

---

# First-order-model

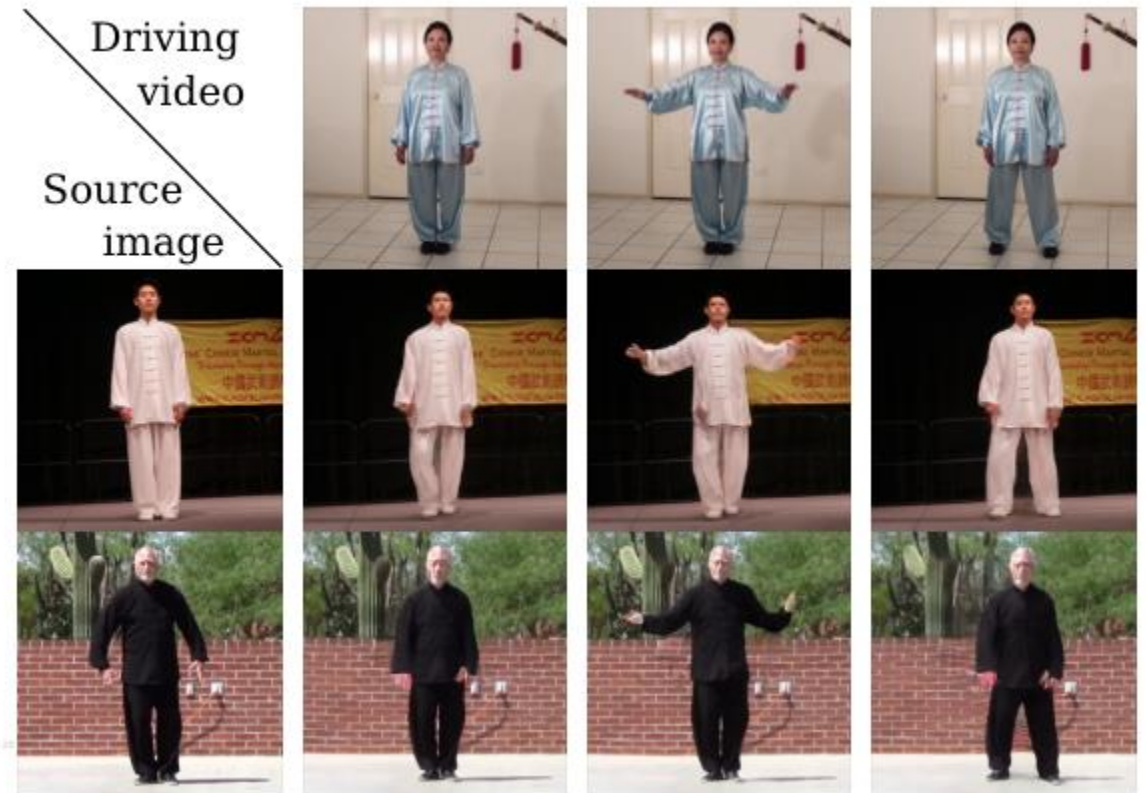
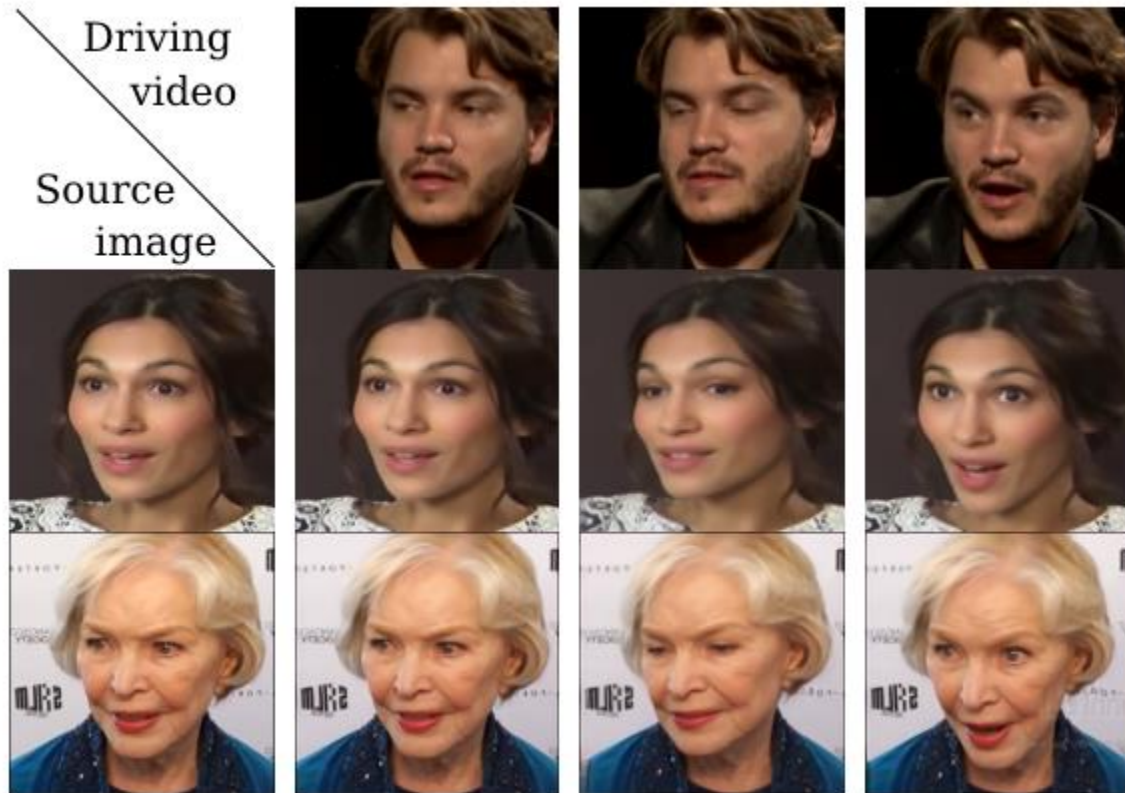
## paper

---

# Introduction

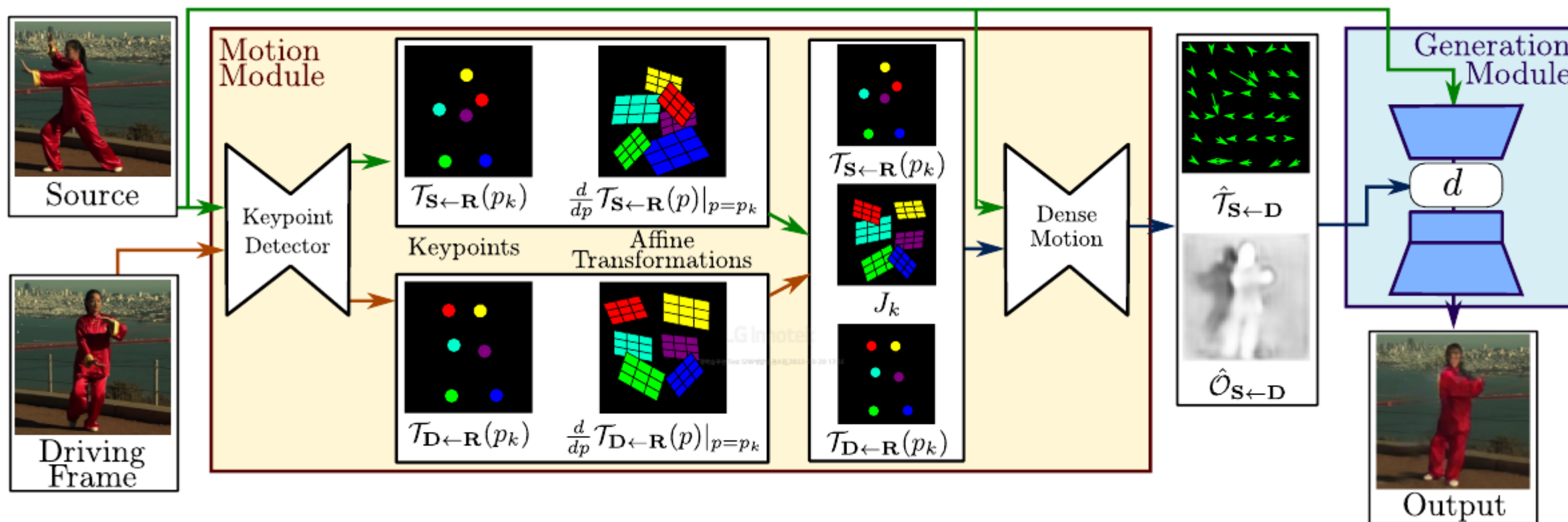
## Source image를 Driving video 처럼 transformation

- Object의 이전 정보에 의존하지 않음 (이전 정보 저장 필요 x)
- Same category에 적용 (ex. Face, human body, robot arms etc)



## Overview

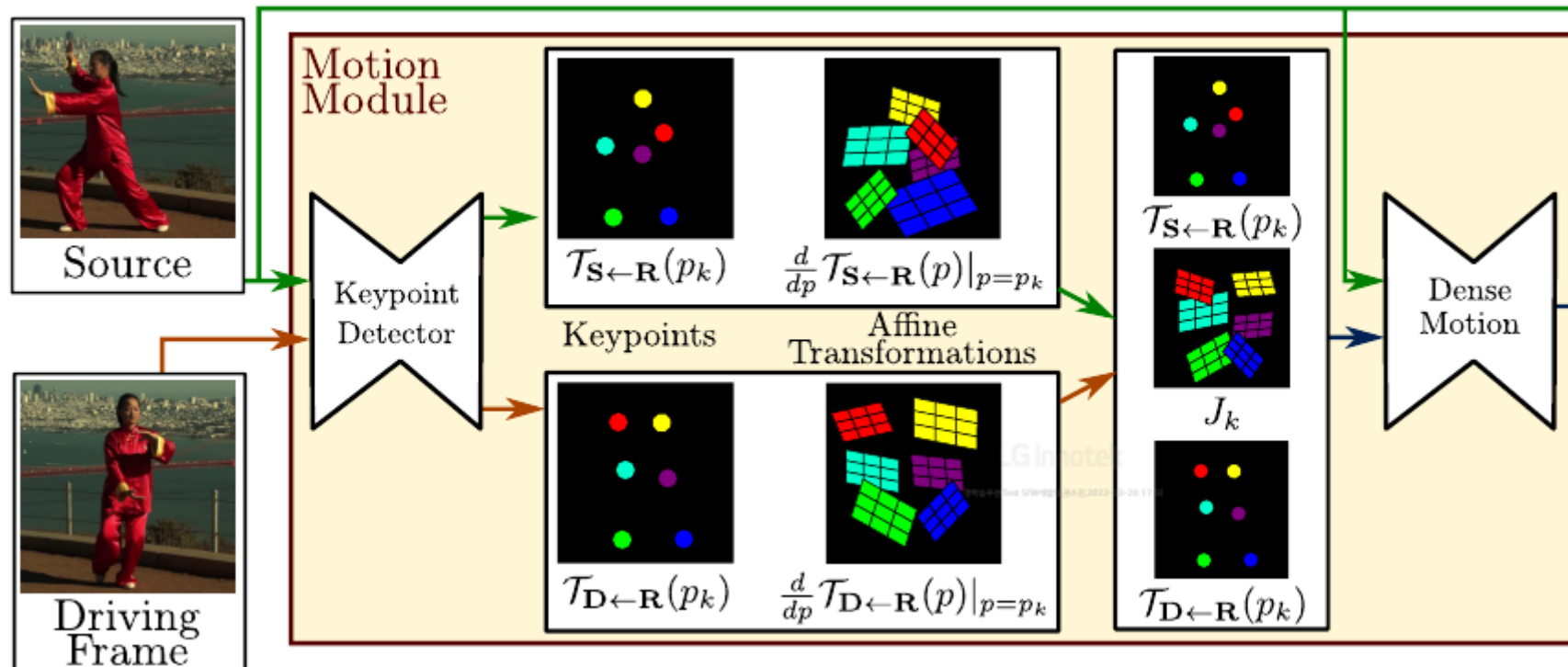
- **Motion Module & Generation Module**
  - Keypoint Detector & Affine Transformation
  - Dense Motion
  - Geraration Module



## Motion module

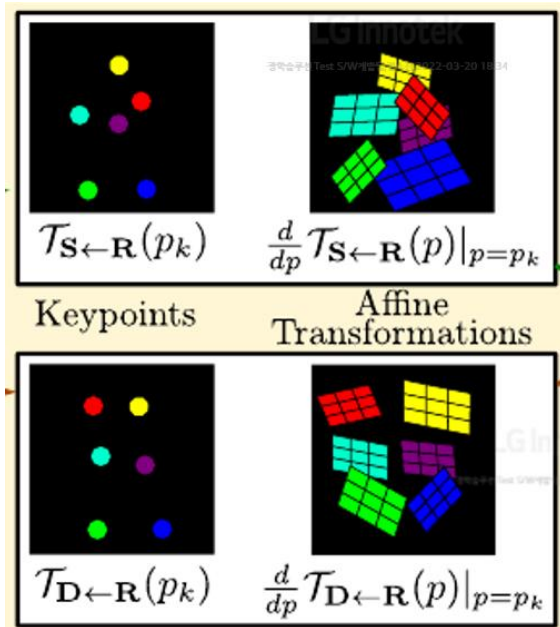
Motion module은 driving video(D)와 Source frame(S) 각각의 dense motion을 예측하는 것이 목표

- Source frame과 driving video의 정보 추출 과정을 서로 독립적으로 만들기 위해 reference frame R을 도입
- Dense motion은 key point와 local affine transformation을 기반으로 예측된다
- Local Affine transformation은 key point 주변의 motion을 모델링한다



# Local Affine Transformation – Taylor Series

테일러 급수가 필요한 이유 : 우리가 잘 모르거나 복잡한 함수를 다루기 쉽고 이해하기 쉬운 다항함수로 대체시키기 위함



1. 각 이미지의 key point K개를 찾음
2. K개의 key point를 이용하여 테일러 급수 전개하여 key point들의 주변부를 예측함

$$f(x) = p_{\infty}(x)$$

$$p_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$
$$= \sum_{k=0}^n \frac{f^{(k)}(a)}{k!}(x-a)^k$$

테일러 급수 예시

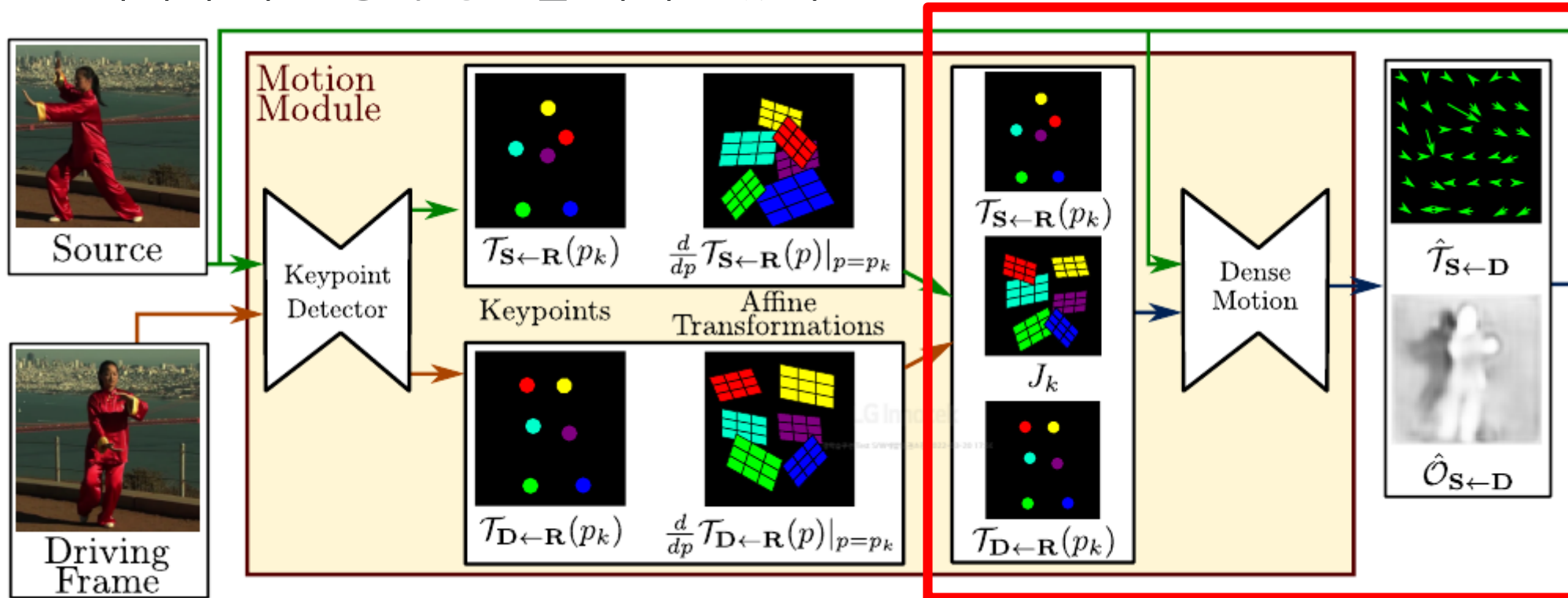
$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

## Dense motion network

Output: Dense motion & Occlusion mask

- Dense motion network는 keypoint detector의 output을 input으로 dense motion(optical flow)과 occlusion mask를 예측한다
- Occlusion mask는 이미지 생성 과정에서 source image의 가려진 부분을 reconstruct 되어야 하는 영역 정보를 가지고 있다.





## Generation Module

- Source image에서 가려진 부분을 생성하기 위해 occlusion map을 이용한다
- Dense motion (optical flow) 과 occlusion map을 이용해 최종 output을 도출함

