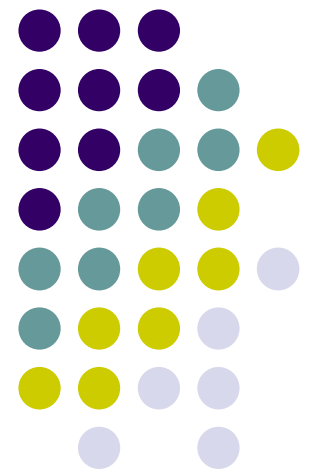


# Towards Surface Image Based Vehicle Tracking

---

Discussion of the first passages  
in the maze towards a PhD

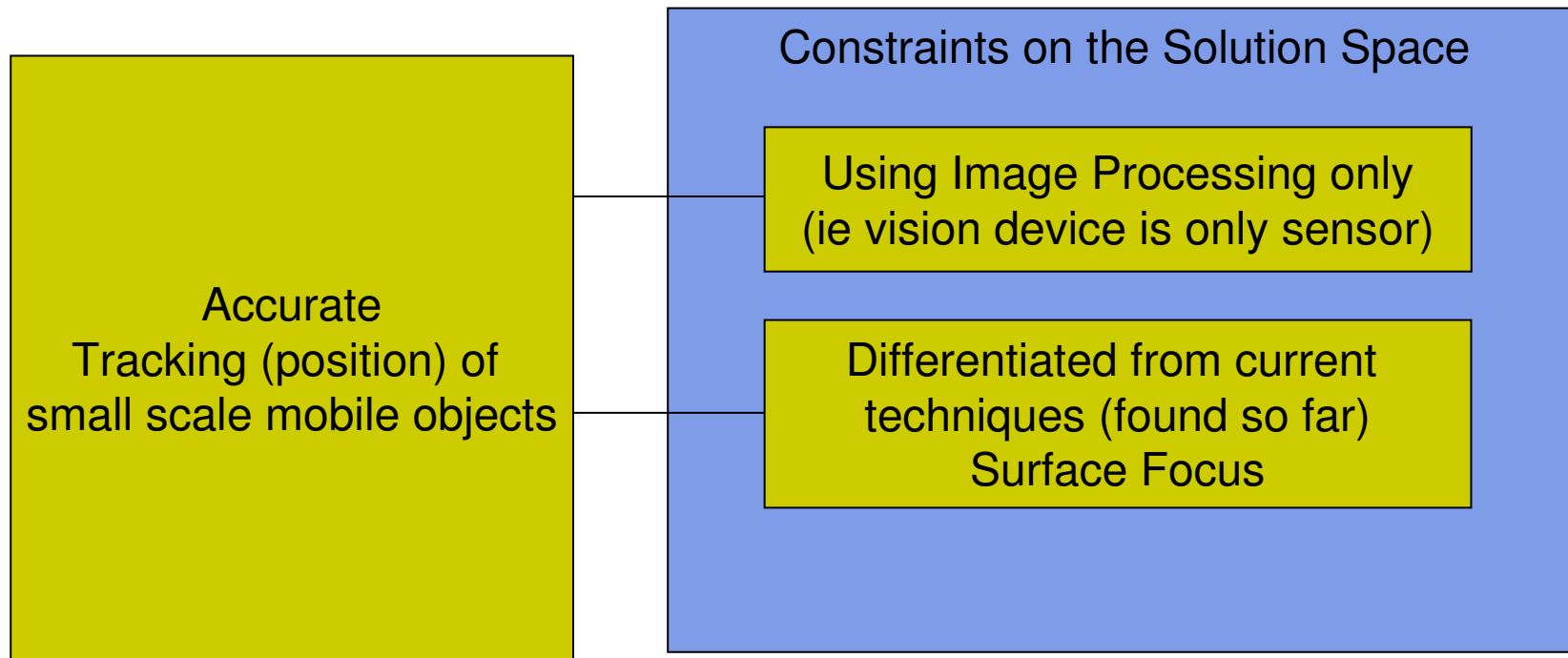
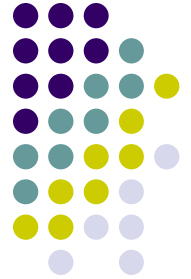




# Agenda

- Problem Statement
- Image Processing Background
- Fixed image plane distance (FIPD) motion tracking
- Experimental Solution
- Results
- Conclusion
- Future Work

# Global Problem Statement



# A survey of hardware techniques

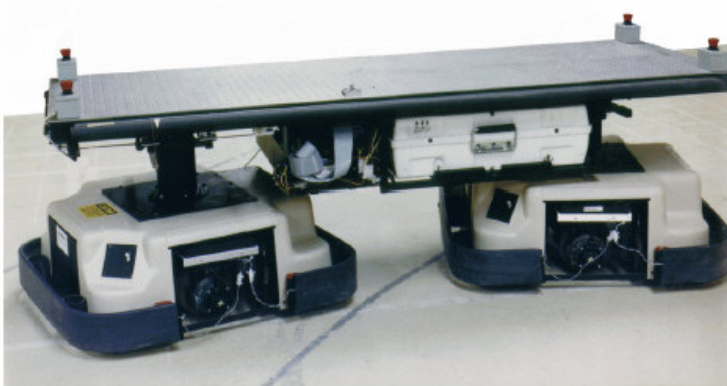


- I: Relative Position Measurements (also called Dead-reckoning)
  - Odometry
  - Inertial Navigation
- II: Absolute Position Measurements (Reference-based systems)
  - Magnetic Compasses
  - Active Beacons
  - Global Positioning Systems
  - Landmark Navigation
  - Model Matching

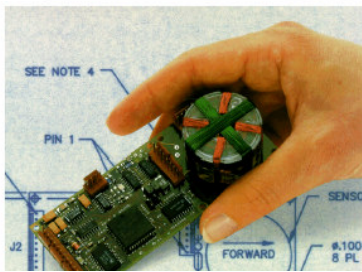
Borenstein et al, 1997



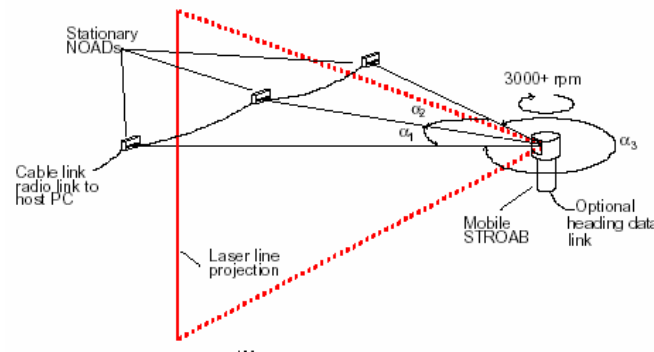
# Some old pictures



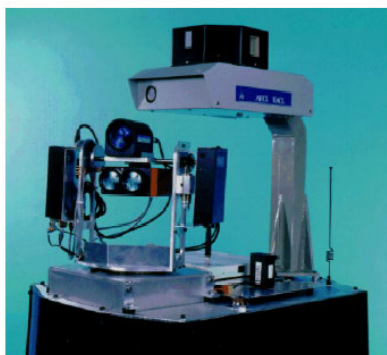
**Figure 7:** A single STROAB beams a vertically spread laser signal while rotating at 3,000 rpm. (Courtesy of MTI Research)



commercially available fully omnidirectional "trucks" mutually correct their



**Figure 8:** Stationary NOADs are located at known positions; at least two NOADs are networked and connected to a PC. (Courtesy of MTI Research, Inc.)



Borenstein et al, 1997



# Comparison Table

| System & Description  | Features                                   | Accuracy – Position [mm]   | Accuracy – orientation [°]                                   | Effective Range | Reference                               |
|---|--|--|--|-----------------|---|
| Odometry on TRC LabMate, after UMBmark calibration. Wheel-encoder resolution: 0.012 mm linear travel per pulse  |  | 4×4 meters square path:<br>smooth floor: 30 mm,<br>10 bumps: 500 mm            | Smooth floor:<br>1-2°.<br>With 10 bumps: 8°                  | Unlimited       | [Borenstein and Feng, 1995]             |
| CLAPPER and OmniMate:<br>Dual-drive robot with internal correction of odometry. Made from two TRC LabMates, connected by compliant linkage. Uses 2 abs. rotary encoders, 1 linear encoder.  |  | 4×4 m square path:<br>smooth floor: ~20 mm<br>10 bumps: ~40 mm                 | smooth floor: <1°<br>10 bumps: <1°                           | Unlimited       | [Borenstein, 1995; 1996]                |
| Complete inertial navigation system including ENV-O5S Gyrostar solid state rate gyro, START solid state gyro, triaxial linear accelerometer and 2 inclinometers                             |  | Position drift rate:<br>1-8 cm/s depending on frequency of acceleration change | Drift: 5-0.25°/s.<br>After compensation drift 0.0125°/s      | Unlimited       | [Barshan and Durrant-Whyte, 1993; 1995] |
| Andrew Autogyro and Autogyro Navigator. Quoted minimum detectable rotation rate: ±0.02°/s. Actual minimum detectable rate limited by deadband after A/D conversion: 0.0625°/s. Cost: \$1000 |  | Not applicable   | Drift: 0.005°/s  | Unlimited       | [ANDREW]                                |
| KVH Fluxgate Compass. Includes microprocessor-controlled fluxgate sensor subsystem. Cost <\$700   |  | Not applicable   | Resolution: ±0.5°<br>Accuracy: ±0.5°<br>Repeatability: ±0.2° | Unlimited       | [KVH]                                   |
| CONAC™ (computerized opto-electronic navigation and control). Cost: \$6,000.  | Measures both angle and distance to target | Indoor ±1.3 mm<br>outdoor ±5 mm  | Indoor and outdoor ±0.05°                                    | > 100 m         | [McLeod, 1993]; [MTI]                   |
| Global Positioning Systems (GPS). Cost: \$1,000 - \$5,000.  |  | order of 20 m during motion, order of centimeters when standing for minutes    | Not applicable   | Unlimited       | Different vendors                       |
| Landmark Navigation   |  | <5 cm  | < 1 deg  | ~10 m           | Different research projects             |
| Model Matching (map-based positioning)  |  | order of 1-10 cm   | order of 1-3 deg   | ~10 m           | Different research projects             |

Borenstein et al, 1997

# Survey of all Image Processing techniques



- Still looking for a good one
- Seems like a worth while exercise - any volunteers?
- Existing techniques found
  - Some look at “off-board” object tracking in video (i.e. camera looking at scene and tracking objects in it)
  - Robot vision techniques process video of surrounding environment.
    - Beacons are found by segmentation, vertical edges, etc
    - Form of model matching/mapping

# Image Processing Background



- Motion from 2D images
- Block Match Algorithm





# Motion from 2D images

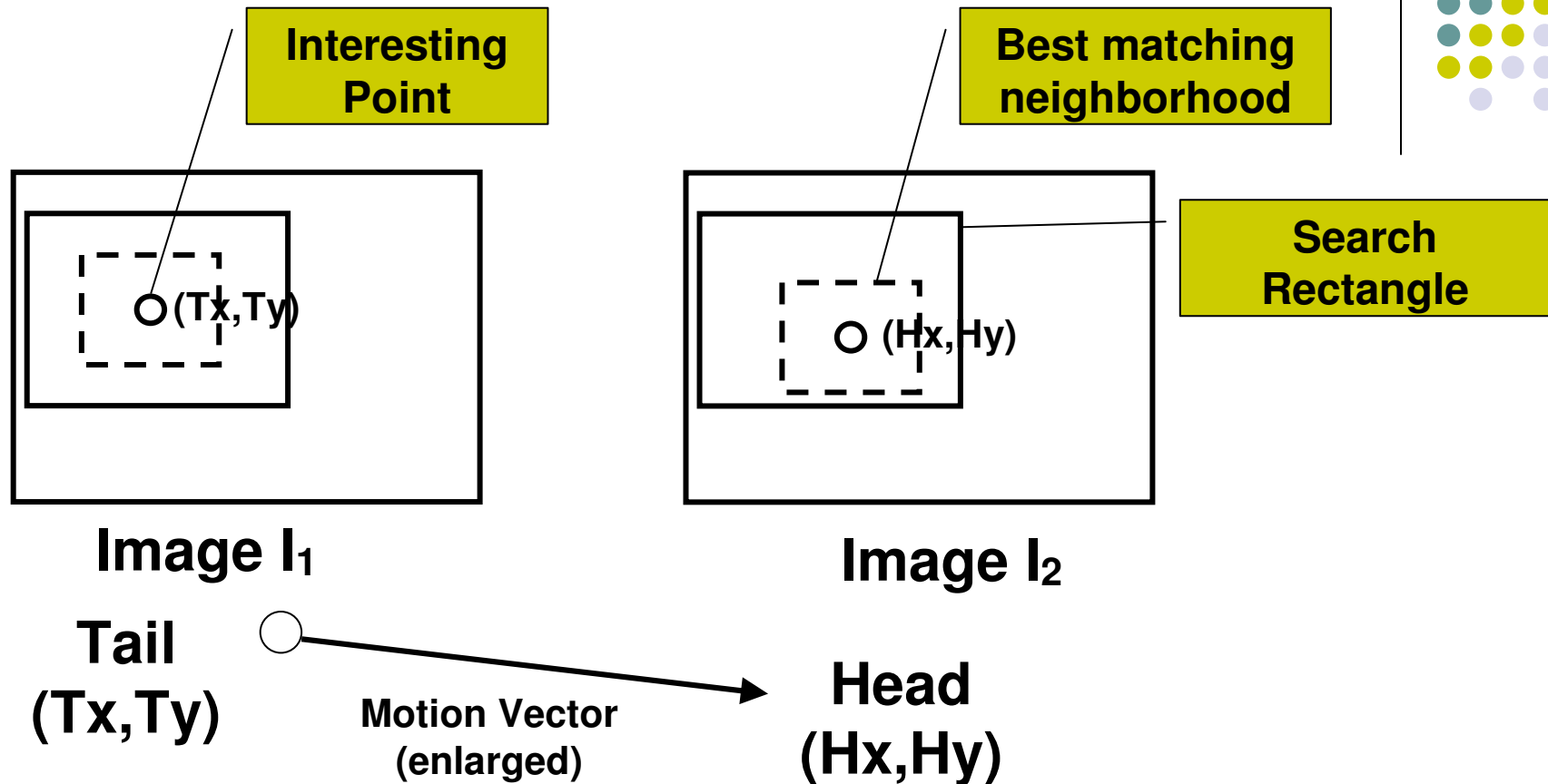
- Motion vectors
  - **Motion field:** 2D array of 2D vectors representing motion of 3D scene points. A vector in the motion field has one of the following two meanings:
    - The tail represents the position of an imaged 3D point at time  $t$  and the head represents the position of the corresponding 3D point at time  $t + \Delta t$
    - The vector represents an instantaneous velocity at time  $t$
  - **Image flow** is the motion field computed under the assumption that image intensity near corresponding points is relatively constant



# Computing Motion Vectors

- Interest Points, eg
  - Centroids of moving regions of segmented image
  - Corner points found from an edge operator
- Interest Operator
  - Thresholded intensity variance through a point  $P$
  - Using texture – variance in an entire  $n \times n$  neighborhood
- To find motion vectors: Given interesting points  $\{P_j\}$  in  $I_1$  at time  $t$ , corresponding points must be identified in  $I_2$  at time  $t + \Delta$ .

# Computing Motion Vectors



For each interesting point  $(T_x, T_y)$  in  $I_1$  a rectangular region of Image  $I_2$  is searched for the best match to a small neighborhood of  $(T_x, T_y)$ . If the match is good, then it becomes the head  $(H_x, H_y)$  of a motion vector



# Computing Motion Vectors

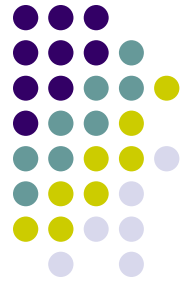
- Cross-Correlation
  - Cross-Correlation of image  $F[x,y]$  and mask  $H[x,y]$  is defined as

$$\begin{aligned} G[x,y] &= F[x,y] \otimes H[x,y] \\ &= \sum_{i=-w/2}^{w/2} \sum_{j=-h/2}^{h/2} F[x+i,y+j]H[i,j] \end{aligned}$$

where  $w$  and  $h$  represents the width and height of the mask

- The minimum value for  $G$  in the search rectangle discussed above gives the best match.

# Block Match Algorithm



- Each frame divided into fixed size macro blocks (MB)
- In MPEG and H.263 each MB is 16X16 pixels
- The search region (SR) is rectangular region larger than the macro blocks.
- For a SR formed by displacing the MB by +/- 15 pixels, a full search (FSA) of the SR to find the best matching block gives  $(15 \times 2 + 1)^2 = 961$  search points (SP).
- Matching per SP is done using the sum of absolute differences (SAD)

$$\text{SAD}[x,y] = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} | I_1[x+i,y+j] - I_2[i,j] |$$



# Optimised BMA

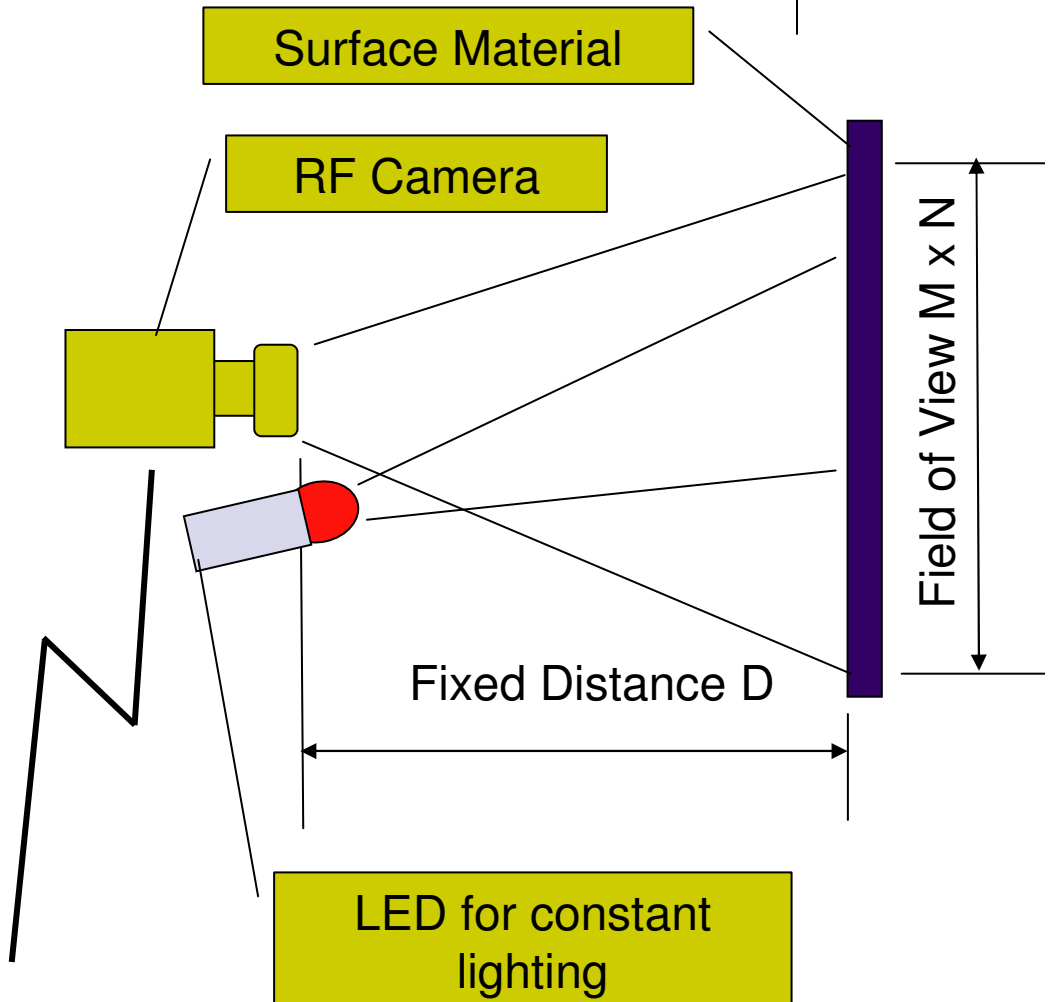
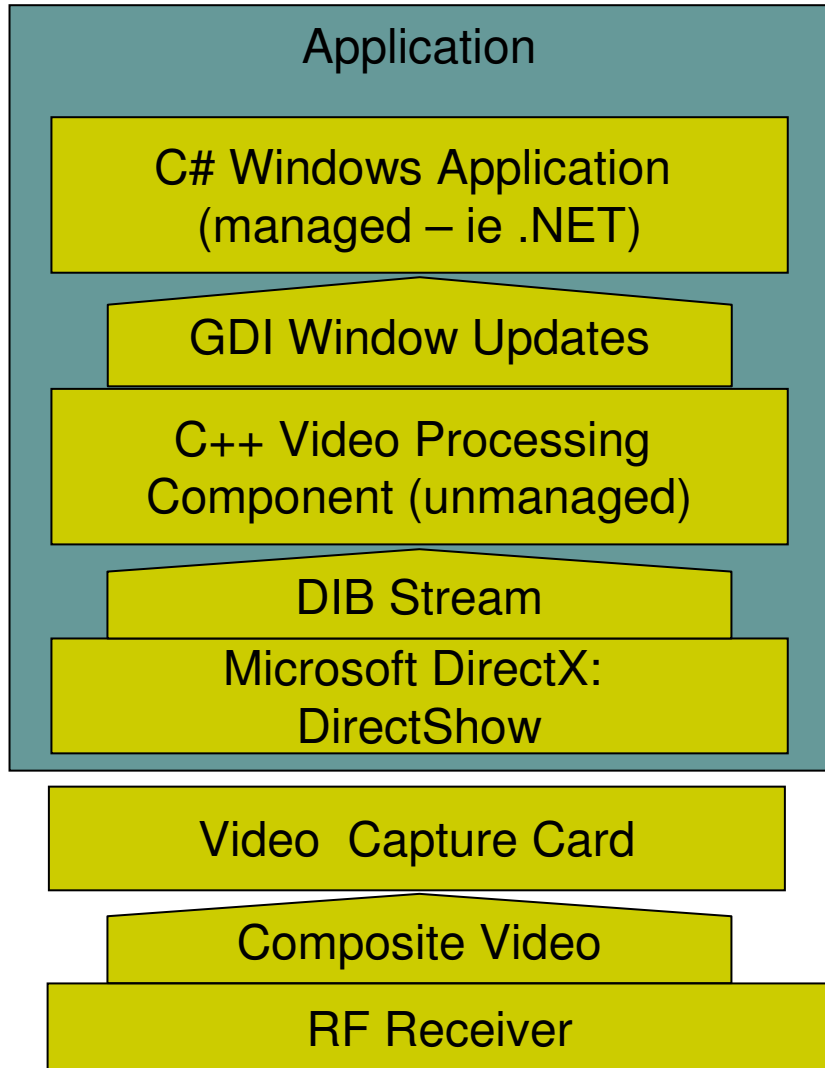
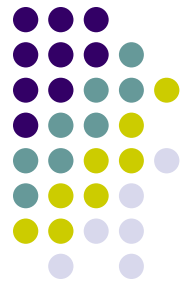
- Using SAD, the amount of RISC-like computations per 16x16 MB for FSA (961 SPs) is:  
 $(2 \times 15 + 1)^2 (2 \times 16^2 - 1) = 491071$
- To reduce the computational effort, non-exhaustive search methods have been developed
  - Three Step Search (TSS) (33 SPs)
  - Modified Motion Estimation Algorithm (MMEA) (25 SPs)
  - Independent Orthogonal Search Algorithm (IOSA) (17 SPs)

# Fixed image plane distance (FIPD) motion tracking



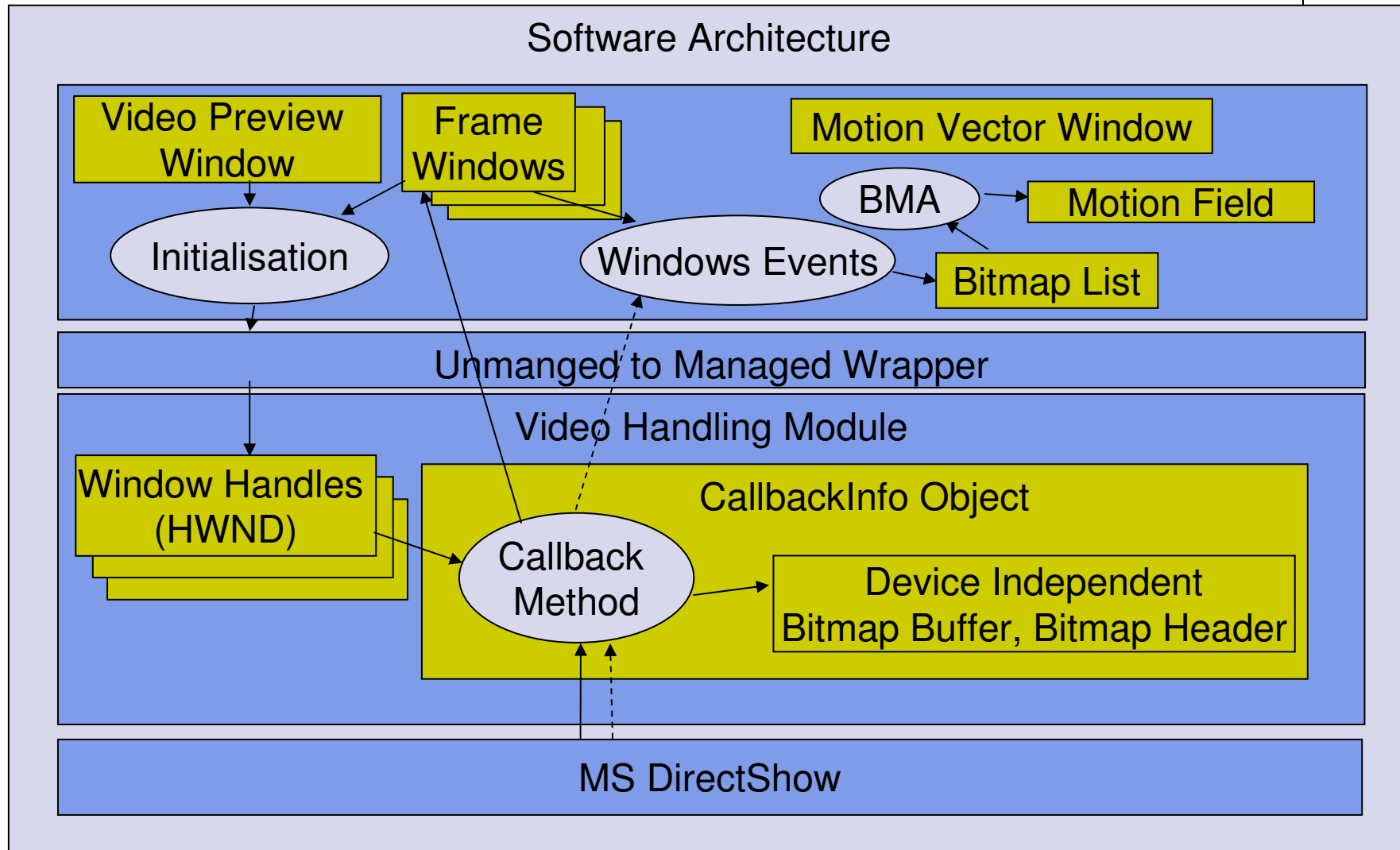
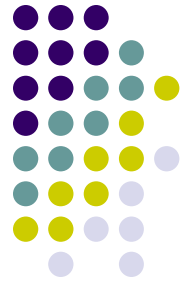
- Coming back to the objective: Surface image motion tracking
- Fixing the distance from surface plane to image sensor plane makes it possible to transduce image motion vectors to physical distance travelled by the object that the sensor has been attached to.
- Related to dead reckoning techniques to track robot motion in the class of odometry

# Experimental Solution





# Experimental Solution



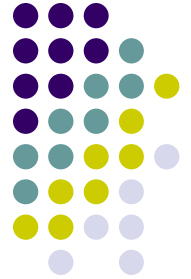
# Application Output



The screenshot shows the 'frmMain' application window with several key components:

- User Input: Geometry of MB**: A control panel with input fields for X (0), Y (0), W (64), and H (64), and a 'GetFrames' button.
- LED Flood light area**: A circular region of interest on the video preview, highlighted in red.
- Video Preview**: A large window showing a video frame with a patterned background.
- Best Motion Vector Window**: A window displaying a single red dot representing the best motion vector.
- Frame Sequence**: A grid of six frames labeled F1 through F6, showing the sequence of frames used for motion vector calculation.
- Output Data**: A list of motion vector parameters: x1: 32, y1: 32, x2: 32, y2: 32, dX: 0, dY: 0.

# Results: F1 → F2



frmMain

$dX = -2$   
 $dY = 0$

X 0 Y 0 W 64 H 64

GetFrames

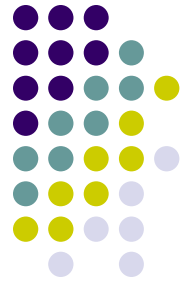
F1 → F2 F3

F4 F5 F6

|    |    |
|----|----|
| x1 | 32 |
| y1 | 0  |
| x2 | 30 |
| y2 | 0  |
| dX | -2 |
| dY | 0  |

ComputeMotion

# Results: F2 → F3



frmMain

$dX = -4$   
 $dY = 0$

X 0 Y 0 W 64 H 64

GetFrames

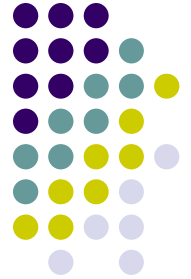
F1 F2 → F3

F4 F5 F6

|    |    |
|----|----|
| x1 | 32 |
| y1 | 0  |
| x2 | 28 |
| y2 | 0  |
| dX | -4 |
| dY | 0  |

ComputeMotion

# Results: F3 → F4



frmMain

$dX = -4$   
 $dY = 0$

X  Y  W  H

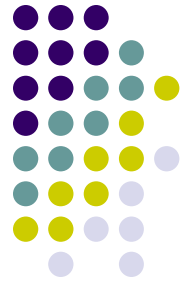
GetFrames

|    |    |    |    |    |
|----|----|----|----|----|
| F1 | F2 | F3 | x1 | 32 |
|    |    |    | y1 | 0  |
| F4 | F5 | F6 | x2 | 28 |
|    |    |    | y2 | 0  |
|    |    |    | dX | -4 |
|    |    |    | dY | 0  |

ComputeMotion



# Results: F4 → F5



The screenshot shows a software window titled "frmMain" with a blue title bar. The main area contains a large image of a patterned surface. A yellow box highlights the displacement values:  $dX = -2$  and  $dY = 0$ . Below this, there are input fields for X (0), Y (0), W (64), and H (64), and a "GetFrames" button. A grid of six frames (F1-F6) is shown, with an arrow pointing from F4 to F5. A table of coordinates is displayed, with the displacement values circled. A red arrow on a white background indicates the direction of motion. A "ComputeMotion" button is at the bottom right.

|    |    |
|----|----|
| x1 | 32 |
| y1 | 0  |
| x2 | 30 |
| y2 | 0  |
| dX | -2 |
| dY | 0  |

# Results: F5 → F6



The screenshot shows a software window titled "frmMain" with a blue title bar. The main area contains a large image of a textured surface. To its left is a smaller inset image. A yellow box with a black border is overlaid on the inset image, containing the text:

$$dX = -3$$
$$dY = 0$$

Below the inset image are input fields for X (0), Y (0), W (64), and H (64), and a "GetFrames" button. Below these are six small frames labeled F1 through F6, arranged in two rows of three. An arrow points from F5 to F6. To the right of the frames is a table of parameters:

|    |    |
|----|----|
| x1 | 32 |
| y1 | 0  |
| x2 | 29 |
| y2 | 0  |
| dX | -3 |
| dY | 0  |

The table is partially enclosed by a black oval. To the right of the table is a white box with a red arrow pointing to the right. Below this box is a "ComputeMotion" button with a dotted border.



# Conclusion

- Coarse tracking of motion working
- Highly optimised BMA not as critical as for image compression, VLSI chips
  - Increase in accuracy
- Theoretical basis of accuracy can be derived from geometry.
  - Function of resolution of imaging device and  $D$  (distance from object plane)





# Further work

- Completing the survey of techniques
- Further displacement accuracy analysis (theoretical basis)
- Combination of segmented image regions and motion analysis and beacon recognition
  - This will enable the unbounded error drift attributable to dead reckoning to be reduced by resets at previously discovered surface beacons
  - Study of different surface image characteristics and its influence on the design of techniques.
- Map discovery (using surface beacons)
- Comparison of the surface based technique with current panoramic image analysis of eg vertical edges.