

Multi Sensor Data Fusion

May 4, 2016, Vikas Agrawal

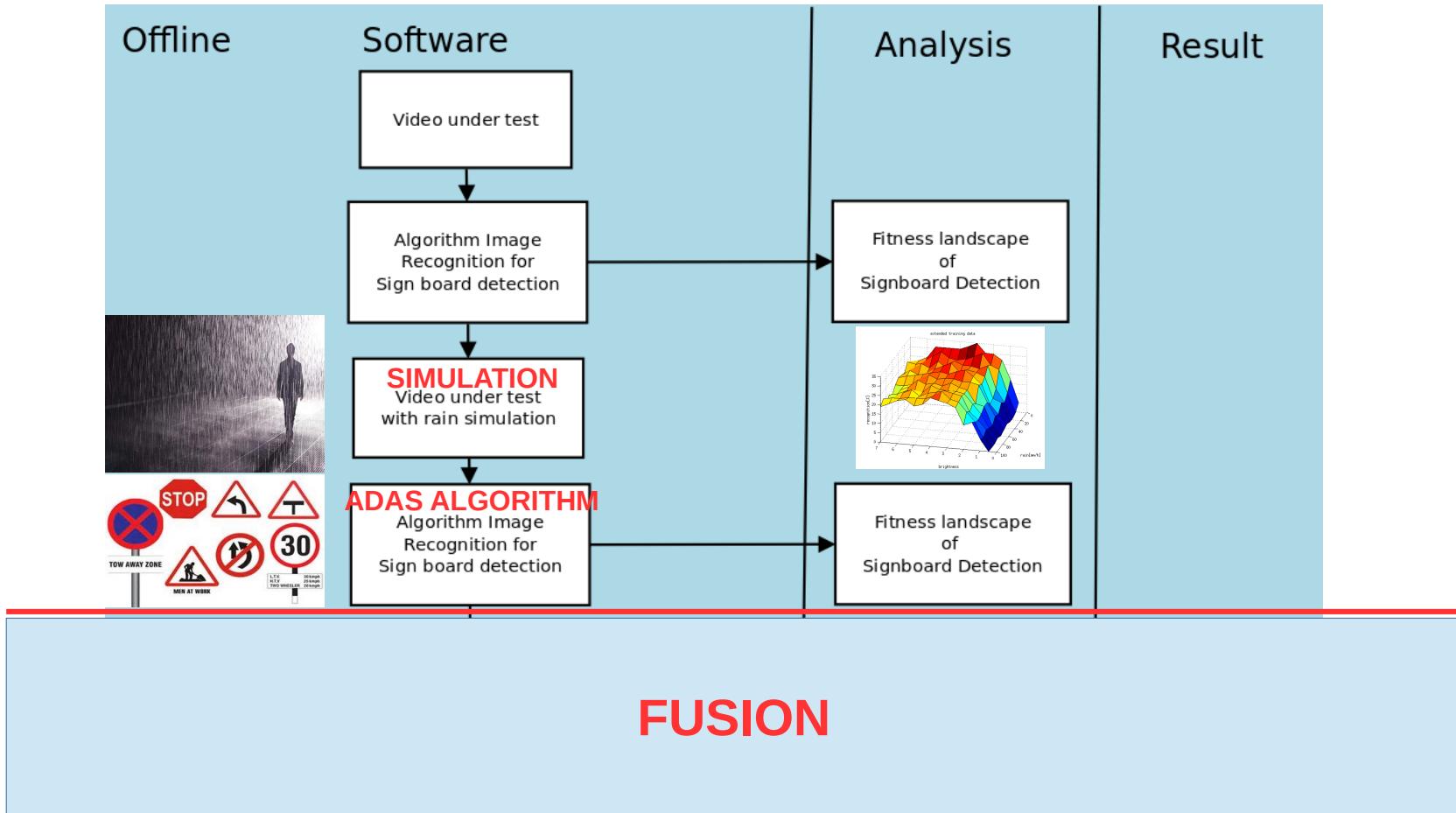


Table of contents

- Motivation
- Sensor Overview
- Multi Sensor Data Fusion Overview
- Alignment Methods
- Some applications of Multi Sensor Data Fusion
- Work at the faculty
- Literature and Tools



Motivation





Sensor Overview

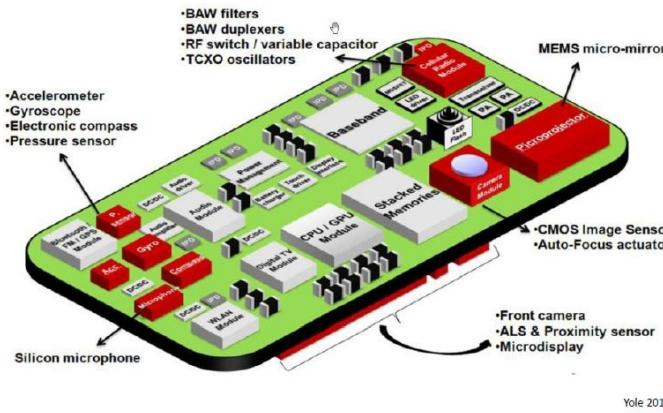


Sensor definition

- **Sensors:** source of informations interacting directly with the environment.
- **Sensor elements:** device perceiving physical properties such as heat, light, sound, pressure, motions etc.
- **Smart sensor:** hardware / software device comprising sensor element, micro-controller, communication controller and associated software for signal conditioning, calibration, diagnostics etc.



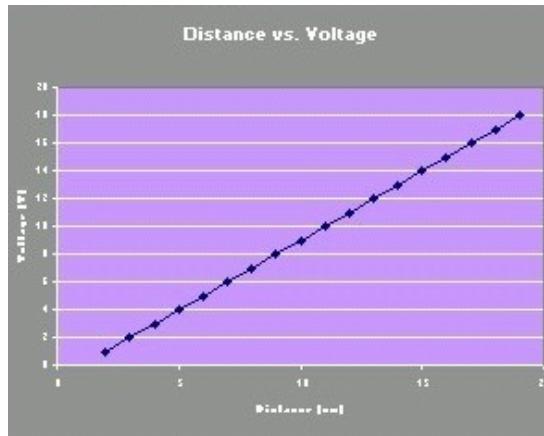
Sensor Characteristics - State



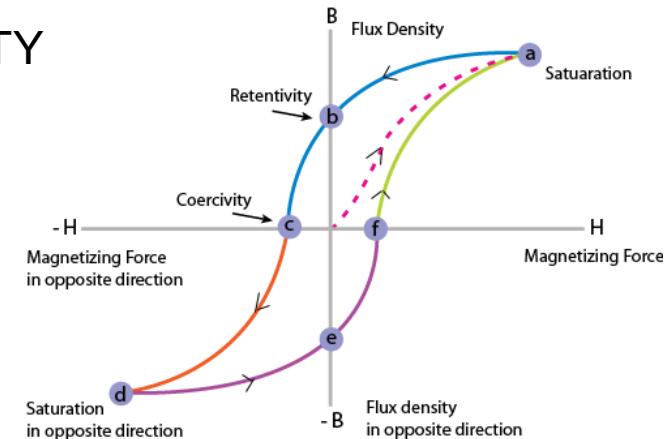
- State - Internal or External
 - Internal sensors measure **internal system parameters** such as position, velocity and acceleration.
 - External sensors measure the **system's dynamicity** such as proximity, external radiations.



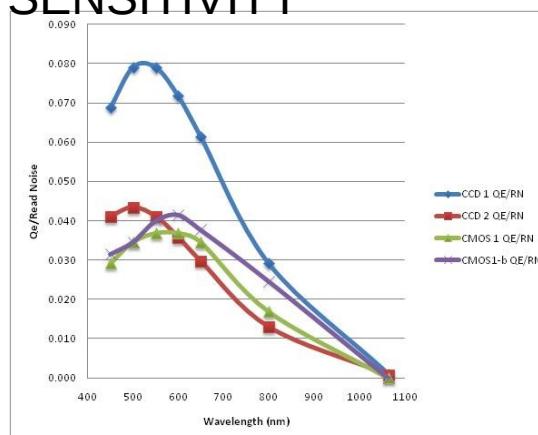
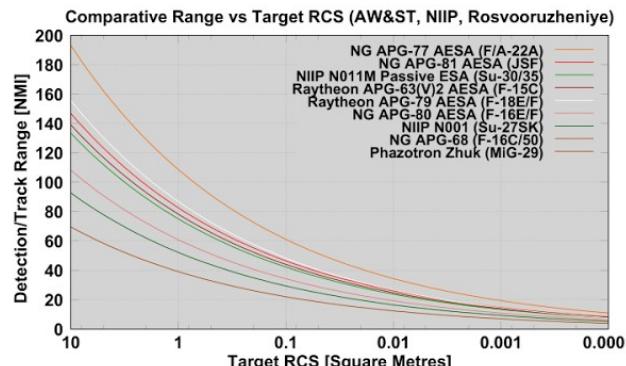
Sensor characteristics - Performance



LINEARITY

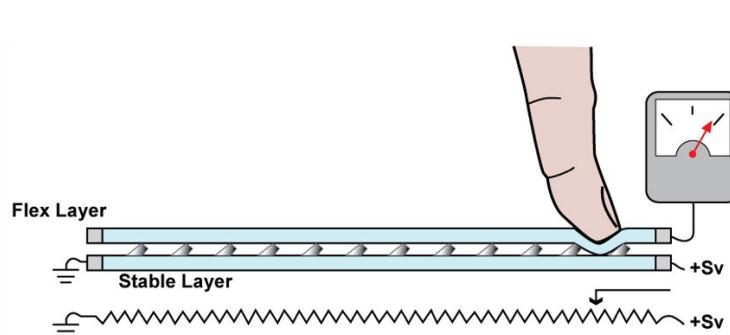


RANGE AND SENSITIVITY

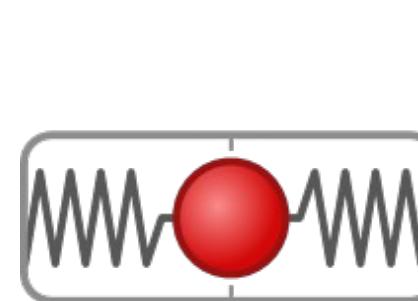




Sensor Characteristics - Measuring



Voltage



Displacement

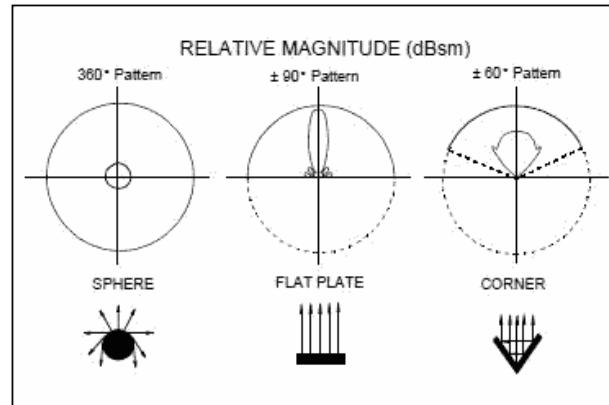
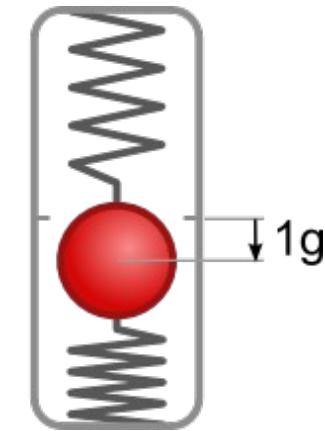
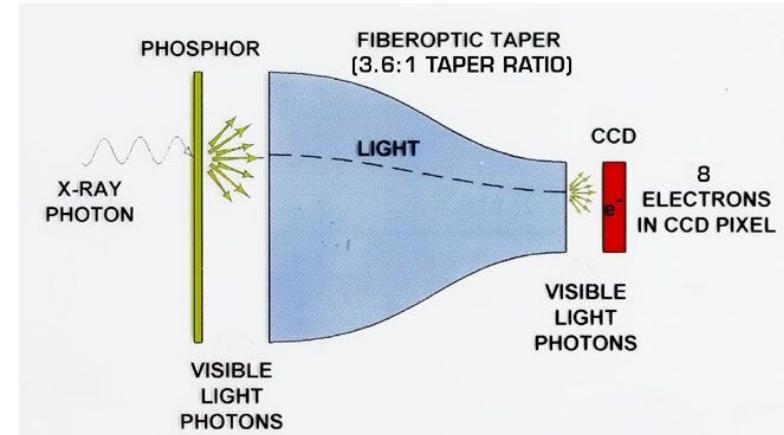


Figure 4. RCS Patterns

Amplitude Wave



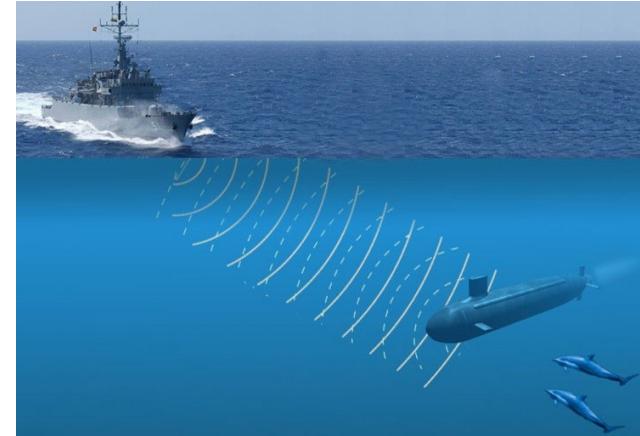
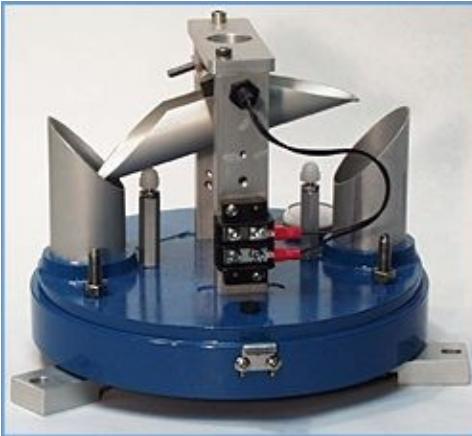
Light Intensity

Uncertainties

- Sensor measurements can only give an estimate of the measured physical property.
 - **Repeatable errors:**
 - **Random error** – Fluctuation in the electrical circuit in the sensor.
 - **Non-repeatable / Systematic errors:**
 - **Calibration Errors** – Errors in calibration of non-linear processes. Hysteresis.
 - **Loading Errors** – Errors during measurement. Temperature change of elements.
 - **Common Representation Format Errors** – Errors or approximation done in transforming representation.



Example Uncertainties



- Rain Gauge suffering from uncertainty at high rain intensities.
- Stray radiations induce uncertainties

Sensor Model

- **Sensor model** : provides coherent description of the sensor ability to extract information from its surroundings. The extracted information helps to calibrate, smooth and predict data.
- **Example:** Model of a robot navigating in an unknown space using ultrasonic sensors
 - The task of predicting Θ is defined mathematically as **estimating the a posteriori probability**, $P(\Theta = \partial | y)$ in a background information I , where ∂ represents the true value of the variable of interest Θ and y denotes the vector of N sensor measurements.



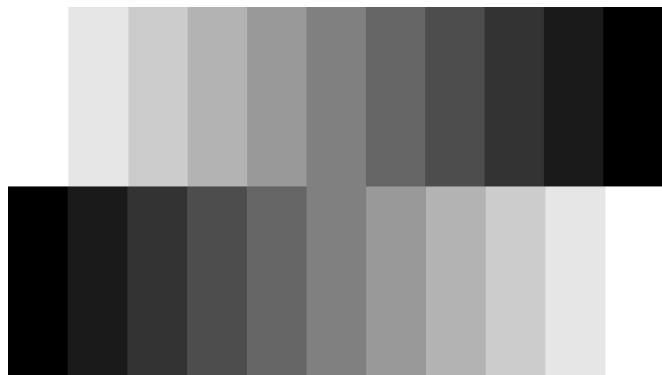
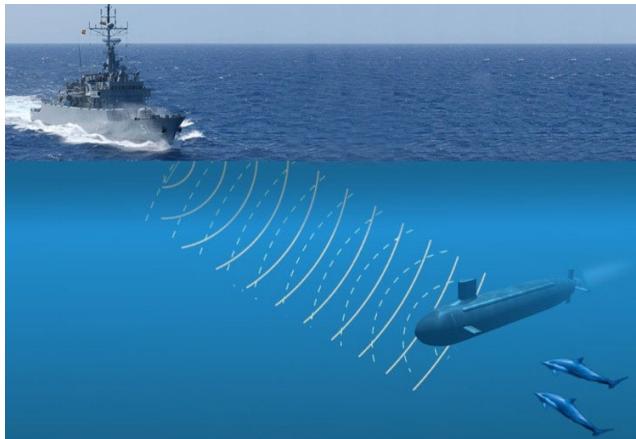
Notation

- $O = < E, x, t, y, \Delta y >$
 - O is the sensor observation and is the **output of the sensor**.
 - E is the name of the **physical property** which was measured by the sensor.
 - x is the **position in space** to which the measured physical property refers to.
 - t is the **time instant** when the physical property was measured.
 - y is the **value of the physical property** measured by the sensor element.
 - and Δy is the **uncertainty** in the measurement.



Examples

$$\mathbf{O} = \langle \mathbf{E}, \mathbf{x}, t, y, \Delta y \rangle$$



- **Time of Flight (Sonar)**

- $\mathbf{O} = \langle \text{ToF}, *, t, y, \Delta y \rangle$
 - Observation is Time of Flight
 - Name is Time of Flight
 - Spatial location independent
 - Time as variable
 - Range Reading (one half of the distance travelled)
 - Uncertainty in the reading.

- **Digital RGB Color Image (Camera)**

- $\mathbf{O} = \langle \text{Intensity}, \mathbf{x}, *, \text{RGB}, * \rangle$
 - Observation is Gray level value
 - Name is Intensity
 - Pixel location
 - Time independent
 - RGB Observation
 - No uncertainty in the reading.



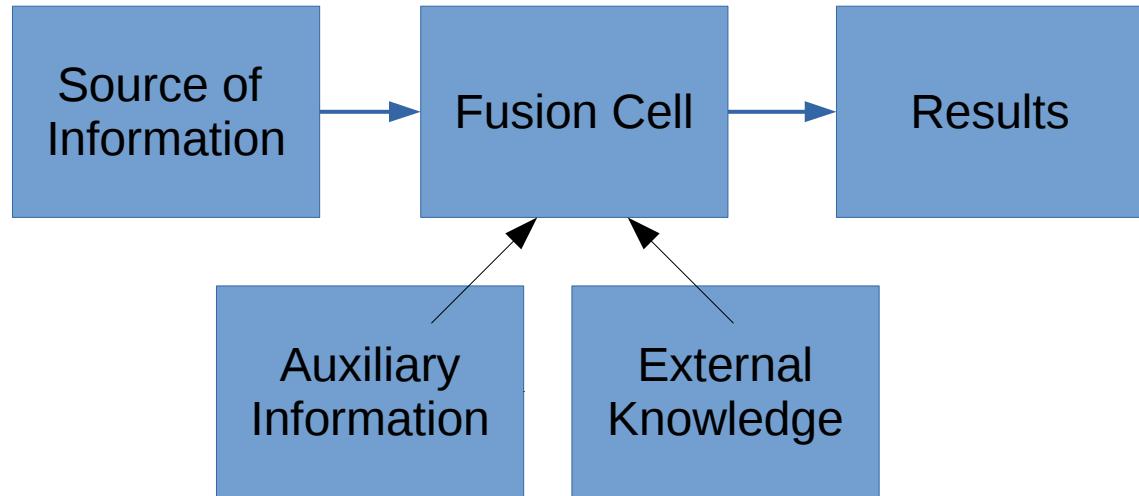
Multi Sensor Data Fusion Overview

Definition of Multi Sensor Data Fusion

- A multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations and their significance.”



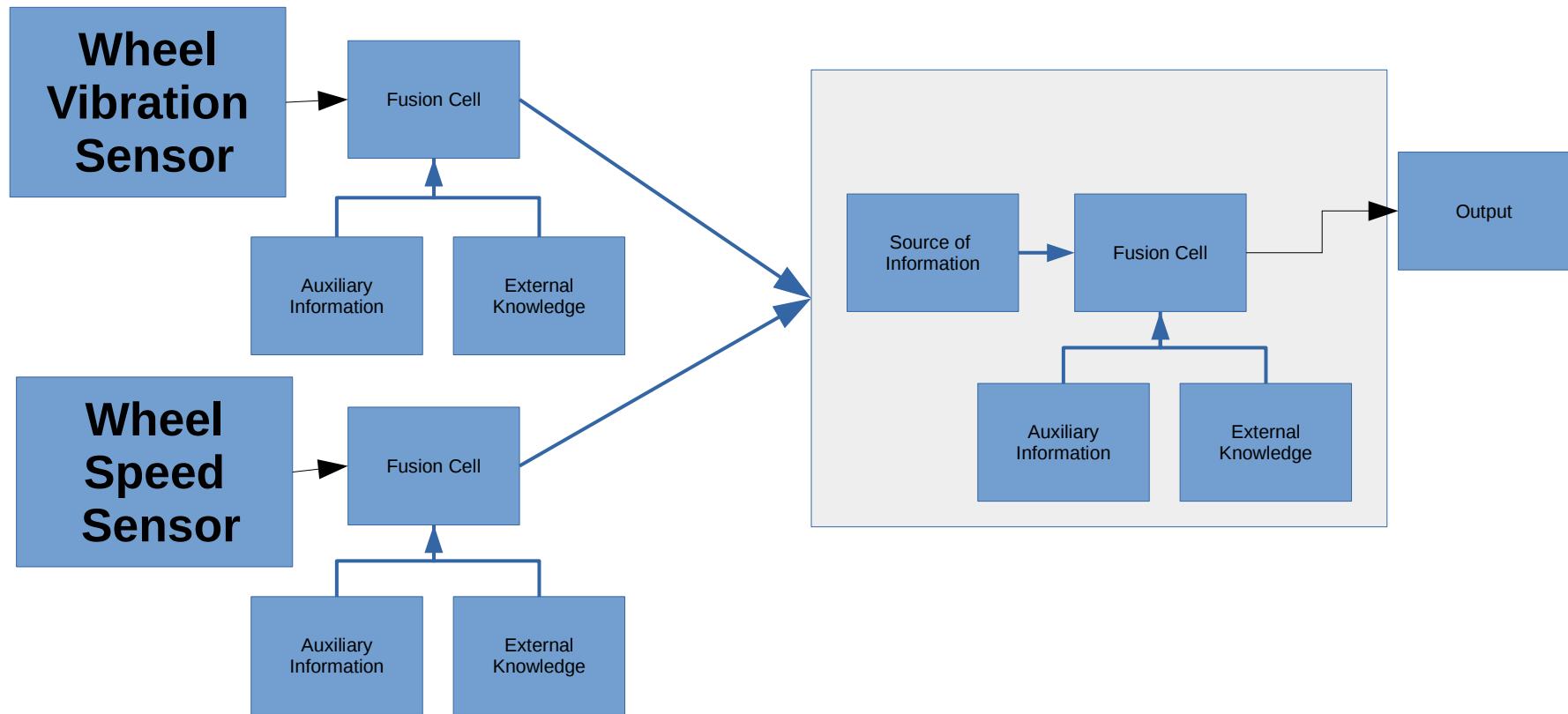
Single Sensor Fusion Cell



- **Source of Information:** Data directly from the sensors
- **Auxillary Information:** Additional data derived by specific processing of the sensor information
- **External knowledge:** Additional data consisting of all the elements of the a priori knowledge

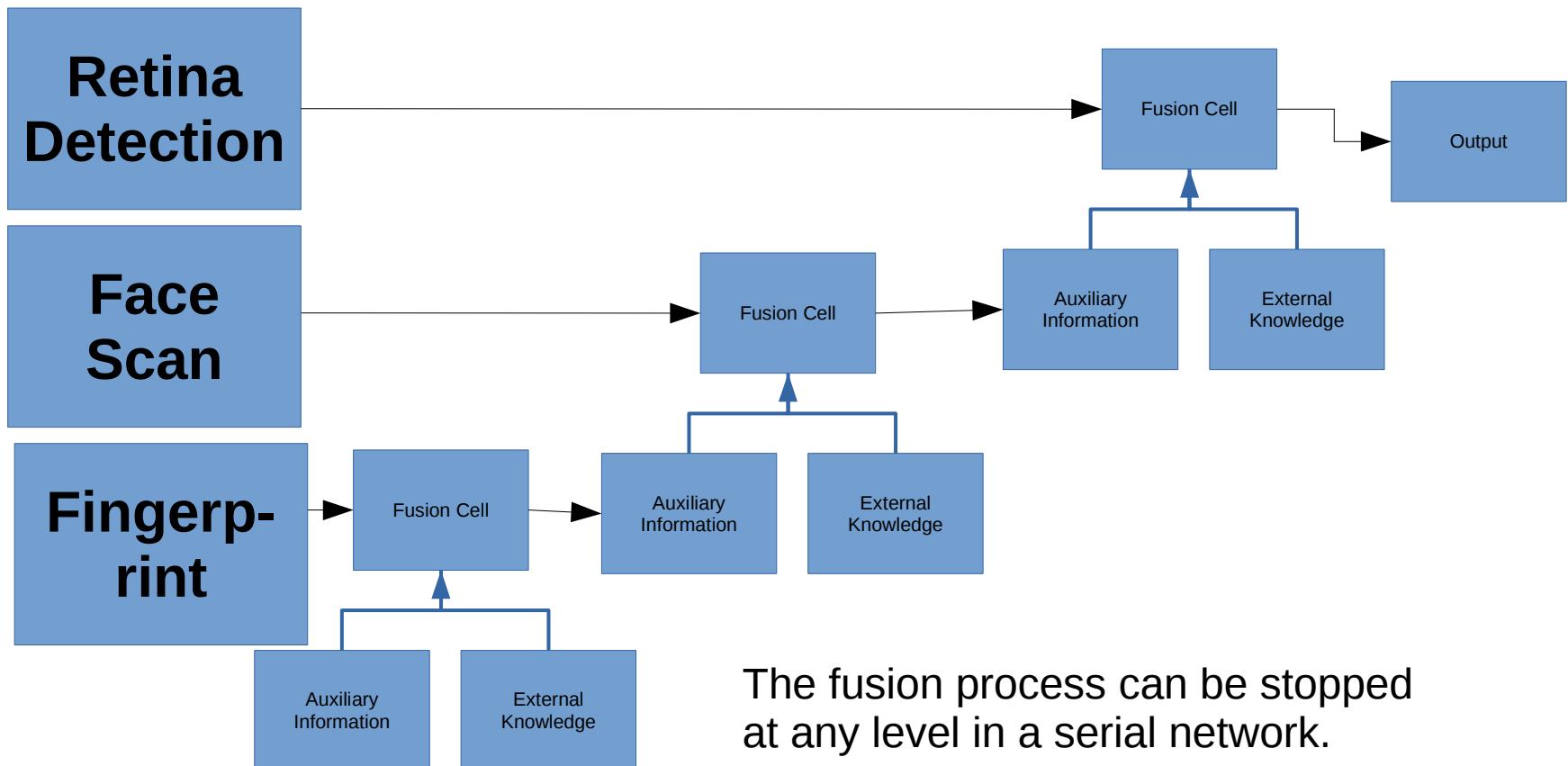


Parallel Network – Indirectly measuring Tyre Pressure



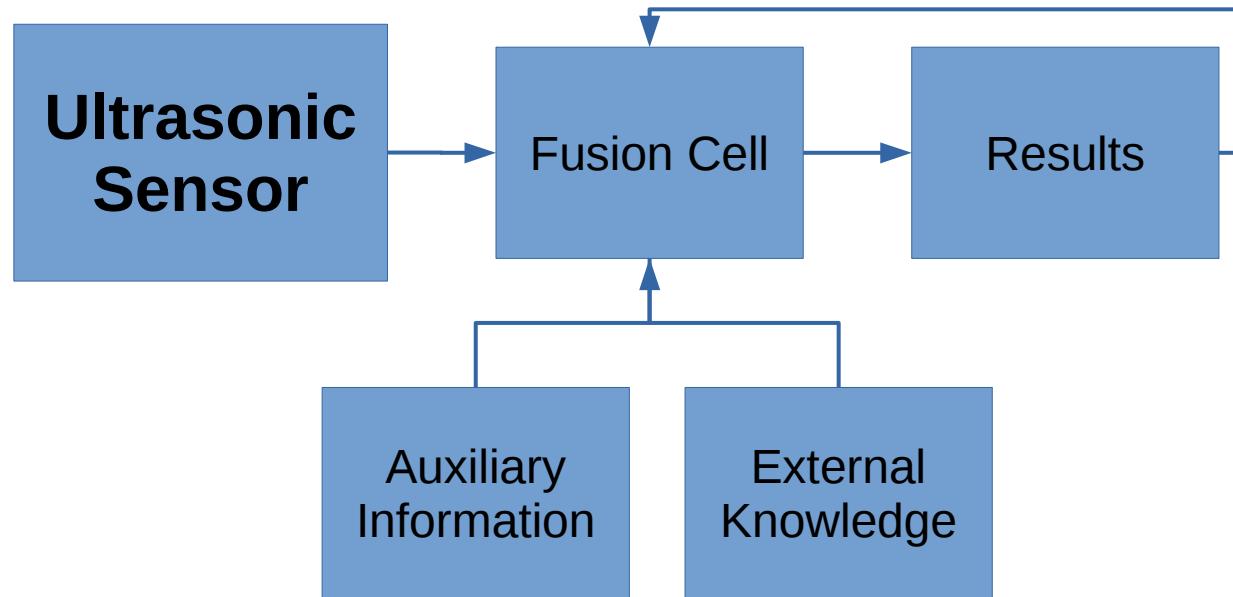


Serial or Cascaded Network – Biometric Identity





Iterative Network – Navigation of a mobile robot



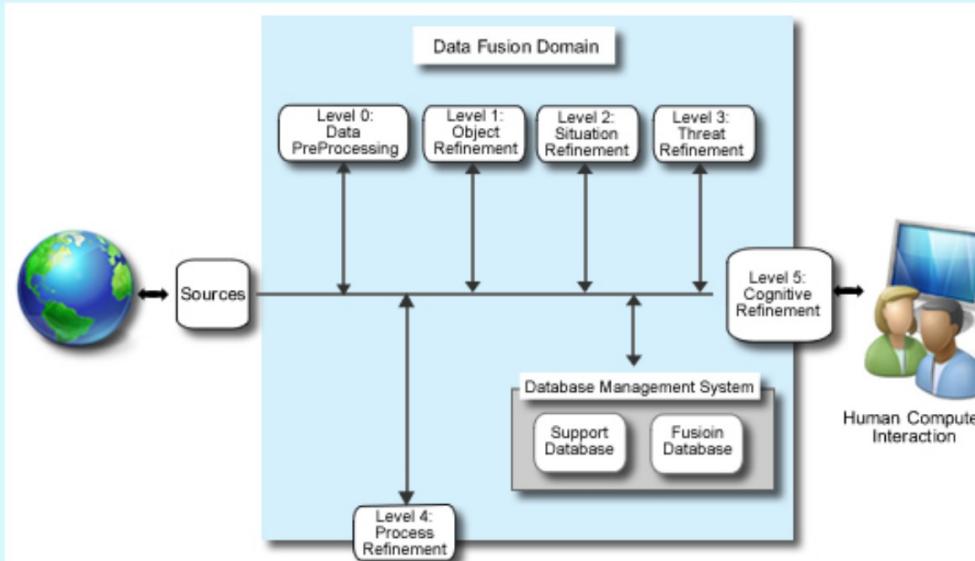
Source of Information is generally data changing through time. The results are fed back as auxiliary information back to the Fusion cell.

Data fusion models

- As seen before, the MSDF is a system-theoretic process (a synergy of sensing, signal, and data processing, estimation, control and decision making). Models to view MSDF
 - JDL Model
 - TRIP model
 - Omnibus model
 - Dasarathy Data Fusion Model



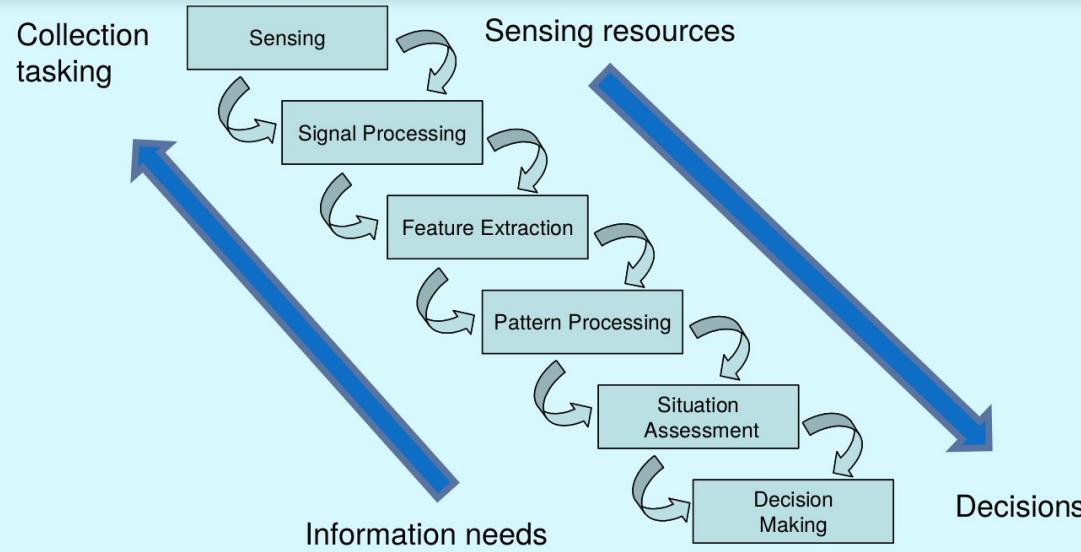
JDL DATA FUSION MODEL



Top level view of the JDL data fusion process model
(Hall and McMullen (2004))



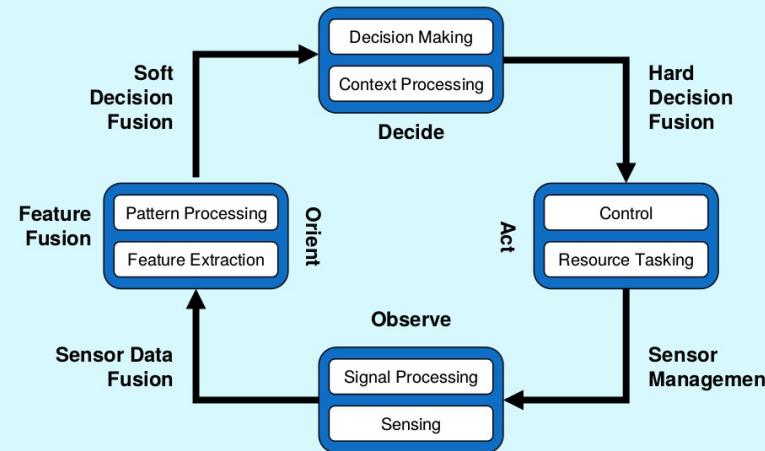
TRANSFORMATION OF REQUIREMENTS FOR THE INFORMATION PROCESS (TRIP) MODEL



The TRIP waterfall model for data fusion system development
(adapted from Kessler and Fabien (2001))



OMNIBUS MODEL



The Omnibus Model for decision making and data fusion
(adapted from Bedworth and Obrien (1999))



DASARATHY'S FUNCTIONAL MODEL

Table 2: Components of Dasarathy's Model

Input	Output	Notation	Analogues
Data	Data	DAI-DAO	Data-level fusion
Data	Features	DAI-FEO	Feature selection and feature extraction
Features	Features	FEI-FEO	Feature-level fusion
Features	Decisions	FEI-DEO	Pattern recognition and pattern processing
Decisions	Decisions	DEI-DEO	Decision-level fusion

Three general levels of abstraction in fusion processing:

- **Data level** - integration of raw observations and can occur only in the case when the observations are of the same type
- **Feature level** - assumes that each stream of sensory data is first analyzed for features, after which the features themselves are fused
- **Decision Level** - based on the fusion of individual mode decisions or interpretations



Data fusion modes

- Low Level Fusion
 - Raw-data fusion produces new raw data set that is expected to be more informative and useful than inputs
- Intermediate level fusion / feature level fusion
 - Mid Level Fusion combines various features such as edges, lines, corners, textures, or positions into a feature map.
- High level fusion / decision fusion
 - Decision Level Fusion use voting, fuzzy logic and statistical methods to combine decisions from several experts.

Data fusion sensor networks

- **Complementary sensor networks** – Sensor data is mutually exclusive.
- **Competitive sensor networks** – Each sensor delivers independent measurements of the same attribute or feature.
- **Cooperative sensor networks** – Data provided by two independent sensors are used to derive information that would not be available from a single sensor.

Data fusion architectures

- Definite arrangement of sensors and sensor-data acquisition systems and signal procesing aspects.
 - **Centralised Fusion Architecture**
 - Mainly for similar sensors, involves time sync and bias correction.
 - Processing at a central node
 - **Distributed Fusion Architecture**
 - Disimiliar sensors with different observation frame.
 - Processing (filtering, calculation of covariance, estimate state vector matrix) done at every node.

Advantages in various levels

- Provides **redundancy**
- Provides **observations**
- Provides agreements and hence **confidence**.
- Increases **dimensionality**.
- Enhances **spatial resolution**
- Extends **temporal coverage**

Drawbacks

- **Sensor Interference:** sensor interfere in the working conditions of each other. Example a radio frequency seeker mounted on a missile are corrupted by receiver noise or RCS (radar cross section) fluctuations.
- **Excessive manipulations** of bad data to convert into good data leads to **unreliable data**.
- **Careful Excessive manipulations** of bad data may convert into good data, but at a very high cost.

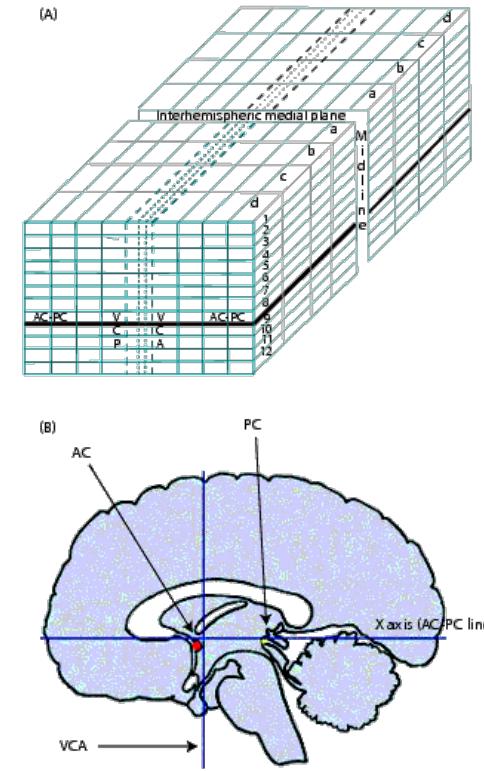


Fusion and Alignment methods



Why Common Representation Data Format

- To make compatible sensor observations.
- Example: In order to compare different brains, a standardised anatomically-based coordinate system or brain-atlas needs to be done
 - The idea is that, in the new coordinate system, all brains have the same orientation and size. This coordinate system is called **Talairach coordinate**.





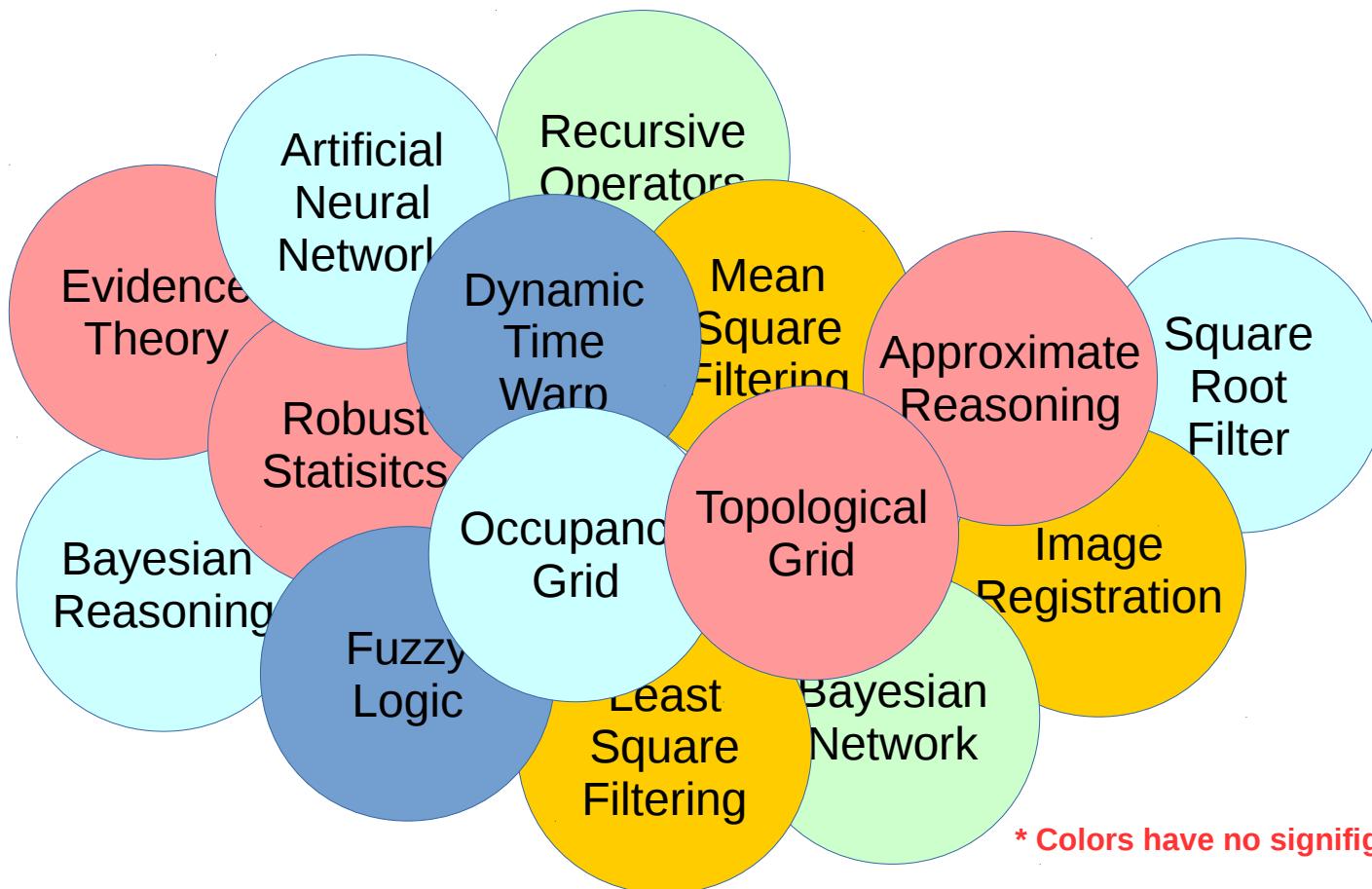
Methods in Common Representation Data Format

$$O = \langle E, x, t, y, \nabla y \rangle$$

- Spatial alignment, x
 - Field of view of each sensors is converted into a common coordinate system.
- Temporal alignment, t
 - Transformation of local times t to a common time axis.
- Sensor value normalization, y
 - Sensor values and their uncertainties are normalized to a common scale.



Quick Overview of various Fusion / Alignment methods *





Mutual Information

- Mutual information **quantifies the entropy** (amount of information) obtained about one random variable, through the other random variable.
- Mathematically, it is determination of similarity between **joint distribution and the product of the factored marginal distribution**.

$$I(X;Y) = \int_Y \int_X p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) dx dy,$$

where $p(x,y)$ is the joint probability density function of X and Y , and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y respectively.

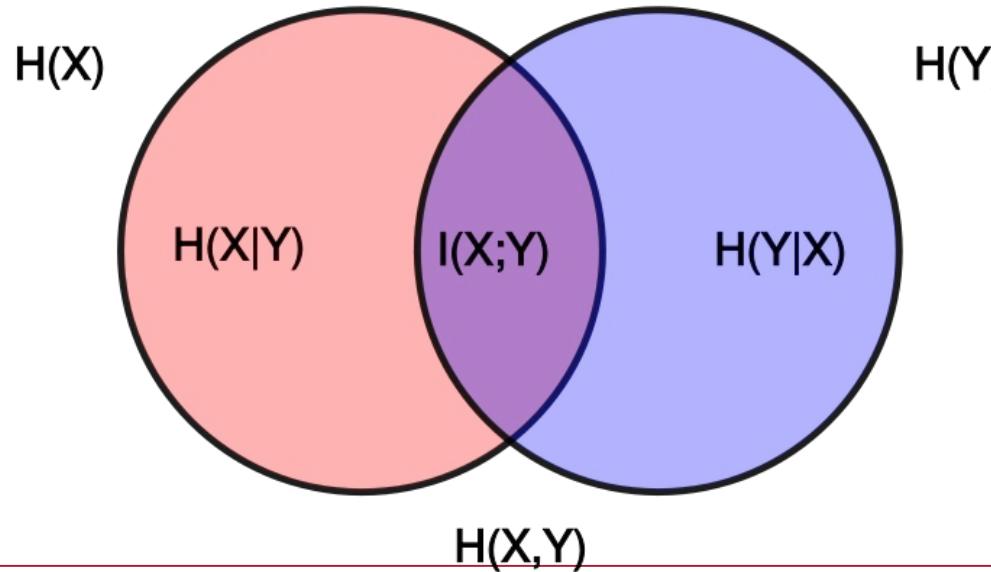
- Mutual information hence between independent random variables is 0

$$\log \left(\frac{p(x,y)}{p(x)p(y)} \right) = \log 1 = 0.$$



Mutual Information - [wikipedia](#)

- **Joint entropy:** $H(X, Y)$
- **Individual entropy:** $H(X)$ and $H(Y)$
- **conditional entropy:** $H(X|Y)$ and $H(Y|X)$
- **Mutual Information:** $I(X; Y)$





Spatial Alignment

- **Transforms local spatial positions** to a common coordinate system in order to maximize the mutual information between the sensor images
- → **to correct misalignment** due to result of lens, sensor distortions, camera perspective or differences between capture devices.
- → **Using Geometric transformations** to the other images so that they align with the reference.



Methods in Spatial Alignment for Images

- Intensity based Automatic Image Registration.
 - maps certain pixels in each image to the same location based on **relative intensity patterns**.
- Control Point Registration
 - allows you to manually **select common features** in each image to map to the same pixel location.
- Phase Correlation and Optimization based Spatial Alignment
 - Estimating the **relative translative offset** between two images using the frequency histogram of the image.

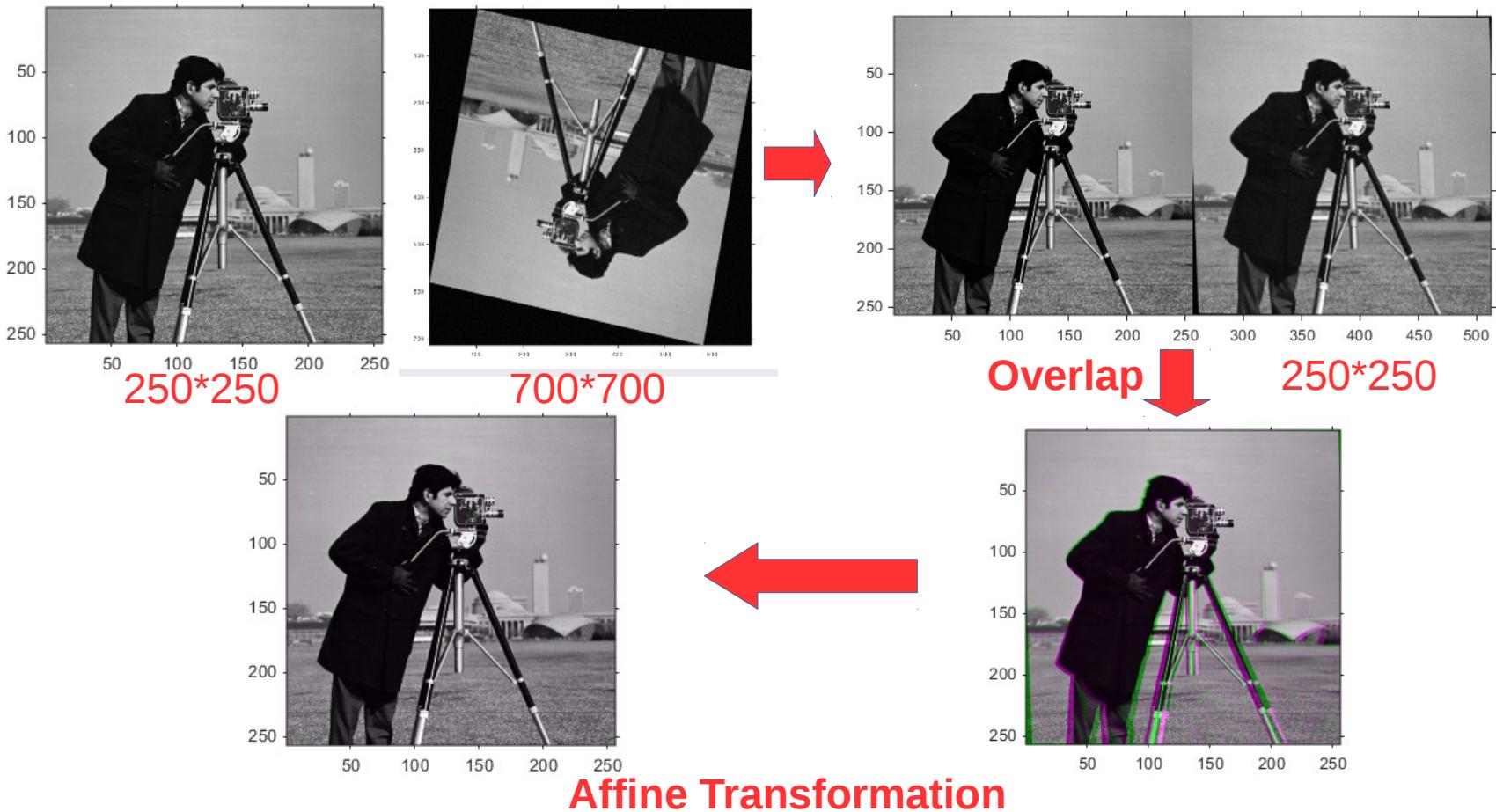
Areas of Application

- Align **satellite images**: discover how a river has migrated
- Align **medical images** captured with different diagnostic modalities (MRI and SPECT) : whether a tumor is visible



Phase Correlation and Optimization based Spatial Alignment

Apply translative offset





Temporal Alignment

- Correct **temporal misalignment**
- assuming that **spatial position is independent**, otherwise use Spatial-Temporal Alignment
- using **Time Axis Transformation**

P_i, Q_j is a set of observations on time axis t_i, t_j measuring y_i, y_j

$$P = (P_1, P_2, \dots, P_M)^T$$

$$Q = (Q_1, Q_2, \dots, Q_N)^T$$

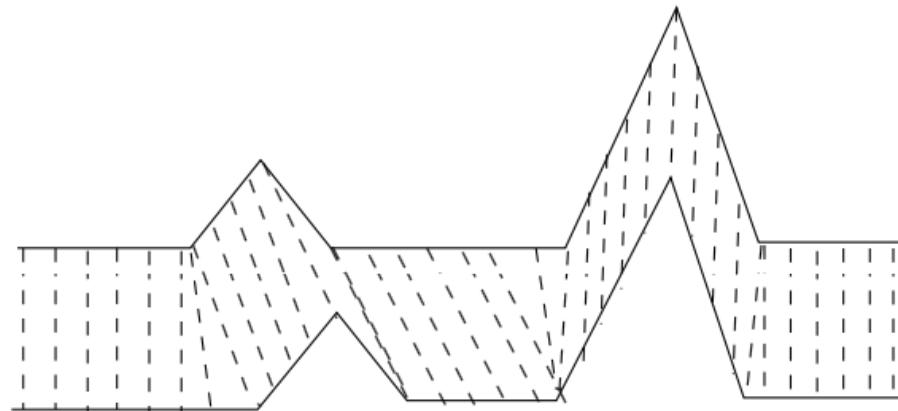
$$\begin{aligned}P_i &= \langle E_i, *, t_i, y_i, * \rangle, i \in 1, 2, 3, \dots, M \\Q_i &= \langle E_j, *, \tilde{t}_j, \tilde{y}_j, * \rangle, j \in 1, 2, 3, \dots, N\end{aligned}$$



Method in Temporal Alignment: Dynamic Time Warping

- **optimal alignment** between two time series P and Q to **minimize the sum of the local distances $d(i,j)$** between the aligned observation pairs (P_i, Q_j)
- Dynamic time Warping is simplified if Euclidean distance is used.

$$d(i, j) = (P_i - Q_j)^2$$





Warping Path

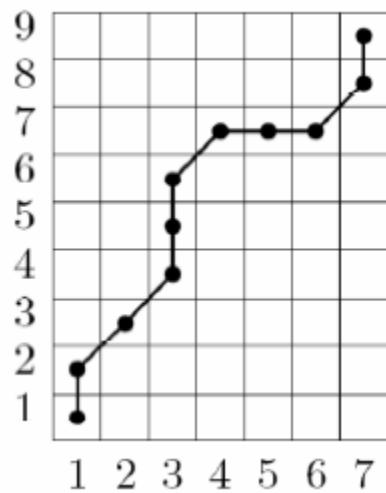
- Path minimizing the sum of local distances, given by,

$$W_{DTW} \equiv W_{DTW}(P, Q) = \min_W \sum_{k=1}^K d(w_k)$$

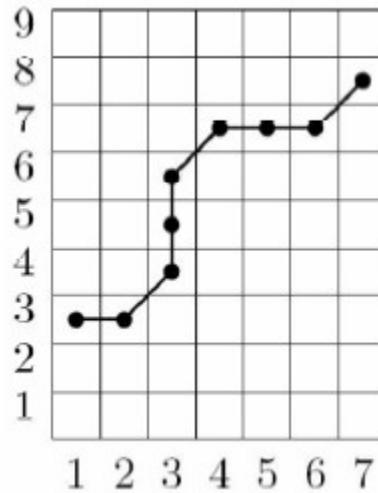
- **Boundary Condition** : Match first observation at P with the first observation at Q and end by matching last observation at P with the last observation at Q.
- **Continuity**: Restricts the allowable steps in warping path to adjacent matrix element.
- **Monoticity**: Force the warping path to increase monotonically in time.



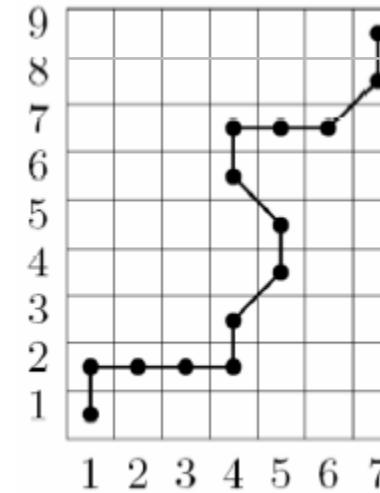
Warping-Path



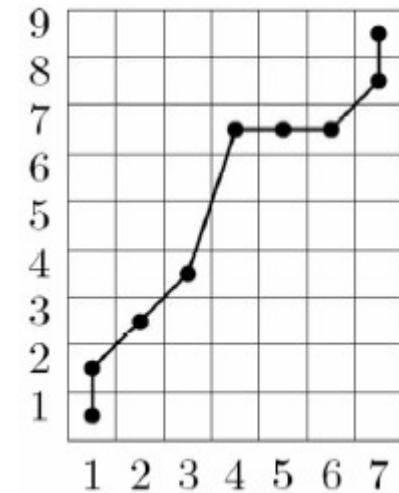
All conditions are met



Boundary conditions are not met



Monotonicity is not met

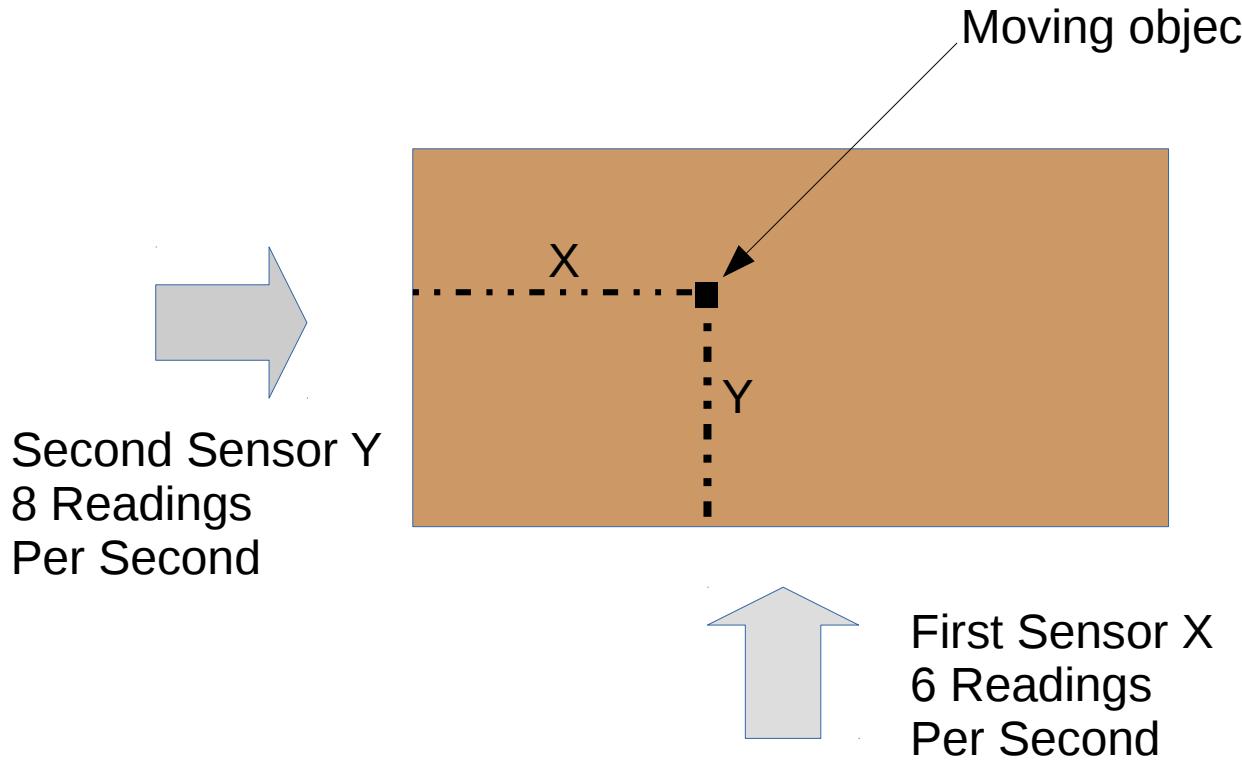


Continuity is Not met



Example: Temporal Alignment

- Two sensors monitoring the x and y position of an object





Calculation: Temporal Alignment

Euclidean Distance = $d(i,j) = (X_i - Y_j)^2$

X	1,7	4,1	3,1	2,2	3,4	1,6		
Y	2,4	3,5	2,7	3,2	4,5	4,7	4,8	5,9
$(1.7-2.4)^2$	0,49	3,24	1	2,25	7,84	9	9,61	17,64
$(4.1-2.4)^2$	2,89	0,36	1,96	0,81	0,16	0,36	0,49	3,24
$(3.1-2.4)^2$	0,49	0,16	0,16	0,01	1,96	2,56	2,89	7,84
$(2.2-2.4)^2$	0,04	1,69	0,25	1	5,29	6,25	6,76	13,69
$(3.4-2.4)^2$	1	0,01	0,49	0,04	1,21	1,69	1,96	6,25
$(1.6-2.4)^2$	0,64	3,61	1,21	2,56	8,41	9,61	10,24	18,49

$$D(3,2) = d(i_k, j_k) + \min(D(i_{k-1}, j_k), D(i_k, j_{k-1}), D(i_{k-1}, j_{k-1})) = \\ 0,16 + \min[0,85, 3,38, 3,87] = d(3,2) + \min(D(2,2), D(2,1), D(3,1))$$

0,49	3,73	4,73	6,98	14,82	23,82	33,43	51,07
3,39	0,85	2,81	3,62	3,73	4,14	4,63	7,87
3,87	1,01	1,01	1,02	2,98	5,54	7,03	12,47
3,91	2,7	1,26	2,01	6,31	9,23	12,3	20,72
4,91	2,71	1,75	1,3	2,51	4,2	6,16	12,41
5,5	6,32	2,96	3,86	9,71	12,12	14,44	24,65

Calculation: Temporal Alignment



Warp path is tracing the recursion **backwards**. Start from 24,65 and find the minimum points till 0,49

Warp Path = [(6,8)->(5,7)->(5,6)->(5,5)->(5,4)->(4,3)->(3,3)->(2,2)->(1,1)]

$$W_{DTW} \equiv W_{DTW}(P, Q) = \min_W \sum_{k=1}^K d(w_k)$$

Sensor Normalization

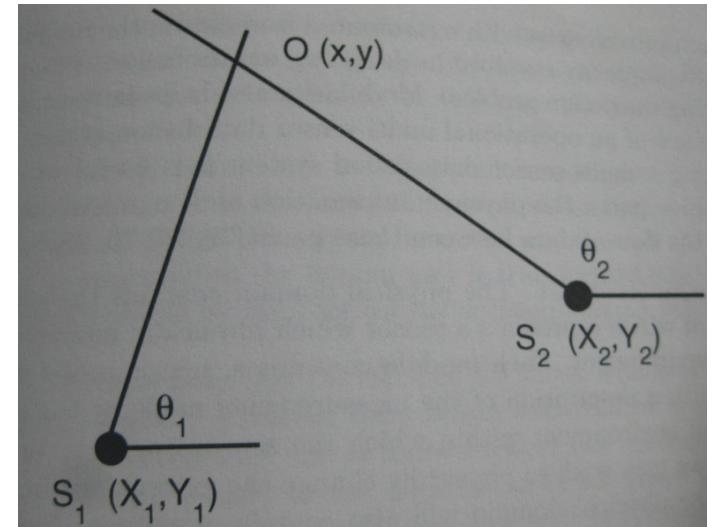
- Normalize the sensor values wrt to a reference point. This part will be skipped.

Localization: Triangulation Cooperative Fusion Method.

- Angle of Arrival Sensors (AOA)
- BTS \leftrightarrow AOA \leftrightarrow MobilePhone
- With 2 Base Terminals, the exact location of the mobile phone can be localised.

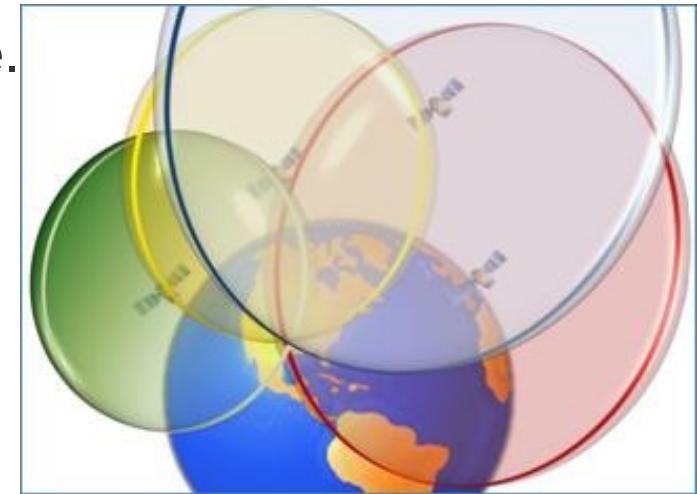
$$x = \frac{Y_2 - Y_1 + (X_1 \tan \theta_1 - X_2 \tan \theta_2)}{\tan \theta_1 - \tan \theta_2}$$

$$y = \frac{Y_2 \tan \theta_1 - Y_1 \tan \theta_2 + \tan \theta_1 \tan \theta_2 (X_1 - X_2)}{\tan \theta_1 - \tan \theta_2}$$



Localization: Trilateration Cooperative Fusion Method.

- Distance between BTS and Cellphone.
 - Radio Signal Strength Sensors (RSS)
 - ToF between BTS and Cellphone.
- $\text{BTS} \leftrightarrow \text{RSS} \leftrightarrow \text{MobilePhone}$
 - With 4 Base Terminal stations, the exact location of the mobile phone can be localised.



$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = r_i^2 \quad \text{mit } i = 1, \dots, N.$$

Linearising the equations by subtracting N^{th} from $[1 \dots (N-1)^{\text{th}}]$ equation, we get.

$$2(x_N - x_i)x + 2(y_N - y_i)y + 2(z_N - z_i)z = (r_i^2 - r_N^2) - (x_i^2 - x_N^2) - (y_i^2 - y_N^2) - (z_i^2 - z_N^2)$$



Example : Localization with 4 towers.

Sphere 1: Center is (x_1, y_1, z_1)

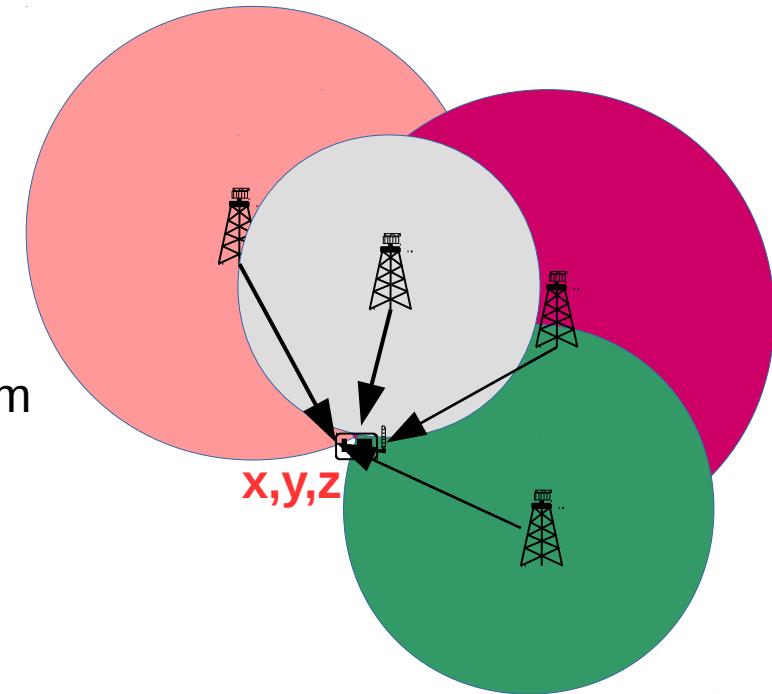
Sphere 2: Center is (x_2, y_2, z_2)

Sphere 3: Center is (x_3, y_3, z_3)

Sphere 4: Center is (x_4, y_4, z_4)

Distance of mobile phone (x, y, z) from each tower.

r_1, r_2, r_3, r_4





- Find x,y,z
- Put x_i, y_i, z_i, r_i in the below formula to obtain x, y, z. The formula is a linear simultaneous equation of the form $Au = B$; u is a column matrix $[x; y; z]$

$$2(x_4 - x_1)x + 2(y_4 - y_1)y + 2(z_4 - z_1)z = (r_1^2 - r_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2)$$

$$2(x_4 - x_2)x + 2(y_4 - y_2)y + 2(z_4 - z_2)z = (r_2^2 - r_4^2) - (x_2^2 - x_4^2) - (y_2^2 - y_4^2) - (z_2^2 - z_4^2)$$

$$2(x_4 - x_3)x + 2(y_4 - y_3)y + 2(z_4 - z_3)z = (r_3^2 - r_4^2) - (x_3^2 - x_4^2) - (y_3^2 - y_4^2) - (z_3^2 - z_4^2)$$

Bayes Theorem to estimate location with Multiple Sensors $S_1 \dots S_M$

$S_1, S_2 \dots S_M$ are sensor sources to **estimate** the position O.

$$\phi_m = \arctan \frac{x - X_m}{y - Y_m}$$

Calculate true bearing from Sensor S_m

$$\theta_m = \phi_m + \omega_m$$

Add random noise

$$p(\theta|\phi) = \prod_{m=1}^M \frac{1}{\sigma_m \sqrt{2\pi}} \exp - \left(\frac{\theta_m - \phi_m}{\sigma_m \sqrt{2}} \right)^2$$

Posteriori Probability Density

$$p(\theta|\phi) = \exp - \sum_{m=1}^M \frac{1}{2\sigma_m^2} \left(\theta_m - \arctan \frac{x - X_m}{y - Y_m} \right)^2 / (\prod_{m=1}^M \sigma_m \sqrt{2\pi})$$

$$p(x, y|\theta) = \pi(x, y|I) \exp - \sum_{m=1}^M \frac{1}{2\sigma_m^2} \left(\theta_m - \arctan \frac{x - X_m}{y - Y_m} \right)^2$$

Inverting the eq.
using Bayes Theorem

$$\hat{x} = \int xp(x, y|\theta) dx dy$$

Estimate locations

$$\hat{y} = \int yp(x, y|\theta) dx dy$$

Formulas not for examination

Inertial Navigation System

- Calculation of angles using rotation sensors like gyroscope.
- Calculation of acceleration using motion sensors like accelerometers.
- Fusion of sensors using Complementary or Kalman Filter

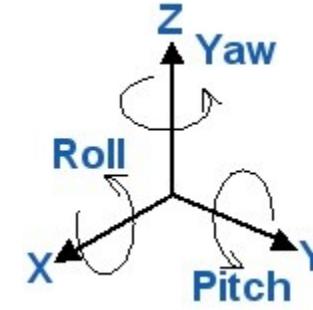
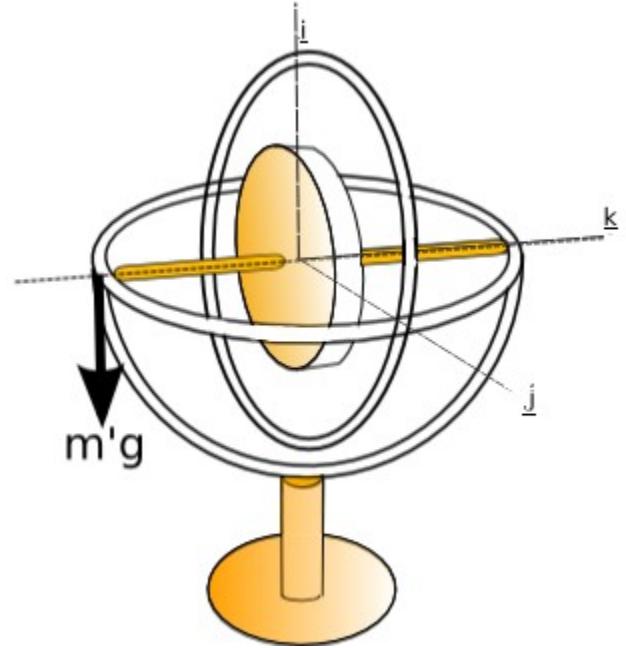


Roll, Pitch and Yaw Angle using Gyroscope

$$roll_{angle} = roll_{rate} * deltaT$$

$$pitch_{angle} = pitch_{rate} * deltaT$$

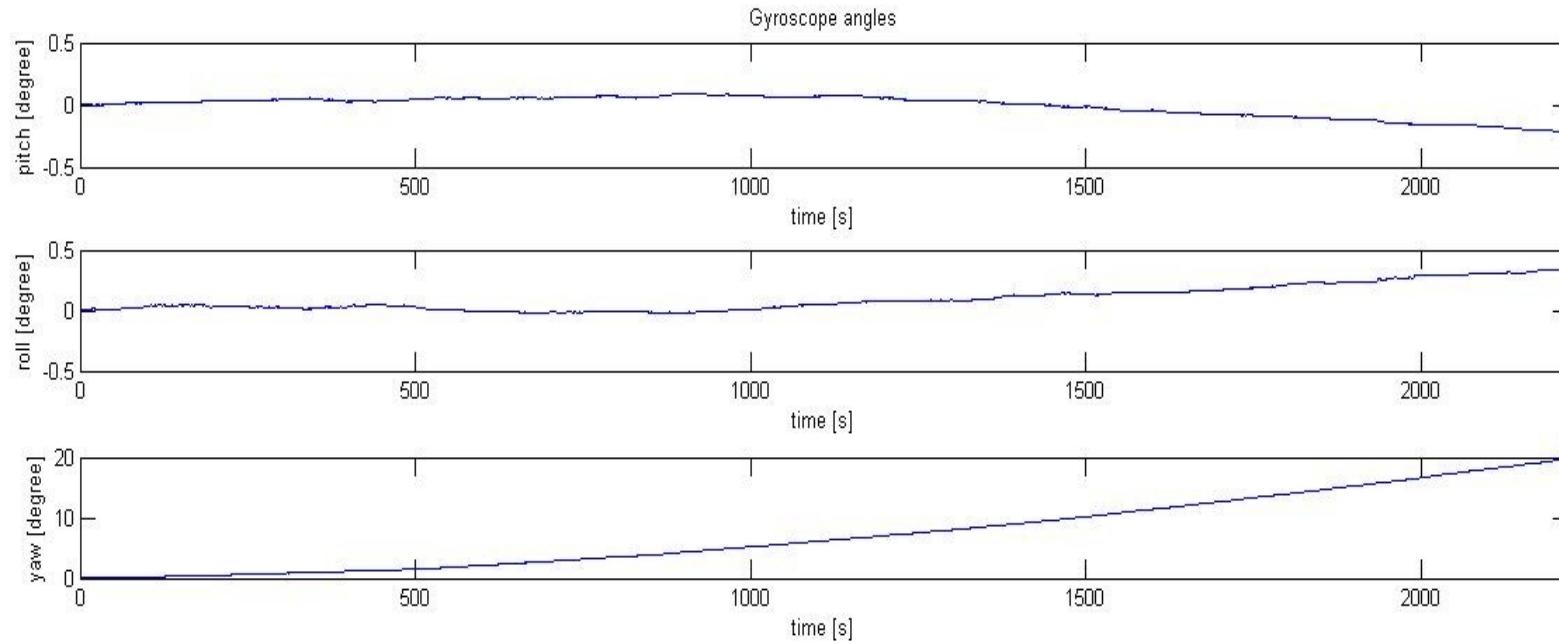
$$yaw_{angle} = yaw_{rate} * deltaT$$





Roll, Pitch and Yaw Angle using Gyroscope

- Large integration drift in static conditions
- Good for fast moving signals





Roll, pitch and yaw angle using accelerometer and magnetometer

$$roll_{angle_rad} = \text{atan2} \left(\frac{acc_y_axis}{acc_z_axis} \right)$$

$$pitch_{angle_rad} = \text{atan} \left(-\frac{acc_x_axis}{acc_y_axis * \sin(roll_{angle_rad}) + acc_z_axis * \cos(roll_{angle_rad})} \right)$$

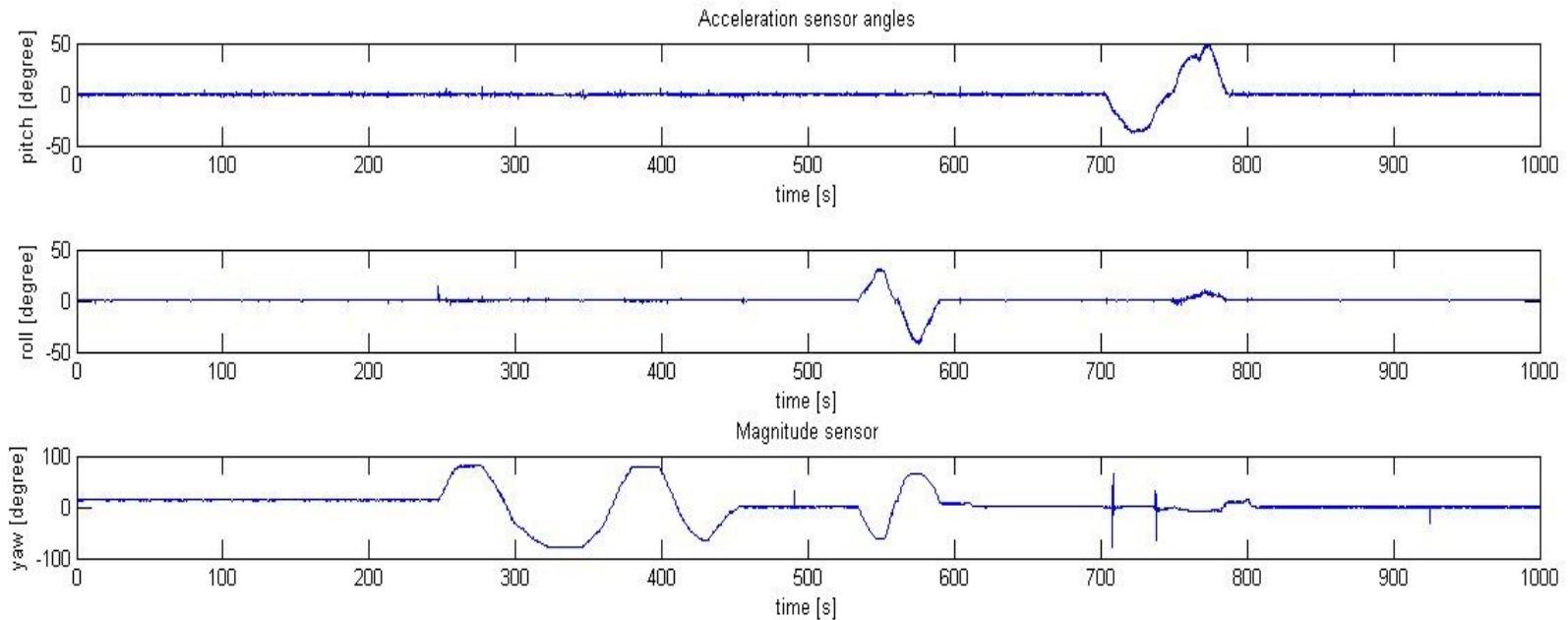
$$\begin{aligned} divider &= mag_x_axis * \cos(pitch_{angle_rad}) + \\ &\quad mag_y_axis * \sin(pitch_{angle_rad}) * \sin(roll_{angle_rad}) + \\ &\quad mag_z_axis * \sin(pitch_{angle_rad}) * \cos(roll_{angle_rad}) \end{aligned}$$

$$yaw_{angle_rad} = \text{atan2} \left(\frac{mag_z_axis * \sin(roll_{angle_rad}) - mag_y_axis * \cos(roll_{angle_rad})}{divider} \right)$$



Roll, pitch and yaw angle using accelerometer and magnetometer

- Large integration drift in dynamic conditions
- Good for slow moving signals

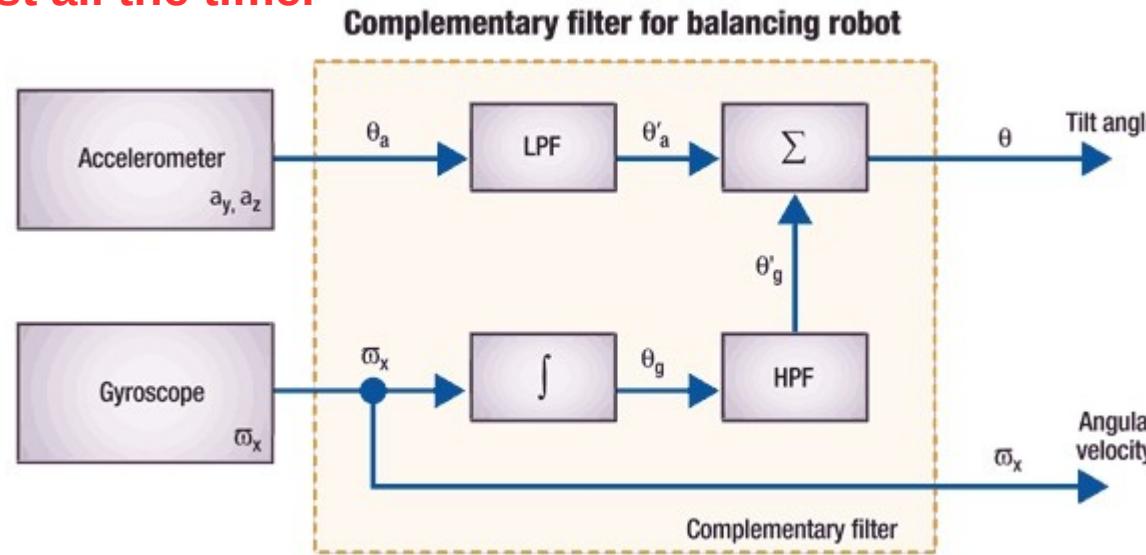


Sensor Fusion with Complementary Filter

- Complementary Filter uses both angular sensors and motion sensors to calculate the final angle.

$$\text{angle} = \text{alpha} * (\text{angle} + \text{integrated_Gyro}) + (1 - \text{alpha}) * \text{Acc_Mag_angle}$$

By fixing high alpha ($> 0,9$), we rely on gyroscope only at very high frequencies. By fixing low alpha ($< 0,1$), we rely on gyroscope almost all the time.



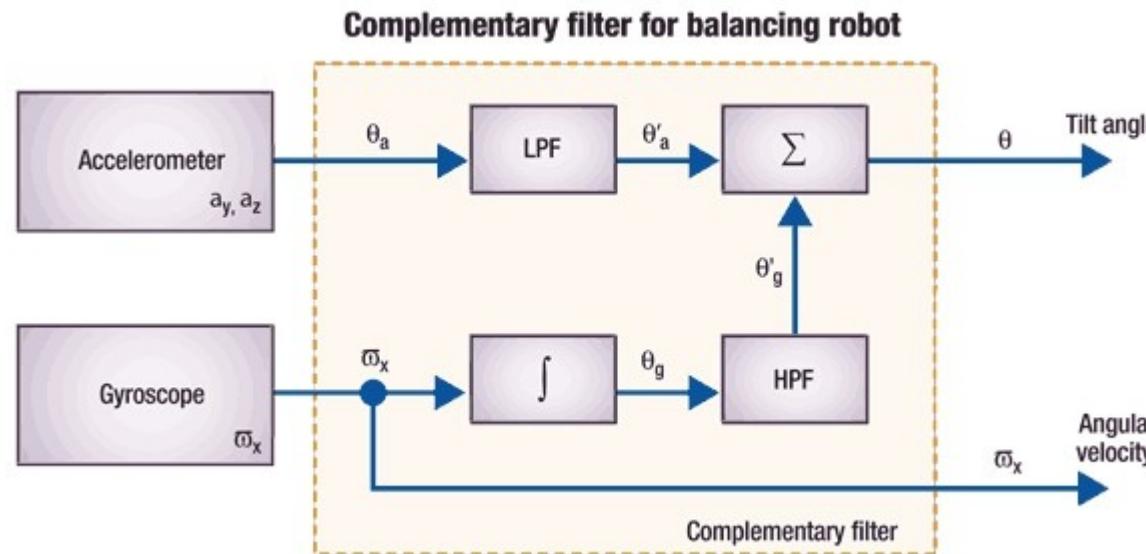


Calculate Filter Coefficients

$$\alpha = \frac{\text{time_constant}}{\text{time_constant} + \text{sample_period}}$$

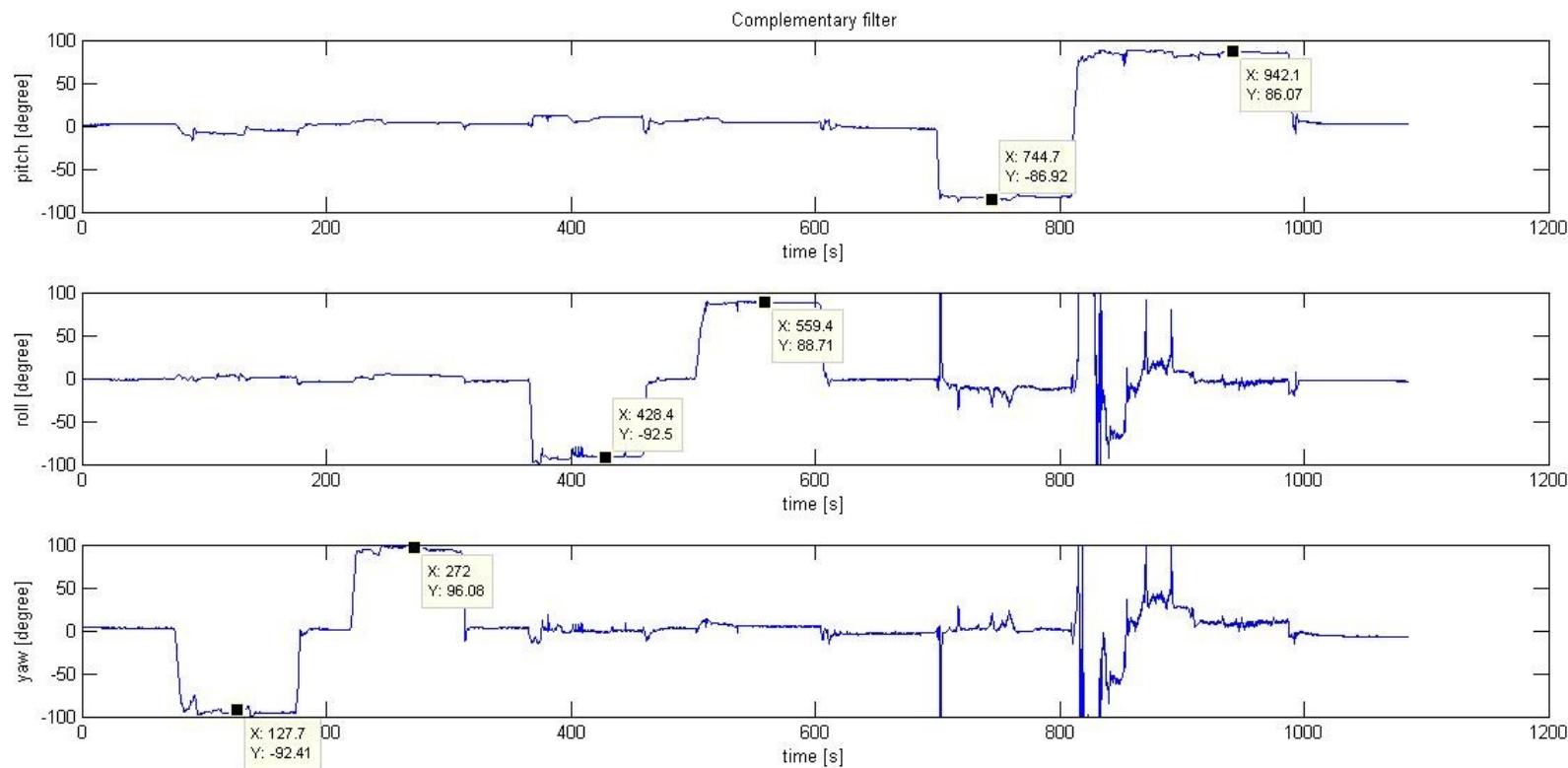
$$\alpha = 0.995$$

$$\text{sample_period} = \frac{1}{800\text{Hz}} = 0.00125\text{sec}$$



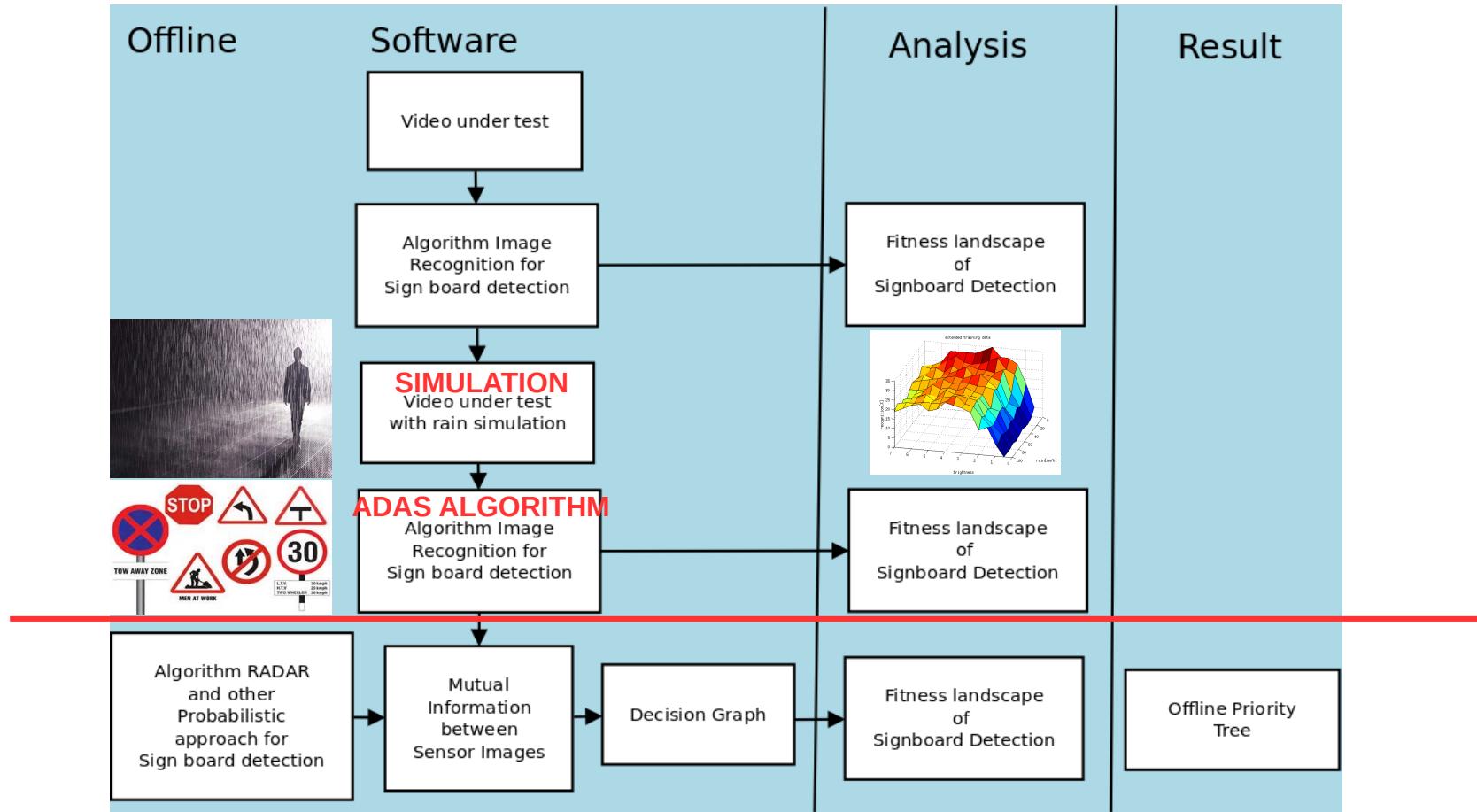


Roll, Pitch and Yaw Angle using Complementary Filter





Motivation



Materials used to create the slides

- Mitchell, HB : Multi Sensor Data Fusion
- Jitendra R. Raol : Multi Sensor Data Fusion with MATLAB.
- Matlab Image Processing Toolbox Help
- Internet: www.wikipedia.com, www.stackoverflow.com
- Hochschule Esslingen : Inertial Navigation System Sensor Fusion.



Thank you.

Vikas Agrawal

**Eberhard Karls Universität Tübingen
Embedded Systems
Phone: +49 7071 - 29 – 75940
Fax: +49 7071 - 29 – 5062
E-Mail: vikas.agrawal@uni-tuebingen.de**