Object Tracking Algorithm Using Particle Filter and Fusion in Multi-camera Datasets

A Dissertation submitted towards the partial fulfillment of
the requirement for the award of degree of

Master of Technology
in
Microwave and Optical Communication

Under the supervision of
Prof. RAJIV KAPOOR
(Associate Professor, Department of ECE)

Department of Electronics & Communication Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Delhi

Submitted to
Dr. Rajiv Kapoor
(Professor ECE)

Submitted by
Divyansh Sharma
MTech MOC
2K15/MOC/07

DELHI TECHNOLOGICAL UNIVERSITY

Established by Govt. Of Delhi vide Act 6 of 2009

(Formerly Delhi College of Engineering)

SHAHBAD DAULATPUR, BAWANA ROAD, DELHI-110042

CERTIFICATE

This is to certify that the dissertation title " **Object Tracking Algorithm Using Particle Filter and Fusion in Multi Camera Datasets**" submitted by **Ms. Divyansha Sharma, Roll. No. 2K15/MOC/07**, in partial fulfilment for the award of degree of Master of Technology inn "**Microwave and Optical Communication(MOC)"** run by Department of Electronics & Communication Engineering in Delhi Technological University during the year 2015-2017, is a bonafide record of student's own work carried out by her under my supervision and guidance in the academic session 2016-17. To the best of my belief and knowledge the matter embodied in dissertation has not been submitted for the award of any other degree or certificate in this or any other university or institute.

Prof. Rajiv Kapoor

Supervisor

Professor (ECE)

Delhi Technological University

Delhi-110042

DECLARATION

I hereby declare that all the information in this document has been obtained and presented in accordance with academic rules and ethical conduct. This report is my own work to the best of my belief and knowledge. I have fully cited all material by others which I have used in my work. It is being submitted for the degree of Master of Technology in Microwave and Optical Communication at the Delhi Technological University. To the best of my belief and knowledge it has not been submitted before for any degree or examination in any other university.

Divyansha Sharma
M. Tech. (MOC)
2K15/MOC/07

Date: July, 2017

Place: Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I owe my gratitude to all the people who have helped me in this dissertation work and who have made my postgraduate college experience one of the most special periods of my life.

Firstly, I would like to express my deepest gratitude to my supervisor **Dr. Rajeev Kapoor**, Associate Professor (ECE) for his invaluable support, guidance, motivation and encouragement throughout the period during which this work was carried out. I am deeply grateful to **Prof. Om Mishra,** Assistant Professor (Dept. Of E.C.E) for their support and encouragement in carrying out this project. I am thankful to Adit Modi, Software Engineer at Successive Software who helped me understand the technical and coding aspect of the project.

I also wish to express my heart full thanks to my classmates as well as staff at Department of Electronics & Communication Engineering of Delhi Technological University for their goodwill and support that helped me a lot in successful completion of this project.

Finally, I want to thank my parents, family and friends for always believing in my abilities and showering their invaluable love and support.

Divyansha Sharma

M. Tech. (MOC)

2K15/MOC/07

CONTENTS

Contents

CHAPTER 1

INTRODUCTION

The application of object tracking has witnessed explosive increase in tracking techniques and algorithms for different sensors, cameras etc. The data which we obtain from these sources is considerably large. But some of these sources and data are less important .By picking out the most credible sources and efficiently merging data from different data sources like cameras in our case, we can eliminate redundancy issues and obtain accuracy and efficiency in object tracking also data .To deal with this we propose two confidence degrees-Internal confidence & External confidence for prediction of credibility level in every camera. We constructed a camera selection approach wherein only reliable and worthy cameras are selected. Also we implemented a novel data fusion method technique by translating actual data of every frame in form of triangular fuzzy number and uncertainty metrics like information entropy for the fused process is calculated. The result observed is that the proposed theory and algorithm and fusion report is effective and efficient for multiple camera dataset. This study introduces a new method to extract reliable and credible camera data from multiple cameras.Using the internal confidence and external confidence we can determine degree of credibility of each camera. Then using fuzzy set logics we fuse the data from multiple cameras to obtain an efficient data table. Then through uncertainty measures like entropy, we deduce that fused data is more reliable. We have implemented this application on object tracking using three cameras at different angles capturing the same scenario to get a better view dataset. Then the information set from the four camera is fused to get a better overall output.

CHAPTER 2

STATE OF THE ART

2.1 Data Fusion

2.1.1 Purpose

Motive of data fusion is aggregating data from several independent sources to produce data more accurate and authentic than while using from independent sources. Input data may not only be originated from different sensors, but also from the same sensor at different time intervals. It makes it possible to exploit experts conclusions made for same data. The advantage of fusion from different sensors is diversity redundancy, and complementarity of various sensors:

The fusion process is bifurcated by the processing stage wherein fusion occurs:

Low-level-fusion -aggregates various sources data to get data in same format as original data, which is spurious and has greater information.

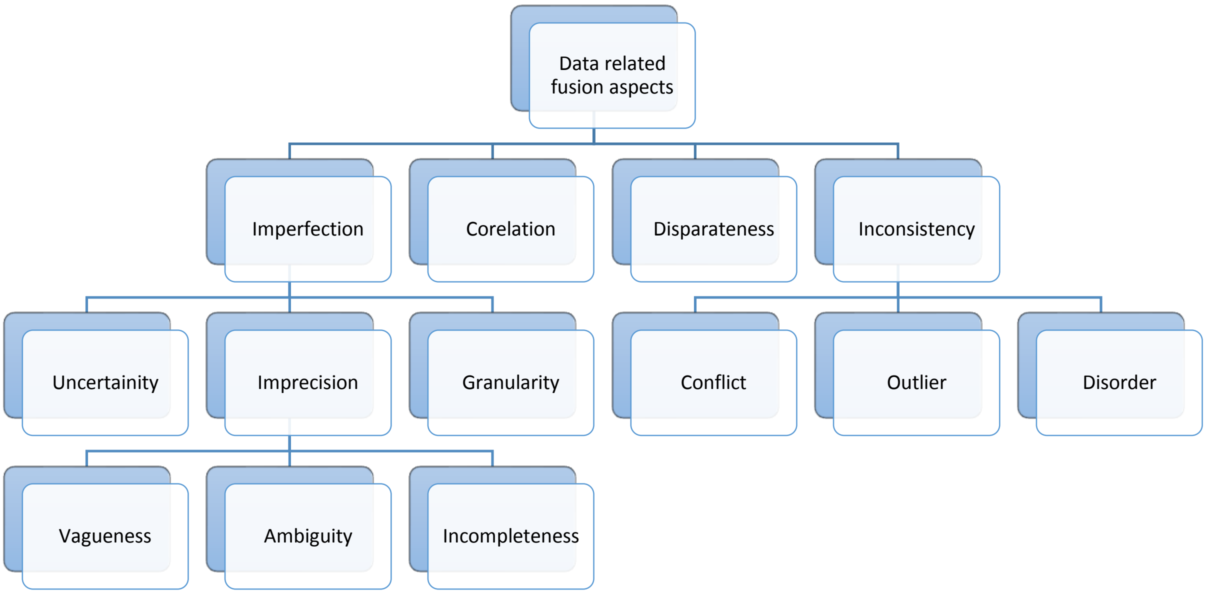
Intermediate-level-fusion or feature-level-fusion -The aim is to get a uniform generic overall feature idea. It clubs various features, exploited from various sources

High-level-fusion or decision-fusion: Uses weights/Scores or decision factor from several observations. Hard fusion deals with fusion of decisions and soft fusion deals with scores and weights.

There are number of matters that make data fusion challenging. The majority of these come from the type of data to be fused, and the nature of application domain as following:

1. Data imperfection: Data produced by sensors is always affected by some level of unreliability. Data fusion algorithms should express these effectively.
2. Outlier's data: Uncertainties in sensors also come from the ambiguities in the environment, and from the inability to differentiate between them [18]. Data fusion schemes should use the redundant data to reduce these effects.
3. Data modality: Sensors might collect same (homogeneous) or different (heterogeneous) data. Both cases should be handled by data fusion algorithm.
4. Data correlation: This is common in distributed fusion settings, e.g. wireless-sensor-networks, e.g. some sensor nodes could be affected by the same noise affecting their measurements. If this data dependency is not acknowledged, fusion schemes, could suffer from over or under reliability in results.
5. Data alignment: Multi source data should be converted from every sensor's local frame to a single common frame before fusion is done. This alignment problem deals with error by individual sources.
6. Data association: multi-target-tracking introduces a big complexity in fusion system i.e. data association problem, which is of two types: measurement-to-track and track-to-track association. The first deals the problem of identifying from which source, each measurement is initiated, while second with differentiating and linking tracks.
7. Operational times: The sensors cover a large environment consisting of varied aspects. In homogeneous sensors, operational frequency of sensors might be varied. This issue should be handled, especially in real-time situations.
8. Dynamic vs. static process: the process can be varying with time. In the second one, it can be important for data fusion schemes to register the immediate history of process into fusion process [21]. Thus how quickly data is captured, changed and updated plays an important role.
9. Data dimensionality: the data to be measured can be pre-processed, locally at every sensor nodes or globally on fusion spot. This stage is beneficial as it

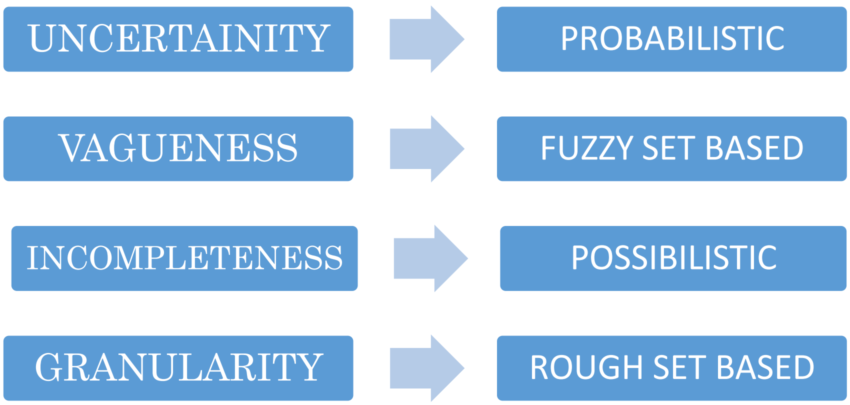
ensure saving on the communication bandwidth and power required for transmitting data.

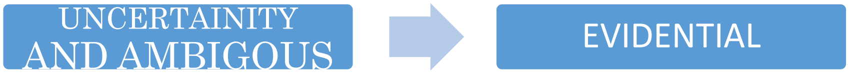


2.2 Data Fusion Algorithms

2.2.1 FUSION OF IMPERFECT DATA

The inbuilt fault of data is the biggest challenge in data fusion. Thus many theories have been developed to tackle this issue.





COMAPARISION OF IMPERFECT DATA FUSION FRAMEWORKS

1. PROBABILISTIC FUSION-

Characteristics: Represents data using probability distributions.It is fused with bayesian framework.

Advantages: Well adopted appraoch for uncertain data.

Limitaations: Not suitable for other data imperfections.

2. EVIDENTIAL BELIEF FUSION:

Characteristics: Takes into account probability mass to henceforth characterise data with the help of possibilities and belief It is fused using Dempsters combination rule.

Advantages: Helps in fusion of ambiguous and uncertain data.

Limitations: Doesn’t take into account other data imperfections. It is not efficient for conflicting data.

3. FUZZY SET BASED FUSION:

Characteristics: Enables representation of vague data.Fusion is done on fuzzy rules.

Advantages: Optimum method for vague data esp generated by humans

Limitations: limited to vague data only.

4. POSSIBILISTIC FUSION:

Characteristics: Represents data using probabilty.Its fusion is based on fuzzy logic.

Advantages: Helps in fusion of incomplete data especially in poorly informed environment.

Limitations: Not commonly used.

5. ROUGH SET BASED FUSION:

Characteristics: Representation of ambiguous data with the help of approximate upper nad lower bounds.

Advantages: Doesnt require any prior knowledge.

Limitations: Needs optimum level of data granuality.

2.2.2 FUSION OF INCONSISTENT DATA

Inconsistencies refer to disordered, conflicting and outlier data. There are many methods to tackle these issues.

1. Outlier Data:

Problem: Leads to inaccurate estimations if fused with correct data.

Solution: Sensor validation Technique and Stochastic adaptive sensor modelling.

Characteristics: It is restricted to prior known failures models.

2. Disordered Data:

Problem: It uses old measurements and estimates to update current estimates.

Solution: Using backward/forward prediction or use state frameworks for delayed estimates.

Characteristics: Assumes linear target dynamics and single lag delays.

3. Conflicting Data:

Problem: Gives Non intuitive output when fused using Dempster's rule.

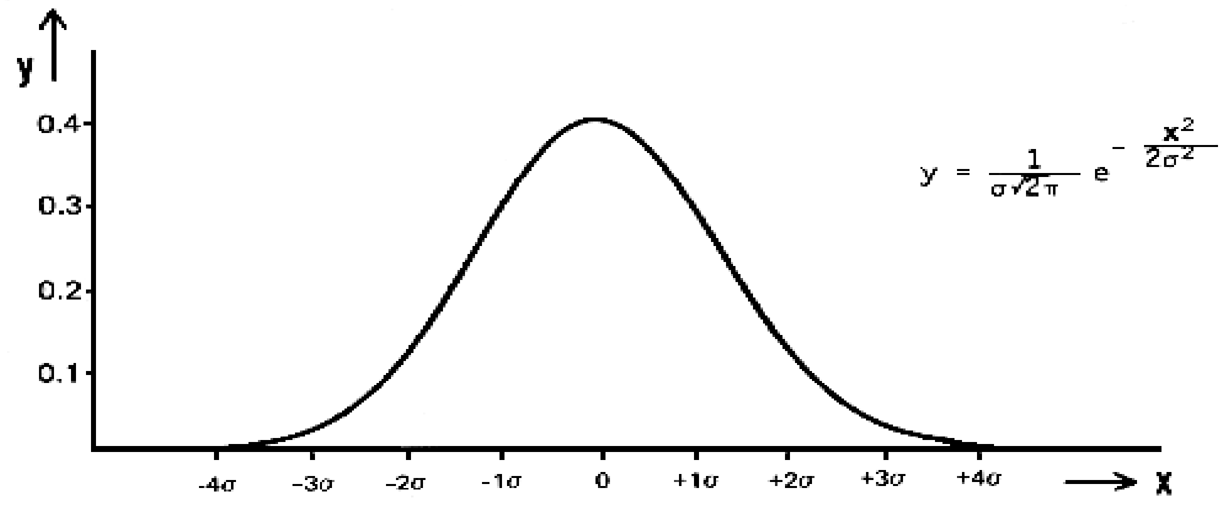
Solution: We can use alternate combination rule or apply correction techniques while using Dempster's rule.

Characteristics: It doesn't has a proper theoretical explanation and adding correction techniques changes validity of Dempster's rule.

2.3 Particle filter

Tracking objects in video involves the modeling of non-linear and non-gaussian systems.

- Non-Linear
- Non-Gaussian



Background-

In order to model accurately the underlying dynamics of a physical system, it is important to include elements of non-linearity and non-gaussianity in many application areas. Particle Filters can be used to achieve this. They are sequential Monte Carlo methods based on point mass representations of probability densities, which are applied to any state model.

Particle Filter is concerned with the problem of tracking single and multiple objects. Particle Filter is a hypothesis tracker that approximates the filtered posterior distribution by a set of weighted particles. It weights particles based on a likelihood score and then propagates these particles according to a motion model.

Mathematical Background:

Particle Filtering estimates the state of the system, x t, as time t as the Posterior distribution:

$$P(x_t \mid y_{0\text{-}t})$$

Let,

$$Est(t) = P(x_t \mid y_{0\text{-}t})$$

Est(1) can be initialized using prior knowledge. Particle filtering assumes a Markov Model for system state estimation. Markov model states that past and future states are conditionally independent given current state.Thus, observations are dependent only on current state.

To implement Particle Filter we need

- State Motion model: $P(x_t \mid x_{t-1})$
- Observation Model: $p(y_t \mid x_t)$:
- Initial State: Est(1)

We sample from the proposal and not the posterior for estimation.

To take into account that we will be sampling from wrong distribution, the samples have to be likelihood weighed by ratio of posterior and proposal distribution:

$$W_t = \text{Posterior i.e.Est}(t) / \text{proposal Distribution}$$

$= p(y_t \mid x_t)$

Thus, weight of particle should be changed depending on observation for current frame. A discrete set of samples or particles represents the object-state and evolves over time driven by the means of "survival of the fittest". Nonlinear motion models can be used to predict object-states. Particle Filter is concerned with the estimation of the distribution of a stochastic process at any time instant, given some partial information up to that time.

The basic model usually consists of a Markov chain *X* and a possibly nonlinear observation *Y* with observational noise *V* independent of the signal *X*.

Particle Filter is concerned with the estimation of the distribution of a stochastic process at any time instant, given some partial information up to that time.

The basic model usually consists of a Markov chain *X* and a possibly nonlinear observation *Y* with observational noise *V* independent of the signal *X*.

Given N particles (samples)

$\{x^{(i)}_{0:t-1}, z^{(i)}_{0:t-1}\}^{N}_{i=1}$ at time t-1,Approximately distributed according to the distribution $P(dx^{(i)}_{0:t-1}, z^{(i)}_{0:t-1} \mid y_{1:t-1})$, particle filters enable us to compute N particles $\{x^{(i)}_{0:t}, z^{(i)}_{0:t}\}^{N}_{i=1}$ approximately distributed according to the posterior distribution $P(dx^{(i)}_{0:t}, z^{(i)}_{0:t} \mid y_{1:t})$.

The basic Particle Filter algorithm consists of 2 steps:

- Sequential importance sampling step
- Selection step

1) Sequential importance sampling

Uses Sequential Monte Carlo simulation. For each particle at time t, we sample from the transition priors.For each particle we then evaluate and normalize the importance weights

2) Selection Step

Multiply or discard particles with respect to high or low importance weights $w^{(i)}_t$ to obtain N particles. This selection step is what allows us to track moving objects efficiently.

CHAPTER 3

METHODOLOGY

3.1 Particle filter Tracking

Our data set consist of 4 videos of a ball in a basket from 4 different directions. The ball is moving in the basket along the edges. This is captured by the 4 cameras from 4 different angles.

We have applied particle filter on the ball movement to track it during its motion.

The particle filter represents the samples from the distribution by a set of particles. Each particle has a log likelihood, which is a probability that the particle is sampled and updated in a particular direction. It is a state estimation method using a lot of particles. It constitute prediction i.e time evolution of particles according to system model and filtering i.e particles with more likelihood are more likely to be picked up.(Reselection of particles according to their likelihood.)

1. Prediction (move particles)

$$\begin{pmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \overset{\text{Location / Speed}}{\begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \dot{x}_{n-1} \\ \dot{y}_{n-1} \end{pmatrix}} + \overset{\text{Noise}}{\begin{pmatrix} v_x \\ v_y \\ v_{\dot{x}} \\ v_{\dot{y}} \end{pmatrix}}$$

2. Filtering (resample particles)

3. Prediction (move particles)

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \dot{x}_{n+1} \\ \dot{y}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \overset{\text{Location / Speed}}{\begin{pmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \end{pmatrix}} + \overset{\text{Noise}}{\begin{pmatrix} v_x \\ v_y \\ v_{\dot{x}} \\ v_{\dot{y}} \end{pmatrix}}$$

<u>State and observables in object tracking</u>

STATE is the location and speed of object.It is an unknown.

X= [x, y, x’, y’]

OBSERVABLE is the colour of pixel on which particles exist.It is known

Y= (r, g, b)

<u>Calculation of likelihood</u>

$$P\left(Y_n \,\middle|\, X^{(k)}_{n\,|\,n-1} \right) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp(-\frac{d^2}{2\sigma^2})$$

d $=\sqrt{(r-255)^2 + (g)^2 + (b)^2}$

When we multiply matrix F = $\begin{pmatrix} 1\,0\,1\,0 \\ 0\,1\,0\,1 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}$ from left the particles move.

We obtain data in the form of X =[x y v_x v_y]

These 4 parameters are the attributes for object tracking done using particle filter.

The data we received from different cameras can be depicted as in following tables

CAMERA 1 (UPSIDED CAMERA)

FRAME	V_y Mean	V_X Mean	L_Mean
Frame1	-3.12	0.04	inf
Frame 2	11.52	10.15	12.55
Frame 3	7.57	5.94	10.95
Frame 4	1.10	2.52	11.56
Frame 5	6.27	-18.21	11.22
Frame 6	0.88	-15.39	11.42
Frame 7	-11.34	-4.94	10.22
Frame 8	-12.23	0.27	10.88
Frame 9	-6.46	26.42	12.83

CAMERA 2 (RIGHT SIDED CAMERA)

FRAME	V_y Mean	V_X Mean	L_Mean
Frame1	1.95	3.06	Inf
Frame 2	6.90	8.43	11.90
Frame 3	-1.91	8.52	11.75
Frame 4	7.70	-3.61	10.85
Frame 5	14.37	-15.93	11.45
Frame 6	0.65	-3.85	11.42
Frame 7	-14.37	-12.03	11.57

Frame 8	-6.72	5.19	12.00
Frame 9	-9.44	20.95	12.93

CAMERA 3 (LEFT SIDED CAMERA)

FRAME	V_y Mean	V_X Mean	L_Mean
Frame1	-0.22	1.15	Inf
Frame 2	25.35	-9.5	12.31
Frame 3	7.19	3.95	12.06
Frame 4	-9.4	-11.31	12.10
Frame 5	-8.56	-19.16	13.16
Frame 6	-4.49	-2.97	12.70
Frame 7	-13.14	9.42	11.72
Frame 8	3.44	3.36	11.47
Frame 9	19.91	26.88	13.08

CAMERA 4 (DIAGONAL SIDED CAMERA)

FRAME	V_y Mean	V_X Mean	L_Mean
Frame1	0.04	-3.39	Inf
Frame 2	-9.39	1.19	11.43
Frame 3	-2.14	-5.42	11.03
Frame 4	-1.83	17.32	11.42
Frame 5	5.30	-3.34	4.07
Frame 6	38.33	-7.03	Inf

Frame 7	17.8	0.52	Inf
Frame 8	1.90	-3.81	11.08
Frame 9	-0.81	-4.41	10.85

3.2 Selection of Reliable Camera Sources

On applying particle filter we have obtained datasets from 4 cameras. On normalizing and applying threshold the datasets become:

CAMERA 1 (UP SIDED CAMERA)

FRAME	V_Y MEAN	V_X MEAN	L_MEAN
Frame 1	1	2	3
Frame 2	2	2	2
Frame 3	2	2	2
Frame 4	2	2	2
Frame 5	2	0	2
Frame 6	2	0	2
Frame 7	0	1	2
Frame 8	0	2	2
Frame 9	1	3	2

CAMERA 2 (RIGHT SIDED CAMERA)

FRAME	V_Y MEAN	V_X MEAN	L_MEAN
Frame 1	2	2	3
Frame 2	2	2	3

Frame 3	1	2	3
Frame 4	2	1	2
Frame 5	3	0	2
Frame 6	2	1	2
Frame 7	0	0	3
Frame 8	1	2	3
Frame 9	0	0	3

CAMERA 3 (LEFT SIDED CAMERA)

FRAME	V_Y MEAN	V_X MEAN	L_MEAN
Frame 1	1	2	3
Frame 2	3	1	2
Frame 3	2	2	2
Frame 4	1	0	2
Frame 5	1	0	3
Frame 6	1	1	2
Frame 7	0	2	2
Frame 8	2	2	2
Frame 9	3	3	3

CAMERA 4 (DIAGONAL SIDED CAMERA)

FRAME	V_Y MEAN	V_X MEAN	L_MEAN

Frame 1	2	1	3
Frame 2	0	2	2
Frame 3	1	1	2
Frame 4	1	3	2
Frame 5	2	1	2
Frame 6	3	0	3
Frame 7	2	2	3
Frame 8	2	1	2
Frame 9	1	1	2

While scrutinizing information sources, there are many factors that affect the authenticity, and reliability of the data. To characterize the reliability of information sources, we define the two source quality factors

1. Internal-confidence degree(IC)-Represent the credibility of the information source itself
2. External-confidence degree (EC) -The degree of mutual support between the sources in a multi-source information system.

The dataset from different cameras may include a lot of features. For the effective use of camera data, unnecessary features should be removed. Here we use concept of attribute reduction. For a multi-camera system, the reduction will be $Red(MI) = \{ Red(AT_1), Red(AT_2), \ldots, Red(AT_s) \}$. We defined measurement which shows the credibility of a source or reasonableness of a particular feature. This is called internal-confidence, and is determined by particular source only.

<u>Internal Confidence:</u>

Uncertainty is an important topic in information source systems. Current uncertainty measures include entropy theory plus significance degree.

Consider a camera information system $I = (U, AT, V)$. For an attribute $A \in AT$, $U/R_A = \{X_1 \ldots X_m\}$,we can find the rough entropy:

Each camera has many attributes, but some of these are unnecessary. To measure the significance of every single feature, we find the relative significance of every feature. Let camera = (U, AT, V) be an camera information system. For any and $a \in A$T, the relative significance of feature in attribute set AT *is calculated.*

$$Sig_{in}(a, AT) = E_r(AT - \{a\}) - E_r(AT),$$

For a given camera I, $a \in AT$, $Sig_{in}(a, AT) > 0$.the attribute is a reduct.

Theorem 1. Let Multi camera Information system = (U, AT, V $_a$). For each single information source let Red (AT) be a reduct of I. The internal-confidence degree of a source is calculated by:

$$IC(I) = \frac{\text{Red}|\ (AT|)\ |}{AT}$$

$0 \leq IC(I) \leq 1$, and that $IC(I)$ is ratio of modulus of $Red(AT)$ and AT . If $IC >$ 0.5, the source is credible and most of the attributes are useful .We keep different thresholds of IC according to the specific needs. IC is an absolute metrics which characterizes one camera in a multi-camera information system, and is calculated by a single camera's attributes. There exists no relationship between the ICs of each camera in the multi-camera information system.

External Confidence:

Theorem 2. Let *Multi camera Information system* = (U, AT_i, V) .For any two single camera Ii and Ij in *Multi camera information system* , the difference between them is calculated as:

$$D_{(I_i, I_j)} = \sum_{k=1}^{|U|}(|[x_k]_{AT_i} \cup [x_k]_{AT_j}| - |[x_k]_{AT_i} \cap [x_k]_{AT_j}|),$$

For this difference between two cameras, we can deduce.

- For every I_i, I_j, $D(I_i, I_j) \geq 0$;
- For every I_i, I_j, $D(I_i, I_j) = D(I_j, I_i)$;
- I_i, I_j, I_k be 3 different camera with feature sets AT_i, AT_j and AT_k, respectively. If relationship $AT_i \subseteq AT_j \subseteq AT_k$ is true, this means that $D(I_i - I_j) + D(I_j - I_k) = D(I_i - I_k)$.

Theorem 3. Let Multi camera Information system = (U, AT ,V) that has s cameras. For every two cameras $I_i - I_j$, the external correlation of I_i and I_j is calculated.

We can deduce that $ec(I_i - I_j) \in [0, 1]$, Also $ec(I_i - I_j) = ec(I_j - I_i)$, plus $ec(I_i - I_i)$ will always be 1 . This quantity is used to define correlation between two cameras in a multi-camera information system. To show the EC degree of a camera in the complete camera system, we state the cumulative EC degree .

Similar to the IC, varied thresholds EC can be given in to specific needs. EC is a relative metrics used to characterize one camera in a multi-camera information system, and is calculated by all cameras. After calculating all EC, we find the relationship between the EC of I_i and I_j, we calculate EC matrix M_{EC} .Its a s - dimensional symmetric matrix. Values on the main diagonal points are 1. The sum of the i th row (or column) is External Confidence

Then the total score weight is calculated.

(TS) Total Score = IC + EC.

3.2 Data Fusion

The reliable sources have been identified after completing the selection procedure. There are many ways for information and data fusion. Here we discuss fusion approach based on the fuzzy sets. We build fuzzy information granules from the camera data for each *x*. Here fuzzy granules are used to replace the object data. This multi-camera information fusion constructs a fuzzy information table, and each value in this table is a fuzzy number.

Let Multi camera Information system = (U, AT_i ,V) , $i = 1, 2, ..., s\}$ consisting of s cameras. For any x , the value of x for any one attribute is depited as $c(x) = (c^1(x), ..., c^s(x))$.

We get data as fuzzy numbers in the form $A_{ij} = (a_{ij}, m_{ij}, b_{ij})$. We calculate the distance(DD) and similarity degree(SD) between two entities,and the rough entropy of the fused system.

Theorem 4. Let Multi camera Information system = (U, AT ,V). The number of features in each camera is m. The relative Minkowski distance (d_M) between x_k and x_l is calculated.

We use Euclidean distance here, and calculate the similarity degree of x_l and x_k.The similarity degree $SI \in [0, 1]$, thus we normalize the distance

$$SI(x_l - x_k) = 1 - d*(x_l - x_k) .$$

Seeing that$0 < S(x_l, x_k) \leq 1$. If d is 0, then x_l is equal x_k . Taking threshold $\delta \in (0, 1]$, we take a new granule with respect to the similarity degree $S(x_l - x_k)$:

$[x_k]^\delta = \{ x_l \in U \mid SI(x_l - x_k) \geq \delta\}$. Thus we calculate the rough entropy and find the uncertainly measures.

The fused table obtained is:

Frame	V_X Mean	V_Y Mean	L Mean
Frame 1	(0.5, 1.5, 2.5)	(1, 2, 2)	(3, 3, 3)
Frame 2	(0, 2, 3)	(1, 2, 2)	(2, 2, 3)
Frame 3	(0.5, 1.5, 2.5)	(1, 2, 2)	(2, 2, 3)
Farme 4	(0.5, 1.5, 2.5)	(-0.5, 1.5, 3.5)	(2, 2, 2)
Frame 5	(1, 2, 3)	(0, 0, 1)	(2, 2, 3)
Frame 6	(1, 2, 3)	(-0.5, 0.5, 1.5)	(2, 2, 3)
Frame 7	(0, 0, 2)	(-0.5, 1.5, 2.5)	(1.5, 2.5, 3.5)
Frame 8	(-0.5, 1.5, 2.5)	(1, 2, 3)	(2, 2, 3)
Farme 9	(0, 1, 3)	(1, 3, 3)	(1.5, 2.5, 3.5)

CHAPTER 4

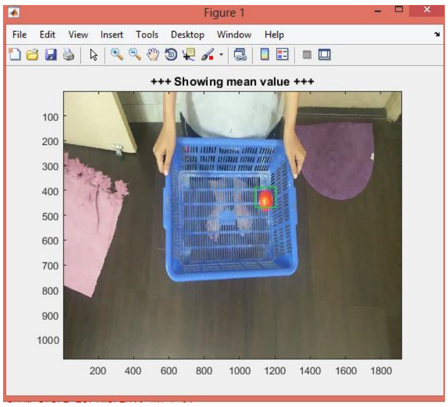
EXPERIMENTAL

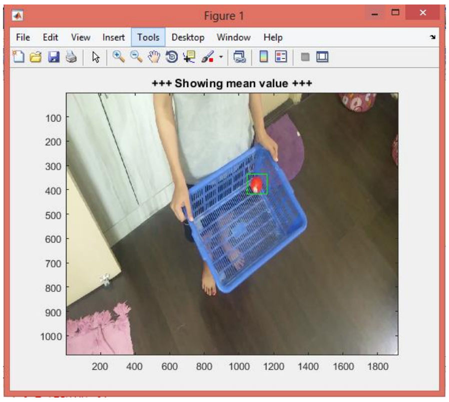
4.1 Particle Filter Tracking:

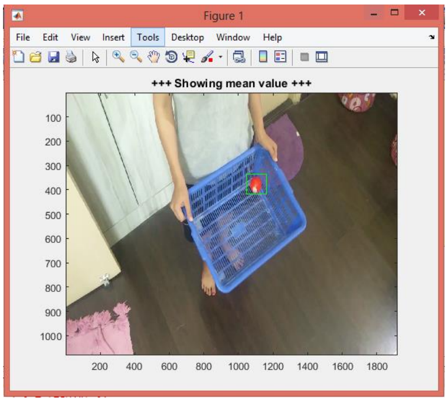
We have used particle filter to track an orange ball in a basket.We have taken 4 videos from different angles and tracked the ball in 4 different scenerios.

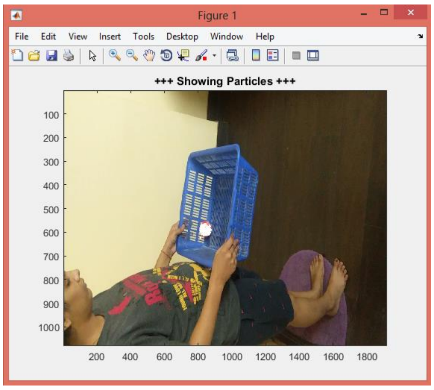
The 4 videos are :

1. Particle filter up
2. Particle filter left
3. Particle filter right
4. Particle filter diagonal









We have used 500 particles. The RGB value for orange is (255 69 0).

The functions used to track the orange ball in particle filter.

1. Create_particles :create 500 particles which constitute the location and speed value of each particle.
2. Update_particles : the values of each particle is updated using Xstd_pos for position and X_std
3. Cal_log_likelihood
4. Resample_particles
5. Show_particles
6. Show_estimated_state :

As output we get the mean location and speed of 500 particles as catalogued in the green box.

The values we obtain are

1. Mean location of 500 particles for each frame
2. Mean Speed of 500 particles for each frame
3. Mean likelihood of 500 particles for each frame

After implementing particle filter we obtain

1. X_up_mean- mean locations and velocity in x and y directions of 500 particles in each frame for upside video.

2. L_up_mean – mean likelihood of 500 particles in each frame for upside video
3. X_left_mean- mean locations and velocity in x and y directions of 500 particles in each frame for leftside video.
4. L_left_mean – mean likelihood of 500 particles in each frame for left side video
5. X_right_mean- mean locations and velocity in x and y directions of 500 particles in each frame for right side video.
6. L_right_mean – mean likelihood of 500 particles in each frame for right side video.
7. X_diagonal_mean- mean locations and velocity in x and y directions of 500 particles in each frame for diagonal side video.
8. L_diagonal_mean – mean likelihood of 500 particles in each frame for diagonal side video.

4.2 Finding reliable camera sources :

Using these attributes (mean velocity and mean likelihood),we fuse the data from 4 cameras and compare the results with individual camera performance

4.2.1 Rough and Information Entropy of each camera:

Rough Entropy of each camera comes out to be:

Camera 1- 0.75

Camera 2- 0.66

Camera 3- 0.22

Camera 4- 0.44

Iformation Entropy of each camera comes out to be:

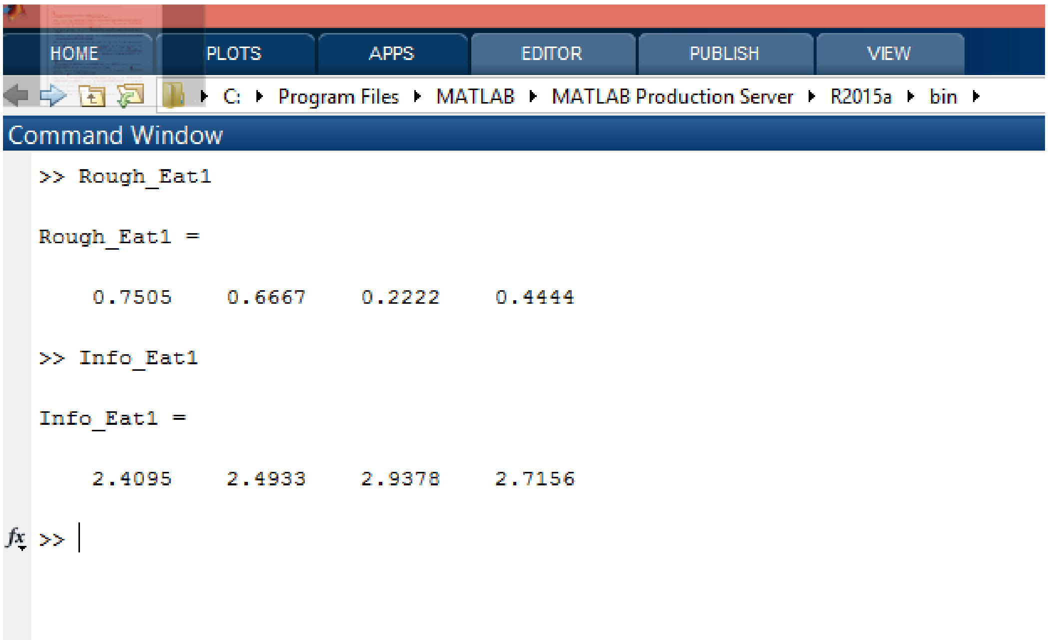
Camera 1- 2.41

Camera 2- 2.49

Camera 3- 2.94

Camera 4- 2.72

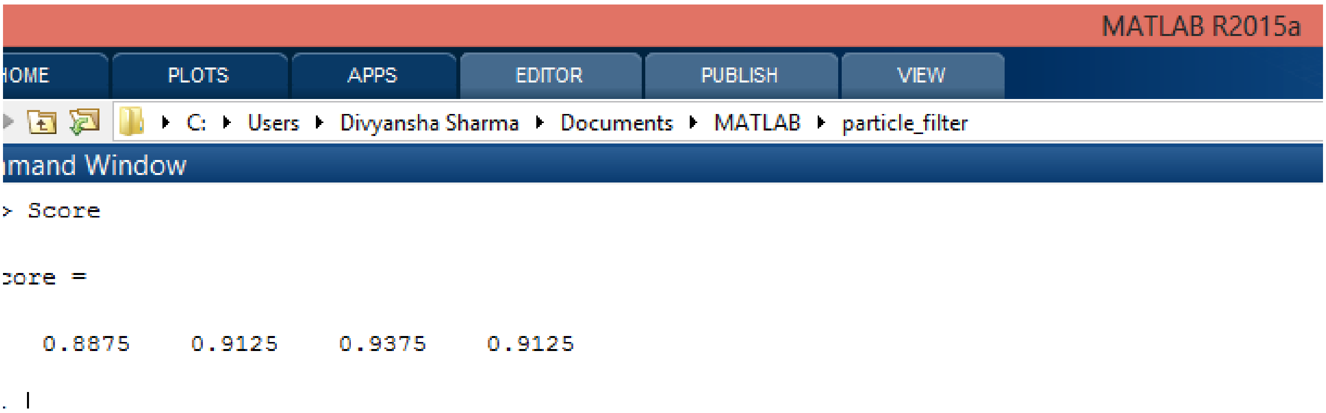
Information entropy should be greater for better performance.



Rough entropy for all cameras is calculated. Its order is 1>2>4>3

Information entropy for all cameras is calculated. Its order is 3>4>2>1

4.2.2 External Confidence and score:



External confidence and score is evaluated for each camera. EC and score order is 3>2 >=4>1.. which implies camera 3 gives the most reliable data followed by 2 & 4 then 1.

4.3 Fused Data:

The data from the camera is fused and we obtain data converted into fuzzy form (a,m,b).

```
F2(:,:,1) =

    0.5000    1.5000    2.5000
         0    2.0000    3.0000
    0.5000    1.5000    2.5000
    0.5000    1.5000    2.5000
    1.0000    2.0000    3.0000
    1.0000    2.0000    3.0000
         0         0    2.0000
   -0.5000    1.5000    2.5000
         0    1.0000    3.0000
```

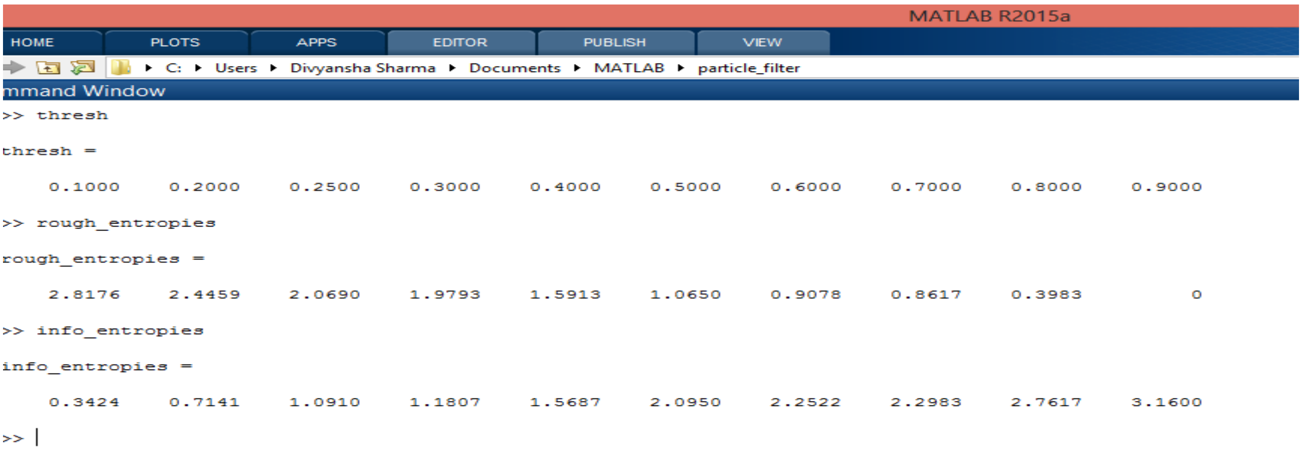
```
F2(:,:,2) =

    1.0000    2.0000    2.0000
    1.0000    2.0000    2.0000
    1.0000    2.0000    2.0000
   -0.5000    1.5000    3.5000
         0         0    1.0000
   -0.5000    0.5000    1.5000
   -0.5000    1.5000    2.5000
    1.0000    2.0000    2.0000
    1.0000    3.0000    3.0000
```

```
F2(:,:,3) =

    3.0000    3.0000    3.0000
    2.0000    2.0000    3.0000
    2.0000    2.0000    3.0000
    2.0000    2.0000    2.0000
    2.0000    2.0000    3.0000
    2.0000    2.0000    3.0000
    1.5000    2.5000    3.5000
    2.0000    2.0000    3.0000
    1.5000    2.5000    3.5000
```

Then for the fused table the entropies are calculated for different thresholds.



The data from 4 cameras is fused using Fuzzy Logic and the entropy of fused table is calculated for different thresholds from the result we conclude that entropy decreases with increase in threshold and the fused entropies are mostly greater than entropies from individual cameras.

These results are tested on different video durations and different number of balls.

CHAPTER 5

CONCLUSION

In this project we have studied object tracking by particle filter and multi camera information fusion through GrC. Particle filter is used to track the object and obtain attribute values of location ,speed and likelihood of particles. For excluding redundant and unreliable data and features, we calculated IC & EC to calculate significance of attributes of different cameras. These quantity measures the absolute and relative credibility of a single camera respectively. Considering these 2 measures a total weight or score of given for each camera. After this we learnt a new fuzzy concept for multi camera information fusion. By translating the multi camera data into fuzzy granules, the feature value of every entity was detected as a triangular fuzzy member. The fused data yields entropies than individual cameras and it found better in terms of object tracking.