# Lecture 10

# Image Alignment and Homography

Multimedia System

Spring 2020

# Image alignment

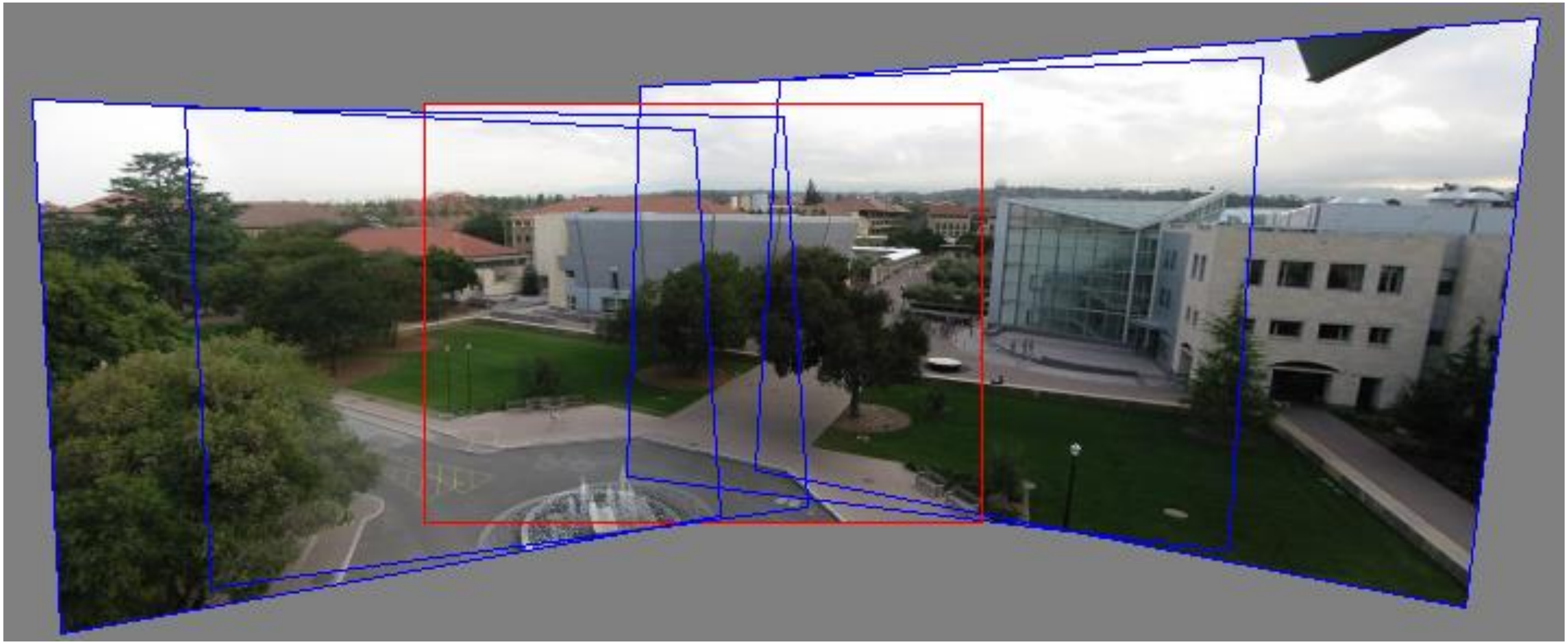
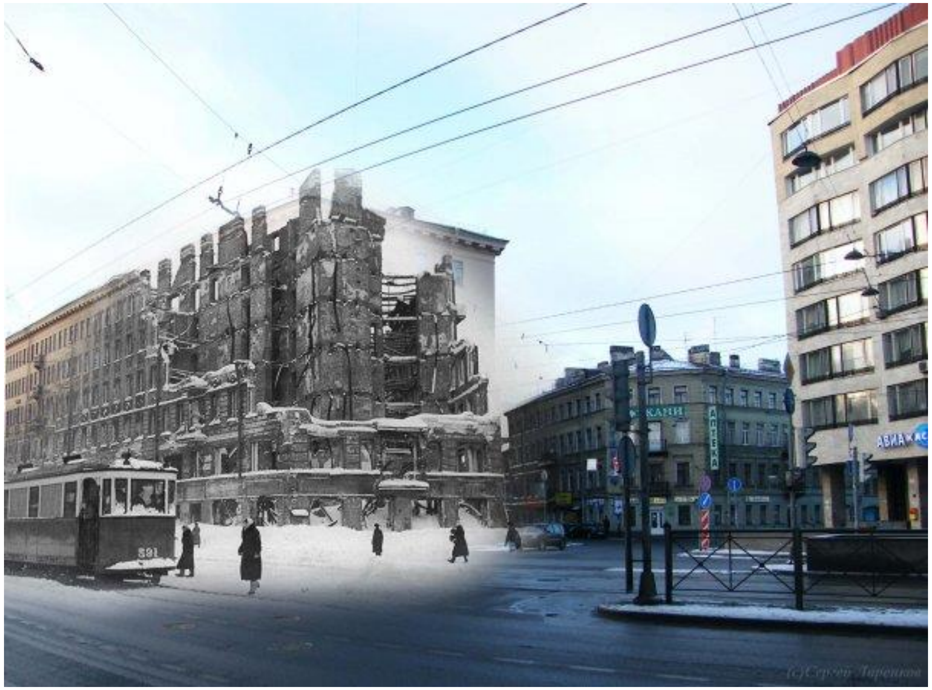


Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fall/

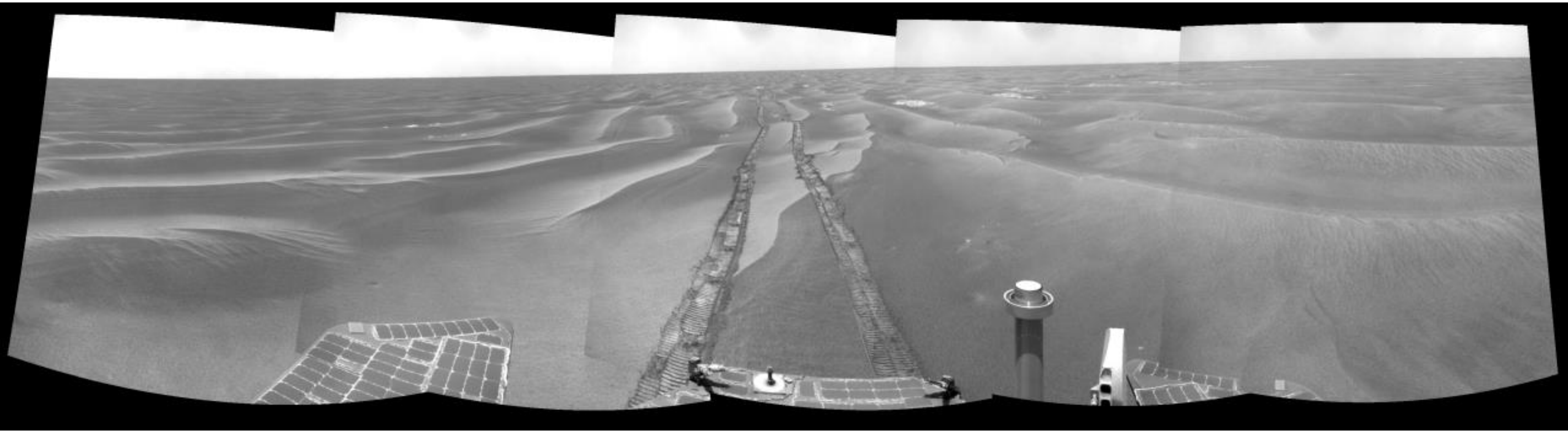
# A look into the past

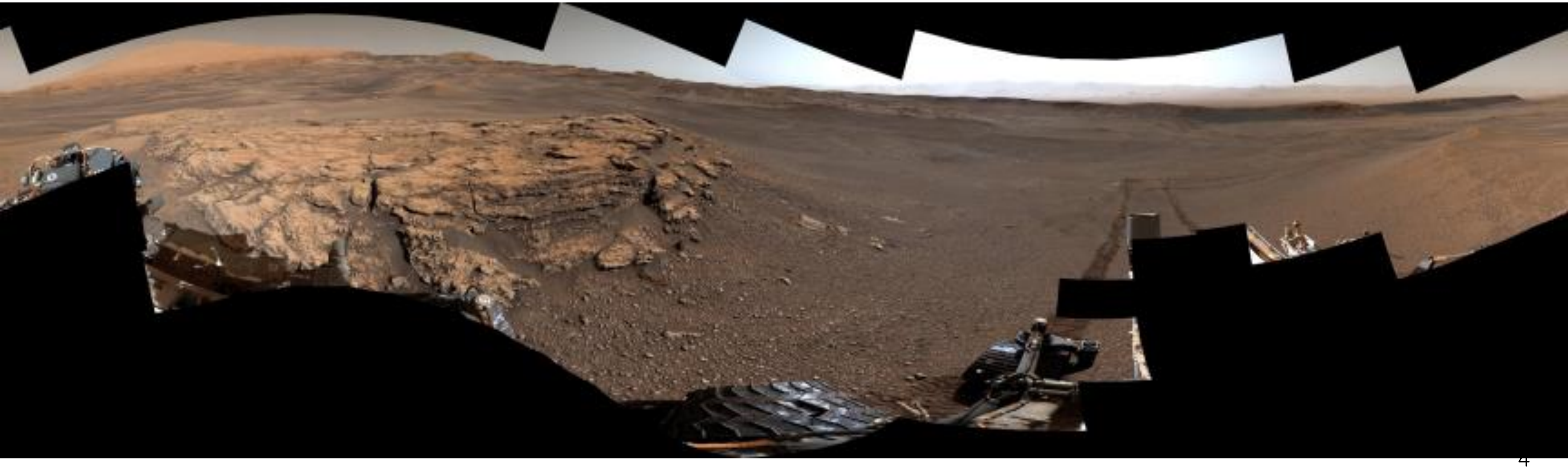
- Leningrad during the blockade



http://komen-dant.livejournal.com/345684.html

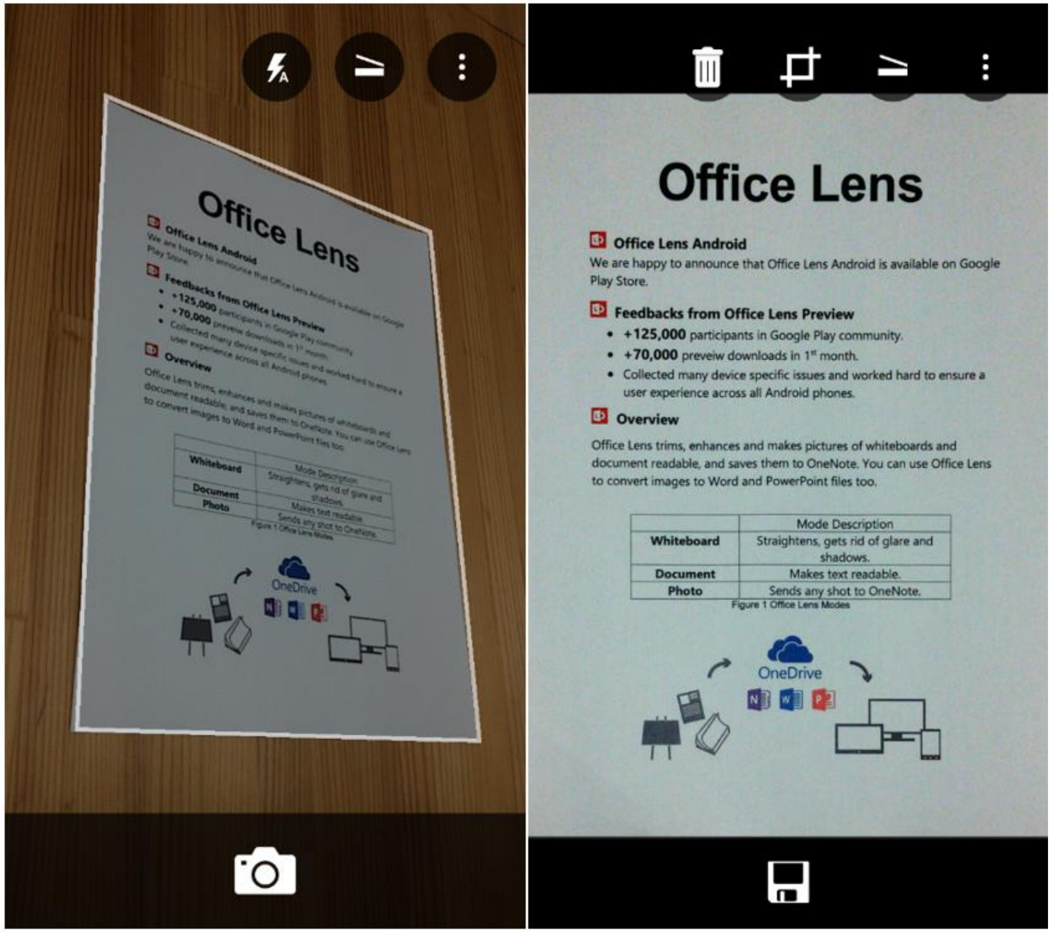
# Images from Mars





# Microsoft Office Lens

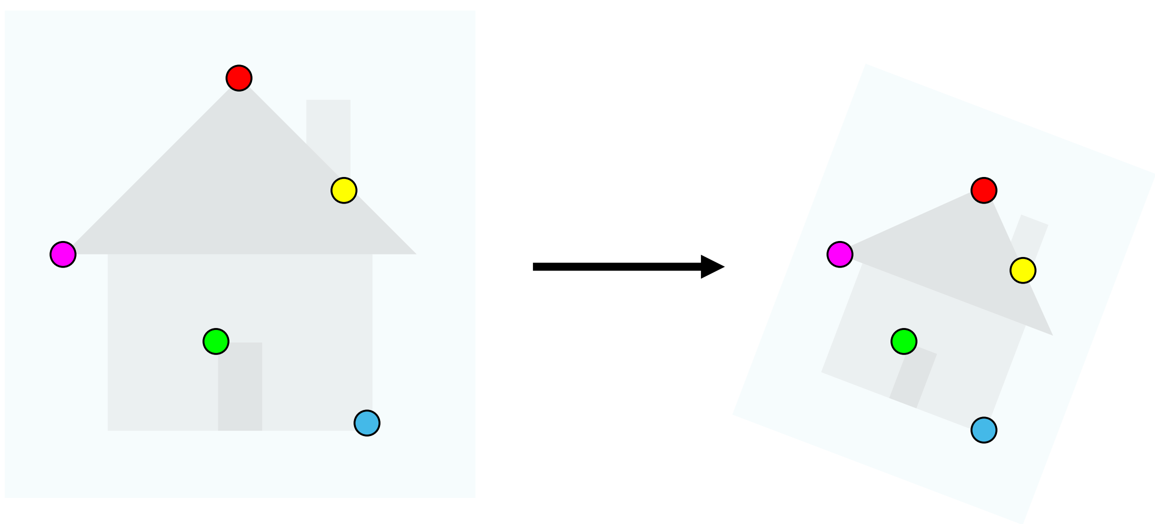
- Smartphone app for image alignment



# Vehicle around view

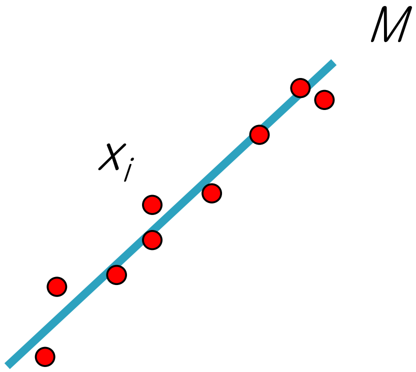
# Image alignment

- Two families of approaches:
  - **Direct (pixel-based) alignment**
    - Search for alignment where most pixels agree
  - **Feature-based alignment**
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment



# Alignment as fitting

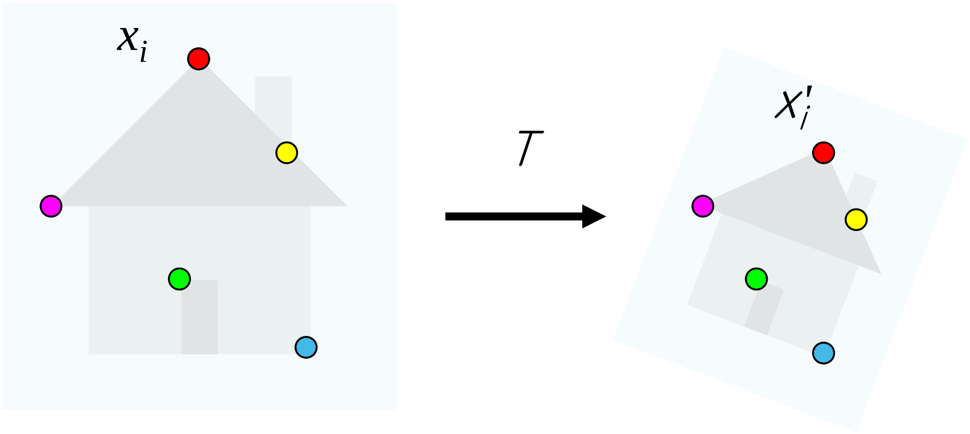
- Example: Fitting a line model to points in 2D space




Find model $M$ that minimizes

$$\sum_i \text{residual}(x_i, M)$$

- Alignment: fitting a model to a transformation between pairs of features (*matches*) in two images

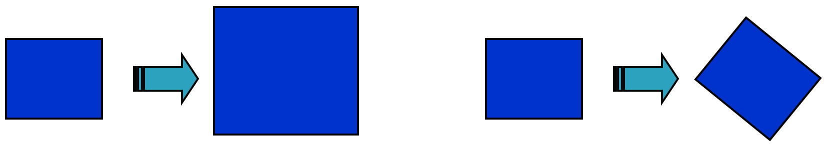



Find transformation $T$ that minimizes

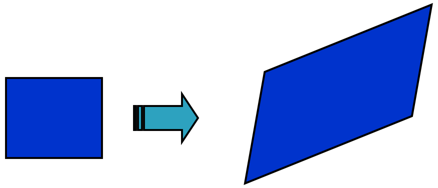
$$\sum_i \text{residual}(T(x_i), x_i')$$

# 2D transformation models

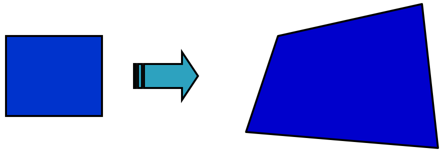
- Similarity (translation, scale, rotation)



- Affine

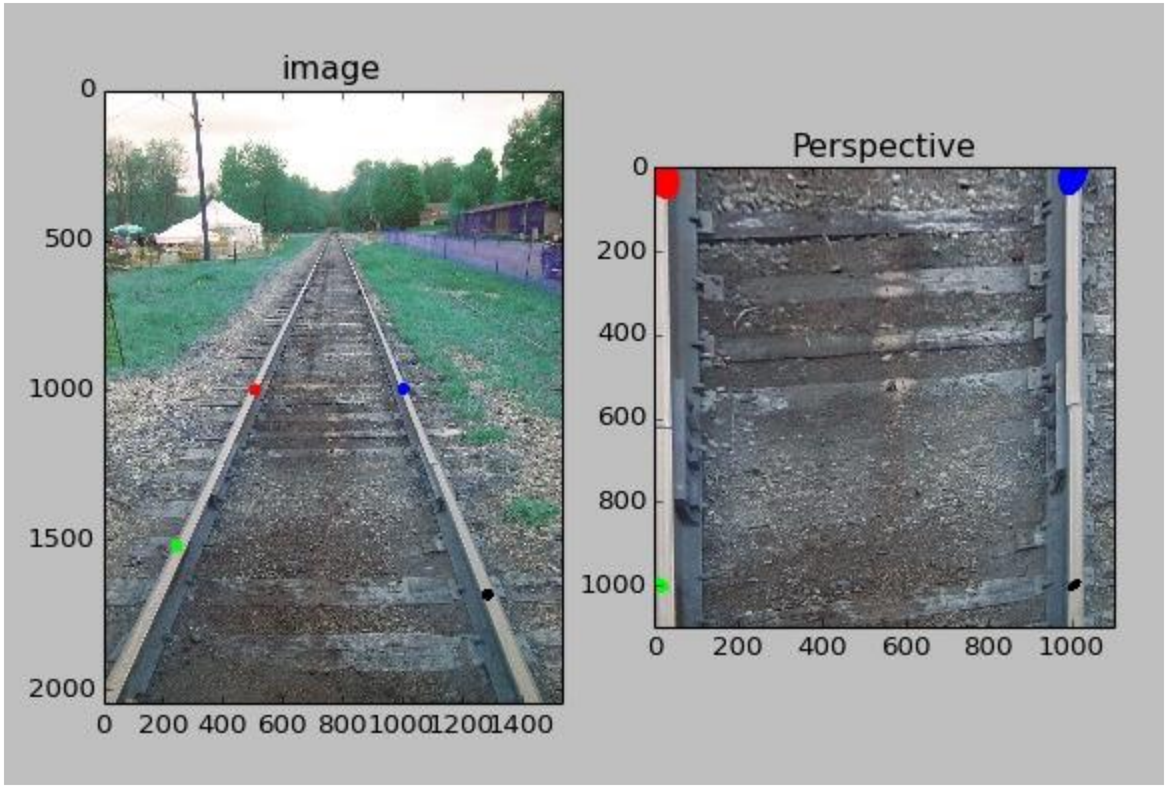


- Projective (homography)



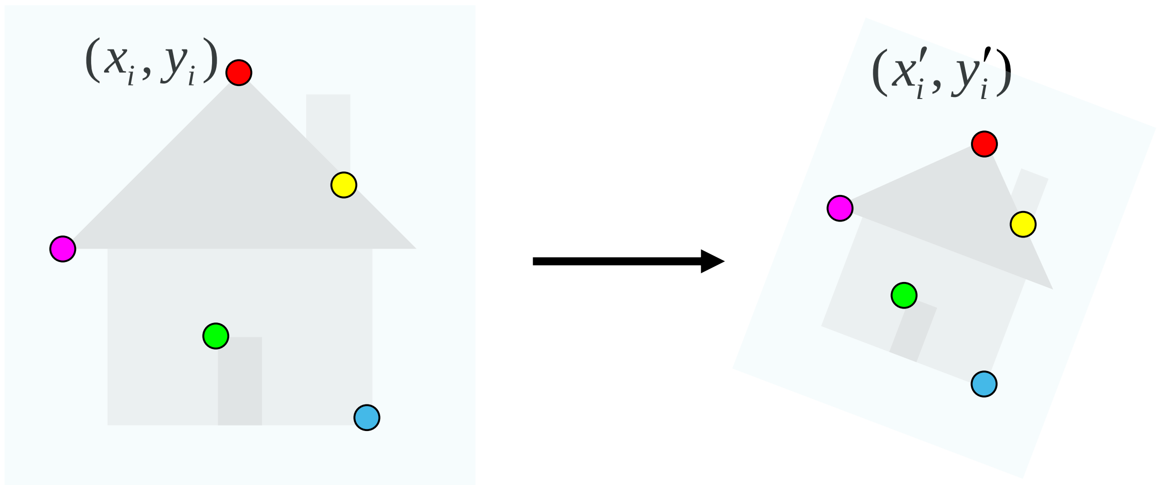
# Let's start with affine transformations

- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models



# Fitting an affine transformation

- Assume we know the correspondences, how do we get the transformation?




$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

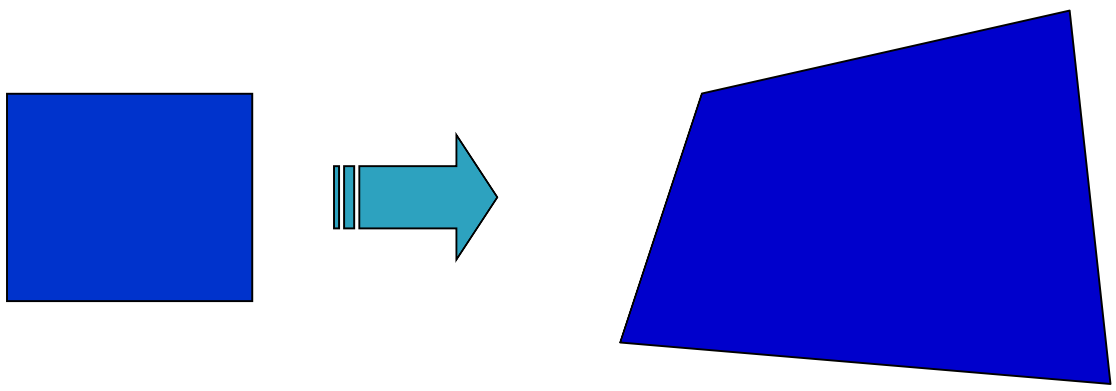
# Fitting an affine transformation

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- Linear system with six unknowns
- Each match gives us two linearly independent equations: need at least three to solve for the tr ansformation parameters

# Fitting a plane projective transformation

- **Homography:** plane projective transformation (transformation taking a quad to another arbitrary quad)

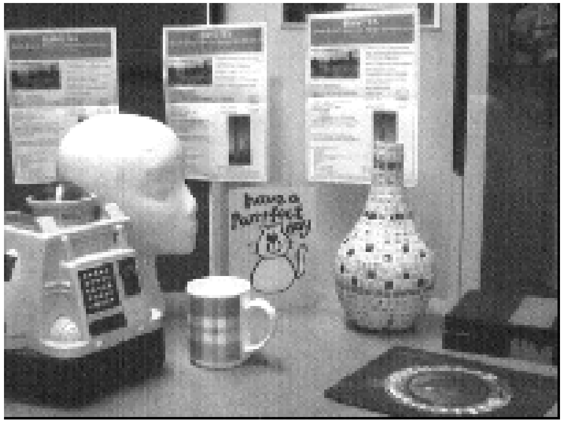
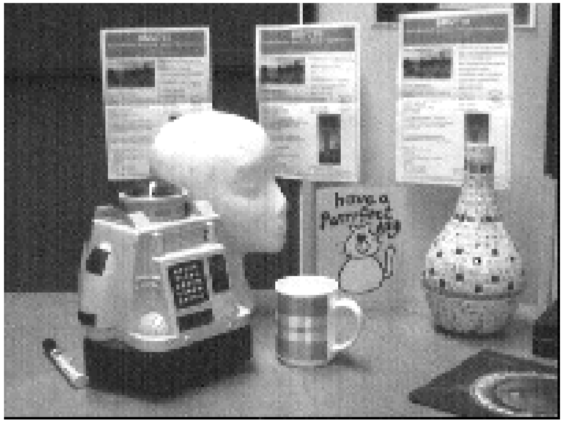


# Homography

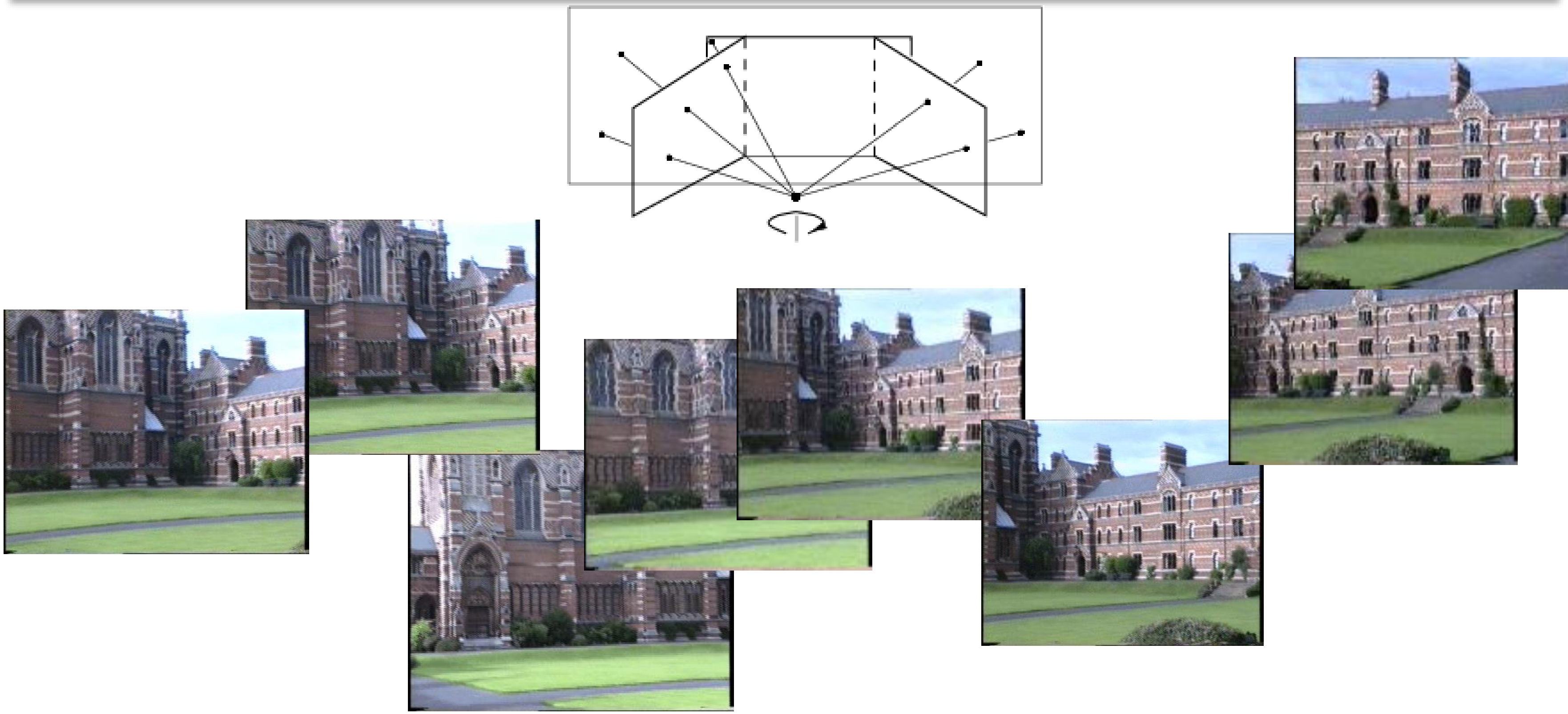
- The transformation between two views of a planar surface

- The transformation between images from two cam eras that share the same center

# Application: Panorama stitching



Source: Hartley & Zisserman

# Fitting a homography

- Recall: homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogeneous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogeneous image coordinates

# Fitting a homography

- Recall: homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogeneous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogeneous image coordinates

- Equation for homography:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Fitting a homography

- Equation for homography:

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\lambda \mathbf{x}_i' = \mathbf{H}\,\mathbf{x}_i$$

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = 0$$

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{h}_1^T \mathbf{x}_i \\ \mathbf{h}_2^T \mathbf{x}_i \\ \mathbf{h}_3^T \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} y_i' \, \mathbf{h}_3^T \mathbf{x}_i - \mathbf{h}_2^T \mathbf{x}_i \\ \mathbf{h}_1^T \mathbf{x}_i - x_i' \, \mathbf{h}_3^T \mathbf{x}_i \\ x_i' \, \mathbf{h}_2^T \mathbf{x}_i - y_i' \, \mathbf{h}_1^T \mathbf{x}_i \end{bmatrix}$$

$$\begin{bmatrix} 0^T & -\mathbf{x}_i^T & y_i' \, \mathbf{x}_i^T \\ \mathbf{x}_i^T & 0^T & -x_i' \, \mathbf{x}_i^T \\ -y_i' \, \mathbf{x}_i^T & x_i' \, \mathbf{x}_i^T & 0^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0$$

3 equations, only 2 linearly independent

# Direct linear transform

$$\begin{bmatrix} 0^T & \mathbf{x}_1^T & -y_1' \mathbf{x}_1^T \\ \mathbf{x}_1^T & 0^T & -x_1' \mathbf{x}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathbf{x}_n^T & -y_n' \mathbf{x}_n^T \\ \mathbf{x}_n^T & 0^T & -x_n' \mathbf{x}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0 \qquad \mathbf{A}\,\mathbf{h} = 0$$

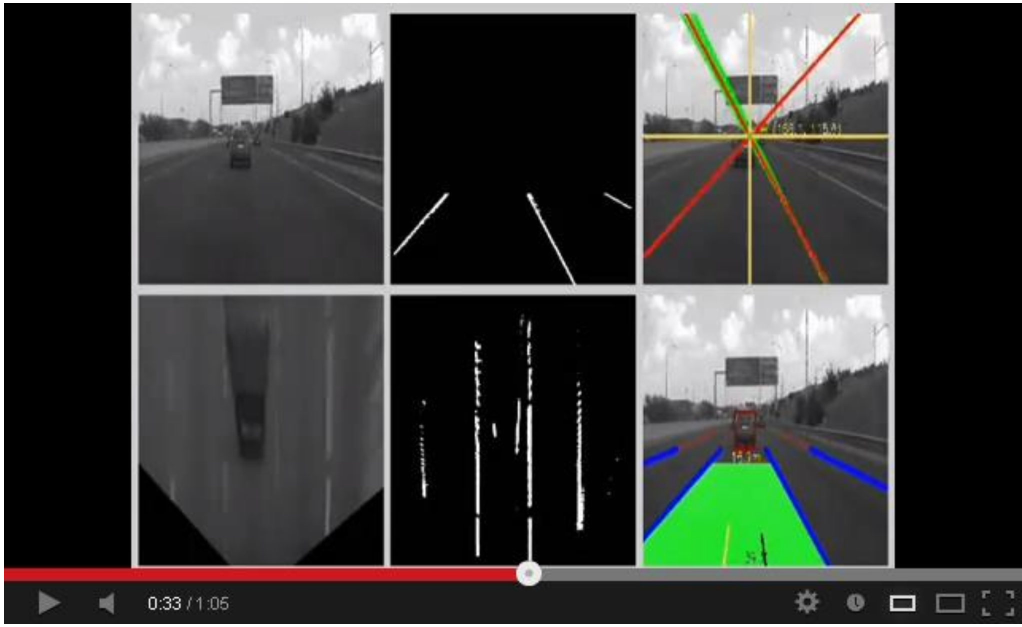
- H has 8 degrees of freedom (9 parameters, but scale is arbitrary)
- One match gives us two linearly independent equations
- Four matches needed for a minimal solution (null space of 8x9 matrix)
- More than four: homogeneous least squares

# Application to ADAS

# Lane Detection & Tracking

- Lane detection - draw boundaries of a lane in a single frame
- Lane tracking - uses temporal coherence to track boundaries in a video sequence

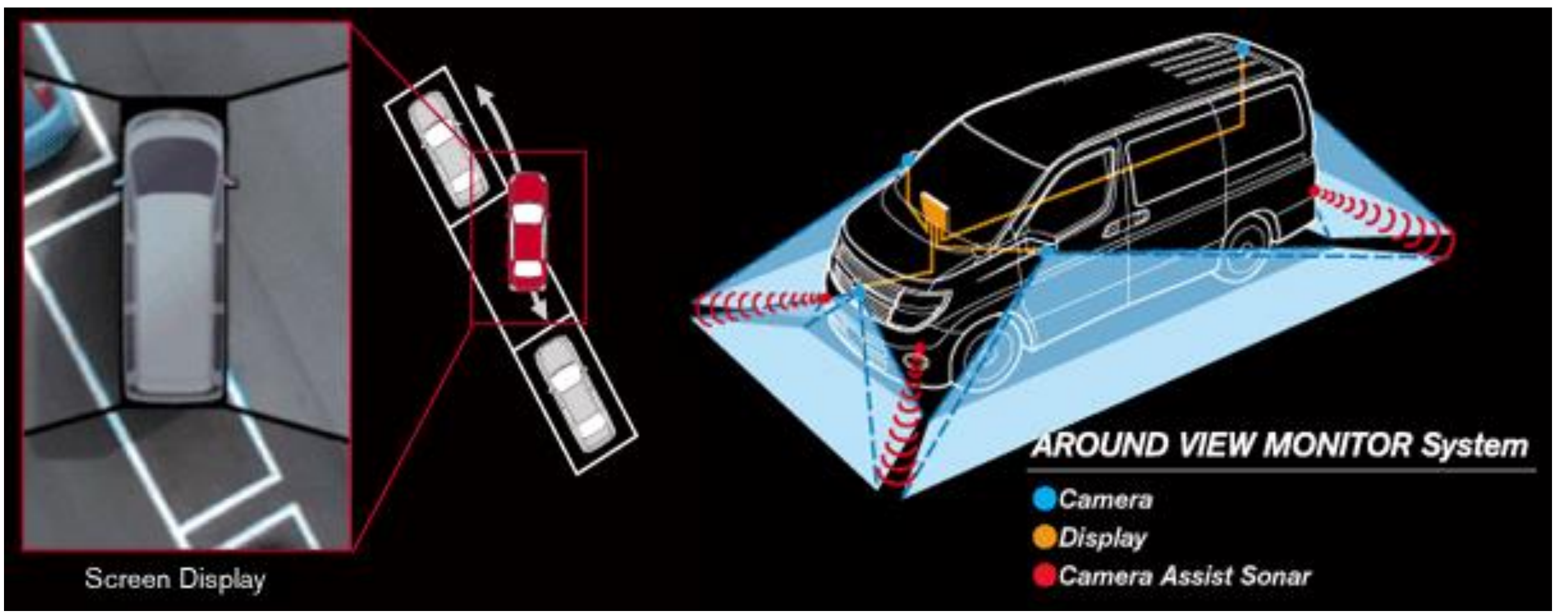
Lane Detection




Lane Tracking

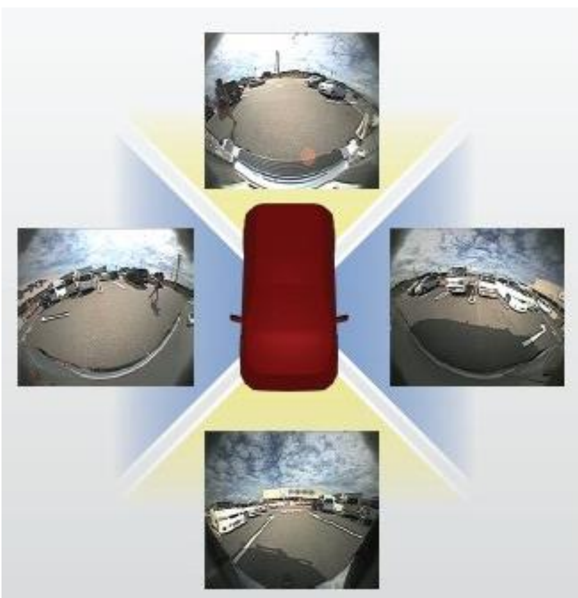
# Bird-eye view

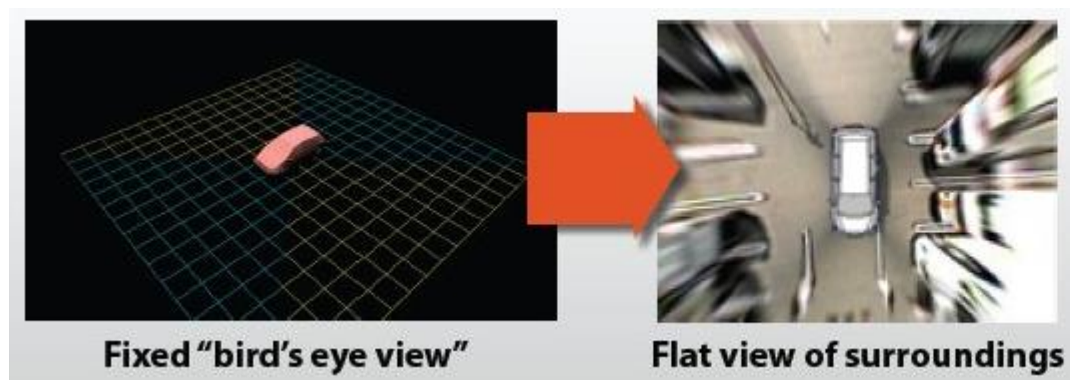
- A technology that assists drivers to park more easily by better understanding the vehicle's surroundings through a virtual bird's-eye view from above the vehicle.
- Same names: surround view, around view



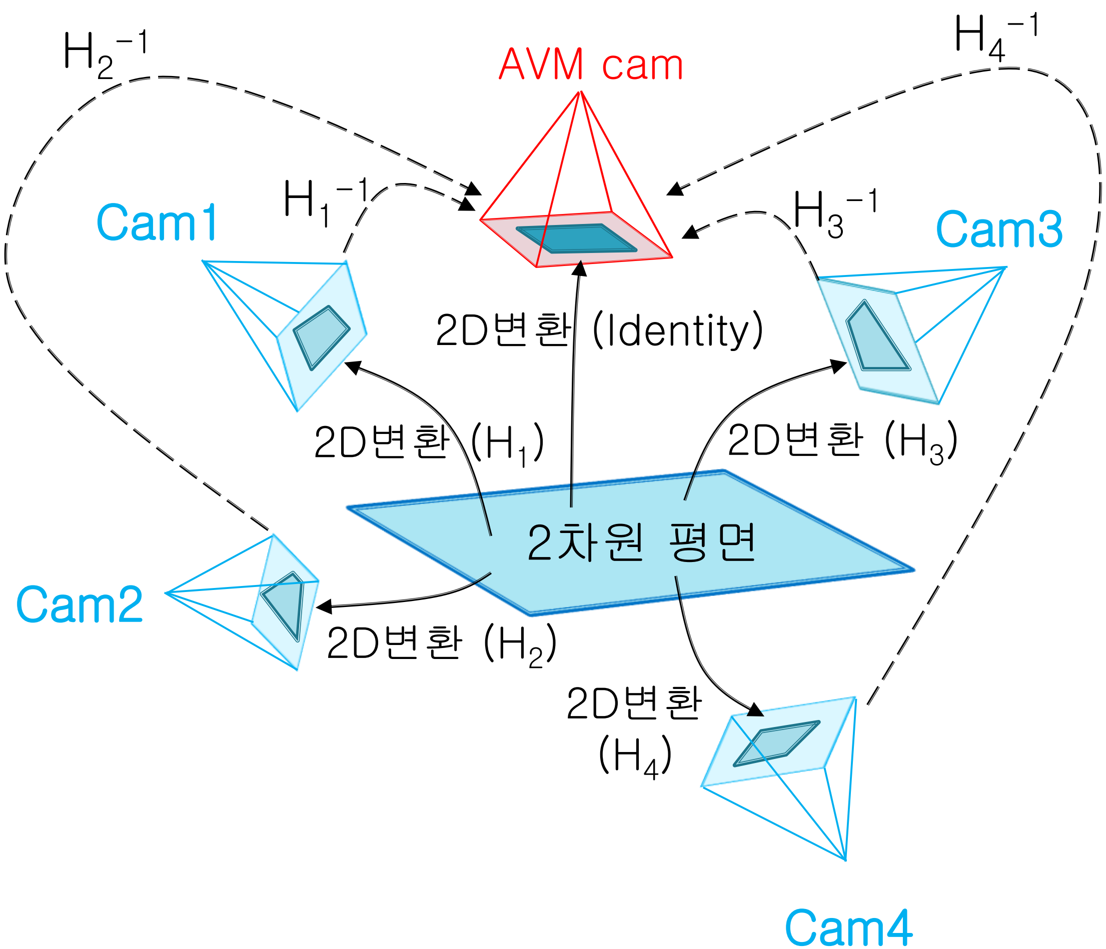
# Bird-eye view

- Transforming n-different views to a common bird-eye view.
  - From n-different views of the ground plane
  - A common view: bird-eye view of the ground plane





# AVM의 컴퓨터비전 이론

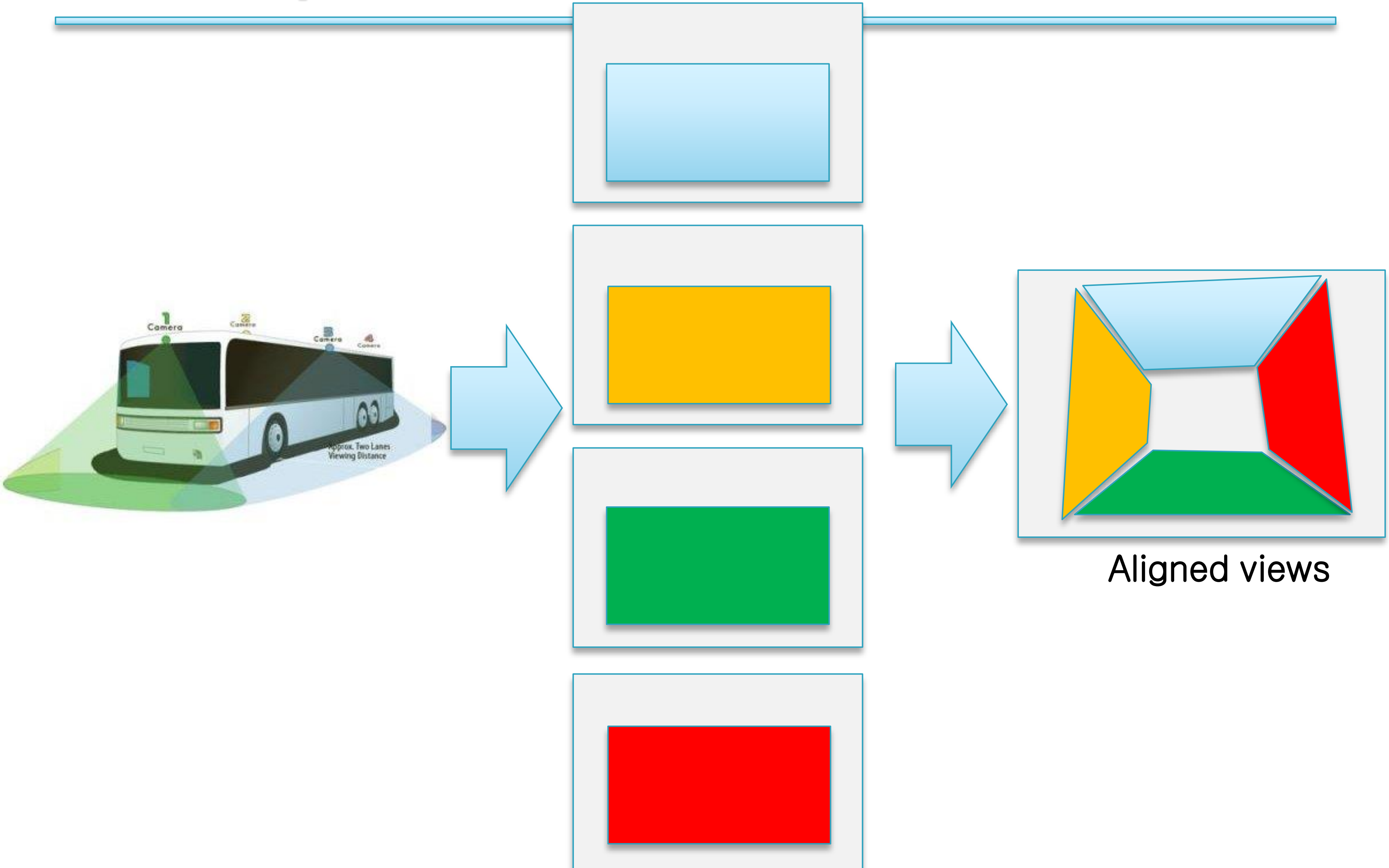



- 지상 평면위의 직선은 카메라 영상에서도 항상 직선 (렌즈왜곡이 없다고 가정)
- 지상 평면위의 사각형은 카메라 영상에서도 항상 사각형 (모양은 변형)
- 지상의 2차원 평면과 카메라의 2차원 영상 사이의 관계는 2차원 호모그래피 변환 (Homography)
  - H =

$$\begin{bmatrix} a & b & c \\ d & \mathrm{e} & \mathrm{f} \\ g & h & i \end{bmatrix}$$

- 지상평면과 카메라 사이의 2D변환을 알고 있다면, 모든 카메라에서 AVM 가상 카메라로 2차원 변환 관계를 구할 수 있음

# Bird-eye view




Aligned views

Different views

# 2D 컴퓨터비전의 한계

- 지상으로부터 높이가 있는 물체의 3차원 정보 획득 불가능