Software Engineering Group L - Batch 2018

User Manual

Part-1

This is an extensive manual for the input image generation.

INTRODUCTION

Background

Considering a target with a Gaussian plume intensity, in the sensor's focal plane

$$S(x,y) = S_{max}e^{-\frac{1}{2}[\frac{(x-x_C)^2}{a_x^2} + \frac{(y-y_C)^2}{a_y^2}]}$$

with the center at (x_c, y_c) and semiaxes as a_x and a_y assumed to be oriented along the sensor's coordinates. The image noise added is additive white gaussian noise with zero mean and a fixed variance.

Aim

Given a source and destination of the target, generate grayscale images which depict the traversal of the target at a generalized sampling period of one second.

PRODUCT & PROCESS

Requirements

1. A Personal Computer with the latest working version of MATLAB.
2. A basic knowledge of MATLAB.
3. A basic knowledge of target detection and tracking.

Quick Start

1. Change working directory of MATLAB to the folder in which the images should be generated for further use.
2. Call the img_gen function with enough input arguments, to generate images and create a few global variables for their usage in montage generation and traversal depiction

USING THE FUNCTIONS

Image Generation

- img_gen(start_pt, end_pt, a_x, a_y, vel, img_edge,targ_int): This function generates grayscale images of a target travelling from ‘start_pt’ to ‘end_pt’ with a velocity ‘vel’. It is based on gaussian plume model and hence takes in semi-axes of the target - a_x and a_y, and the maximum intensity of the target, ‘targ_int ’. This function also needs the length of the edge of the image that has to be generated. The output is images generated from the given parameters stored as a PNG file under the name ‘test_i.png’ where i varies from zero to n-1 where ‘n’ is the total number of images generated. Every image is an inverted cartesian plane with origin at the centre of the image and y-coordinate increasing in the downward direction.

Image Representation

- Display_img(file_path): This function generates a mesh diagram of the target at various instances during the traversal. The function takes in the absolute path of the folder where the images are generated. The mesh diagrams contains peaks which indicates the target and plots the position of the target in every alternate image for small number of images and in every fourth image when there are large number of images

- Img_coll(file_path): This function also takes in the absolute path of the folder where the images are produced. This function displays the generated images in a rectangular montage. The output is an image with all the target images merged into one chronologically. This allows one to clearly understand and visualize the course of the target over all the images created by the img_gen function.

- streak(file_path): This function takes in the absolute path of the folder in which the images are generated. This function creates an image which depicts the path traversed by the target i.e. it creates a white streak to show the path.

- vid_gen(file_path) : This function generates a video of the trajectory followed by the target, using the images generated by the img_gen() function. The input required for this function is the absolute path of the folder of in which the images are generated and creates a video named “target.avi”.

PRECAUTIONS :

- Users must call the image generation function before they use other functions in order to get a correct output

- The users should delete the images already present before every image generation.

- Make sure that there are no images(PNG files) ,which are not related to the application, starting with ‘test_’ in the working directory.

- Clear all the variables by using keyword 'clear' on the workspace before every tracking.

RESOURCES

Mailing lists

For any more information on the system contact on the following email id.

- togarusuryateja1999@gmail.com

References

Tracking and data fusion: A handbook of algorithms - Yaakov Bar-Shalom.

Software Engineering Group L - Batch 2018

Part-2

This is the extensive manual for using the centroiding system to obtain info about the target.

INTRODUCTION

Aim

Given any image which contains a target, we calculate its location and provide its co-ordinate (w.r.t centre of the frame). Variance in the centroid is also calculated along with contrast of the image.

PRODUCT & PROCESS

Requirements

1. A Personal Computer with the latest working version of MATLAB.
2. A basic knowledge of MATLAB.
3. A basic knowledge of target detection and tracking.

Quick Start

1. Change working directory of MATLAB to the folder containing all the scripts or copy all the scripts to the current working directory.
2. Generate images using img_gen function to store global variables which are then used for centroiding, centroid variance and contrast.
3. Run the appropriate commands or function in the Matlab command window to store the image as a matrix having its elements as intensities.

USING THE FUNCTIONS

- imread(image_name) : This is an inbuilt Matlab function .it converts a given image into a matrix displaying the intensity at each pixel.
- Centroid(Intensity_location): This function calculates the centroid of the target. It takes the intensities of pixels of the frame containing the target and returns its location (coordinates w.r.t centre of the frame). The frame should be of odd grid. If Gray image is not provided then before centroid calculation it is converted into a Gray image.
- gpt_variance(Intensity_location, centroid, variance): This function calculates the variance of the centroid of the Gaussian plume target. It takes the intensities of the pixels of the frame containing the target, centroid of the target, variance of the noise introduced in the frame as inputs and returns the variance of the centroid. The semi axes and maximum intensity of the target are already stored as global variables from img_gen.
- gpt_contrast(Intensity_location): This function calculates the contrast of the Gaussian plume target. It takes the intensities of pixels of the frame containing the target and returns its contrast. The semi axes and maximum intensity of the target are already stored as global variables from img_gen.
- Target_stats(img_name, variance): This function calculates the centroid, variance of the centroid and its contrast. It takes the name of the image and variance of noise as inputs.The semi axes and maximum intensity of the target are already stored as global variables from img_gen so it is required to first generate the images so as to get correct values.
- centroid_variance(centroid, size, frame_SNR, Contrast, total_target_intensity): This function calculates the variance of the centroid of the target. It takes intensities of pixels of the frame containing the target, centroid of the target, variance of the noise introduced, total target related intensity as inputs and returns the variance of the centroid.
- cen_graph(variance) :
  This function calculates the centroid, variance in the centroid and contrast of the target of all the images/frames (named as test_N where N is the nth image/frame) present in the current directory . It also shows a graph of Centroid Vs Centroid Variance (for both X and Y coordinates) to compare and confirm parabolic nature of the centroid variance w.r.t its centroid. It takes variance of the white noise introduced as input and returns above mentioned information for all images/frames. The semi axes, maximum intensity of the target and total number of images generated are already stored as global variables from img_gen so it is required to first generate the images so as to get correct values.

PRECAUTIONS :

- The users should delete all the images and clear all the variables by using keyword 'clear' on the workspace before every tracking.

RESOURCES

Mailing lists

For any more information on the system contact on the following email id.

- ayush.prasad9@yahoo.com

References

A handbook of algorithms . Tracking and data fusion. - Yaakov Bar-Shalom.

Software Engineering Group L - Batch 2018

Part-3

This is the extensive manual for using the tracking system.

INTRODUCTION

Aim

Given a set of infrared images , we use them to predict the position of the moving object using White noise model .

PRODUCT & PROCESS

Requirements

4. A Personal Computer with the latest working version of MATLAB.
5. A basic knowledge of MATLAB

Quick Start

4. Change working directory of MATLAB to the folder containing all the scripts or copy all the scripts to the current working directory.
5. Run the appropriate script or function in the Matlab command window.
6. Change the inputs of img_gen in final_2 for different end and start points,velocities etc.

USING THE FUNCTIONS

- imread() : This is an inbuilt Matlab function .it converts a given image into a matrix displaying the intensity at each pixel.

PRECAUTIONS :

- The users should delete all the images and clear all the variables by using keyword 'clear' on the workspace before every tracking.

RESOURCES

Mailing lists

For any more information on the system contact on the following email id.

- dhaarna1999@gmail.com

References

A handbook of algorithms . Tracking and data fusion. - Yaakov Bar-Shalom.

Part 4

This is the extensive manual for using the Kalman Filter

Background

This part is considering the estimation of the state vector of a stochastic linear dynamic system. The state estimator for discrete-time linear dynamic systems driven by white noise — the (discrete-time) Kalman filter.

Under the Gaussian assumption for the initial state (or initial state error) and all the noises entering into the system, the Kalman filter is the optimal MMSE state estimator. If these random variables are not Gaussian and one has only their first two moments, then, in view of the discussion from the Kalman filter algorithm is the best linear state estimator, that is, the LMMSE state estimator.

Aim

To develop a system that can demonstrate the results of the Kalman Filter in some basic implementations and then apply the same filter to the data modeled.

PRODUCT & PROCESS

Requirements

1. A Personal Computer with the latest working version of MATLAB.
2. A basic knowledge of MATLAB

Quick Start

1. Change working directory of MATLAB to the folder containing all the scripts or copy all the scripts to the current working directory.
2. Run the appropriate script or function in the matlab command window.

USING THE FUNCTIONS

Demonstrate Working Of Kalman Filter

- Script : Kalman_sample
- Predict the position and velocity of a moving train 2 seconds ahead, having noisy measurements of its positions along the previous 10 seconds (10 samples a second). Ground truth: The train is initially located at the point x = 0 and moves along the X axis with constant velocity V = 10m/sec. Approach: We measure (sample) the position of the train every dt = 0.1 seconds. But, because of imperfect aperture, weather etc., our measurements are noisy, so the instantaneous velocity, derived from 2 consecutive position measurements (remember, we measure only position) is inaccurate. We will use Kalman filter as we need an accurate and smooth estimate for the velocity in order to predict train's position in the future.

Apply Kalman Filter On the Gaussian Plume Object Modeled

- Script : Kalman_application
- Predict the position and velocity of the gaussian plume target created using the above functions. The input is taken as consecutive images of the motion of the target and then centroiding algorithms are applied on the images. Based on these parameters the Kalman Filter is used to track and predict the position if the object.

RESOURCES

Mailing lists

For any more information on the system contact on the following email id.

- namansinghal198@gmail.com

References

- ESTIMATION WITH APPLICATIONS TO TRACKING AND NAVIGATION - Yaakov Bar-Shalom