Visual Color Based Object Tracking Using Particle Filter

Pernilla Wikstrom[1] pwikst@kth.se

***Abstract*— This paper presents results of the project in the course Applied Estimation at KTH, Stockholm. The project presents a method for visual color based object tracking using particle filter. By pre-processing the data with help of a color based RGB computer vision technique the filter tracks objects through noisy scenes. Colors will be compared from frame to frame for tracking. By using an arbitrary scene the motion model is set to be Gaussian. Two sets of re-sampling methods has been studied, multinomial and systematic. The result verifies that the tracking algorithm has fulfilled its mission to track an object in a video.**

Keywords: computer vision; object tracking; particle filter; weighting.

I. INTRODUCTION

The capability of tracking objects has exploded over the last decades. Object tracking in videos involves modeling of non Gaussian and non linear distributions which is non trivial. By creating solutions of the image processing problems, object tracking is used for surveillance, human computer interaction and video communication such as safety systems for vehicles or robot navigation. Vehicles with camera systems are often able to use lane detection, pedestrian detection and/or traffic sign alerts. Through object tracking robots can navigate to see features in localization methods. The outcome of using these video tracking algorithms contributes to the safety of vehicle environments, it changes the possibility of automation in industries and keep our facilities safe.

The common element of these systems integrates camera systems that provide data used in image processing. The challenge in this paper is to find image treatments that allow for efficient object recognition without being too computationally heavy or complex in its algorithm. As well as implementing and parametrisering the particle filter which is a sample based method that represent a particular density. It is a nonparametric filter and is used for non Gaussian and non linear distributions that represents the density with a state and weight [1]. The idea is to compute the posterior distribution of the states of some Marcow process, given a noisy scene and partial observations. The mission of the algorithm is to find a set of particles that gives us that representing density located in the target distribution. The basic concept is to sample from the proposal distribution which corresponds to the proposal belief $\bar{bel}(x_t)$. Followed by computing importance weights which tells the likelihood of each particle. By using the proposal belief and the importance weights it is possible to compute the target distribution which is the true posterior distribution that we try to approximate and corresponds to the target belief, $bel(x_t)$. The particle filter is described in the basic algorithm 1 [1] where the keys are related to sequential weighting, sampling and selection. The hypothetical state $x_t^{[m]}$ is approximated as the state transition probability given the control u_t and posterior particle $x_{t-1}^{[m]}$. To make this step, it is necessary to be able to sample from that distribution. The set of particles is then, after M iterations estimated as the filter representation $\bar{bel}(x_t)$. The weight or the important factor is obtained through the probability of the measurement z_t given the state $x_t^{[m]}$.

Algorithm 1: Particle Filter X_{t-1}, u_t, z_t)

Result: Approximate the belief(x_t) by the set of particles $\bar{X}_t$ that gives the density around the target.

$\bar{X}_t$ = $X_t = \phi$

for *m=1:M* **do**

  sample $x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]})$ {`Prediction` }

  $w_t^{[m]} = p(z_t|x_t^{[m]})$ {`Weighting` }

  $\bar{X}_t$ = $\bar{X}_t$+ $\langle x_t^{[m]}, w_t^{[m]} \rangle$

end

for *m=1:M* **do**

  draw i $\propto w_t^{[m]}$ {`Re-sampling` }

  add x_t to X_t

end

return X_t

Then the powerful step of the particle filter algorithm will be executed, the re-sampling step. Before the re-sampling, the particles were distributed as filter representation $\bar{bel}(x_t)$. By re-sampling particles with probability proportional to $w_t^{[m]}$, the resulting particles are distributed to the product of the proposal probability and the importance weights. The weights of the particles will be increased in the re-sampling process. We add particles in the more likely areas, then re-sample, use the motion model and change the distribution. The weights of some particles drops to nearly zero while others would become very large. This means that the outcome of re-sampling, results in particle areas with higher probability and those with lower probability will get lost. When the re-sampling is done the set of weighted particles is the Bayes filter posterior $bel(x_t) = \eta p(z_t|x_t^{[m]})\bar{bel}(x_t)$ including the state and the weight.

The new belief will be added to the set of particles. Thus, one must be careful to avoid particle deprivation which likely occurs if there are not sufficient particles in the region with sufficient probability of being. It could be caused by

*This work was coworked by Martti Yap, student at System, Controls and Robtics, KTH.

[1]Pernilla Wikstrom, MSc. System, Controls and Robotics student, KTH, Stockholm.

an unlucky series of random sampling of a finite number of particles. It means that we might not have particles where we want them to be. It could occur for a small range of particle as much as for a large number. Another issue while re-sampling is that random processes might result in wrong approximation and as the algorithm is recursive it could affect the result and appear as large offsets. The issue is called sampling variance and can be avoided by introducing more particles.

However, the contribution in this paper is an implementation of a vision color based object tracker using a particle filter. A collection of data is used from Temple Color 128 website [4] where videos in RGB are provided with frame object boxes. The code is written in Matlab.

A. RELATED WORK

Particle filter used for object tracking have typically been applied in color based framework. *A Color Based Particle Filter* is a paper [3] which compares two methods, the color based particle filter algorithm and the mean shift algorithm which is a common approach in object tracking. The mean shift algorithm is a segmentation technique that finds features or colors, uses a random window, find its density center and eventually perform mean shifts of the density center until they have converged. However, the color based method must have prior knowledge of the initialization, which mean either manually choosing a target histogram which requires an operator interaction or an object recognition algorithm that finds interesting targets. The color based particle filter algorithm appeared to give faster computations for frames where the tracker had lost the object. Unlike their solution of the Bhattacharyya distance which is a measured distance between two distributions (the amount of overlap between two statistical samples or populations [7]) our approach is to measure the euclidean distance in color space RGB of from the previous frame to the current frame.

Another approach is to use the condensation algorithm which is a Bayesian recursive estimator that uses sequential Monte Carlo Importance Sampling. The purpose is to detect and track the contour of objects that moves in a noisy environment. However, paper [5] (*Visual object tracking in 3D with color based particle filter*) explains how their algorithm provides feedback of the closeness of particles to the 3D location of the colored object, which is an interesting approach. Similar to their Gaussian motion model, our solution presents a random motion model as the intention of object tracking could be any moving object in an arbitrary video.

B. Outline

Section II explains the implementation of the method. The particles filters weighting, motion model, re-sampling and accuracy will be explained. In section III the successful experimental results will be explained. Then, in section IV the summary and conclusion of the project will be presented.

II. METHOD

The implementation is a basic computer vision method and must include decisions of re-sampling method, number of particles, how to weight particles, additionally a motion model. The implementation in this paper is made with possibilities of changing some settings from case to case. Which mean it is possible to adjust the settings if the video has one camera angle with only one moving object, or if the camera moves as much as the object. Number of particles can be adjusted as well. The motion model and the weighting is based on the computer vision solution which compares pixels in RGB space from frame to frame of the video. Two methods of finding changes of frames has been studied. The first method (m_1) computes an average color RGB of the previous frame F_{t-1} and check for each pixel p_{ij} the euclidean distance of the current frame F_t to the average of the previous frame F_{t-1}. Then hypothetically, the solution assign each pixel of the current frame F_t with corresponding euclidean distance. The second method (m_2) computes the euclidean distance in RGB of pixel p_{ij} from the previous frame F_{t-1} to the current frame F_t. Method m_2 is a solution which finds the RGB change of each pixel of the current frame F_t and hypothetically tracks quick motions in pixels.

A. Weighting

The weights of the particles has been studied in two methods, one (w_2) where each particle x_t^m position in the frame F_t is weighted by its euclidean color distance from all pixels in the frame F_t. The second method (w_3) is similar but takes the motions/euclidean distances mentioned (m_1,m_2) into account. Each frame is multiplied by its change in color space RGB. Pixels with longer distances results in higher weights and pixels with shorter distances will be weighted lower. The intuition behind is, if a particle has a longer distance from the average background, it is more likely to have a position of the object.

B. Motion

When each particle is assigned with a weight the particle filter must predict the particles motion model. As the project has developed, there is no specified object to analyze and as the camera angle could be either static or moving, the motion model is considered best as Gaussian motion.
As the desired object normally is focused in the middle of a frame, the motion model must consider that particles that goes towards corner might be unlucky series of numbers. Therefore, to prevent particles with positions close to frame boundaries or even outside, a threshold for Gaussian positions is implemented. This means if particles goes towards the corners the variance of the Gaussian motion is increased by a large scalar in order to get a larger search radius of possible positions and hypothetically particles tracks the target object in focus again. Additionally an edge scaling function is implemented to prevent particle positions around

the corners. The method consist of a Gaussian scaling frame multiplied with the current video frame such that pixels around the corners slowly converge to same RGB values and pixels near the frame center keep their colors in RGB.

C. Re-sampling

The implementation includes both multinomial and systematic re-sampling. The Multinomial re-sampling algorithm 2 [2] is the most simple method. It draws particles independently using random numbers r_m, computes the cumulative distribution function *CDF* of the weights of the particle set for each iteration, resulting in M numbers/calculations respectively.

Algorithm 2: Multinomial Re-sampling

$X_t = \phi$
for *m = 1:M* **do**
  CDF(m) = $\sum_{i=1}^{m} w_t^i$
end
for *m=1:M* **do**
  $r_m = rand\{0 \leq r_m \leq 1\}$
  i = argmin$_j CDF(m) \geq r_m$
  $X_t = X_t \cup < x_t^i, \frac{1}{M} >$
end
return X_t

The systematic re-sampling algorithm 3 [2] cumputes the *CDF*, only one random number followed by drawing the particles. It is an alternative method which offers better speed and additionally better variance.

Algorithm 3: Systematic Re-sampling

$X_t = \phi$
for *m = 1:M* **do**
  CDF(m) = $\sum_{i=1}^{m} w_t^i$
end
$r_0 = rand\{0 \leq r_0 \leq 1\}$
for *m=1:M* **do**
  i = min j: CDF(j) $\geq r_0 + \frac{m-1}{M}$
  $X_t = X_t \cup < x_t^i, \frac{1}{M} >$
end
return X_t

D. Accuracy

All videos used in the project are downloaded from *Temple Coloe 128* [4]. The data is provided with all frames corresponding object boxes. To validate the particle filter, the algorithm measure the *RMSD- root mean squarred deviation* that assess how well the reconstruction of an image perform relative to the original image. A mean particle of the particle set $\hat{X}_t$ of a frame holds as $\hat{y}_t$ and will be compared with the box center y_t of the video object from the first time frame $t = 1$ until the last time frame T.

$$RMSD = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_y)^2}{T}} \tag{1}$$

Another accuracy validation is done by computing the percentage of the mean particle $\hat{y}_t$ being positioned inside the object box along the video duration.

III. RESULTS

The results are divided into three different data sets, ice skater, singer and jogger respectively. The accuracy and the *RMSD* of the particle filter will be presented.

A. Data set 1, ice skater

The chosen parameters of the particle filter applied on the ice skater was set to:

- Variance, $\sigma = 2$,
- Variance scalar, $C_{sigma} = 80$,
- Blurring filter variance, $\sigma_{blur} = 1$,
- Number of particles, $M = 3000$,
- Frame changes mode, m_1,
- Weight mode, w_3,
- Systematic Re-sampling.

By using the tuned parameters the filter behaves well and tracks the ice skater. In the beginning of the video the particles are uncertain of the true target as seen in Figure 1. The green box shows the object box which was given in the data set. The yellow marks represents particles and the red mark is the mean of all particles which in the current frame is inside the object box. The particles find motions in the pixels but unfortunately as clear to the reader, not the desired target.

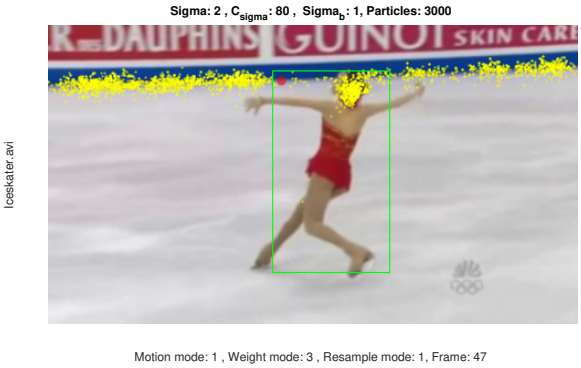

Fig. 1. Data set 1, ice skater, time frame 47.

After running a few frames the ice skater is on her way from the corner in time frame 294 as seen in Figure 2. The particle distribution has converged and become certain of the true target, the ice skater.

The ice skater moves from the corner and the particles successfully moves with the ice skater to time frame 400 as seen in Figure 3 when the ice skater spins a pirouette.

B. Data set 2, singer

The scenery of data set 2 with the singer is more complex. As the camera moves and the scene of the concert in some frames get lots of shifted light, the frames shifts background from black to white. To successfully track the object with a larger uncertainty, the variance and the variance scalar must be increased. The parameters and settings of the second data set with a singer on a scene was set to

- Variance, $\sigma = 5$,

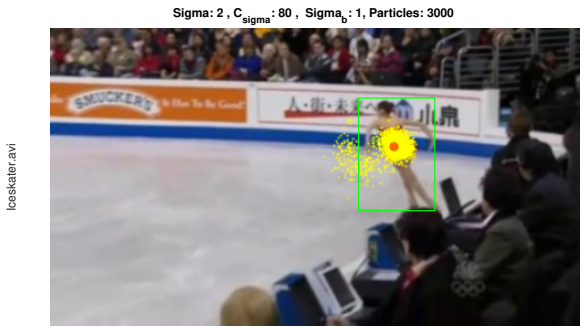

Fig. 2. Data set 1, ice skater, time frame 294.

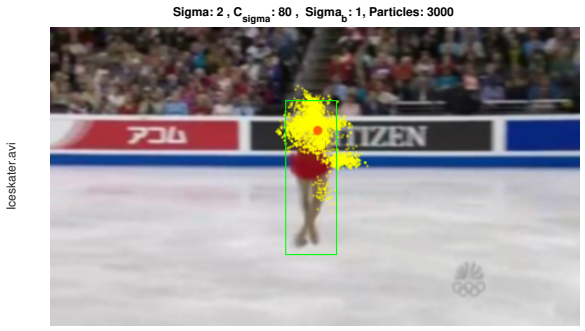

Fig. 3. Data set 1, ice skater, time frame 400.

- Variance scalar, $C_{sigma} = 100$,
- Blurring filter variance, $\sigma_{blur} = 1$,
- Number of particles, $M = 3000$,
- Frame changes mode, m_1,
- Weight mode, w_3,
- Systematic Re-sampling.

As the object now in Figure 4 has a white dress and the background is considered to be white, the longest distance of colors in each pixels appears to be next to the target object. The result mean that frame 100 has failed to find the true target.

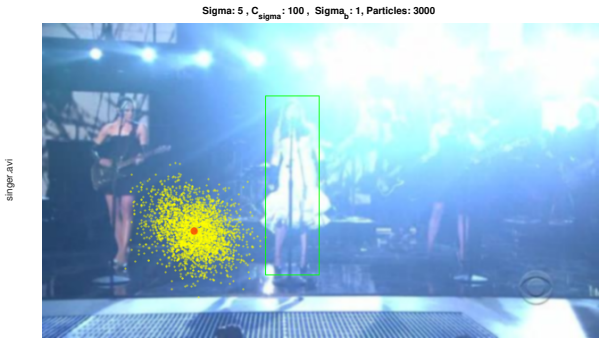

Fig. 4. Data set 2, singer, time frame 100.

By using the big variance scalar function which is used in the motion model, particles randomly get new states and in a reasonable time frame, the tracker finds the object back again as seen in Figure 5.

Fig. 5. Data set 2, singer, time frame 352.

C. Data set 3, jogger

The tuning of the jogger gave fortunate results by using same parameters and settings as the ice skater. Figure 6 shows the particle distribution of the filters behavior applied on data set 3, jogger of the time frame 308. The reason why particles only find the second jogger, is that the second jogger wears a white t-shirt which is less like the average color in the frame by that time. Therefore particles in that position increases their weights.

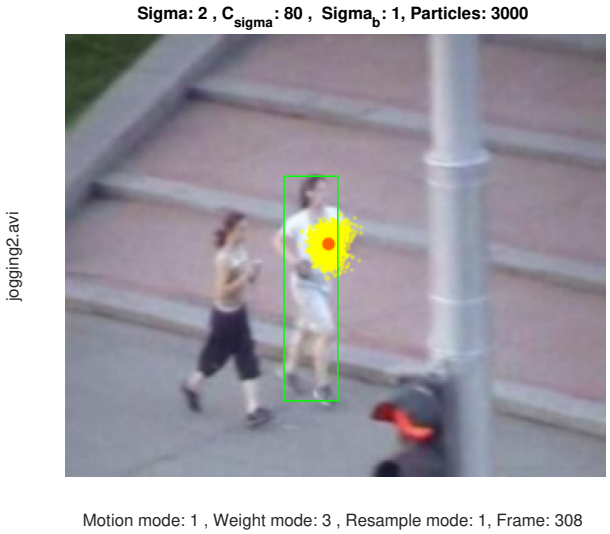

Fig. 6. Data set 3, jogger, time frame 308.

D. Accuracy and RMSD comparison

To find the most suitable settings of the particle filter applied on the data sets, a comparison of weight model and re-sampling method were made.

1) Data set 1, ice skater: By observing results of the accuracy as in Figure 7 the best setting according to their performance would be multinomial re-sampling with weighting method w_3, whereas weighing method w_2 gave the least accuracy. One can see the the systematic re-sampling is less sensitive of changing the weighting method.

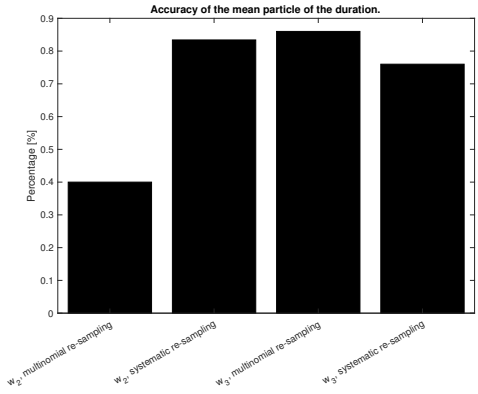

Fig. 7. Accuracy of data set 1.

On the other hand it is essential to look at the horizontal and vertical precision of the particle filter as seen in Figure 8. The result shows that weighing method w_2 compared to w_3 when applying the systematic re-sampling method results in smaller *RMSD* in the most sensitive regions.

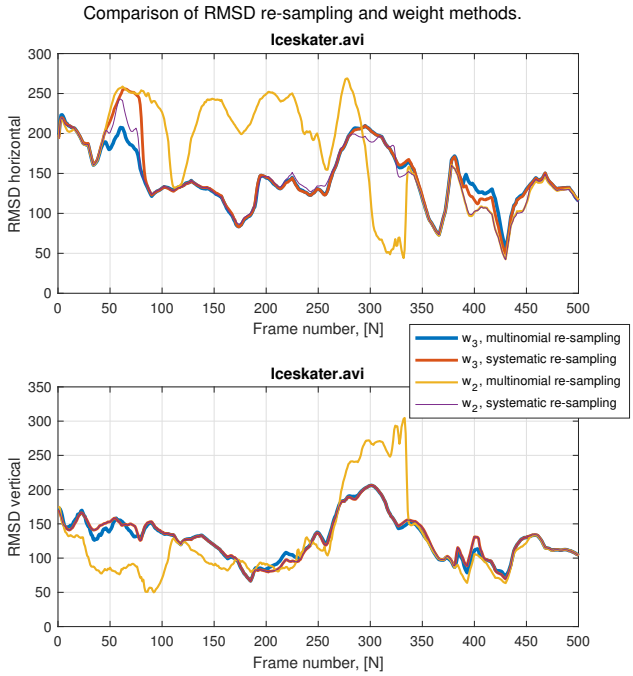

Fig. 8. Root mean squared deviation of data set 1, horizontally and vertically respectively.

2) Data set 2, the singer: The accuracy results of methods applied on the singer is seen in Figure 9 where the weighting method clearly is more essential. That is because of the already mentioned complexity of frames with lighted scene resulting in shifting background colors. By using the average background weighting method w_2 it is possible to achieve an accuracy close to 100% of the video duration.

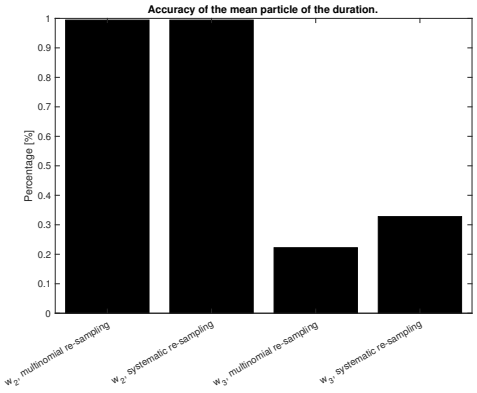

Fig. 9. Accuracy of data set 2.

Thus, the *RMSD* seen in Figure 10 indicates lower precision, both horizontally and vertically by using weighting method w_2. This implies on a larger cloud of particles when the cloud is accurate.

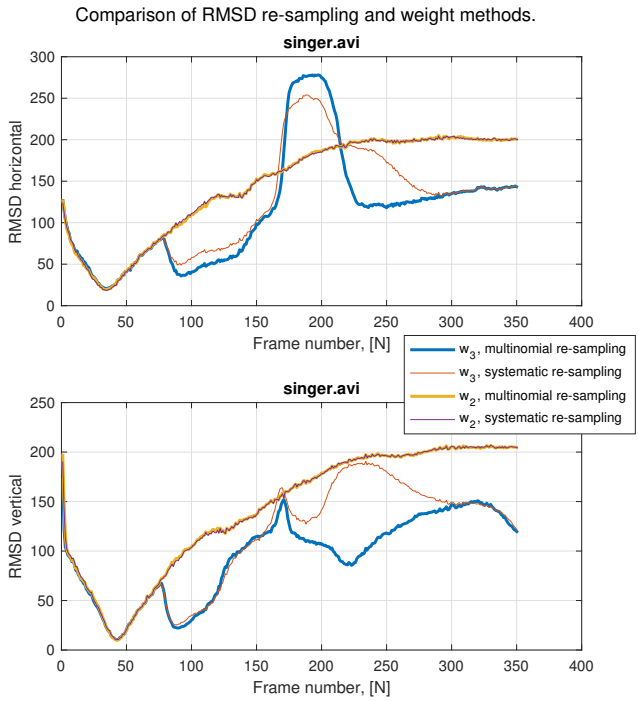

Fig. 10. Root mean squared deviation of data set 2, horizontally and vertically respectively.

3) Data set 3, jogger: The accuracy of the settings applied on Data set 3, the jogger seen in Figure 11 are similarly high for all settings, above 90 %. The object is easy to find as it is white. In color space the white pixels has a large distance

to the average color as in weighting model w_2. In pixel velocity/change the most shifting pixels are located around the true target object.

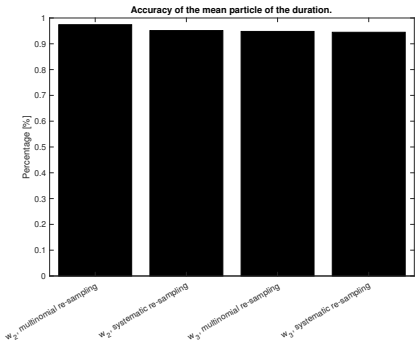

Fig. 11. Accuracy of data set 3.

As the object tracking is easy by its beneficial data set 3, the jogger the result of the *RMSD* behaves as expected seen in Figure 12, all methods results in similar precision.

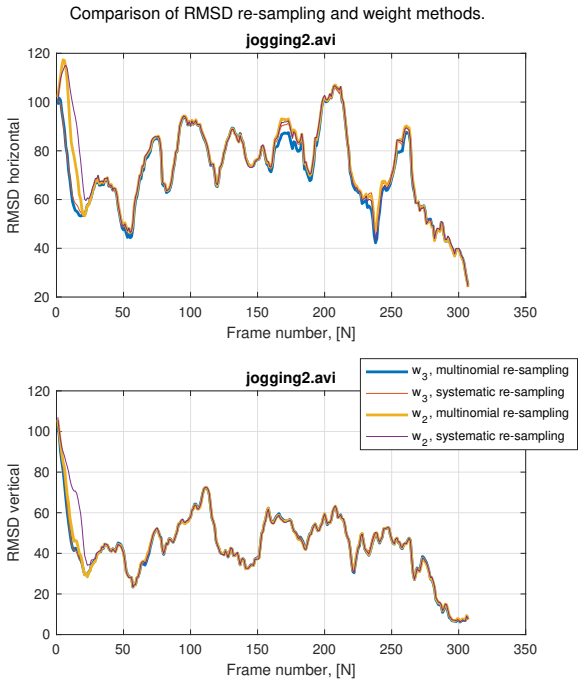

Fig. 12. Root mean squared deviation of data set 3, horizontally and vertically respectively.

IV. SUMMARY AND CONCLUSION

To summarize, a particle filter algorithm was successfully implemented. The results were as expected we found an image treatment that allowed for efficient object recognition without being too computationally heavy or complex in its algorithm. The parametrisering implementation of the particle filter were studied to give a deeper understanding. There are many improvements that can be done. The basic computer vision technique is not suitable to an arbitrary visual color motion tracker. The algorithm has difficulties finding more than one motion. One lesson learned was that the edge scaling function was not useful due to the accuracy of details in this implementation. Neither was the Gaussian blurring that was implemented to speed up the computations.

However, the result shows that the implementation could be applied on data sets where the object is known of being in contrast with the background. Preferably only one camera angle such that the camera is static and only one target motion. As expected, the more simple multinomial re-sampling method generally resulted in less accuracy and precision of the object tracker. The systematic re-sampling method proved more fortunate results and less dependent of the weighting model. The Gaussian motion model with the idea of extending the searching radius of particles that closes the frame corners made it possible to prevent particle deprivation.

A. Future work

Future work would be to test the particle filter with a more robust image processing technique, by using a histogram filter to identify the object. The algorithm would be less affected of a shifting background. Perhaps only using one re-sampling method, preferably the systematic. Another interesting angle would be to try to optimize our implementation using the Kullback-Leibler divergence, *KLD* [6] which measures the spread of the distribution and subsequentially adjust number of particles. More spread of particles contributes to an increased number of particles. This could possibly strengthen the certainty of the true target when the spread is smaller.

REFERENCES

[1] Dieter Fox Sebastian Thrun Wolfram Burgard. *Intelligent Robotics and Autonomous Agents*. Probabilistic Robotics-Mit, 2005. ISBN: 9780262201629.

[2] John Folkesson. *Applied Estimation, Course material of lab 2 - Particle Filter.*

[3] Luc Van Gool Katja Nummiaro Esther Meier. *A Color based Particle Filter.*

[4] E. Blasch P. Liang and H. Ling. *Temple color 128*. URL: http://www.dabi.temple.edu/~hbling/data/TColor-128/TColor-128.html?fbclid=IwAR0Snz-nfZCvj404TlREgnIV7Bzsdrgjkrtp0kKSgRZ8Qdo_QM1TLMryur4. (accessed: 21.12.2019).

[5] Vincente Matellan Pablo Berrera Jose M. Canas. *Visual object tracking in 3D with color based particle filter.*

[6] ScienceDirect. *Kullback-Leibler Divergence*. URL: https://www.sciencedirect.com/topics/engineering/kullback-leibler-divergence. (accessed: 04.01.2020).

[7] Wikipedia. *Bhattacharyya distance*. URL: https://en.wikipedia.org/wiki/Bhattacharyya_distance. (accessed: 22.12.2019).