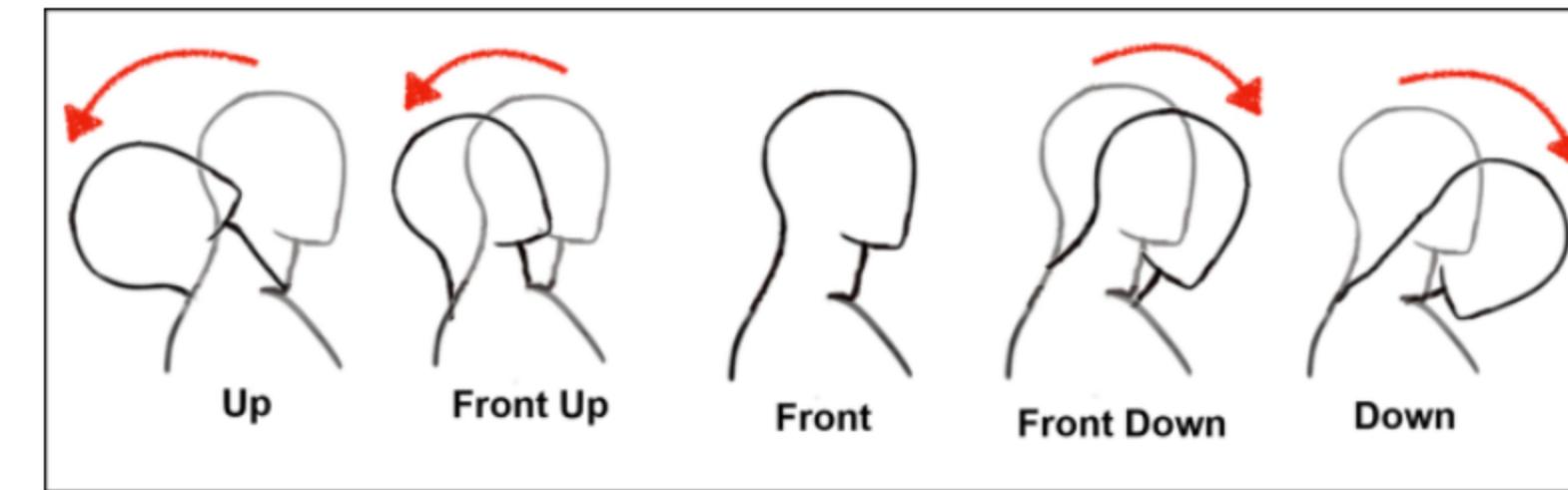
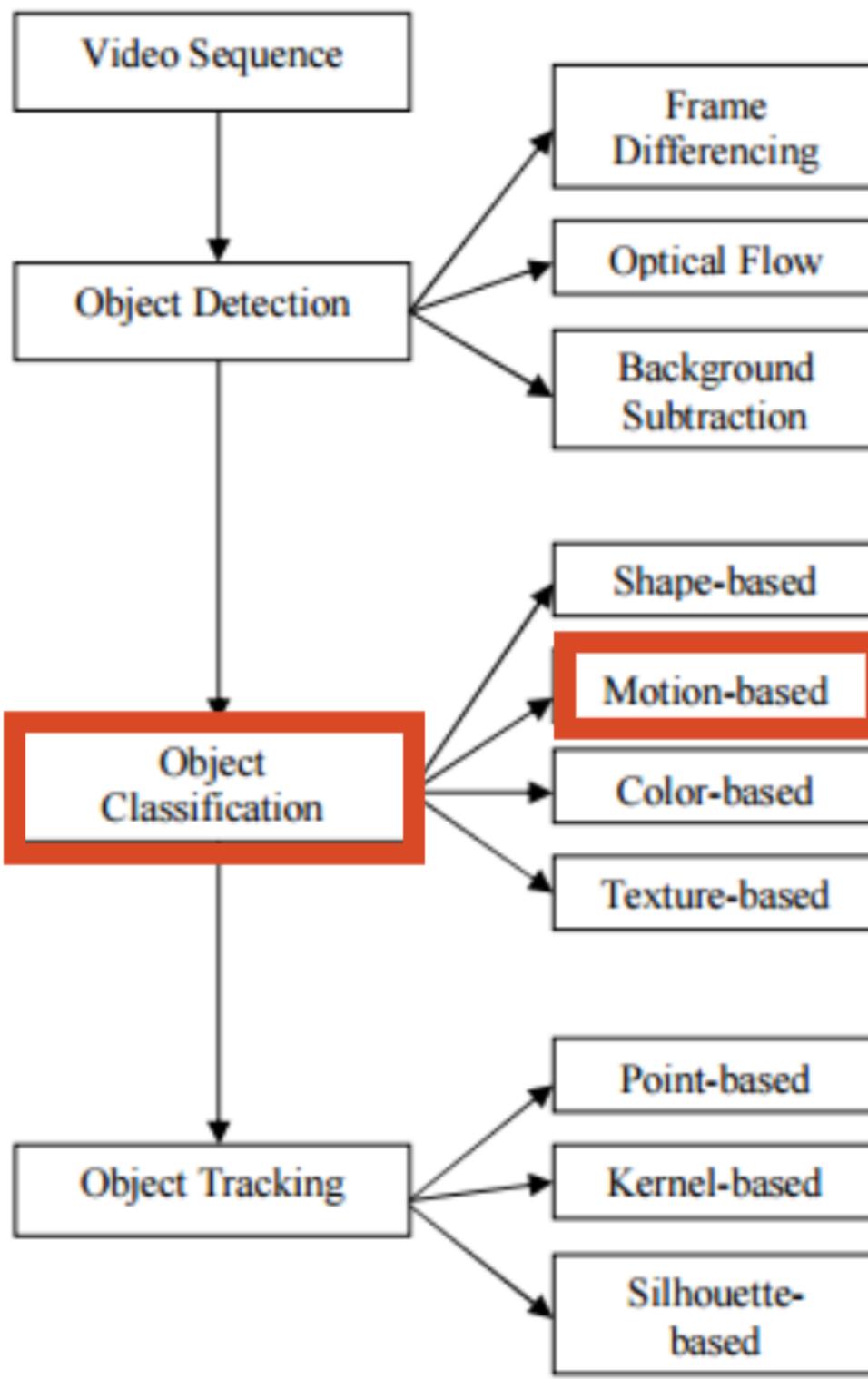


\* Shape-based Classification  
motion 영역의 shape을 통해, 이동 물체를 분류해낸다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking



## \* motion-based Classification

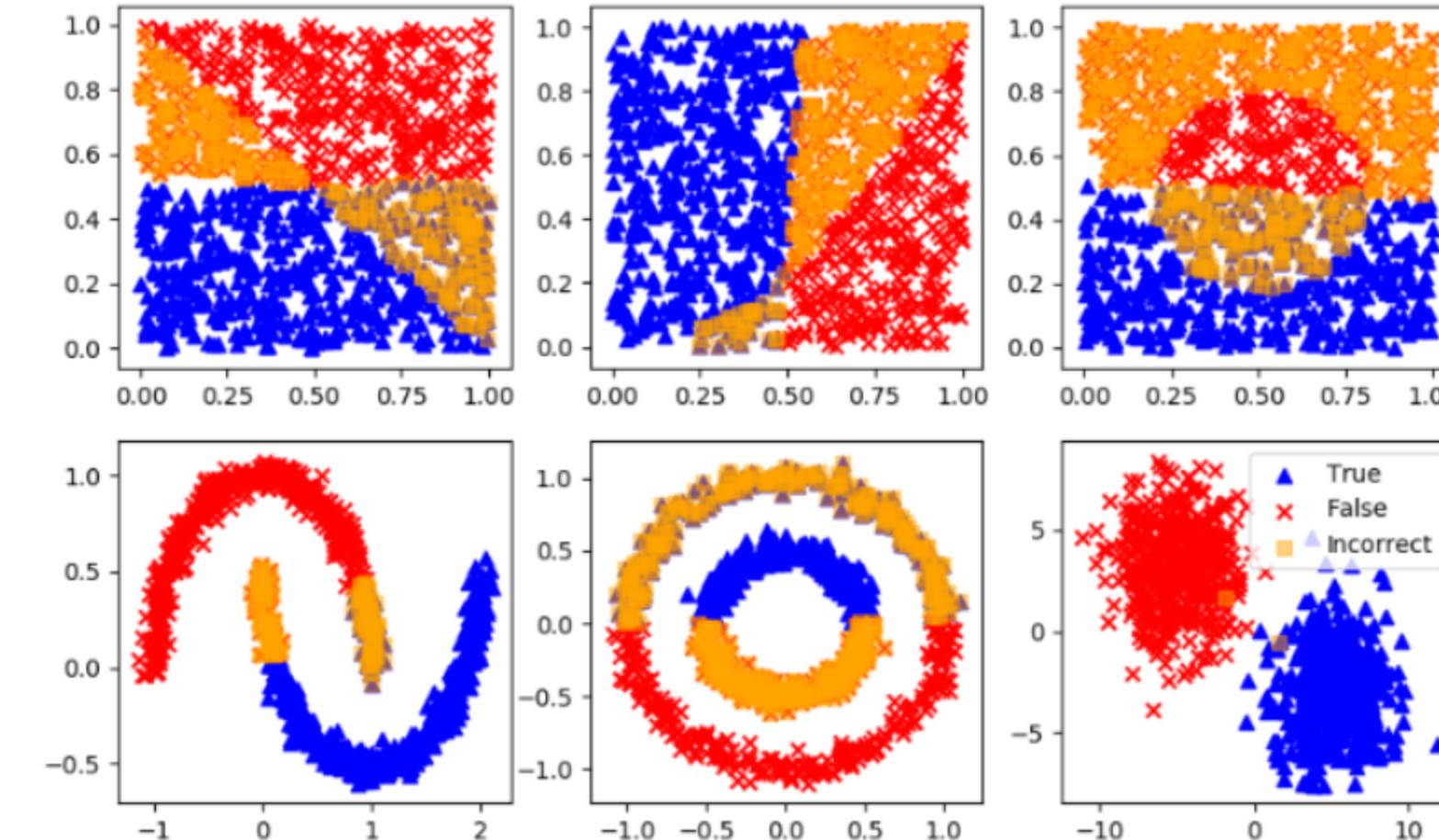
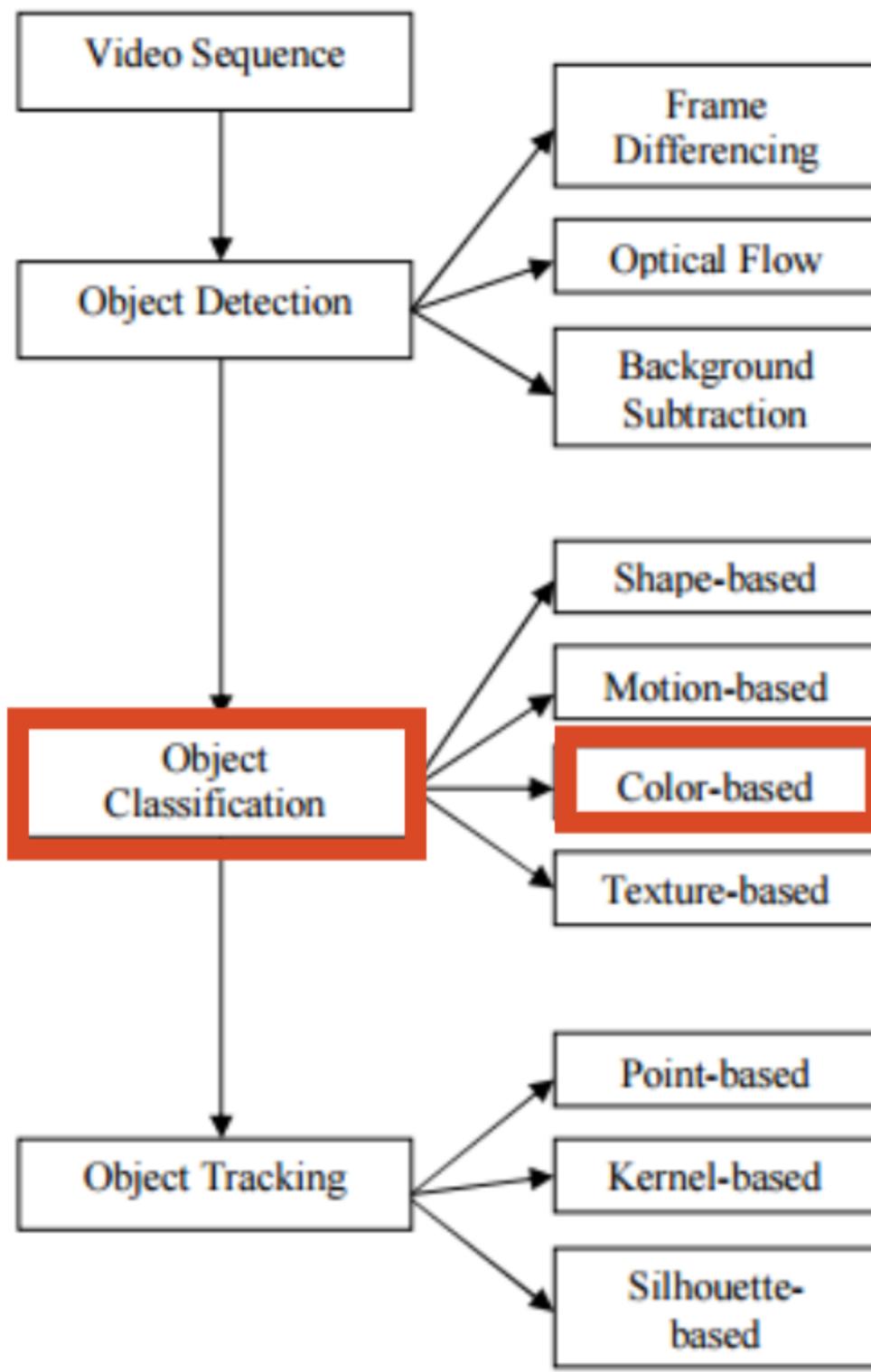
motion의 움직임을 통해, 물체를 분류하고, 움직이는 객체가 다른 요소와 연결관계가 있는지 분석하는데 유용할 수 있다.

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> Object Tracking



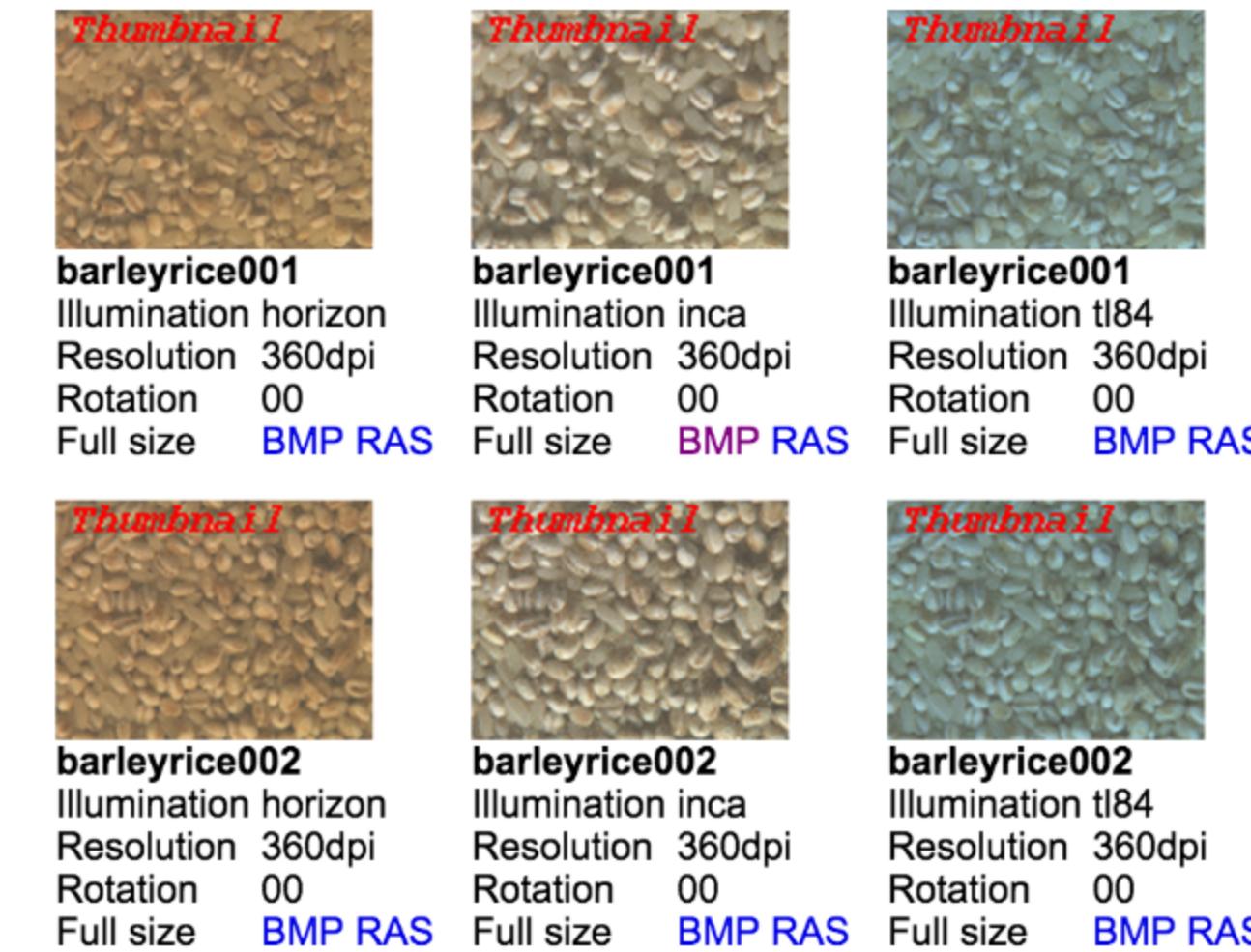
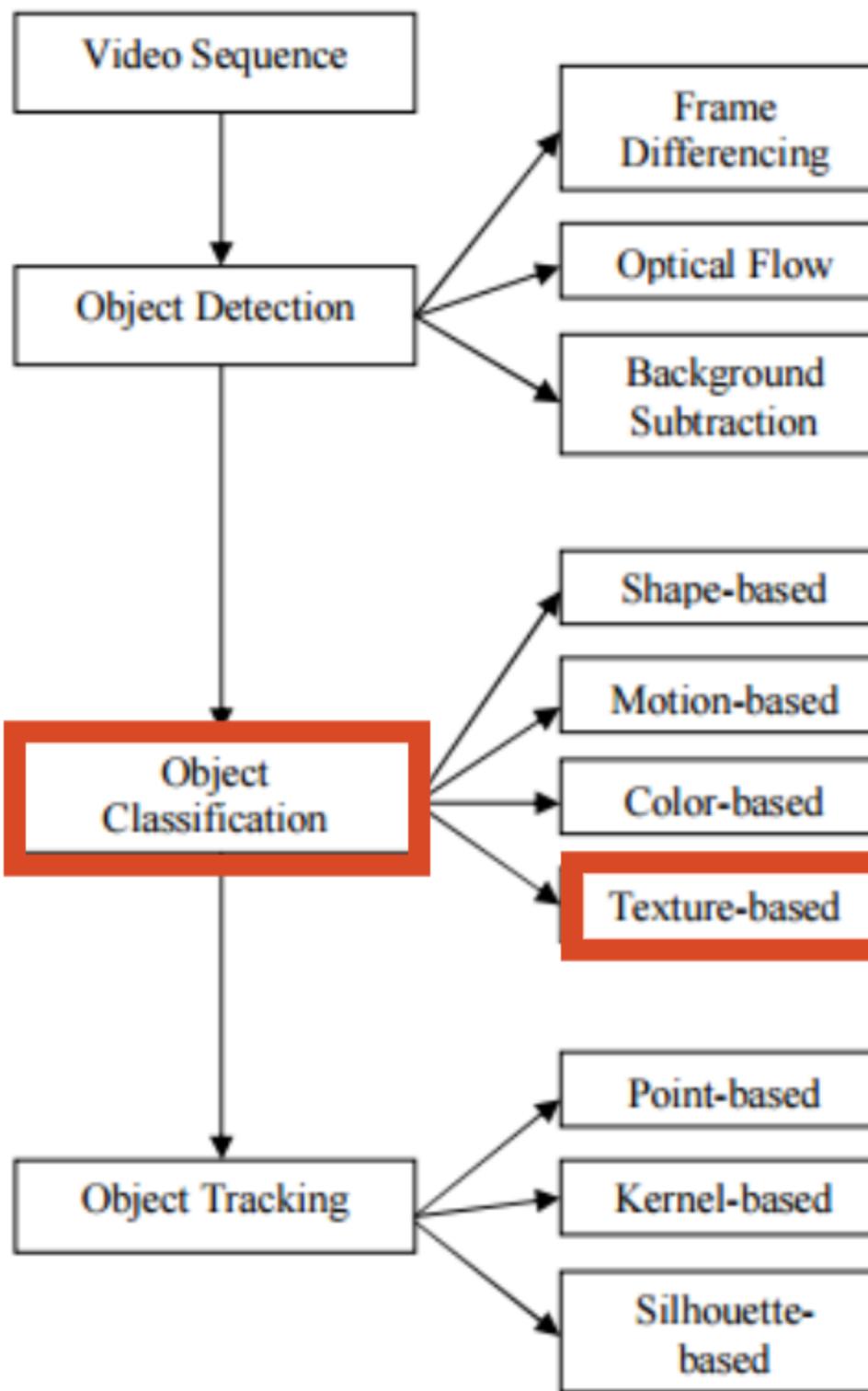
## \* color-based classification

색 정보를 이용하여 객체를 분류한다. 그러나, 이 경우 viewpoint 변화에 밀접하게 연결되기 때문에 항상 적절한 color 정보를 주는 것이 아님에 유의해야 한다. 가우시안 혼합모델이 background와 object에 대한 color distribution 의 예이다.

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object  
: Object Detection -> Object Classification -> Object Tracking



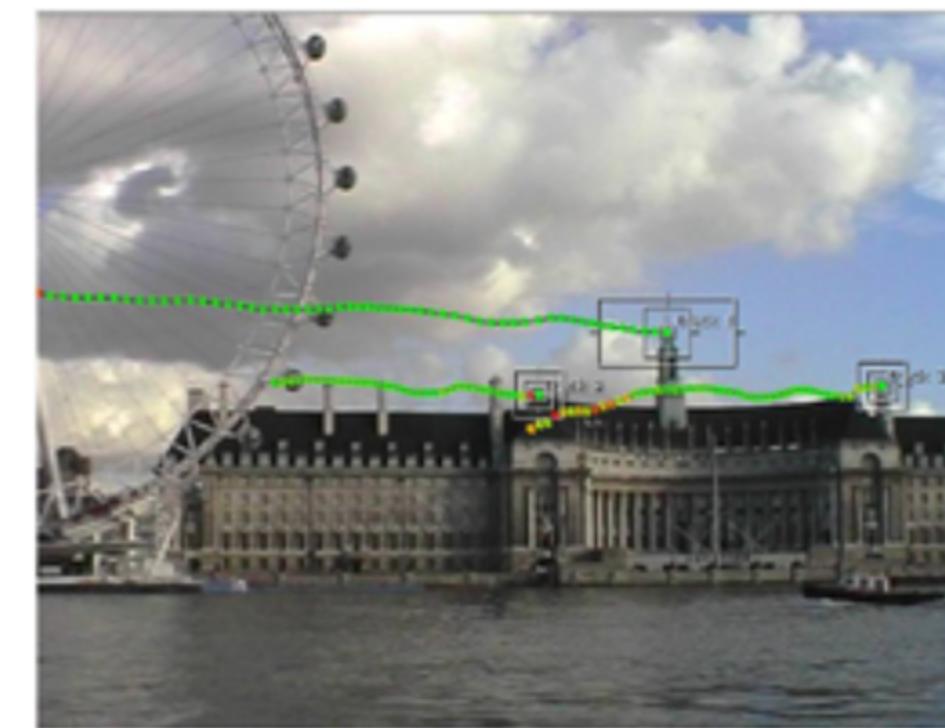
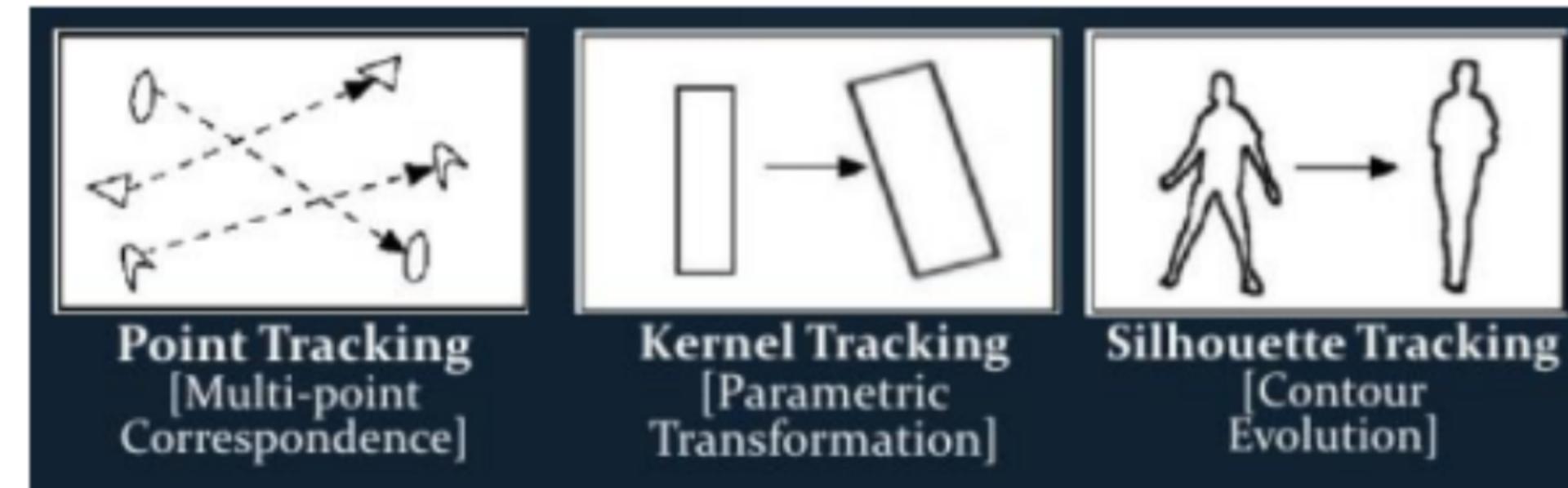
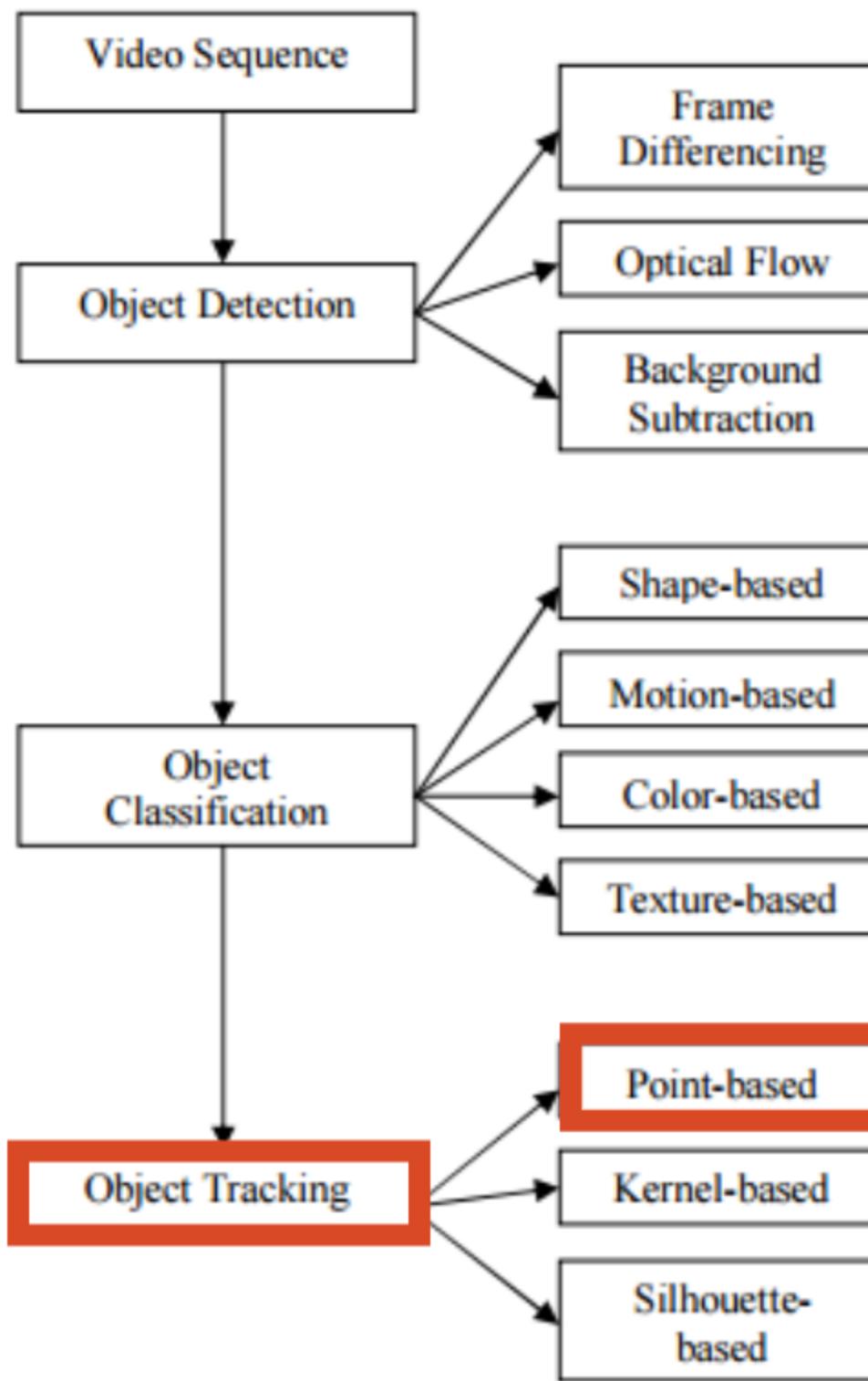
\* texture-based classification

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



\* point tracking

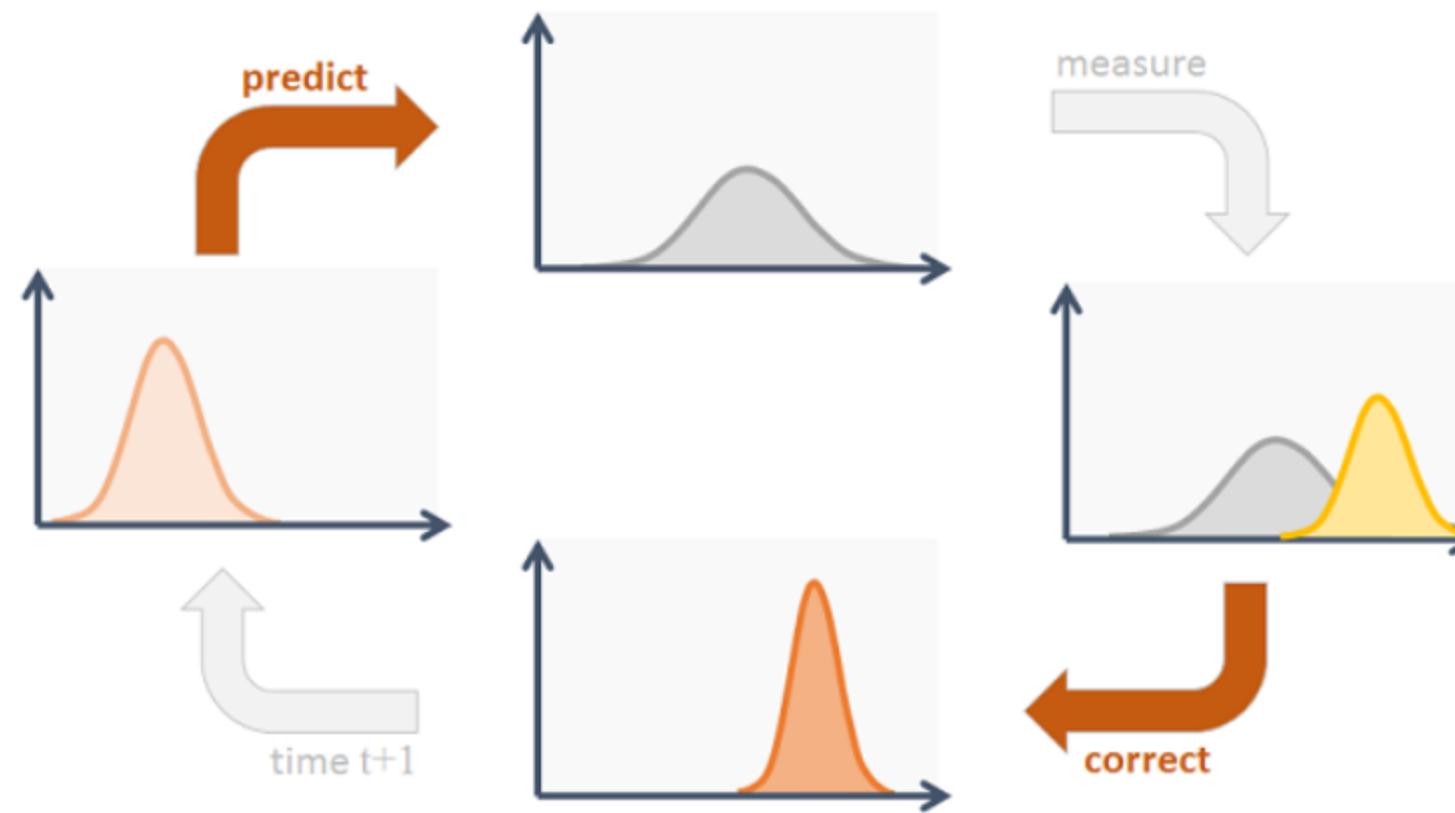
연속된 프레임에서 감지된 객체는 점으로 표시되며, 점을 연결하여 나타나는 방향성으로 객체가 움직일 것이라고 예측할 수 있는 방법이다.

Fig 1: Basic steps for tracking an object [8]

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



관련 영상

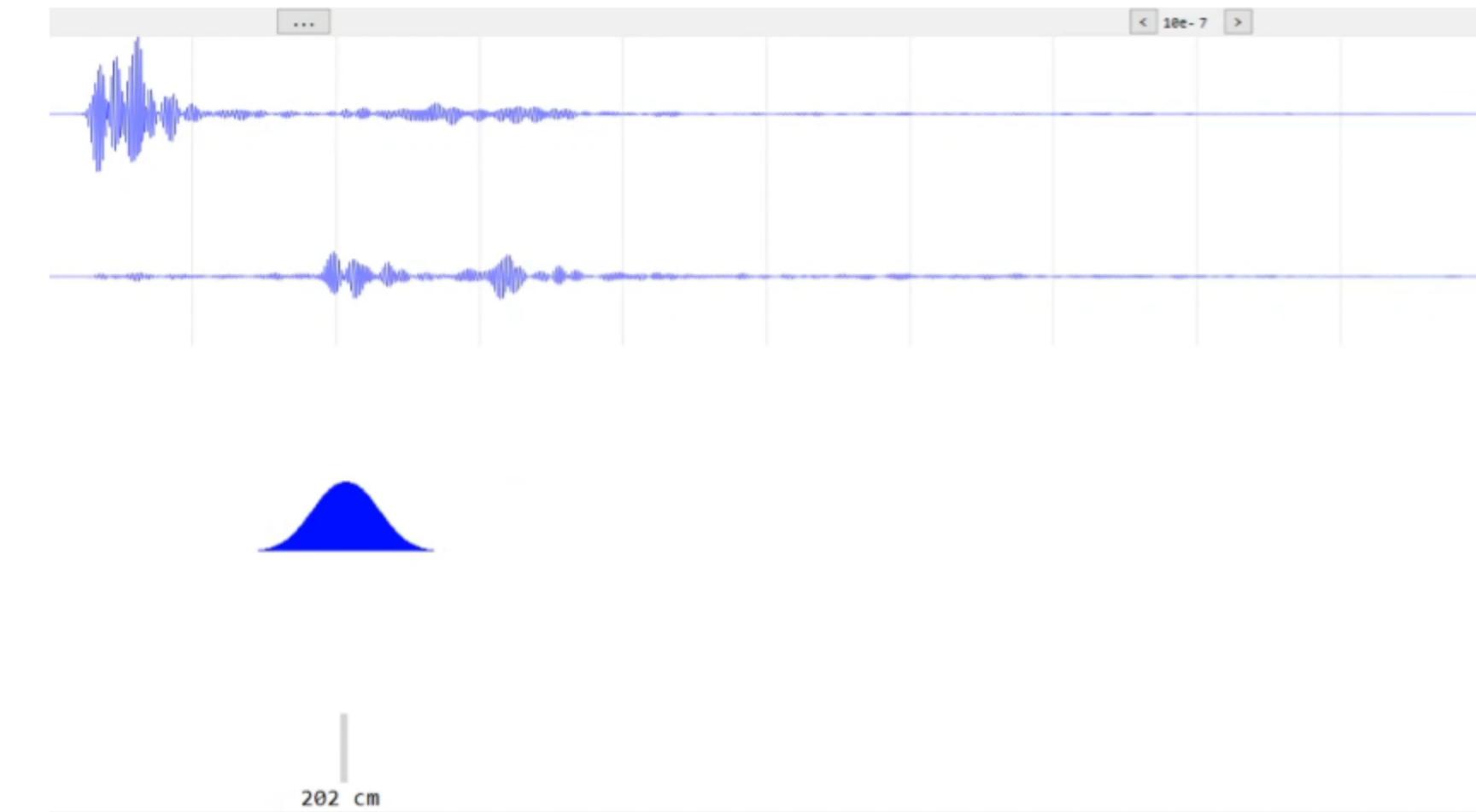
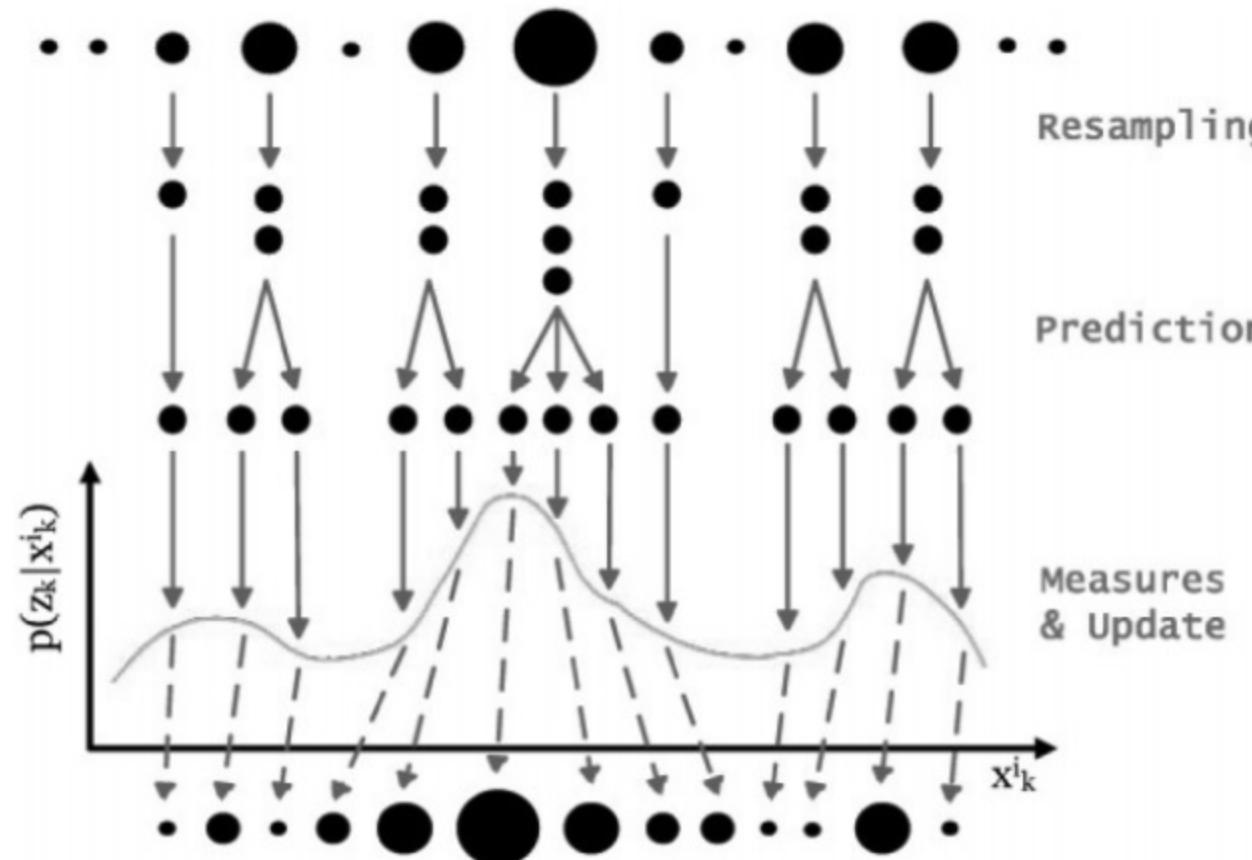
## \* Kalman Filter – point tracking method 1

- 컴퓨터 비전, 네비게이션, 경제학 등에서, 시계열 예측과 관련된 문제 해결에 주로 사용된다.
- Detection과 Prediction을 이용하여, 현재의 상황에 대해 추측하고자 한다.
- 재귀적으로 작동하며, real time으로 동작한다.
- 이전상태의 분포와 측정한 분포 사이에서 새로운 분포를 추정하는 것과 같다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



## \* Particle Filter – point tracking method 2

- kalman filter와 마찬가지로, 노이즈가 있는 환경에서 측정된 데이터를 필터를 사용하여 실제 위치를 추정하는 도구이다.
- 보통 가우시안 분포가 아닌 측정 데이터를 다루기 위해 사용된다.
- 검은색 동그라미는 particle을 의미하며, 동그라미가 다른 것은 각 동그라미의 weight가 다른 것을 표현한 것이다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**

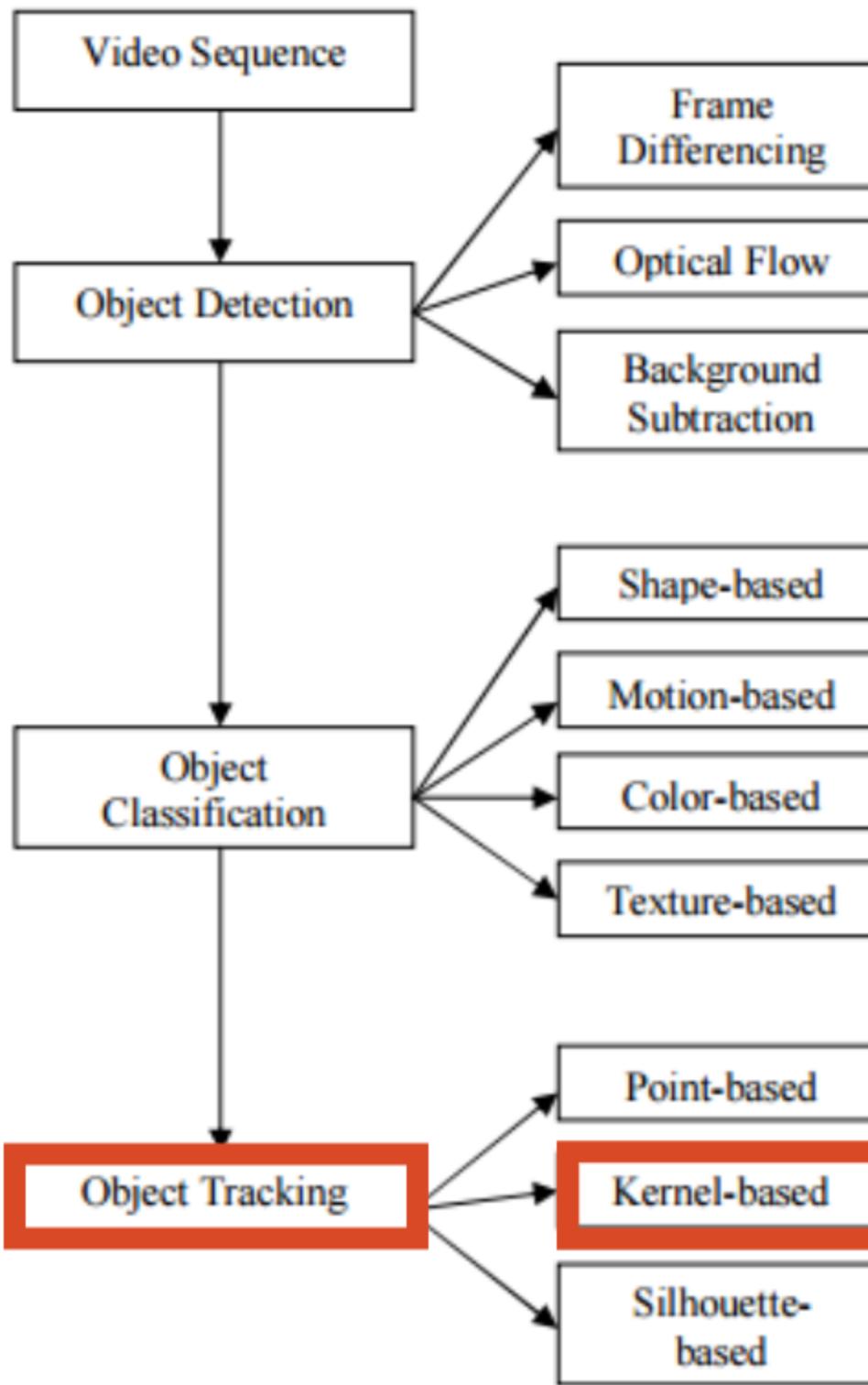
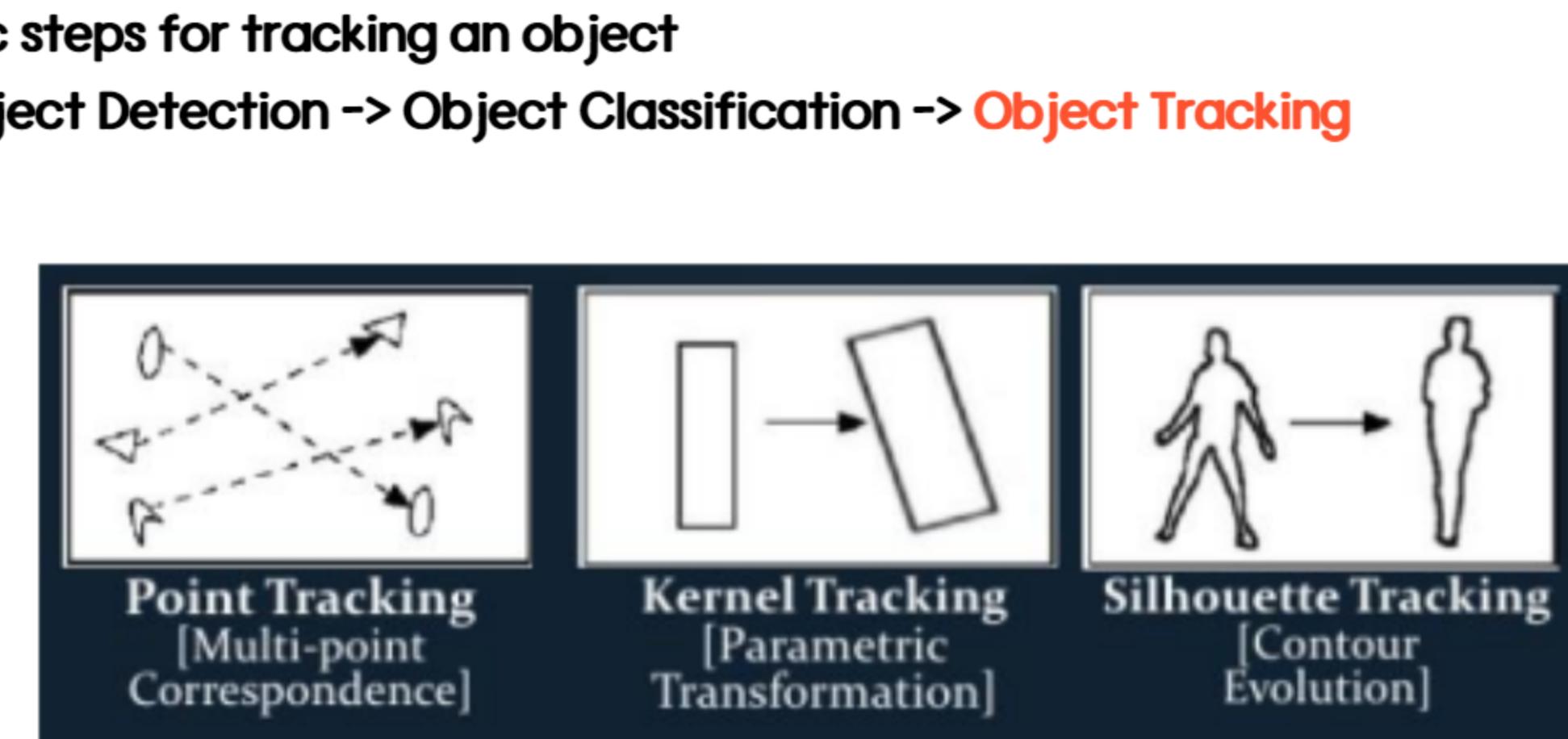


Fig 1: Basic steps for tracking an object [8]



\* **kernel tracking**

커널은 객체의 모양을 의미하며, 직사각형의 템플릿을 의미한다. 객체는 연속 프레임에서 커널의 움직임을 계산하면서 추적한다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



image

템플릿

Detected Point



메시 얼굴 검출

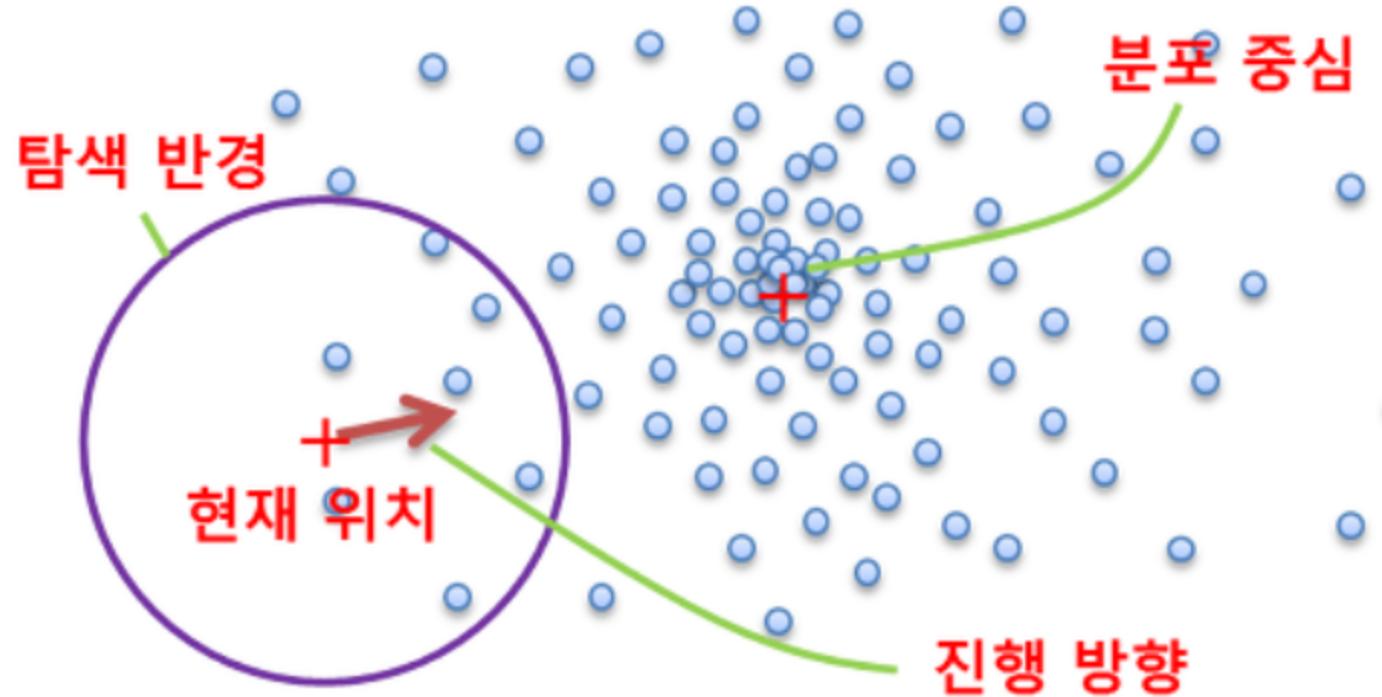
## \* Simple Template matching – kernel tracking method 1

- 템플릿 매칭은 입력 영상에서 템플릿 영상과 일치하는 부분을 찾는 기법이다.  
템플릿=커널로 생각해도 무방
- ccoff 연산을 이용하면, 밝기보정을 통해 매칭이 가능하기 때문에,  
노이즈나 밝기 변화에 강하다.
- 픽셀값들을 곱하고 더해서 유사도를 판단하는 방식이기 때문에,  
회전, 크기변환에는 민감하다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



\* Mean Shift method – kernel tracking method 2

- 어떤 데이터 분포의 peak 또는 무게중심을 찾는 방법으로, 현재 자신의 주변에서 가장 데이터가 밀집된 방향으로 이동하다보면, 분포의 중심을 찾는다는 개념

- mean shift algorithm의 과정)

1. 현재 위치에서 반경  $r$  이내의 데이터들을 구한다.  
 $(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots$
2. 1에서 구한 값들의 무게중심 좌표로 현재 위치를 이동시킨다.
3. 1과 2의 과정을 반복하면서 위치변화가 거의 없을 때까지 반복한다.



# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**

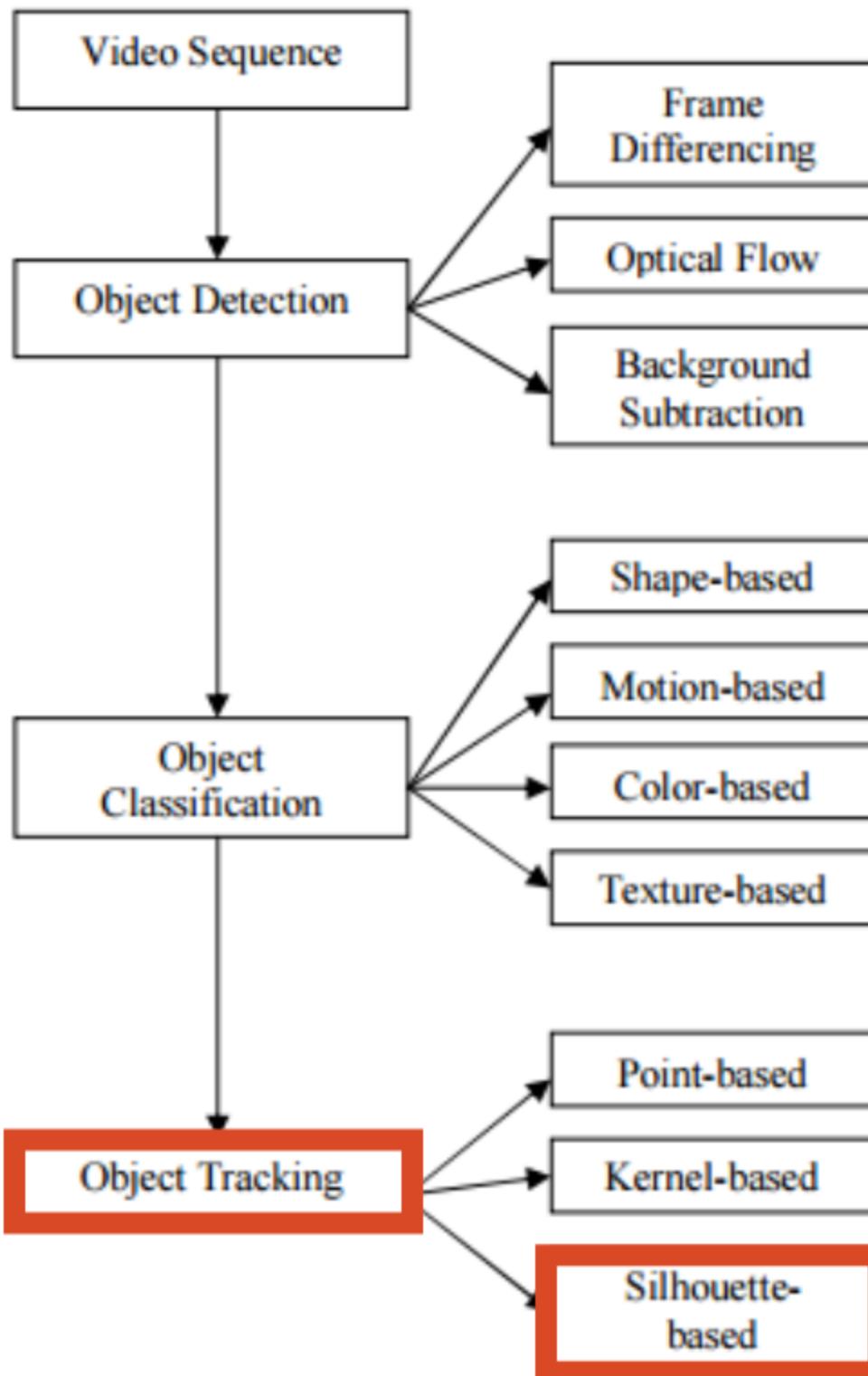
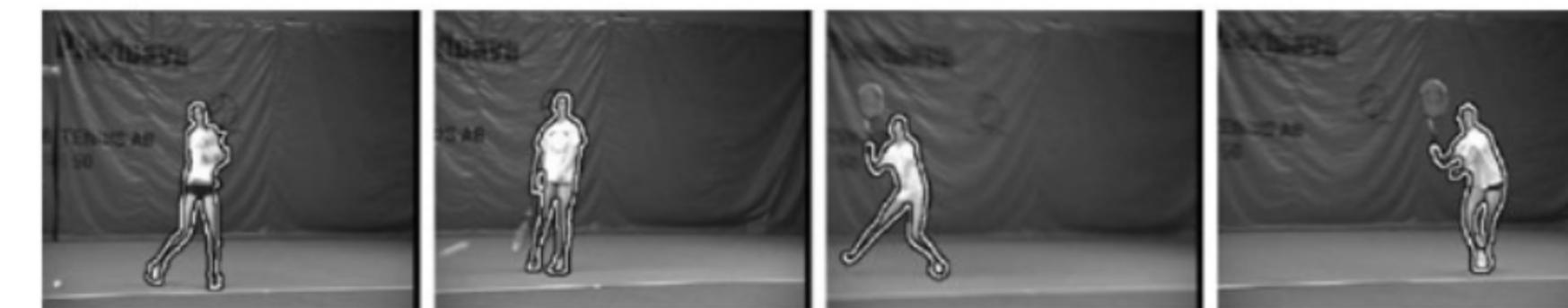
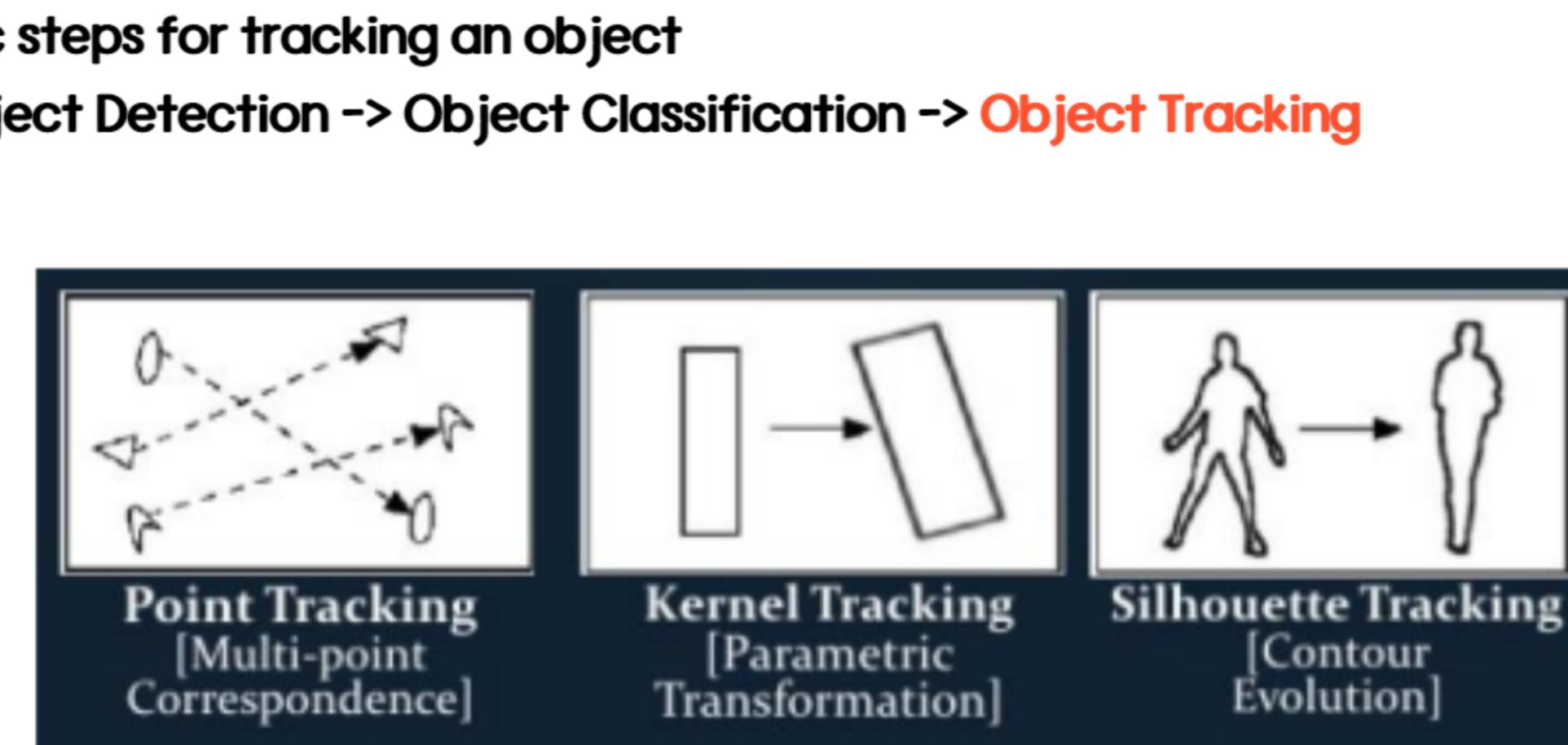


Fig 1: Basic steps for tracking an object [8]



(a)



(b)

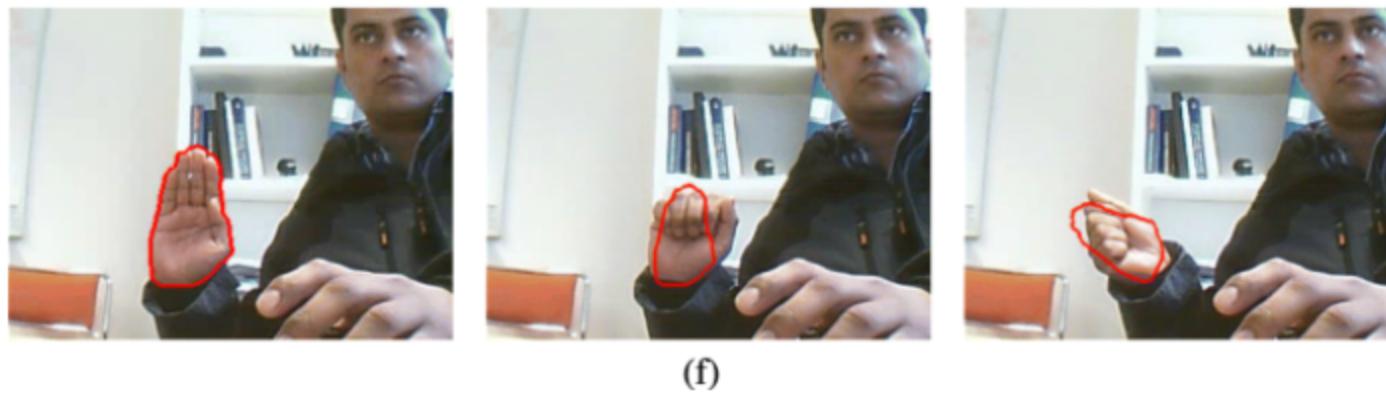
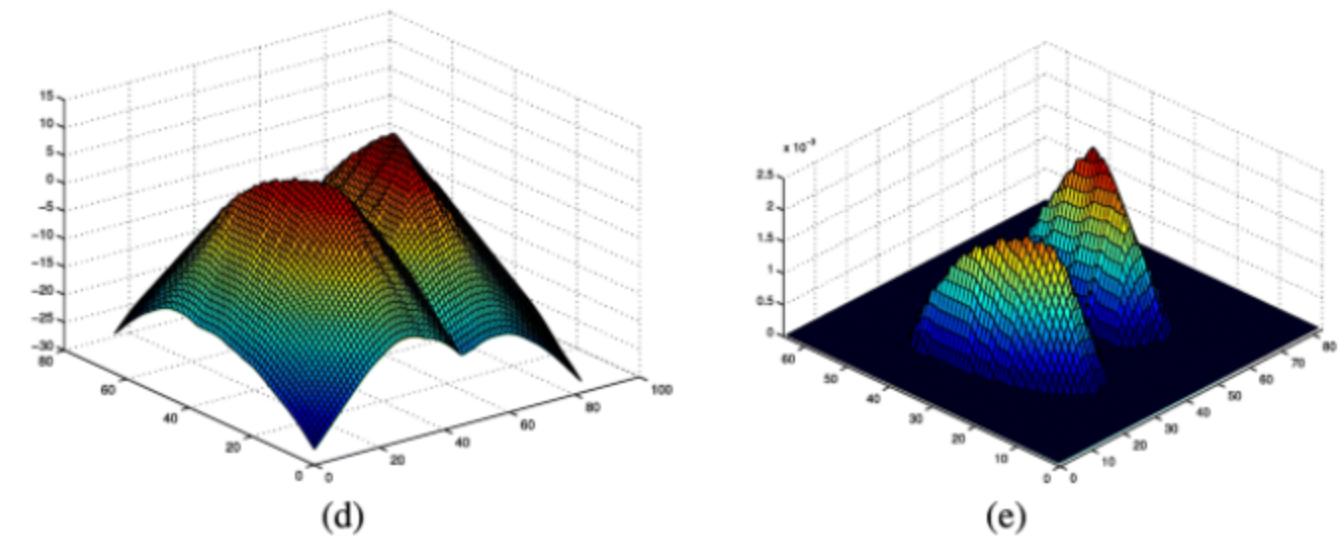
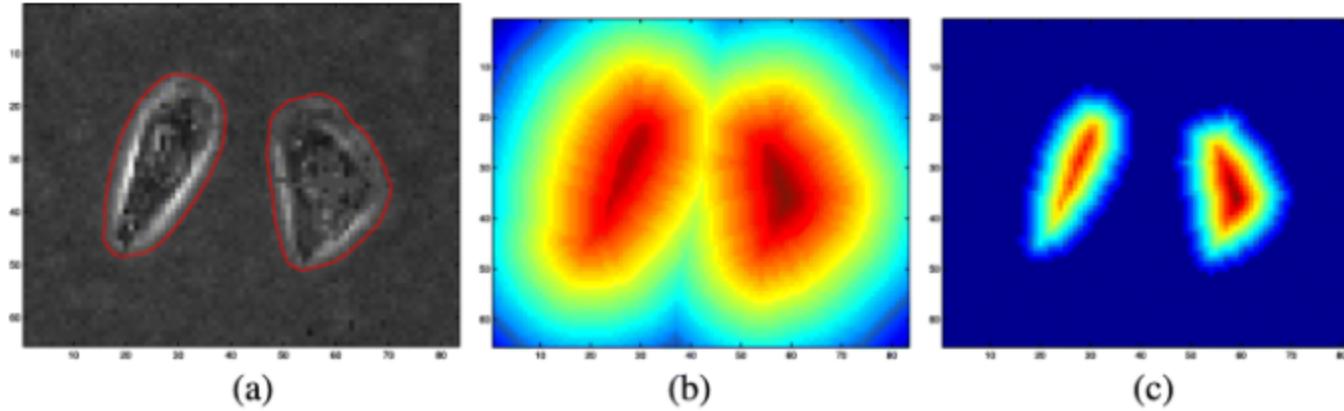
\* **silhouette tracking**

주로 손, 손가락, 어깨와 같은 복잡한 모양을 추적할 때 사용한다.

# Object Detection

Basic steps for tracking an object

: Object Detection -> Object Classification -> **Object Tracking**



## \* Contour tracking – silhouette tracking method 1

- 이전 frame의 기본 윤곽선을 새로운 프레임에 계속 반복적으로 그리면서 표현
- 다양한 객체 모양을 처리할 수 있다는 점의 유연성이 강점이다.
- two different approaches
  1. state space models to model the contour shape and motion
  2. directly evolves the contour by minimizing the contour energy