Motion Tracking and Event Understanding in Video Sequences

Isaac Cohen

Elaine Kang, Jinman Kang

Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA

Objectives

- Infer interesting *events* events in a video
  - Some examples: people meeting, exchanging objects, gesturing….
  - Events may take place over a range of time scales
  - Requires object recognition
- Provide a convenient form to define events
  - An *event recognition language* (ERL) that can be compiled automatically to produce recognition programs
- Compute a *structured representation* of video
  - Events and objects
  - Spatial and temporal relations between them

Example Video

Video Description

- Descriptions useful for query, annotation and compression
- Symbolic descriptions
  - Events: Name, actors (objects), reference objects, duration, place
  - Sub-events and relations between them
  - Object descriptions
    - Trajectory, Shape, Appearance
    - Background objects

Example Video

Video with inferred annotations

Challenges

- Translation from signals to semantic information
  - Signals are ambiguous (one to many mapping)
- Image sequence analysis
  - Detection and tracking of moving objects
- Generic and specific object recognition
  - Object appearances change with many variables (view point, illumination, occlusion, clothing…)
- Inference of events from object trajectories and identities

Topics

- Moving blob detection and tracking
  - Static and moving cameras
  - Perceptual grouping for tracking
- Detection and tracking of objects
  - Segmentation of blobs into objects
  - Tracking of articulated objects
- Event recognition
  - Definition, compilation, computation
- Miscellaneous
  - Activities, evaluations, future plans

Motion Detection (Static Cameras)

- Construct an adaptive model of "background"
  - Each pixel modeled as a multi-dimensional Gaussian distribution in color space
  - Updated when new pixel values are observed

- Extract "foreground"
  - A pixel is foreground if sufficiently different than current background model
  - Marked pixels grouped into connected components

Background Learning

- Maintain an adaptive background model for the entire region of awareness
- Model each pixel as a multi-dimensional Gaussian distribution (μ,σ) in RGB space
- Update background when a new pixel values are observed

Background Learning

- Maintain an adaptive background model for the entire region of awareness
- Model each pixel as a multi-dimensional Gaussian distribution (μ, σ) in RGB space
- Update background when a new pixel value x is observed:

$$\mu \leftarrow \alpha x + (1-\alpha)\mu$$

$$\sigma^2 \leftarrow \max\left(\sigma^2_{\min}, \alpha(x-\mu)^2 + (1-\alpha)\sigma^2\right)$$

where α = learning rate

Foreground Extraction

- Detect changes produced by occluding elements
- Compare observed pixel values to the current distribution
- A new pixel observation x is marked as foreground if:

$$(x-\mu)^2 > (2\sigma)^2$$

- Group foreground pixels in connected components - layered description (sprites)

Detection Example

Note: blobs may not correspond to single objects

Detection (Moving Camera)

- Compensate for the camera's motion by an affine transformation of images
  - Multi-resolution, robust feature-based matching
  - Accurate when depth variations in background and image area of moving objects are small

- Detection of moving objects by
  - Residual motion
  - Color-based classification

Parameter Recovery

- Recover 6 affine parameters : (a,b,c,d,t_x,t_y)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

->Solving a set of linear equations

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 \end{pmatrix}_i \begin{pmatrix} a & b & T_x & c & d & T_y \end{pmatrix}^T = (x_1\ y_1)_i^T$$

- Feature based approach:
  - Feature points are extracted by a corner detector
  - RANSAC for *outliers* removal
  - Linear Least Square Method using *inliers*

Panning Camera

Mosaic shown from the first camera view point

Hand-held Camera

Limitations

- Sudden illumination changes:
  - The layer model is learnt over a sliding window
- Fragmented detection:
  - neighborhood properties are not considered
- Mis-registration of images
  - Large depth variations in background and large image area of moving objects
- Registration in the image joint space

Outlier Detection

- e-RANSAC
  - RANSAC (Fischler-Bolles) is a standard tool for robust parameter estimation
  - Enhanced by controlling feature sampling
- Tensor voting
  - General purpose grouping methodology
  - Encodes local properties and their uncertainties as *tensors*
  - Propagates local estimates to neighbors
  - Propagated information combined as a weighted *tensor* addition
  - Provides a *saliency* map, outliers have low saliency
  - Used to group points lying in salient planes in the joint image space

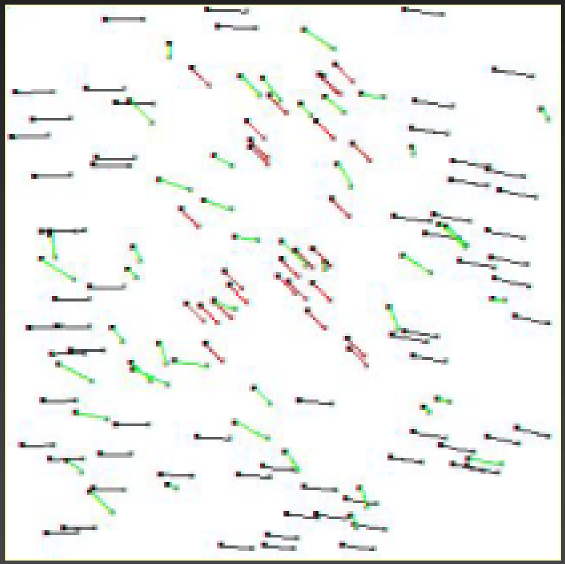
Robust Affine Motion Estimation in Joint Image Space

- Parametric motion estimation
  - Widely used for video processing: image mosaics, compression, and surveillance
  - Affine motion model (six parameters)
    - commonly used due to simplicity and the small inter-frame camera motion.
- Issues
  - Correspondence-based parameter estimation often fails in the presence of many mismatches and multiple motions
  - Robustness of estimation relies on outlier removal

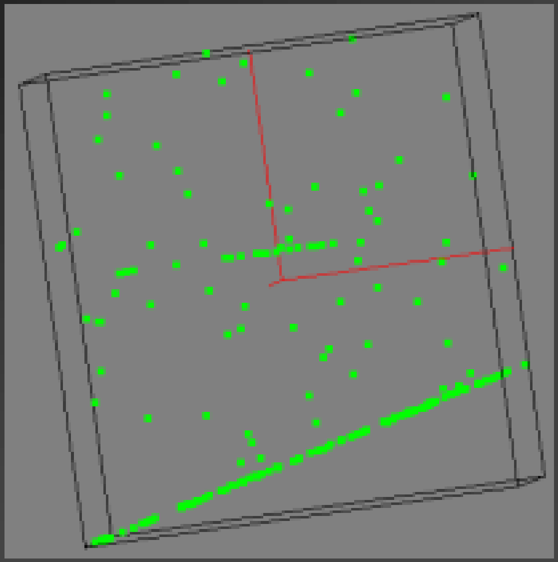
Goal and Approach

- Goal
  - Outlier removal for robust parameter estimation
- Approach
  - Representation of correspondences in decoupled joint image spaces
  - Analyze the metric of the affine parameters in the defined space
  - Tensor voting-based outlier removal
  - Direct parameter estimation from correlation matrix of inliers

Illustration of Our Approach



Initial correspondences by affine motions



Views in 3D: Affine motions form planes in decoupled joint image space

Initial Correspondences

↓

Tensor encoding in joint image space

↓

Second order Sparse Tensor Voting

↓

Inlier Detection and Grouping

↓

Parameter Recovery

↓

Affine Parameters

Affine Joint Image Space

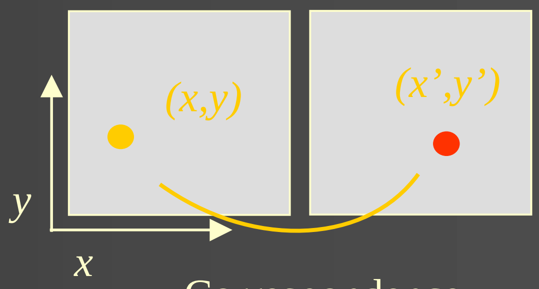
- Joint Image Space : (*x, y, x', y'*)
- Affine Transformation in the joint image space :

$$\begin{pmatrix} q^T & 1 \end{pmatrix} C \begin{pmatrix} q \\ 1 \end{pmatrix} = 0$$

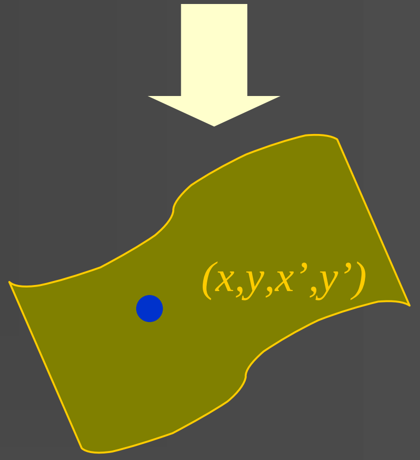
$$C = p^T p$$

$$p = \begin{pmatrix} a & b & -1 & 0 & t_x \\ c & d & 0 & -1 & t_y \end{pmatrix}^T$$

$$q = \begin{pmatrix} x & y & x' & y' \end{pmatrix}^T$$



Correspondence



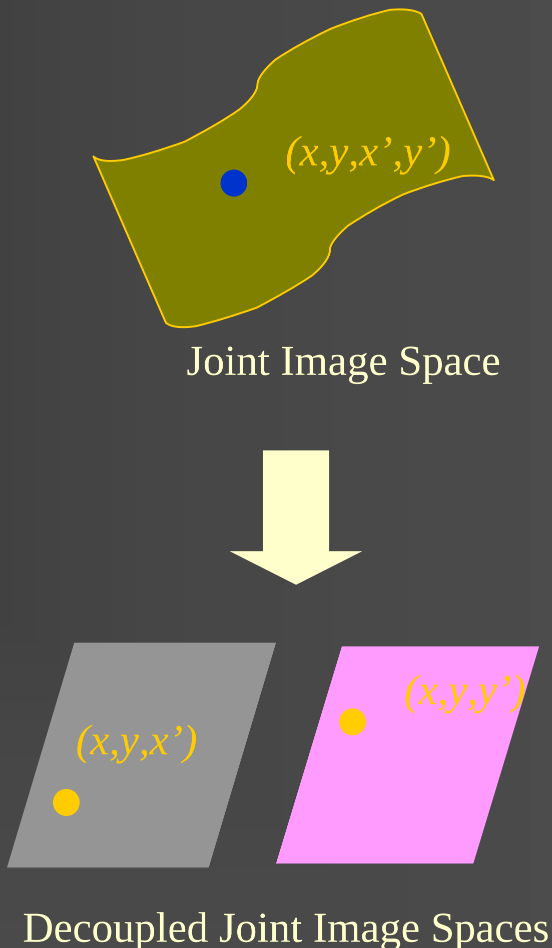
Joint Image Space

- The equation : a quadric in the 4 dimensional joint image space of (*x, y, x', y'*)
- A 5x 5 matrix *C* is rank 2

USC

Decoupled Affine Joint Image Space (1/2)

- The parameters (a,b,t_x) and (c,d,t_y) are independent
- We can ***decouple*** the joint image space into two spaces
- Decoupled joint image spaces
  - Defined by (x, y, x') and (x, y, y')
  - Dimension reduction
  - Isotropic and orthogonal spaces
  - Allow to enforce affine constraint during tensor voting



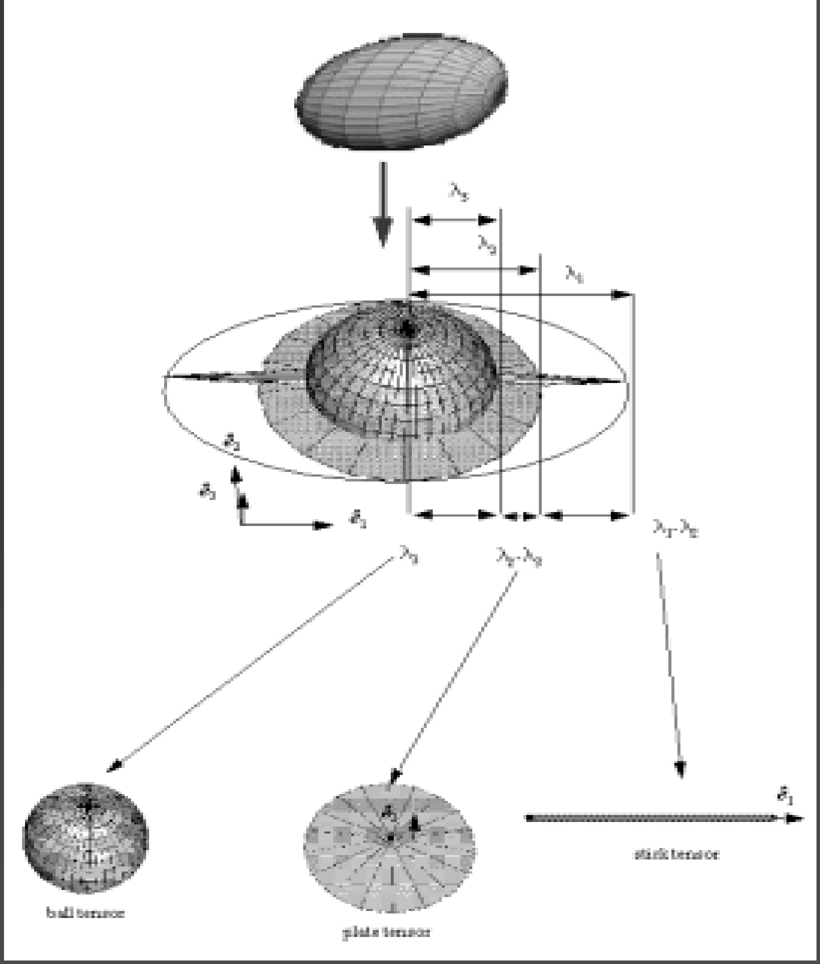
Decoupled Affine Joint Image Space (2/2)

Decoupled Joint Image Spaces	$q_x=(x,y,x')^T$	$q_y=(x,y,y')^T$
Parameter Vectors	$p_x=(a,b,-1,t_x)$	$p_y=(c,d,-1,t_y)$
Affine Transformation Equations	$A_x = p_x \begin{pmatrix} q_x \\ 1 \end{pmatrix} = 0$	$A_y = p_y \begin{pmatrix} q_y \\ 1 \end{pmatrix} = 0$

- Consists of all points $(q_x,1)$ or $(q_y,1)$
- All points $(q\ ,1)^T$ lie on a 2D plane given by equation A_x or A_y parameterized by p_x or p_y
- If a correspondence is correct, it lies on a 2D plane
- Outlier/Inlier detection is to extract points on a plane
- Properties of a plane :
  - (a, b, –1) and (c, d, -1) define orientation of the plane
  - *tx* and *ty* define the perpendicular distance between the plane and the origin of the space

Generic Tensor Voting (1/2)

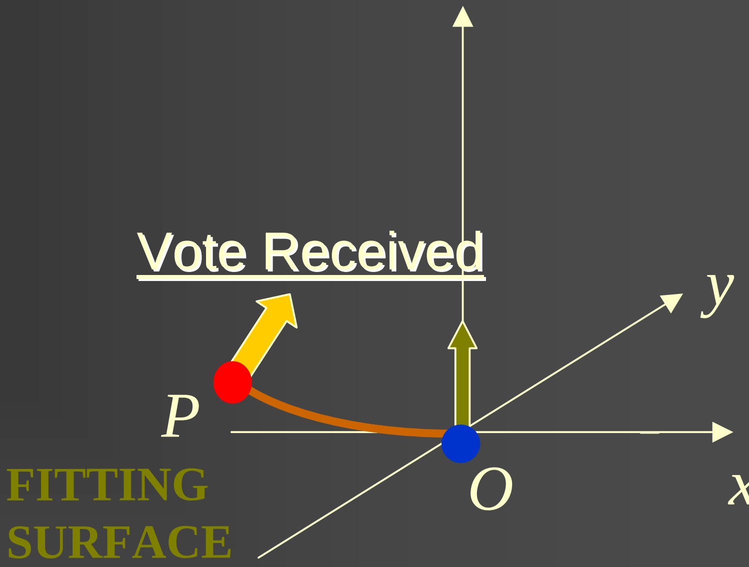
- Data representation: Second order symmetric tensors
  - **shape**: orientation certainty
  - **size:** feature saliency
- In 3D
  - 3 eigenvalues (λ_{max} λ_{mid} λ_{min})
  - 3 eigenvectors ($\mathbf{e}_{max}$ $\mathbf{e}_{mid}$ $\mathbf{e}_{min}$)
  - Surface extraction
    Saliency : $\lambda_{max} - \lambda_{mid}$
    Normal: $\mathbf{e}_{max}$

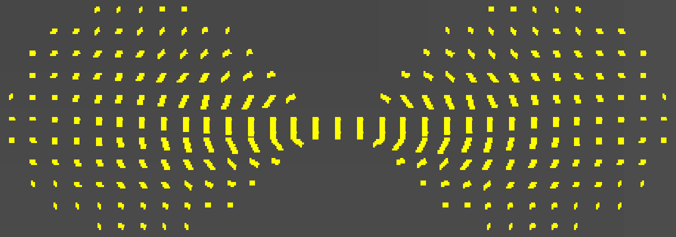


USC

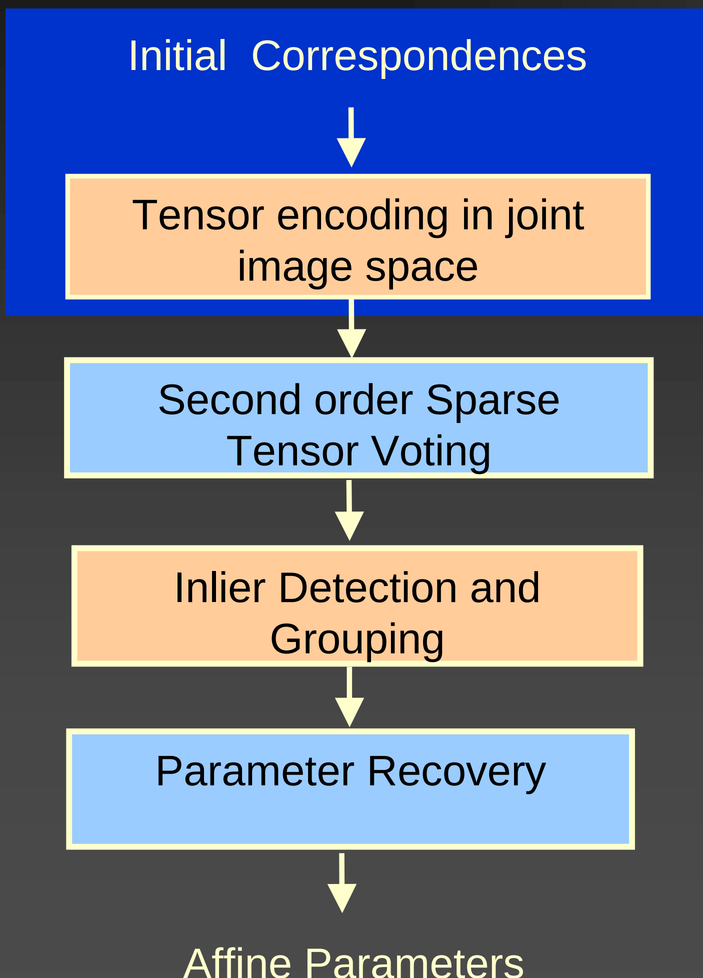
Generic Tensor Voting (2/2)

- Communication: **Voting**
  - non-iterative, no initialization
- Constraint representation: Voting fields
  - tensor fields encode smoothness criteria
- Each input site propagates its information in a neighborhood
- Each site collects the propagated information



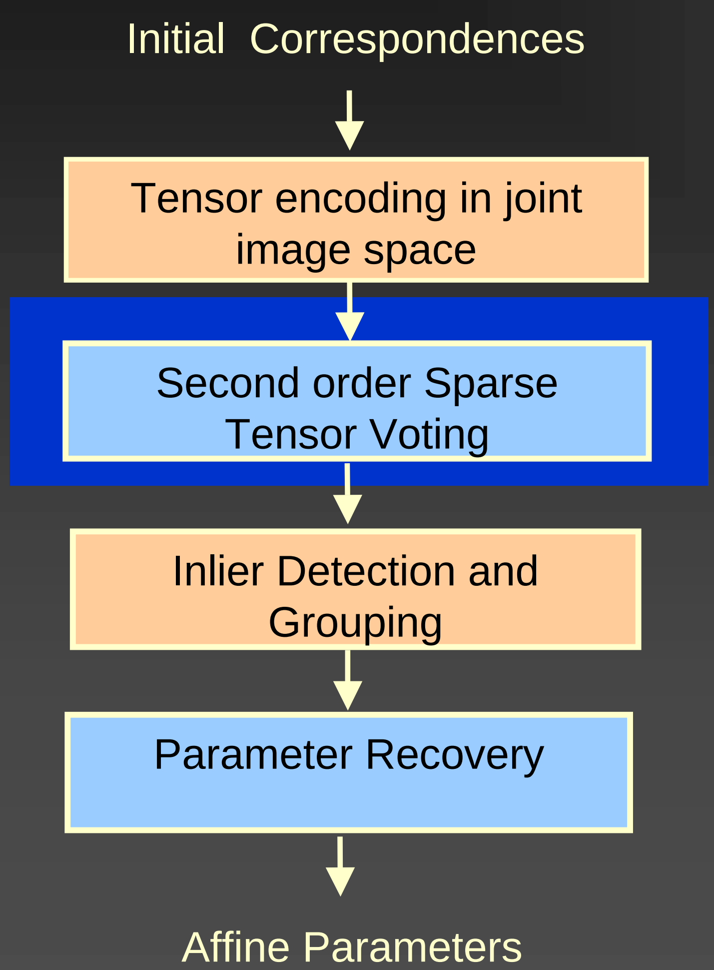


Initial Tensor Encoding



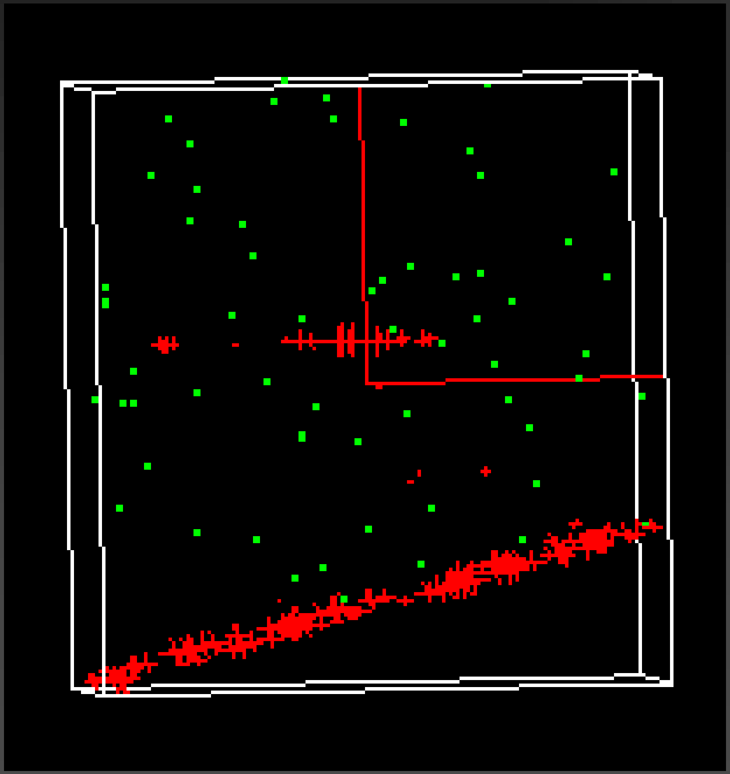
- Initial Correspondence
  - Harris Corner & Cross Correlation
- Stick tensor encoding
  - Use prior knowledge and assumption
    - The normal direction for each plane
      - $n_x = (a,b,-1)$, $n_y = (c, d, -1)$
    - Assume
      - $a=1$, $b=0$, $c=0$, $d=1$, $t_x = (x-x')$ and $t_y = (y-y')$
    - Therefore, the normal direction for each plane is initialized by
      - $n_x = (1, 0, -1)$
      - $n_y = (1, 0, -1)$
  - Give preference of normal direction of the plane
  - Prevent from generating unecessary votes

Plane Extraction by Tensor Voting



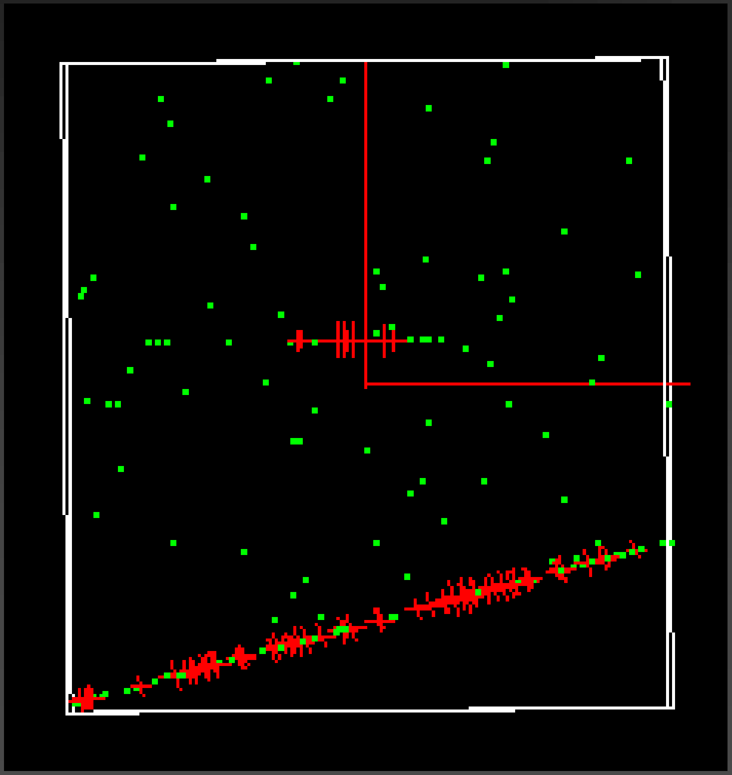
- Tensor Voting for Plane Extraction
  - Vote with planar field for plane extraction
  - First sparse voting
    - Extract the normal direction of the 2D plane encoded by e_1 with saliency $\lambda_1 - \lambda_2$
    - Remove random correspondences
  - Second sparse voting
    - Enforce the normal direction extracted from the first voting

Plane Extraction by Tensor Voting



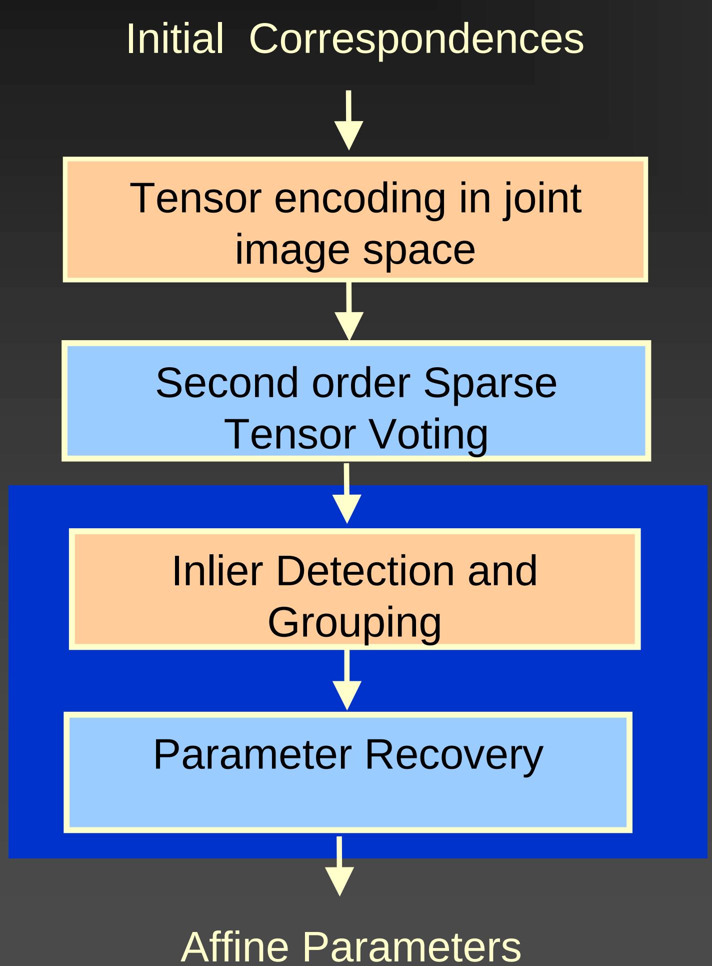
- Tensor Voting for Plane Extraction
  - Vote with planar field for plane extraction
  - **First sparse voting**
    - Extract the normal direction of the 2D plane encoded by e_1 with saliency $\lambda_1 - \lambda_2$
    - Remove random correspondences
  - Second sparse voting
    - Enforce the normal direction extracted from the first voting

Plane Extraction by Tensor Voting



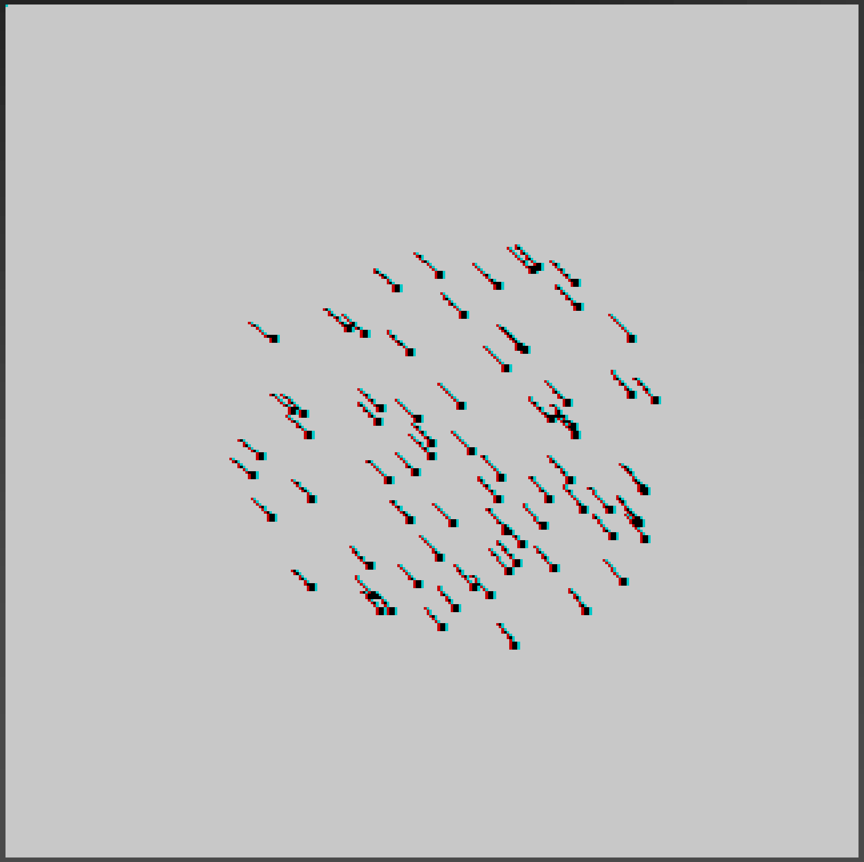
- Tensor Voting for Plane Extraction
  - Vote with planar field for plane extraction
  - First sparse voting
    - Extract the normal direction of the 2D plane encoded by e_1 with saliency $\lambda_1 - \lambda_2$
    - Remove random correspondences
  - **Second sparse voting**
    - Enforce the normal direction extracted from the first voting

Inlier Detection & Grouping



- Inlier detection and grouping
  - Inliers : Points having saliencies higer than a threshold (median)
  - Two inlier grouping criteria
    - Normal direction e_1 and locally smooth displacement *(x-x')* or *(y-y')*
- Parameter estimation :For each set of grouped inliers
  - Compute the correlation matrix : M_x and M_y
  - To estimate parameters *a, b*, t_x, the eigen vector corresponding to the smallest eigen value of M_x is used

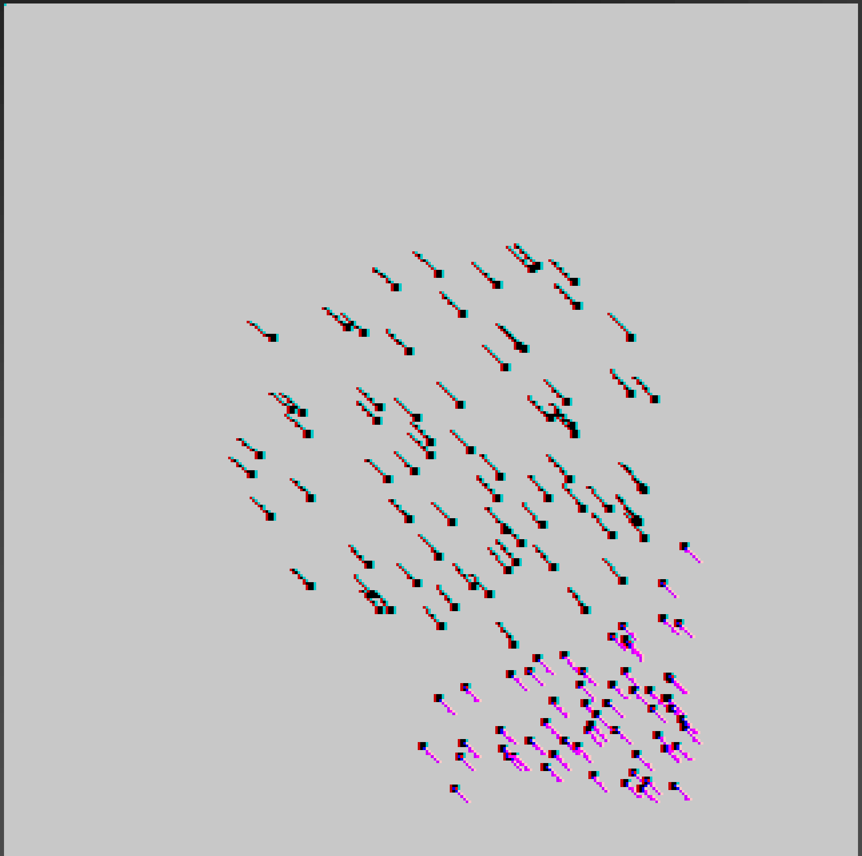
Experimental Result(1/2)



Motion 1 :

Foreground motion with rotation and translation

Experimental Result(1/2)



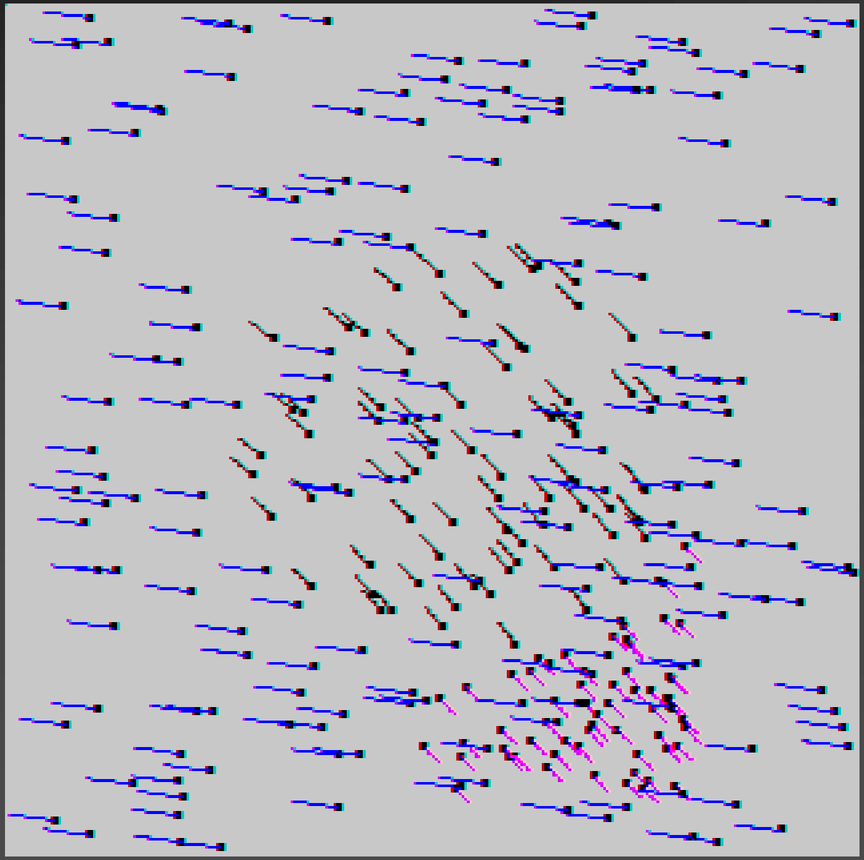
Motion 1 :

Foreground motion with rotation and translation

Motion 2 :

Conflicting foreground motion with translation

Experimental Result(1/2)



Motion 1 :

Foreground motion with rotation and translation

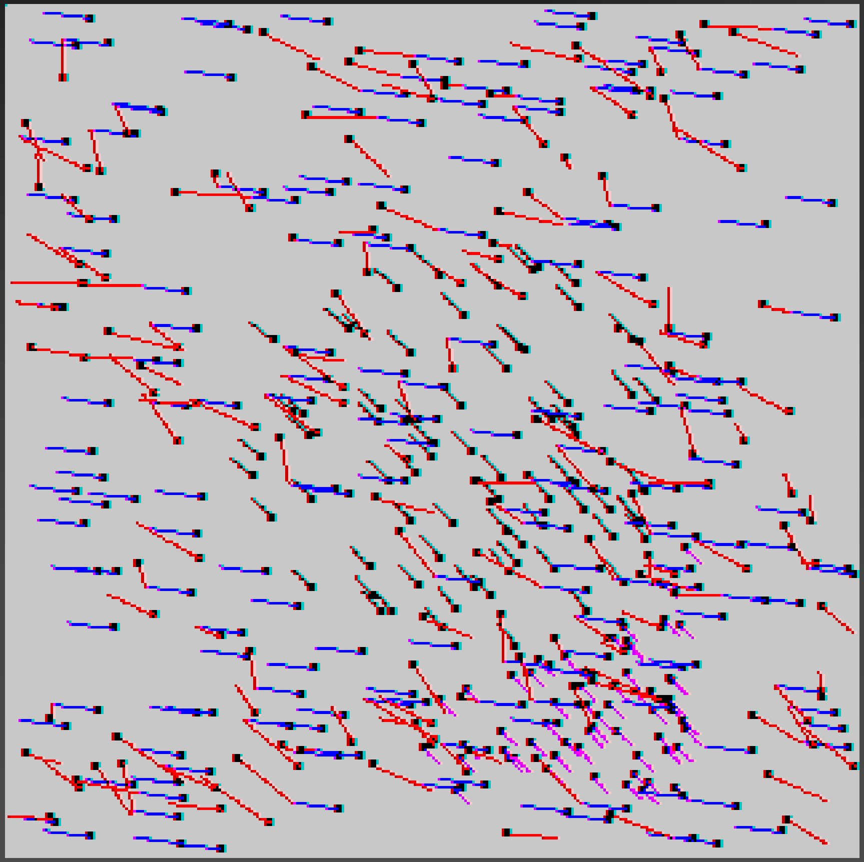
Motion 2 :

Conflicting foreground motion with translation

Motion 3 :

Transparent background motion with translation

Experimental Result(1/2)



Motion 1 :

Foreground motion with rotation and translation

Motion 2 :

Conflicting foreground motion with translation

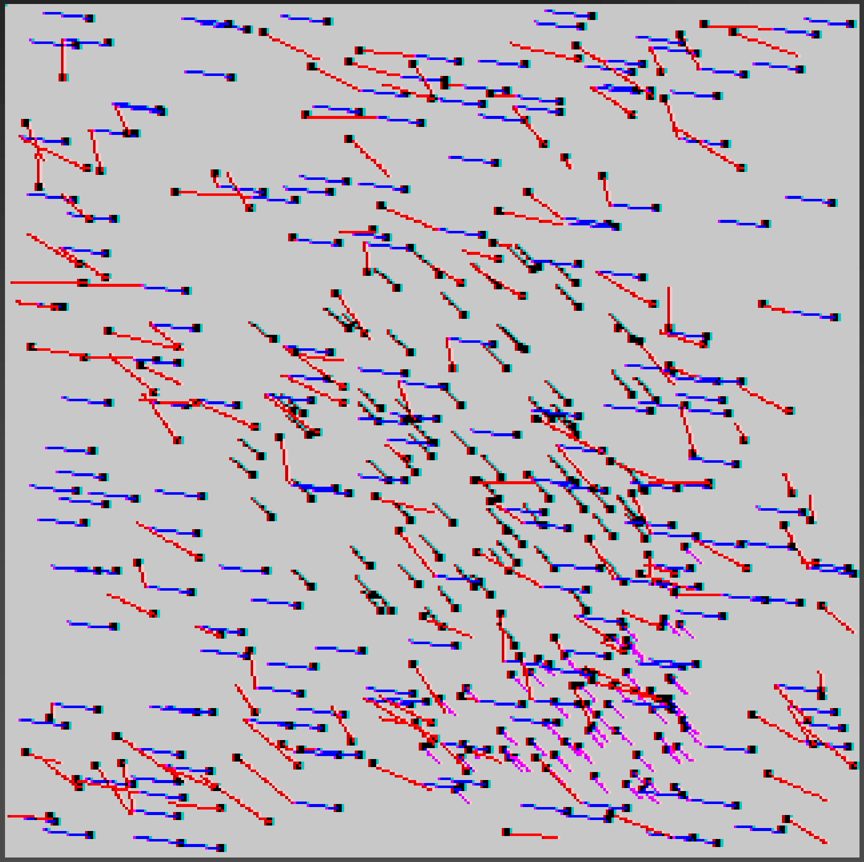
Motion 3 :

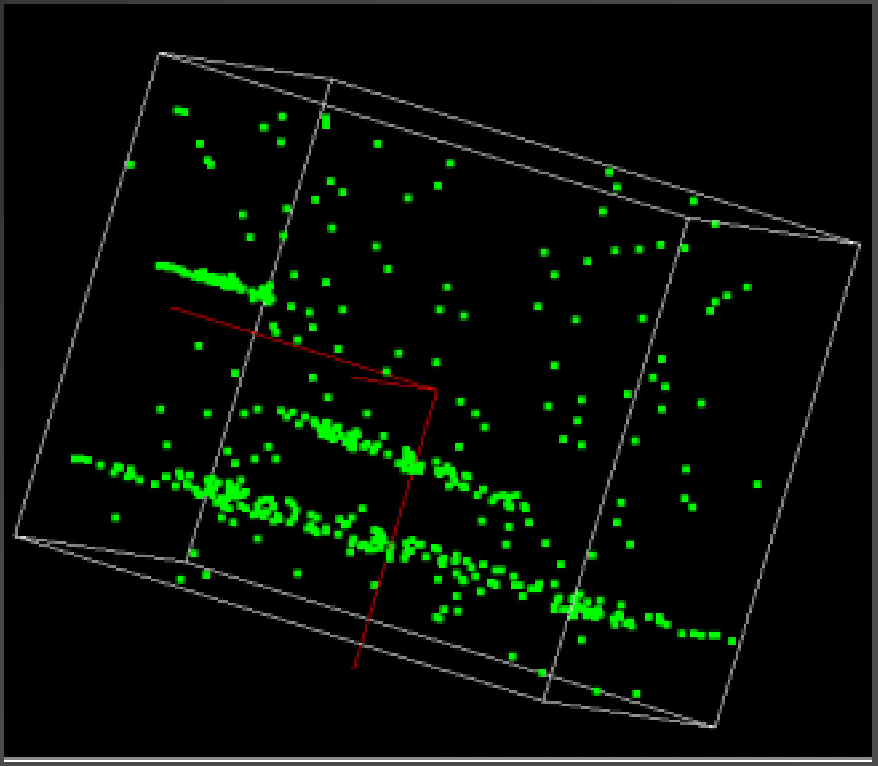
Transparent background motion with translation

Motion 4 :

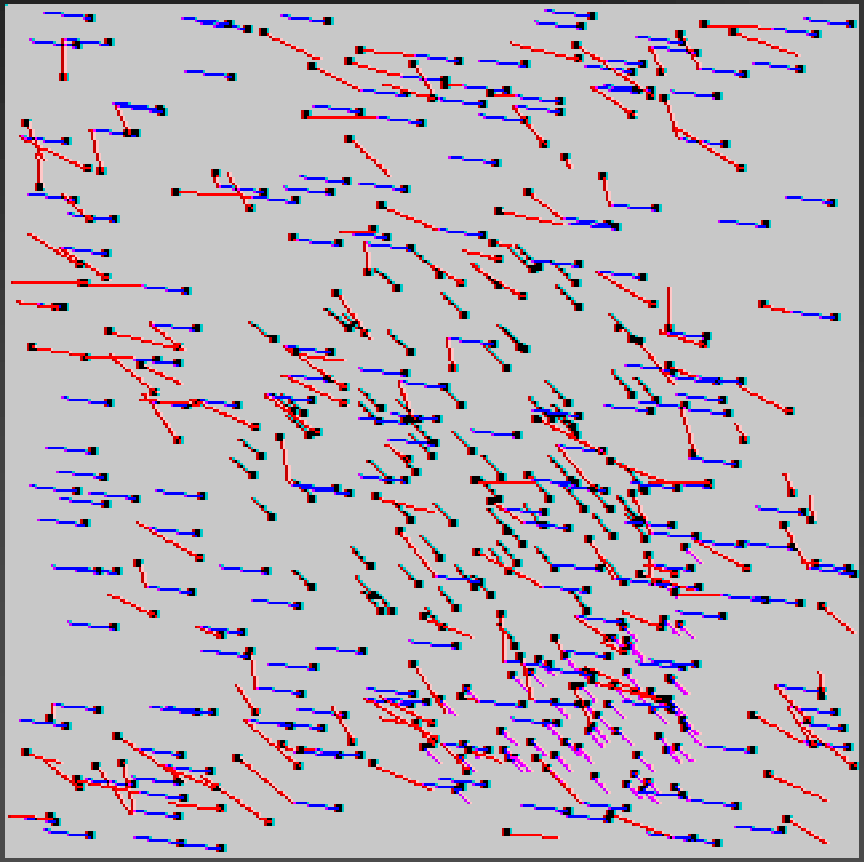
Random motion

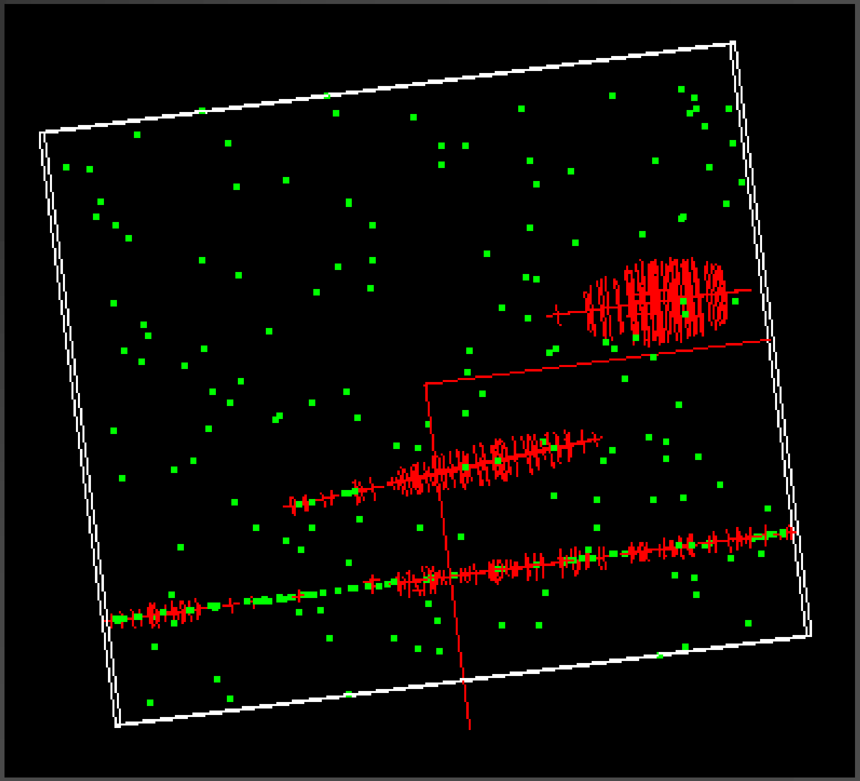
Experimental Result(1/2)



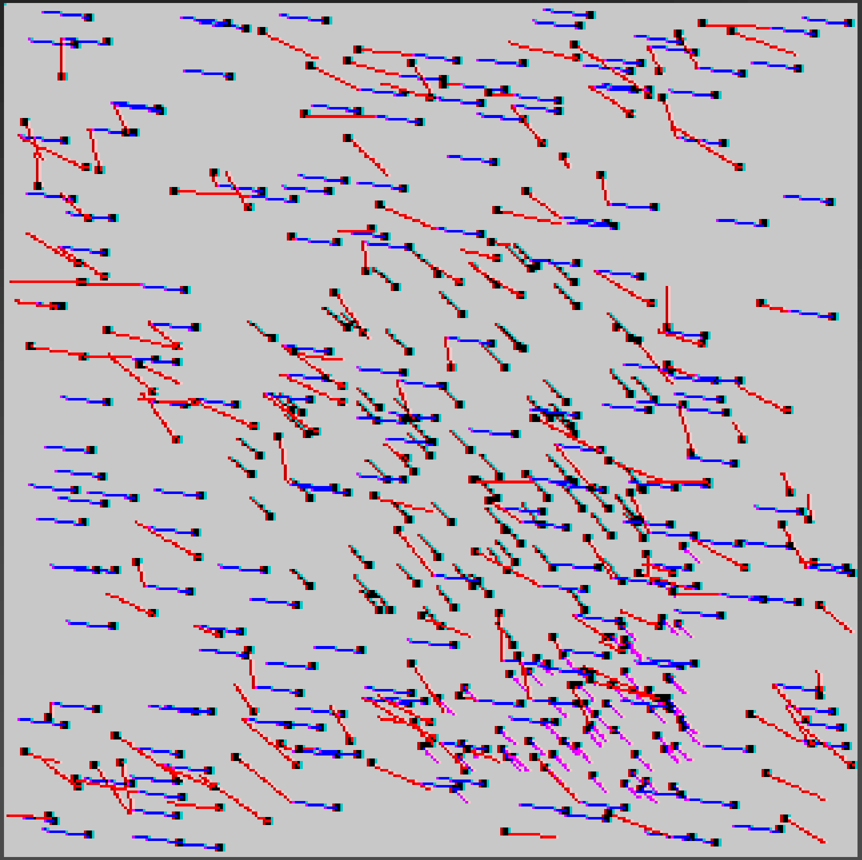


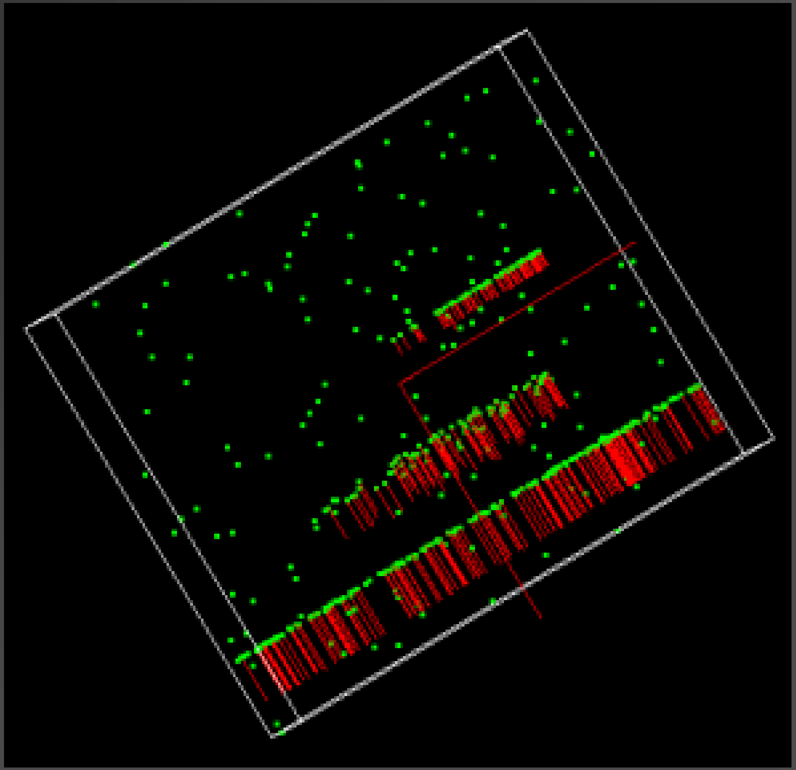
Experimental Result(2/2)





Experimental Result(2/2)



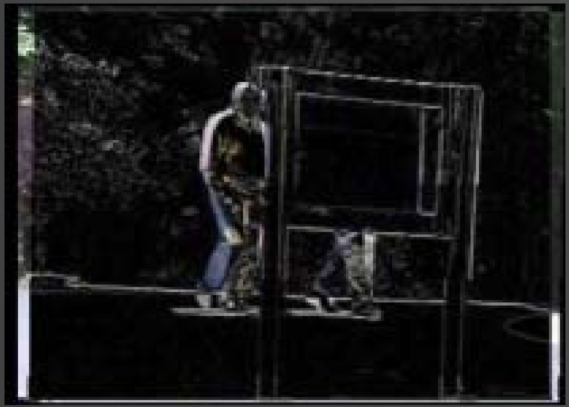


Result (1/3)

Input frames

Initial correlation-based correspondences

Residuals:
by RANSAC(top)
by our method (bottom)



Result (2/3)

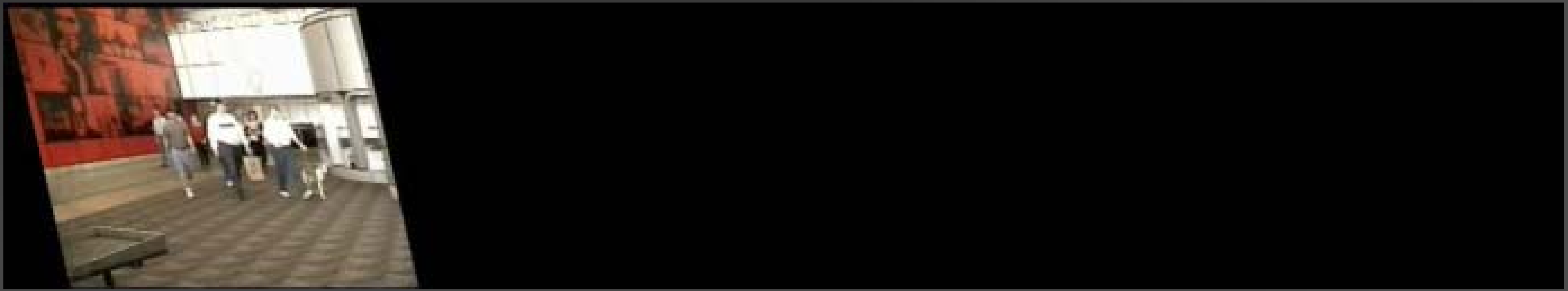
Input frames

Initial correlation-based correspondences

Residuals :
by RANSAC(top)
by our method (bottom)

Result (3/3)

Input Video Sequence

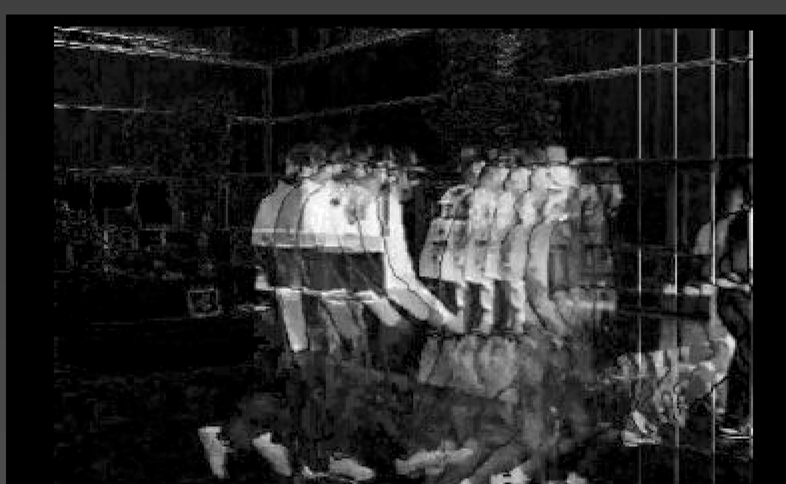


Stabilization by using our method

Closer Look

Input Frames

Accumulated Residuals by Our Method



Accumulated Residuals by RANSAC

Summary and Future Work

- Robust affine parameter estimation in the presence of many mismatches
  - Representation of correspondences in a decoupled joint image space
  - Outlier removal by using Tensor voting
  - Detect multiple motion layers by extracting multiple planes
- Extension to eight parameter projective case
  - Analyze the geometric structure of projective transformation in joint image space

Detection of Moving Objects

Two-step approach:

- ***image stabilization*** to compensate for the motion induced by the observer
- ***moving region detection*** using residual flow in two consecutive frames

Detection of Moving Objects

- Layered representation of scenes
  - Static component of the scene: background layer
  - Moving objects: foreground layers
- Detecting moving objects
  - Color-based and pixel-based temporal distribution
    - Color model : RGB Gaussian pixel-based model
- Layer extraction
  - Background layer : μ
  - Foreground layers : pixels outside of 2σ range

Vehicle Detection

Image sequence

Moving objects

Panning Camera

Hand-held Camera

Entourage Results

Mosaic

Foreground regions

Summary: Moving Blob Detection

- Static cameras
  - Background learning method is effective but can be affected by rapid illumination changes, common in some indoor videos
- Moving Cameras
  - Ego-motion computation effective when scene objects are far or camera motion has no translation

Tracking Moving Regions

- ***Tracking*** is performed by establishing correspondence between detected regions
  - Region similarity approach
  - Tensor Voting perceptual grouping approach
  - Using Geometric context

Tracking Moving Regions

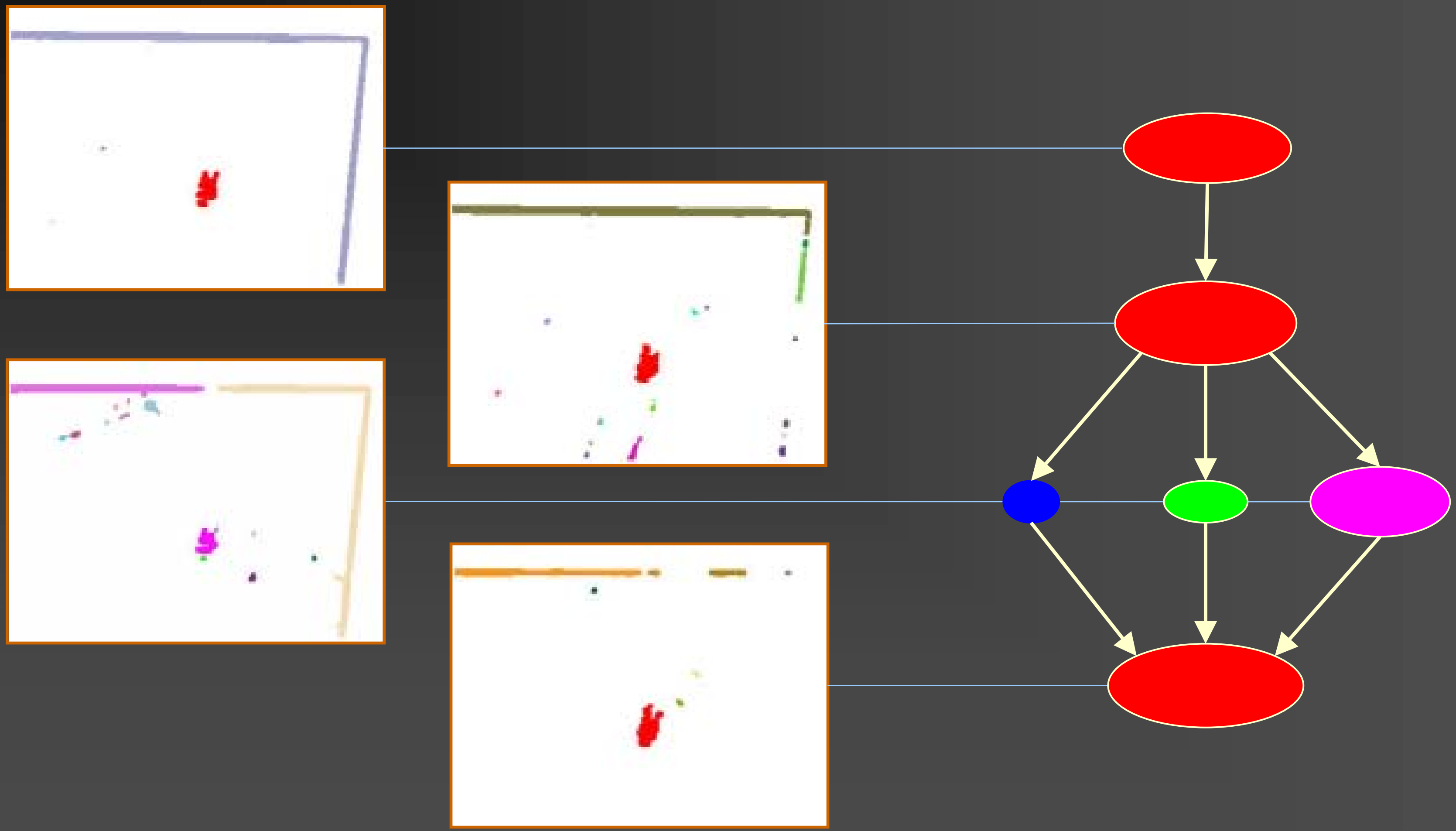
- Detection is performed using two consecutive frames
- Tracking is performed by establishing correspondence between these regions
  - template inference
  - temporal coherence
  - temporal integration

Graph Representation of Moving Regions

- *A node*:
  a detected moving region

- *An edge*:
  a possible match between two regions

Graph Construction



Objects Trajectories

- Tracking is performed by establishing correspondences between these regions
  - template inference
  - temporal coherence (>5 frames)
- Implemented as a graph search:
  - node: moving region
  - edge: between matched regions

Objects Trajectories

- Extract optimal path along each graph's connected components
  - Multiple objects tracking
  - Temporal integration
- Optimality criterion:
  - Associate to each edge a cost
  - Characterize optimal paths

Optimality criterion

- Associate to each edge a cost:

$$c_{ij} = \frac{C_{ij}}{1 + d_{ij}^{\ 2}}$$

  - similarity between the regions:
    - correlation C_{ij}
    - shape, gray or color distribution...
  - proximity between regions:
    - distance d_{ij}
- Local optimum

USC

Optimality criterion: Temporal Integration

- Characterize each node by its length:
  - length of the maximal path ending at this node

$$l_i = \max\left\{ l_j,\, j \in successor(i) \right\} + 1$$

- Associate to each edge the cost:

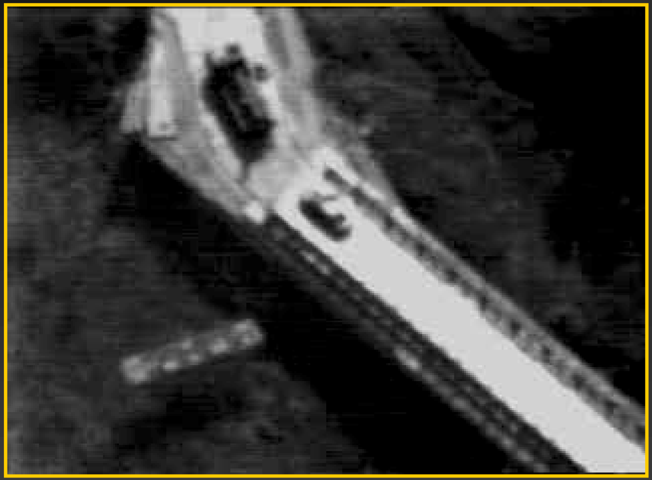
$$\Gamma_{ij} = l_j c_{ij}$$

Vehicle Tracking

Image sequence

Vehicle trajectory

Human Tracking



Objects trajectories

USC

Convoy Tracking

Region Similarity Approach

- Regions are matched based on similarities
  - Gray level distributions, Distances
- Graph representation of moving regions
  - Node: Detected moving region
  - Edge: A possible match between two regions
- Inference of median shape from graph
  - Temporal coherence
- Extract optimal paths along connected components of the graph
  - Multiple objects tracking

Multi-Object Tracking

- Graph description well suited for:
  - Complete description of moving objects properties
  - Identifying temporal changes of objects:
    - occlusions
  - Tracking arbitrary number of moving objects

Multi-object Tracking

Tracking by Similarity Graph

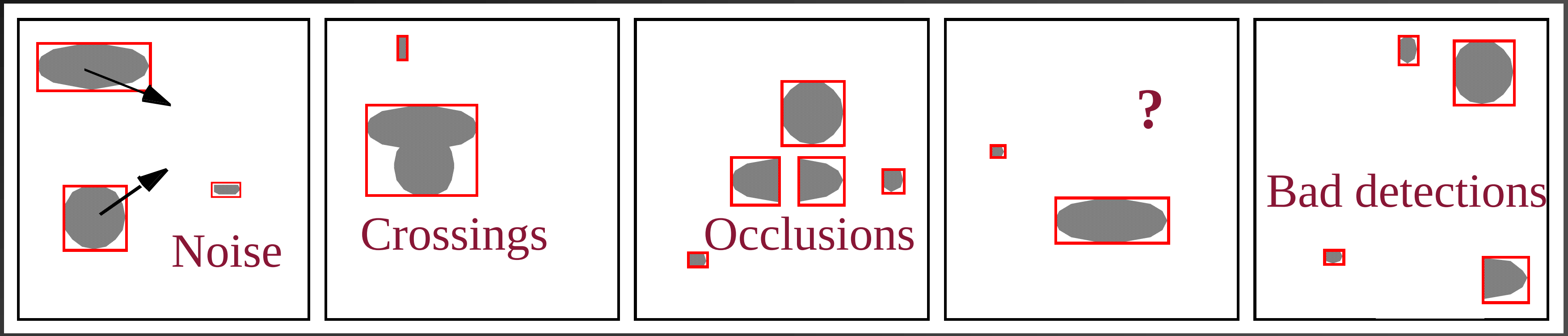
- Detected regions are matched based on similarities
  - Gray level distributions, distances in the image
- Graph representation of matched regions
  - Node: detected moving region
  - Edge: A possible match between two regions
- Extract optimal paths along connected components of the graph
  - Allows for multiple object tracking

Tracking Moving Blobs

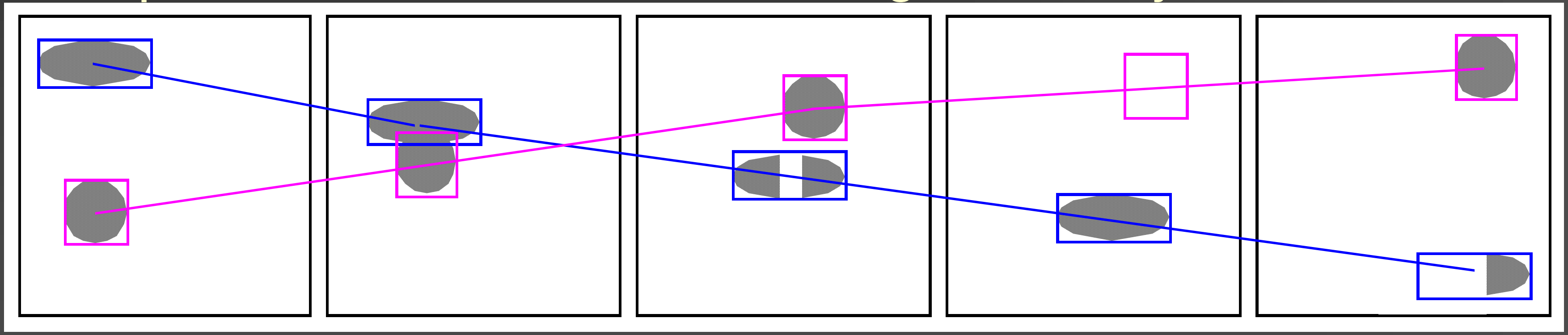
- Blobs split and merge due to occlusion, similarity with background, noise blobs appear…
- Use of region similarities and trajectory smoothness can help infer good trajectories
- Perceptual grouping using graph representation and tensor voting

Some Tracking Problems

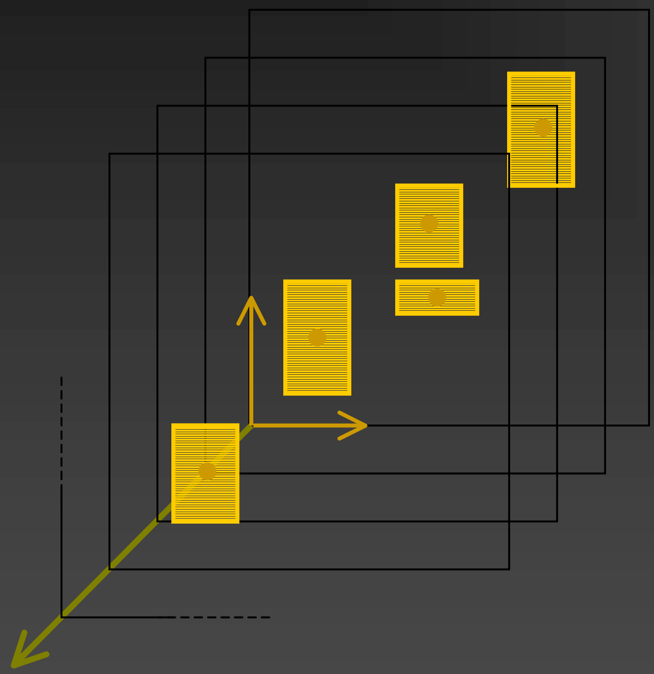
- Input: A set of moving regions

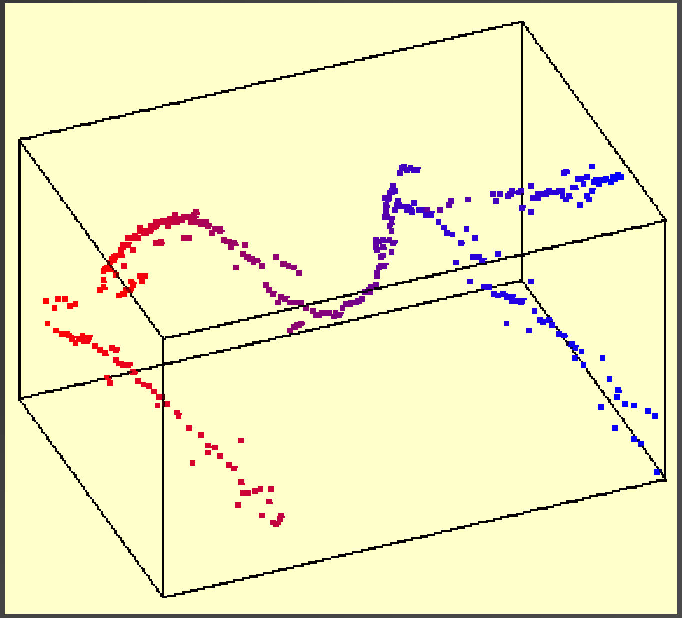


- Output: Identification and Tracking of the objects



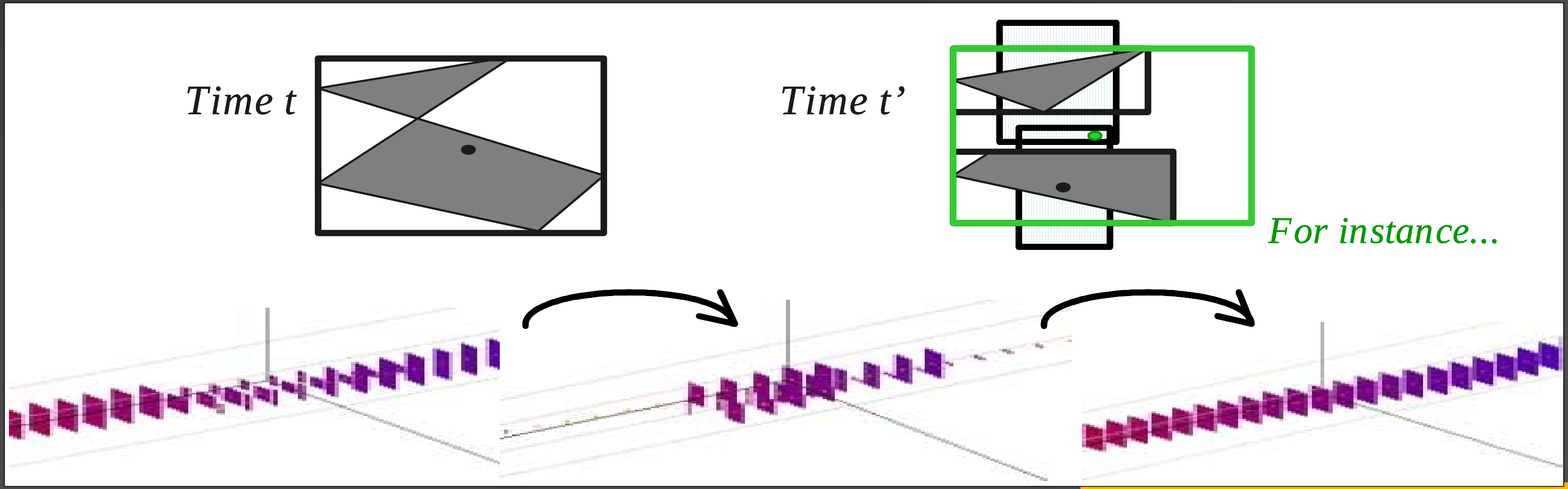
2D+t Representation





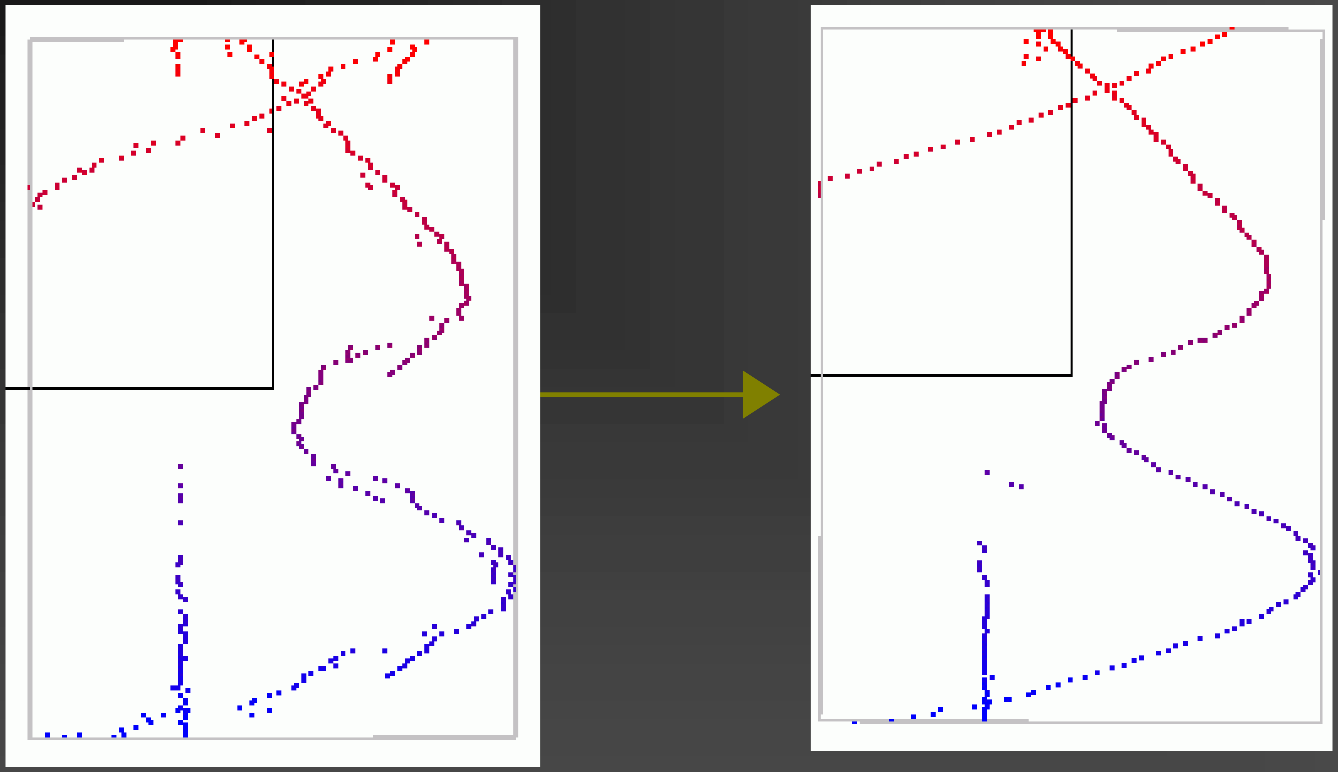
Tracking by Tensor Voting

- Propagate motion and shape information into uncertainty regions around paths in the graph
- Accumulate votes by tensor voting process
- Includes process for merging and splitting of blobs to yield smooth trajectories



USC

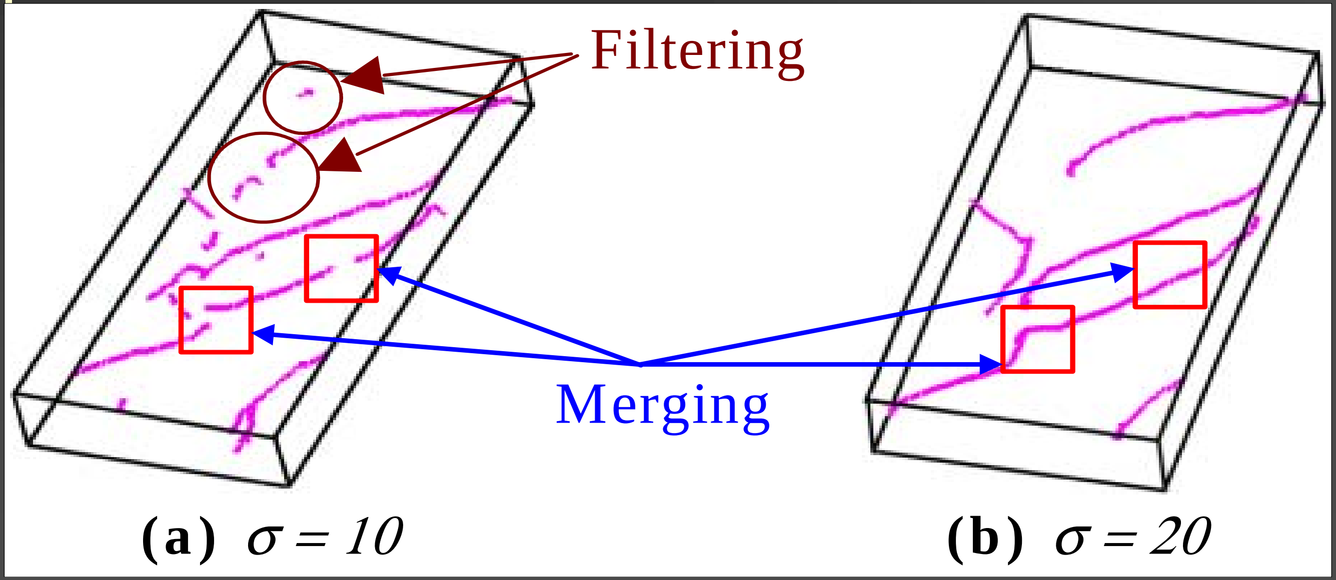
Grouping Example



Multi-Scale Processing

Gap completion depends on the scale of the voting field (σ)

- Larger σ provides more filtering and continuity but is more expensive and tracks are less precise



Multiple Objects Tracking

Tracking with multiple cameras

- Motivation
  - Multiple cameras can provide larger field of view and reduce effects of occlusion
- Issues
  - Registration of multiple views
    - Calibration of different cameras
  - Synchronization of different video streams
    - Different frame rates
  - Grouping of trajectories across views

Approach

- Infer trajectories in each camera view
  - Tracking by Tensor Voting
- Register cameras views using a homography
  - Use of the ground plane
  - Misalignment can occur if trajectories are not on the ground plane or the streams are not synchronized
- Synchronize the multiple video streams
  - Register and merge trajectories in 2D+t using a homography
- Current implementation not real-time (~1 fps)

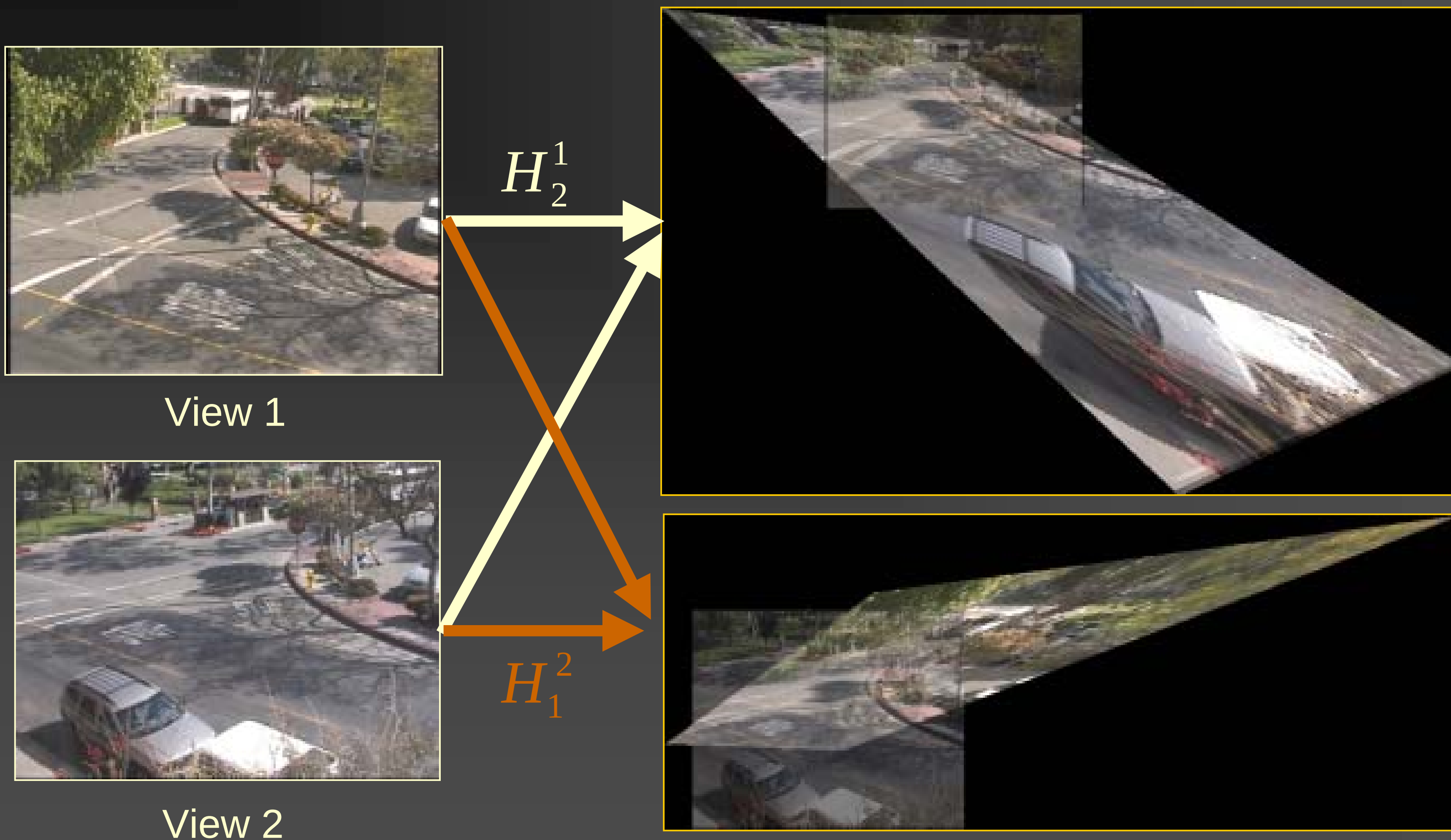
Two Video Streams

Two partially overlapping views

Registration of Multi Camera

- Registration of views using ground plane
  - Use of a projective transform
  - Requires at least 4 pairs of corresponding points on the ground plane

Cameras registration

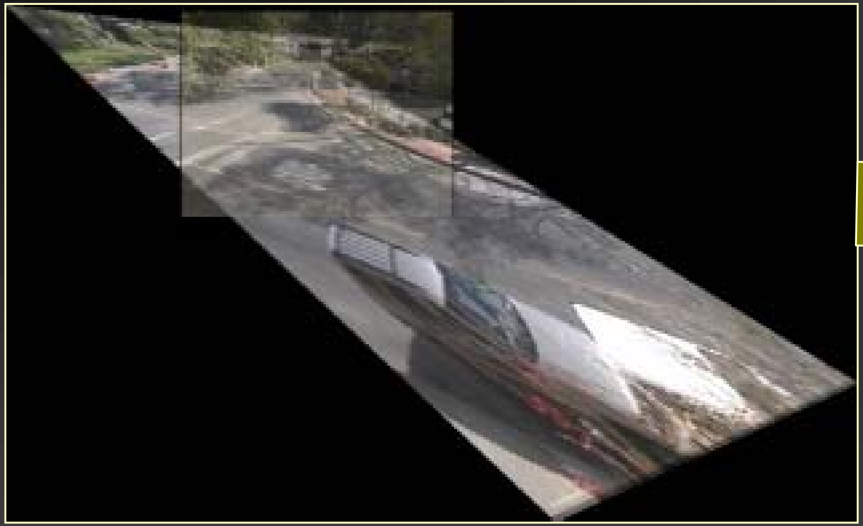


Registration of Trajectories

- Use of a homography for registering cameras views
  - Misalignment can occur if:
    - Trajectories are not on the ground plane
    - Input streams are not synchronized
    - Shutter speeds of cameras are different
- Use of objects trajectories for registering the views
  - Videos synchronization using observed trajectories
  - Align objects trajectories

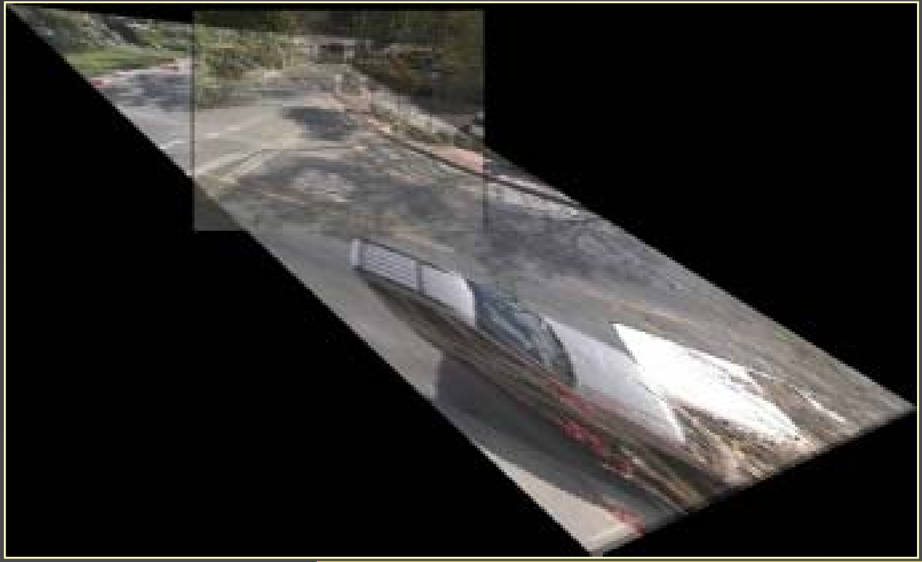
Registration of Video Streams

Projective Registration



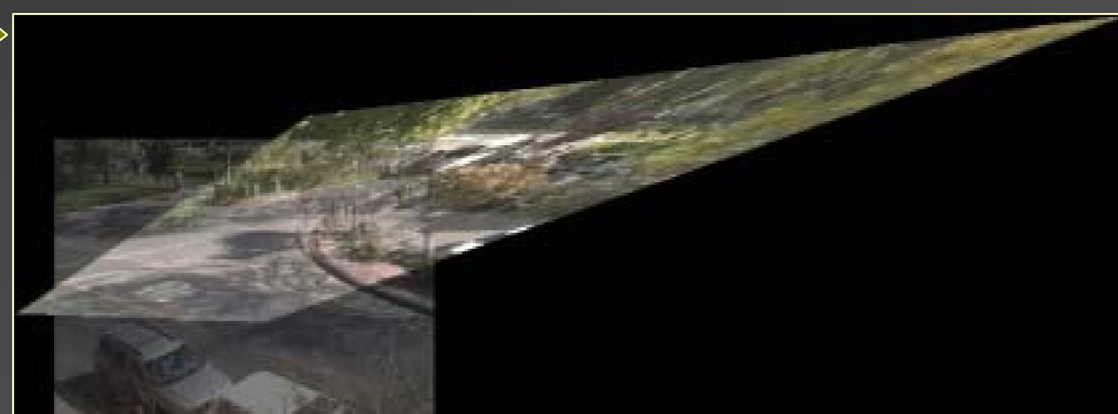
Registered Views

Registered Trajectories



Reverse View

Registered Views



Registered Trajectories

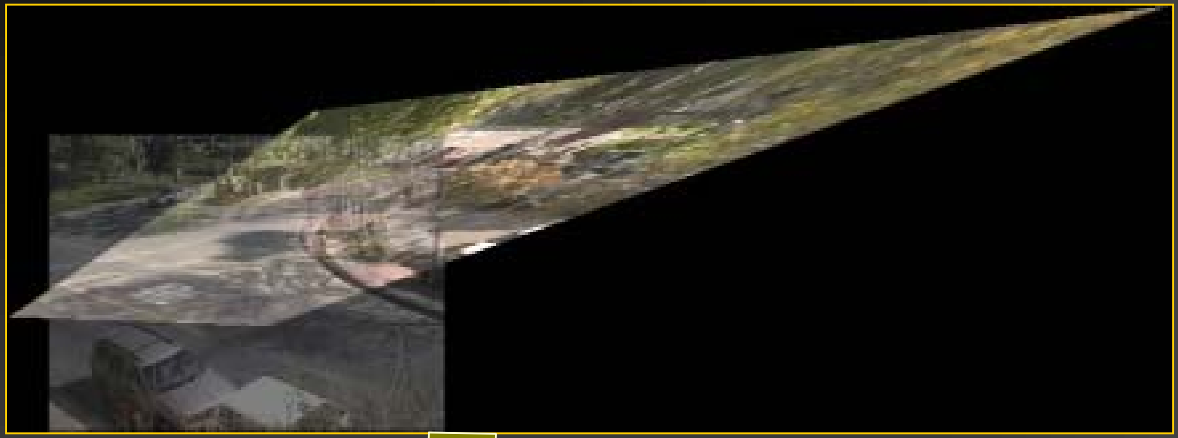
Another Example

Projective Registration

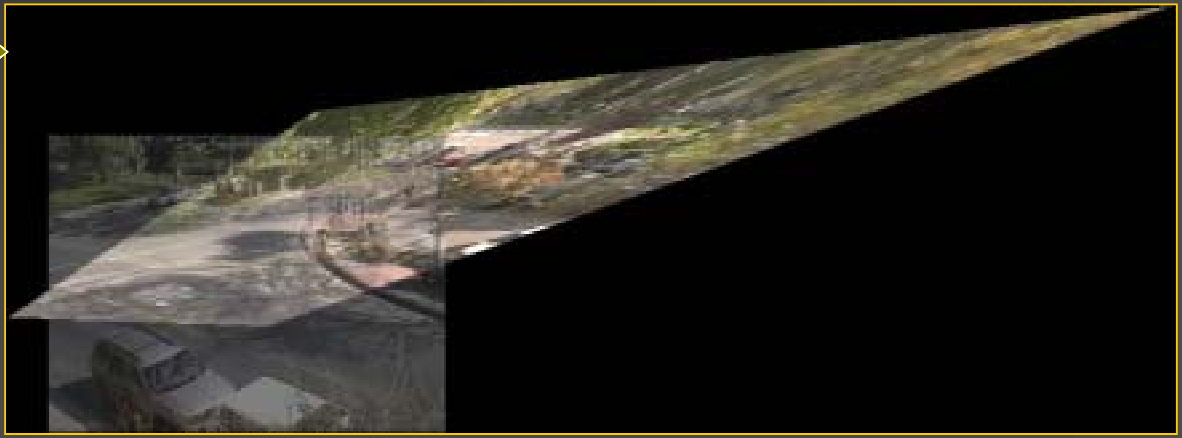
Registered Views

Registered Trajectories

Reverse View



Registered Views



Registered Trajectories

Tracking Using Multiple Cameras: Summary

- Ability to merge trajectories across views
  - Reduces effects of occlusions
  - Increases the field of view
  - Speed: 1 to 1.5 Frames per seconds

- Future Extensions
  - Inference of the 3D tracks
    - Use camera calibration from vanishing points
    - Use multi-view geometry
  - Automatic synchronization of trajectories
    - Accommodate different shutter speeds

Summary: Blob Tracking

- Real-time tracking with hierarchical approach and graph-based methods
  - Region similarity exploited but not continuity
- Perceptual grouping using tensor voting
  - Slower, but better tracks
- Merging multiple camera tracks is in early development stage
- Future work
  - Improve efficiency of perceptual grouping
  - Develop adaptive multi-scale approach
  - Integrated detection and tracking from multiple cameras

Essay #1

- We have presented two detection algorithms:
  - Background learning method for stationary cameras
  - Residual motion for moving platforms

Describe these algorithms and suggest an approach for combining the two (I.e. a background learning method for moving platforms).

Essay #2

- Various methods are used for estimating camera motion: Direct parameter estimation, RANSAC…

  Can you describe the joint image space algorithm and explain why registering two images amounts to identify planar patches in the joint image space

Essay #3

- Describe the tracking algorithms presented:
  - Graph based blob tracking
  - Perceptual grouping based tracking

And explain what are the advantages in combining both for tracking moving objects in a scene.