Object Tracking using a Color-Based Particle Filter with an Adaptive Target Color Distribution

EL2320 - Applied Estimation

Miguel Allan Garcia Naude (Author)
KTH Royal Institute of Technology
magn2@kth.se

Matthew William Lock (Group Partner)
KTH Royal Institute of Technology
mwlock@kth.se

***Abstract*—Several methods exist to track objects. The particle filter has proven to be useful in non-linear and non-Gaussian tracking problems. The use of color features for object is advantageous as it can provide a framework that is rotation and scale invariant and robust with respect to partial-occlusion of an object and noise. However, color feature based algorithms are sensitive in particular to illumination and appearance changes of the tracked object. A review and implementation of an existing object tracking algorithm using a color-based particle filter is conducted. This algorithm attempts to solve the problem of illumination and object appearance changes by adapting the color distribution of the target during gradual changes. Tracking is achieved by comparing the color distribution of a target to that of hypothetical states (particles). Color distributions are represented using histograms in the HSV color space, in an attempt to reduce the algorithms sensitivity to illumination changes. It is shown that color-based tracking is sufficient for use in ideal environments where the target has a distinct color distribution, and is robust to non-rigidity of objects, as well as partial occlusion. However, it appears as though methods to address slowly a adapting target are insufficient in cases where noise and occlusion is slowly introduced and is ineffective in cases of sudden illumination and appearance changes.**

***Index Terms*—Particle filter, color-histogram, color distribution, object tracking, dynamic target tracking**

I. Introduction

In a world that is becoming increasingly digitized, camera technology finds itself being integrated into various different areas. As a result, tracking and detection of objects has become more prevalent due to its uses in many important applications, namely security and surveillance, autonomous vehicles obstacle avoidance, traffic monitoring, video communication and compression, etc [1], [2]. The problem lies in tracking objects in complex environments. In the real world, the appearance of the object subject to changes due to illumination variance, pose of the object, partial or full occlusion, or due to the non-rigidity of the object [3]. It is therefore desirable to develop a robust object tracking method that is successful in tracking under these changes. Thus methods that allow for a dynamic target appearance have been proposed, whereby the color representation of the target is adapted during gradual changes in target appearance.

The tracking of objects in real time to determine a range of motion parameters, can be achieved in a variety of different ways. Typically these are separated into two areas, *bottom-up* or *top-down* approaches [4]. Where *bottom-up* approaches firstly segment the image into regions that describe objects, which are then used for tracking, while the *top-down* approach uses the image to verify individual object hypotheses which are generated [4]. Methods such as Mean-Shift Algorithm typically fit into the *bottom-up* approach, while Background Subtraction and Bayesian based filters, such as the Kalman and Particle Filters fit into the *top-down* approach [2], [5]. It is worth noting the limitation of the Background Subtraction and Kalman Filter techniques, as the first can only be performed using a stationary camera and the latter requires that its motion model is linear and its noise Gaussian [2], [6]. However, in the case of the Kalman filter, one could investigate the use of an Extended Kalman Filter or Unscented Kalman Filter for the tracking of objects with non-linear motion models.

For these reasons, the use of a Particle Filter (PF) is ideal, as it performs well in solving problems of non-linear and non-gaussian systems [7], [8]. It is also widely used due to its ability to perform multi-modal processing, allowing it to track multiple strong hypotheses [7]. For the purpose of object tracking, particle filters were traditionally employed in a combination with edge-based image features [8]. The alternative to a PF with edge-based image features is a PF with color-based image features, which is the focus of this paper. This method makes use of the color distribution of an object, named the target model, which is then tracked using the PF by comparing its color distribution with the distribution of the particles/samples [8]. The color-based method is advantageous in that it provides robustness against partial occlusion and noise, and are rotation and scale invariant, however it struggles against illumination changes. On the other hand, the edge-based method is illumination invariant, but struggles in noisy environments [2], [8].

Alternatively, one could employ Gaussian Mixture Models (GMM) to represent the color of the target object as apposed to histograms. However, it is preferable to use image histograms, as only objects/regions in the image that have similar histograms are matched with the target histogram, whereas any region in the image that contains only one of the colors of the mixture will match when making use of GMM [8].

When using a color-based model, one should consider that the color of the target object is able to vary over time due to changes in illumination, visual angle, target object orientation and camera parameters [4]. Therefore it is appropriate that the color model of the target should be dynamic, such that it is adapted during periods of stable image observations, in order to deal with these changes in appearance [4].

This paper focuses on findings related to tracking using a color-based Particle Filter and discusses the implementation of various methods of static and adaptive/dynamic color-based target tracking using a PF to result in scale, illumination and rotation invariant target tracking that can perform well under partial-occlusion [8] [4] [2], [3]. Each of these different methods are evaluated by tracking different types of target objects in various videos. By comparing the qualitative results of each method, it is shown that color-based particle filter with an adaptive target model can provide a functional method for tracking objects under ideal conditions, but it is sensitive to sudden changes in target appearance and environmental changes, and the gradual introduction of unwanted features, such as noise and occlusion. It should be noted that ground truth data was unavailable for each dataset and therefore the performance of each tracking method is evaluated qualitatively through visual inspection.

The outline of this paper is as follows. The paper begins by introducing key points of the particle filter, after which the use of color distributions and histograms to represent objects is described. The paper then covers the use of color information with a particle filter. This is followed by a description of methods to address changing object appearance. The methodology pertaining to the implementation is then explained, followed by a section on experimental results obtained.

II. Particle Filter

Introduced in 1993, the Particle Filter serves as a non-parametric implementation of the Bayes Filter [6], [9]. It has become increasingly popular for solving real-time tracking problems, particularly when there is plenty of computational power available [9]. This is as it does not require a linear model and is not subject to Gaussian noise constraints. The PF also allows for multiple hypotheses to be considered; for instance a robot localisation problem in a highly symmetric environment where the robot could hypothetically be in multiple locations.

The PF uses a set of samples, known as particles, that are drawn from a posterior $bel(x_t)$ to represent this posterior [6]. Each particles serves as a hypothesis as to what the true state of the object may be at some time [4]. This set of particles is denoted as

$$X = \{x_t^{[1]}, x_t^{[2]}...x_t^{[m]}...x_t^{[M]}\} \tag{1}$$

The concept of the PF is to approximate the probability distribution of the objects state using this set of particles, where the likelihood of each hypothetical state denoted the weight, is included in the particle set [6]:

$$X = \{x_t^{[1]}, w_t^{[1]}...x_t^{[M]}, w_t^{[M]}\} \tag{2}$$

This weight is given by:

$$w_t^{[m]} = p(z_t|x_t^{[m]}) \tag{3}$$

,the probability of the measurement z_t under the particle $x_t^{[m]}$ [6] These weights are normalized, such that $\sum_{m=1}^{M} w^{[m]} = 1$.

In the case of localisation, where the position of the target is unknown, the sample set is typically initialised with all the particles having the same associated weight, $w^{[m]} = \frac{1}{M}$, where M is the total number of particle. With the positions of the particles uniformly distributed over the possible state space.

The sample set then traditionally evolves by propagating each particle through the possible state space at discrete time intervals, according to a pre-defined motion model.

Each particle is then assigned a weight according to equation 3. M samples are then drawn from the sample set with replacement, with the probability of drawing a sample given by its weight, $w^{[m]}$ [4], [6]. For tracking purposes, estimation of the mean state of the target object can then be performed using:

$$E[X] = \sum_{m=1}^{M} w^{[m]} x^{[m]} \tag{4}$$

There exist different ways to perform the sampling step of the PF. The traditional way of sampling (Multinomial Resampling) typically results in a loss of diversity in the particle population due to repeated resamping [6]. Systematic Resampling (Low Variance Sampling) is a method typically used to reduce sampling error (variance reduction in the state hypothesis) [6]. As a result multiple hypotheses can exist for longer.

III. Color Distributions

As mentioned previously a color-based PF makes use of the color histograms to represent the color distribution of a target object and the individual particle region hypotheses, which are later used to determine the likelihood of these hypotheses. The use of color-based image features is advantageous with regards to robustness against partial-occlusion, rotation and non-rigidity of the object [8].

Various different types color models exist to mathematically describe colors in images. Each of these models represent colors differently with regards to their characteristics: hue, saturation or chroma, and brightness or value. Examples of common color models include the RGB, CYMK, NTSC and the HSV/HSL color models. The last is an adaption of the RGB color model, consisting of three channels; Hue, Saturation and Value/Lightness, that is designed to be more intuitive and interpretable [10]. For the application of a color-based PF, the use of an HSV color space is ideal due to the separation of illumination and chrominance. This can be useful when dealing with environments where the illumination varies [11].

The color distribution model implementation that is discussed below was developed by Nummiaro et al [8]. Importantly, it is noted that if a particle represents a region of pixels, then the importance of an individual pixel to describe objects can be related to its distance from the region centre. For this, a weighting function is used

$$k(r) = \begin{cases} 1 - r^2 & : r < 1 \\ 0 & : \text{otherwise} \end{cases} \tag{5}$$

Where r is a representation of the distance from the region center. Therefore pixels closer to the pixel centre are more relevant, increasing the reliability of the color distribution when pixels close to the region edge belongs to the background or becomes occluded [8].

The color distribution of a region $p_y = \{p_y^{(u)}\}_{u=1,..,m}$ with center y is calculated using

$$p_y^{(u)} = f \sum_{x_i \in R} k \left(\frac{||y - x_i||}{a} \right) \delta[h(x_i) - u] \tag{6}$$

[8]

where a defines the region size of interest and $\delta[h(x_i) - u]$ is the Kronecker delta function that assigns the color of a given pixel at location x_i to a bin u of the m set of bins. Finally, f is a normalisation factor that results in

$$\sum_{u=1}^{m} p_y^{(u)} = 1 \tag{7}$$

This results in a color histogram of an image region that can be compared to other color histograms using different distance measures. For tracking purposes these histograms are used as observations in each frame of a video sequence. The Comparative study of histogram distance measures for re-identification by Marin-Reyes et al [11] agrees with Nummiaro et al [8] that the Bhattacharyya coefficient should be used to measure the difference between two color distributions. This a common method used to find the distance between two distributions. Given two discrete color densities represented by histograms $p(u)$ and $p(v)$ the coefficient is defined as

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}} \tag{8}$$

Where $0 \leq \rho[p, q] \leq 1$. The magnitude of $\rho[p, q]$ correlates to the similarity between the two color distributions. The Bhattacharyya distance, indicating the distance between two color distributions [8] is given by

$$d = \sqrt{1 - \rho[p, q]} \tag{9}$$

IV. Color-Based Particle Filter

As mentioned previously, a PF draws from a set of particles that each represent a hypothetical state of the tracked object, where the probability that the particle is sampled is given by its weight. In the case of the color-based PF the Bhattacharyya distance, as defined in equation 9 between the color distributions of the target object and a region around a hypothetical state given by a particle, is used to calculate the weight, $w^{[m]}$ associated with that particle. Thereby updating the priori distribution.

The regions around particles from which color distributions are calculated, can be represented in various ways. Depending on the structure of the object to be tracked some approaches may be better than others. A simple approach would be to represent the region around a particle using a rectangle, with the rectangle centre being the particle. For objects that are not rectangular or do not have sharp vertices, this may result in a large number of background pixels being included, reducing the reliability of the color distribution.

Instead, a different approach, where regions around particles and the target are represented by ellipses can be used [8]. This approach is more reliable with regards to the resulting color distribution and is more computationally efficient. These elliptical region are given by

$$\frac{(x - x_i)^2}{H_x^2} + \frac{(y - y_i)^2}{H_y^2} = 1 \tag{10}$$

Where the centre of an ellipse in the image frame is given by the particle position (x, y), and the length of the half axes is given by H_x and H_y respectively.

A single particle is therefore defined as follows:

$$x_m = \{x, y, H_x, H_y, w_m\} \tag{11}$$

Motion Model

The particle set X is propagated using some motion model. The model below is commonly used for such applications [2], [4], [8]

$$x_t = A x_{t-1} + w_{t-1} \tag{12}$$

Here A defines the deterministic system model and w_{t-1} is the process noise. In this case, A is the identity matrix and w_{t-1} is random Gaussian noise, $w_{t-1} \sim \mathcal{N}(0, \sigma^2)$. This is known as the random walk model. It is computationally efficient, as velocity does not need to be included in the state vector.

A first order model for object tracking can also be achieved by adding velocities in the x and y directions to the state.

Measurement Model

As mentioned above, the weight of the particles is defined using the Bhattacharyya Distance between the color distribution of the target object and a region around a particle. The color histograms of the target and the elliptic region of the particle are calculated using equation 6. Here

$a = \sqrt{H_x^2 + H_y^2}$, as specified by Nummiaro et al [8]. The Bhattacharyya distance d between the respective histograms is then calculated and the weight of the particle is defined using a Gaussian

$$w^{[m]} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} \tag{13}$$

Where σ is the measurement noise that represents the uncertainty surrounding the similarity between the color distribution of the target and that of the hypothesis.

V. Adaptive Target Distribution

The PF can perform well in the case of tracking objects under partial and temporary occlusion, as less likely particles are able to exist in the sample set temporarily. However, due to changes in camera angle or parameters, object orientation or illumination, the color-based PF may struggle to track the object due to it being unable to adequately match the color distribution of any of the hypotheses with that of the original color distribution of the object.

Slowly Target Varying Color Distribution

To overcome the gradual change in color distribution of the target object, a method whereby each bin of the target object histogram is updated was proposed in a second paper by Nummiaro et al [4]. The color histogram of the target is updated as follows

$$q_t^{(u)} = (1 - \alpha)\ q_{t-1}^{(u)} + \alpha\ p_{E[S_t]}^{(u)} \tag{14}$$

Where $q_t^{(u)}$ is the updated target histogram, $q_{t-1}^{(u)}$ the previous target histogram and $p_{E[S_t]}^{(u)}$ is the histogram of the region around the mean state of the particles. α weighs the contribution of the mean state histogram against the previous target histogram.

To prevent the target histogram from updating when the target is lost, occluded or the image is too noisy, it is stated that the target should only be updated when a threshold on the similarity between the mean state histogram and the original target histogram is met [4].

$$\pi_{E[X]} > \pi_T \tag{15}$$

Where $\pi_{E[X]}$ is the observation probability of the mean state, while π_T is the set threshold.

Rapidly Varying Target Color Distribution

Visual tracking in real world environments is typically extremely challenging, even if the tracking algorithm is robust against non-rigidity and orientation changes of the target object. Robust tracking under sudden illumination poses a more difficult problem.

In the method proposed by Nummiaro et al [4], a limitation is that there is a trade-off between how sensitive the model should be to changing color distribution and how quickly the target color distribution should update. Thus it can fall short of tracking an object under sudden appearance changes.

Another method for tracking with an adaptive target color distribution was proposed by Li et al [3]. In this approach, the authors break down the state space into two parts, a kinetic part (K) and a color model part (M). The state space of a particle is thus represented using

$$x_t^{[m]} = \{K_t^{[m]}, M_t^{[m]}\} \tag{16}$$

The color model part of the state space consists of two variables, $M_t^{[m]} = \{\alpha, \alpha'\}$, where α weighs the contribution of the mean-state histogram to the new target histogram as in equation 14, and α' represents its rate of change [3]. These two variable as thus also estimated in the sequential Monte Carlo particle filter.

VI. Methodology

The tracking of target objects in video sequences was performed using a Color-Based PF, based on the method defined by Nummiaro et al [8]. However, instead of handling color information in the RGB domain (like in [8]), color information is used in the HSV domain due to it being more suitable for environments with changing illumination.

A. Target Object Identification

Here, the color distribution of the target object and its original size are specified. If only tracking of the target from a starting position, and not localisation of the target is desired, then its initial position is also specified. The color histogram of the target is determined by creating an image mask over the objects position in an image frame. The RGB information of the target is then converted to HSV. The color distribution of the target is then calculated using equation 6, where number of bins used per channel (H-channel, S-Channel & V-channel) was typically $(16, 16, 1)$, but could be different depending on the object.

B. Particle Filter Initialisation

The particle filter is typically initialized with between 150 and 400 particles depending on the size of the target object. Originally, the regions around the target object and the particles were implemented simply using rectangular regions, where the centre of the region was given by the position of the particle. This approach resulted in reduced reliability of the color distribution of the regions as they often included a large number of background pixels. Instead, an where regions around particles and the target are represented by ellipses was adopted. This approach is more reliable with regards to color distribution and is more computationally efficient.

The state vector of each particle is given by

$$x_m = \{x, y, H_x, H_y, w_m\} \tag{17}$$

Where its constituents are as explained in the earlier section. If localisation of the object in the space is desired, the initial position of the particles is uniformly distributed over the possible image space. If the object is only to be tracked, the particles are initialized around the initial position of the object. The parameters describing the length of the ellipse half axes were defined based on the size of the target object. Lastly, the particles are initialised with equal weights of $\frac{1}{M}$.

For the purpose of simplicity, the motion model used was the random walk model, as given in equation 12. The generality of the this model allows it to be applied to a wide range of object tracking applications. The state parameters describing the elliptic region around the particle are propagated in the same way as that of the particle position, by adding Gaussian noise to the previous state. Hence the process covariance matrix is 4x4 diagonal matrix, that is tuned manually.

C. Observations and Weight Assignment

As with that of the target object, the color histogram of each particle is calculated using equation 6. To reiterate, this creates a weighted histogram where pixels closer to the centre of the particle are weighted higher when defining the color histogram. The number of bins used for each channel is the same as that for the target histogram. Setting the V-channel to only have one bin attempts to make the observation more robust with regards to illumination changes [2].

During initial experimentation, the distance between the color distribution of the target and a particle region was calculated using the Euclidean distance. After conducting more reading, it was decided that the Bhattacharyya distance would be used to compare two distributions. Therefore the distance between two color distributions is calculated using equation 9, after having calculated the Bhattacharyya coefficient of each distribution using equation 8. The particles are then assigned weights using the the Gaussian formula 13. The σ parameter of the Gaussian formula correlates to the variation in magnitude of the weights and it was randomly chosen. These weights are then normalized.

It should be noted that it is assumed that the color distribution of particles that move out of frame of the image remains the same.

D. Resampling

Resampling can be achieved in a number of ways. In our implementation we make use of Systematic resampling to try to reduce the loss in variance of the particle states. This improves the chance of recovery if the tracker temporarily loses the target.

E. Position & Size Estimation

The position of the target is then estimated by using the mean state, $E[X]$ of the particle positions as given by equation 4. The size of the target is approximated by using a similar method as directly above. Instead, the mean state of the particle ellipse parameters H_x and H_y are calculated using

$$E[X] = \sum_{m=1}^{M} w^{[m]} H_x^{[m]} \tag{18}$$

The resulting ellipse provides a rough estimate on the size of the target object.

F. Adaptive Target Color Distribution

In order to try compensate for gradual changes in appearance of the target object we implemented the method proposed by Nummiaro et al [4], whereby the color histogram of the target is updated according to equation 14. The threshold required and α value were manually set. The use of the adaptive target color distribution can easily be toggled off by setting $\alpha = 0$.

VII. Experimental Result

To show the performance of the implementation of the color-based PF, both with and without an adaptive target color distribution, the methodology above was applied to a number of video sequences. The point of these experiments is to assess the performance of the tracking algorithm under differing conditions such as ideal conditions, partial occlusion, varying illumination, non-rigidity of the target and the presence of multiple strong hypotheses. The goal in each experiment is to track a predefined target for the duration of the video. In some experiments, the particle positions were uniformly distributed on initialization, to allow it to localize the target prior to tracking, while for others the particles are initialised around the position of the target. As explained previously, it would be extremely cumbersome to generate ground truth data for each video sequence, and thus the performance of the algorithm implementation is judged qualitatively.

Each of these experiments were processed using the HSV color space with $(16, 16, 1)$ bins. In the first three videos an image scale of 0.3 is applied, while an image scale of 0.6 is applied in the last video. It is found that reducing the scale of the images both increases the computational efficiency and the performance of the algorithm. The parameters for each experiment are summarized in table I.

The first experiment conducted is the localisation and tracking of a basketball as it travels towards the basket [12], shown in figures 1. This was conducted using 400 particles. For the adaption of the target color distribution, a observation probability threshold and alpha value of 0.45 and 0.05 were used respectively.

Figure 1 indicates how the algorithm is able to accurately locate an object with a distinct color distribution, although it should be mentioned that it was significantly more difficult to locate the target with less than 300 particles.

The inconsistency of the results should be noted, as the particles sometimes converged to incorrect locations where orange existed, such as the basketball hoop, or the persons shoes. This could be improved by making use of more particles, but at a large computation expense.

Experiment	Image Resolution	Number of Particles (M)	Process Noise w_t (R)	α	Observation Probability Threshold
Basketball Tracking	780x1280	400	0.3 $\times$ diag([20 20 5 5])	0.05	0.45
Car Tracking	1080x1920	200	0.3 $\times$ diag([50 50 20 20])	0.05	0.45
Bee Tracking in Minecraft	1080x1920	200	0.3 $\times$ diag([50 50 20 20])	-	-
Person Tracking	324x576	200	0.6 $\times$ diag([50 50 20 20])	-	-

TABLE I: Parameters used for the different experiments

Figure 2 illustrates how the algorithm is able to accurately track a rigid target with a distinct color distribution in a space where it is not occluded. Although it may not appear so, the target color distribution changes significantly at the top of the arc of its path. This is illustrated in figure 3, which shows that the observation probability of the mean state decreases significantly around frame 60. Momentarily the algorithm is less confident that it is tracking the target (second image - frame 80 in figure 2; however, it later regains confidence (frame 120).

Fig. 1: Localisation of a Basketball, M = 400 Particles. The green ellipse in the second frame represents the mean state.

Fig. 2: Successful Tracking of a Basketball, M = 400 Particles. Frames 35, 80, 120 & 160. The presence of a red ellipse indicates that the algorithm is confident that it is tracking the target object.

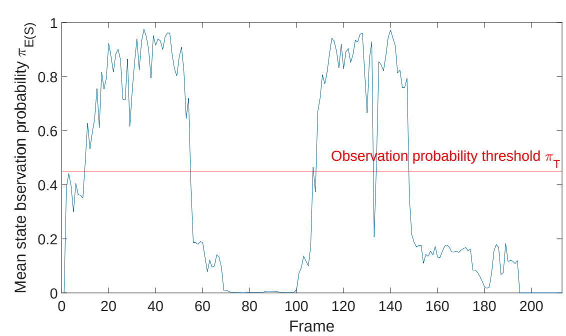

Fig. 3: Observation Probability of the Mean State for each frame

The second experiment involves the tracking of a car on a busy road [13] and is shown in figures 4 and 5. This a good example of tracking an object of changing scale. The experiment made use of 200 particles to track the car both without (figure 4) and with (figure 5) an adaptive target color distribution. Since the size of the car is originally very small, the particles are initialised on the location of the car to avoid needing excessively many particles for localisation.

In both cases, the mean-state remains on the car for the duration of the video, with the size of the mean-state ellipse increasing with the increasing size of the car. However, when an adaptive target distribution is used, it was observed that the size of the mean-state ellipse increases beyond the size of the car, and eventually began to resemble the color distribution of the background. This can be explained by the fact that noise and occlusion (as seen in frame 680) are gradually introduced into the target color distribution. This is seen as the mean-state ellipse gradually grows and begins to integrate increasing amounts of the road into the target distribution. In frame 680, it can be seen that the tracking algorithm is confident it is tracking the car, even though the car is partially occluded by a motorcycle. The mean-state ellipse has grown such that the algorithm now considers large portions of tarmac and the motorcycle as part of the target, which occurs as the occlusion took place very gradually. Therefore the adaptive target distribution model does not perform sufficiently in cases where noise or occlusion are gradually introduced.

With the increasing size of the mean-state ellipse, the computation time between frames increased due to the increasing number of calculations. Therefore the computational efficiency of the algorithm with the adaptive target color distribution reduced largely over time.

Fig. 4: Tracking of a yellow car of changing scale without use of adaptive target color distribution, M = 200 Particles. Frames 30, 300 & 680

Fig. 5: Tracking of a yellow car of changing scale with use of adaptive target color distribution, M = 200 Particles. Frames 30, 300 & 680

The third experiment involves the localisation and tracking of a bee in a video sequence of a video game that was self generated. Illustrations of this experiment are shown in figures 6 and 7. For tracking purposes only, the filter was initialised with 200 particles. Localisation could be achieved, although it proved unreliable as a large number of particles were needed due to the small size of the bee and because multiple different objects in the image have similar color distributions to that of the target.

This experiment proved interesting for various reasons. Firstly it demonstrated that color distribution is not a very reliable method for object identification, as several objects may possess the very similar color distributions. This is seen in figure 6 where the algorithm shifts from tracking one bee to tracking another. This is due to a fast rotation of the original target bee, while the other bee with within close proximity. As a result, the particles converge to and identify the second bee as the target as its orientation results in its color distribution being more closely related to that of the originally defined target.

To prevent the chances of the tracker from switching from one bee to another, the process noise pertaining to the position of the particles can be decreased, not allowing the particles to propagate far from their previous position (to prevent a particle from directly falling on the other bee). Multi-hypothesis tracking can also be improved by increasing the measurement noise allowing for longer tracking of more than one strong hypothesis.

Although this was conducted without an adaptive target color distribution, it would be indifferent as the change in color distribution of the original bee is not gradual enough.

Secondly, the experiment demonstrated that the algorithm is able to track an object under partial occlusion (assuming the target object has a distinct enough color distribution). This is shown in figure 7, where the bee is partially occluded by grass, after-which the tracker recovers its position.

When conducting this experiment while using the adaptive target color distribution, the target color distribution gradually began to resemble the color distribution of the background and eventually failed to track the bee.

Frame 325

Frame 345

Fig. 6: Tracking of a bee in Minecraft. This figure illustrates the change in the object being tracked due to the color distribution of another bee more closely resembling the color distribution of the target for a number of frames, M = 200 Particles.

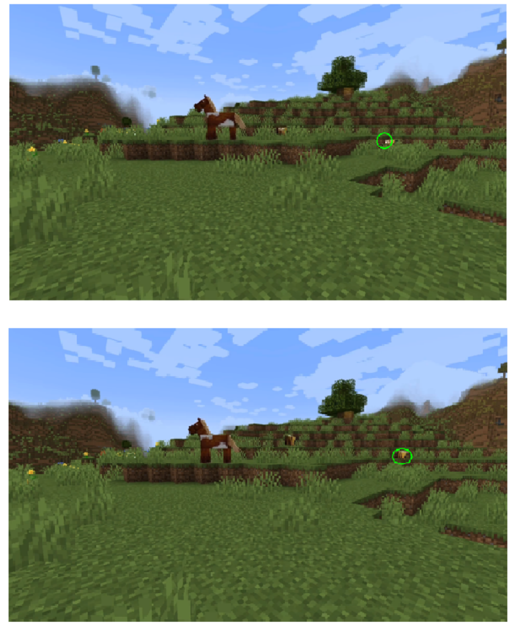

Fig. 7: Tracking of a bee in Minecraft. This figure illustrates the ability of the algorithm to track a target under partial occlusion, M = 200 Particles.

The last experiment involved tracking a person in a cluttered environment under rapidly changing illumination conditions. This was conducted using 200 particles which were initialised on the target for tracking. The target is tracked successfully until the illumination of the area changes drastically, thereafter the algorithm fails to detect and track the target. This is judged to be due to the mean-state color distribution changing too rapidly and thus the target distribution is not updated accordingly. It demonstrates that the implementation of the algorithm is not robust under rapidly changing illumination conditions. The results of this experiment are illustrated in figure 8. As can be seen in the third image (frame 80), the particles are unable to detect the target after decreased illumination.

The decrease in similarity between the mean-state histogram and the target histogram is shown by the observation probability plot in figure 9. On frame 72 of the video sequence, there is the sudden change in illumination and the observation probability drops considerable. After which, the tracker cannot recover.

This last experiment can be compared to an experiment performed by Mukhtar et al [2], in which they claim to successfully track a person under changing illumination. While the results appear successful, the experiment does not appear challenging enough in that there is not a sudden large change in illumination. In their proposed algorithm, they instead make use of color information in the YIQ form and utilise background subtraction together with the particle filter, which may lead to improved results [2].

Fig. 8: Tracking person in cluttered environment. Fails to track person under rapidly changing illumination, M = 200 Particles. Frames 2, 50 & 80.

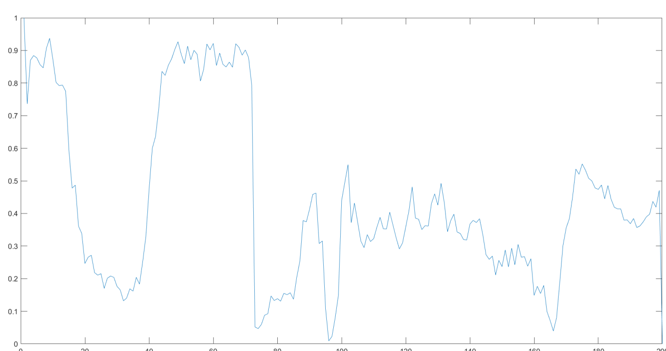

Fig. 9: Observation Probability of the mean-state per frame, while tracking a person in a cluttered environment under sudden illumination changes, M = 200 Particles.

Having conducted these experiments, the following should be expressed.

Two regions having the same color distribution does not necessarily mean they represent the same object, as multiple different object are able to have the same color distribution. Unless the object has a distinct color distribution in the environment, color distribution is not a reliable method for object identification. If the object does not have a distinct color distribution or is very small, the particles should be initialised on the target for tracking.

It was recognised that the level of quantisation of the color histograms has an effect on the performance of the tracker. Too few bins results in colors that may appear rather different being assigned to the same bin, while too many bins causes a noisy distribution and a lower computational speed, both reducing the performance of the algorithm.

While, using the random walk model appears sufficient as a motion model in simple applications, it is believed that the tracking performance could be improved considerably if a more complex motion model that more closely resembles the motion of the target object, is implemented. This could assist in tracking the target under complex conditions, such as full occlusion for a more extended period of time. Although, this will result in more computation load, as velocities will need to be added to the state of each particle.

Shortcomings of the gradually adapting target color distribution can be observed. It is insufficient for tracking of objects when noise or occlusion are gradually increased. It also fails to track the target object under sudden illumination changes. This is made difficult by the trade-off between an increasing sensitivity (decreasing the threshold parameter of the algorithm) and an increasing color distribution adaption speed (increasing alpha).

Lastly, it was observed that higher image resolution yielded worse tracking of objects. Hence higher resolution data should be scaled down to yield better results.

VIII. Summary and Conclusion

This paper covers the implementation of color-based particle filter for object tracking proposed by Nummiaro et al [4] with minor modifications with respect to the color space used. This algorithm also included a model for adapting to gradual appearance change of the tracked object. It can be said that this implementation could successfully track fast moving and non-rigid objects, with distinct color distributions under partial occlusion and in cluttered environments. However, the algorithm failed to successfully track the target under sudden appearance changes. Under gradual appearance changes the tracking of the object proved better. Although, for increased sensitivity of the adaption model, the estimation of the target size often grew larger than the target object itself, specifically when undesirable features are introduced gradually, eventually leading to slower computation and failure to track the object successfully.

The algorithm was implemented using the HSV color space as opposed to the RGB, in an attempt to increase performance under changing illumination; though, it appears indifferent for sudden large illumination changes.

It is observed that the execution time of the algorithm depends mainly on the number of particles, the size of the tracked object and the image resolution. For future work, it is recommended that a motion model more closely resembling the motion of the target, as it can improve performance under complex conditions. The use of an algorithm that can adjust its number of particles may also prove useful. The presence of more particles initially would allow for better localisation of the target, and the ability to decrease the number of particles during tracking will decrease the computational load. There are also many redundant computations that occur with respect to particles that lie on the same point or within close proximity. To decrease the computational load, this should be addressed. Lastly, an algorithm for better adaption of the target under sudden appearance changes should be investigated and implemented. We attempted to implement the method proposed by Li et al [3], but it showed no major improvement over the method developed for a gradual change in appearance of the target. There was insufficient time to investigate why this was the case and hence it should be revisited in future. This would likely lead to the largest improvement in the performance of the tracker.

References

[1] A. Vahab, M. S. Naik, P. G. Raikar, and P. S. R, "Applications of object detection system," *International Research Journal of Engineering and Technology*, p. 4186, 2008. [Online]. Available: www.irjet.net

[2] A. Mukhtar and L. Xia, "Target tracking using color based particle filter," in *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2014, pp. 1–6.

[3] X. Li and N. Zheng, "Adaptive target color model updating for visual tracking using particle filter," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, Oct 2004, pp. 3105–3109 vol.4.

[4] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "Object tracking with an adaptive color-based particle filter," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 353–360.

[5] A. Ali and K. Terada, "Object detection and tracking using kalman filter and fast mean shift algorithm," vol. 1, 2009, pp. 585–589.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2005.

[7] S. Srinivasan, "Particle filter : A hero in the world of non-linearity and non-gaussian," 8 2019. [Online]. Available: https://towardsdatascience.com/particle-filter-a-hero-in-the-world-of-non-linearity-and-non-gaussian-6d8947f4a3dc

[8] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "A color-based particle filter," 01 2004.

[9] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[10] H. Kumar, "Color and color spaces in computer vision," 1 2020. [Online]. Available: https://kharshit.github.io/blog/2020/01/17/color-and-color-spaces-in-computer-vision

[11] P. A. Marín-Reyes, J. Lorenzo-Navarro, and M. C. Santana, "Comparative study of histogram distance measures for re-identification," *CoRR*, vol. abs/1611.08134, 2016.

[12] WIRED, "Why shooting 95% from the free-throw line is almost impossible (ft. steve nash) — wired," 2019. [Online]. Available: https://www.youtube.com/watch?v=BKIOqbx3sbU&ab_channel=WIRED

[13] P. Security, "4k camera example for traffic monitoring (road)," 2015. [Online]. Available: https://www.youtube.com/watch?v=jjlBnrzSGjc&ab_channel=PanasonicSecurity