Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 2469 – 2485
© TÜBİTAK
doi:10.3906/elk-2011-45

Real-time motion tracking enhancement via data-fusion based particle filter

Tuğrul TAŞCI[*], **Numan ÇELEBİ**
Information Systems Engineering, Faculty of Computer and Information Sciences, Sakarya University,
Sakarya, Turkey

Received: 22.12.2020 • **Accepted/Published Online:** 03.05.2021 • **Final Version:** 23.09.2021

Abstract: Motion tracking is a well-defined yet application-specific problem of computer vision field, mostly entailing real-time constraints. Methods addressing such problems are expected also to ensure achievements such as high accuracy and robustness. A probabilistic estimation-based approach is proposed in this paper, in order to enhance the real-time motion tracking process of an RGB-Depth device, in terms of accuracy. A novel method is presented for tracking hand-palm of a moving human subject to this end, under a sequence of assumptions such as indoor environment, single object, smooth movement and stable illumination. Tracking accuracy is improved within a particle filter framework by fusing device output with the newly extracted information from RGB and depth images. Experimental results are shared revealing the advantages of the proposed method over the built-in device algorithms. The results demonstrate that the proposed method produces smaller RMSE values both for single implementations and multiexecution trials without violating real-time constraints.

Key words: Motion tracking, data-fusion, particle filter

1. Introduction

Today, there are numerous systems and services available, enhancing the quality of daily life including governments' services, scientific research and military development. Rapidly changing and evolving technology engenders entirely new expectations in minds, desire of private sector initiatives to become prominent in increasingly globalizing competition and the country strategies aiming globally powerfulness force those systems to have novel and more efficient functions. In this context, new problems arise continuously over time that need to be solved in real-time or online with reasonable accuracy depending on the sector. One of the most significant requirements for the subject systems is the subsistence of some mechanism allowing the solution architects to know the current status of the system in terms of planning the appropriate prospective actions. However, this is not possible in most cases, due to the process and measurement noises causing not to obtain status information directly, accurately and completely. From this point of view, there appears a certain need for filtering the noise through estimation of some valuable system parameters by means of robust methods searching for convenient denouements in high dimensional solution spaces.

Motion tracking is based on the principle of extracting significant information from data obtained mostly in real-time, particularly through optical sensors. The use of optical sensors in motion tracking is a well-established area of research that has been subject to numerous applications over the last two decades including video surveillance, content-based image retrieval, articulated body motion analysis and classification, sign

[*]Correspondence: ttasci@sakarya.edu.tr

language and mimic recognition and virtual/augmented reality. Comprehensive information on computer vision-based motion tracking since the year 1980 can be found in the survey studies [1–3]. Those studies have been usually investigated in two main categories in terms of data acquisition, namely active and passive systems.

While both active and passive systems are for certain type of applications, in today's world, the number and variety of applications based on passive systems is much greater than the ones based on active systems. Passive motion tracking systems are categorized as appearance and model based. Though these two approaches differ in terms of problem modelling, their aim is identical: achieving an acceptable solution within a very large solution space. However, regardless of the approach and method chosen it is generally a huge challenge to get such a solution without any restrictions related to the problem, scene, subject or environment. Single moving subject, stationary camera, occlusion-free scene, soft motion, stable illumination, known first position and colored clothes are the most utilized assumptions.

Human motion tracking with real-time constraints for many applications and the large solution space have led researchers in recent years to the methods based on probabilistic inference. Probabilistic inference has an intrinsic filtering function allowing to shrink solution space by using the information obtained in the previous time steps, taking advantage of the transition relation between the consecutive system states and measurements for the time-varying dynamic systems which can readily be considered as covering motion tracking problem. Bayesian methods starting from basic Kalman filter to particle filter variants are deemed as principal group of methods based on probabilistic inference. In the early studies, Kalman filtering method has been extensively used for motion tracking purposes. Due to the restrictions on linearity with Gaussian noise, Kalman filter-based methods are substituted by some variants over time supporting nonlinear system modelling such as extended Kalman and unscented Kalman filters. In recent years, probabilistic sampling-based Bayesian methods, mainly particle filter, has gained respectable amount of attention among researchers due to the fact that it has the capability of overcoming high dimensionality through importance sampling. Likewise, particle filter-based methods allow the measurements to be obtained from multiple sensors and combining them within a probabilistic framework causing innumerably but basic calculations to be performed which is indeed not a problem with current computing resources.

In this study, we demonstrated that Microsoft's Kinect for Windows, specifically designed for real-time gaming experiences, which monitors the human skeleton system in real-time with a model, produces wrong results especially in case of occlusions. We proposed a tracking method based on particle filter to enhnace tracking performance. We applied our method for real-time tracking of the center of hand-palm moving on a nonuniform background. One of the contributions of this work is to show that it is likely to improve tracking results using a sensor-fusion based particle filter method instead of Kinect's built-in functions. This study includes all necessary steps for tracking human motion with particle filter and can be used as a reference source in diverse fields which is considered as another contribution to the literature.

2. Background work

Online visual tracking has many real-world applications such as human-computer interaction, video surveillance, automatic robotics etc. There are many researches to solve such kind of problems including applications related to diverse fields. Those are mainly categorized into four approaches in the field literature: stochastic, deterministic, productive, and discriminative [4]. Although each approach has distinct advantages and disadvantages over the others, there are common problematic issues that ultimately lead to the curse of dimensionality for almost all methods such as complex background, illumination changes, shape deformation, occlusion [5], abrupt and rapid movements as well as real-time constraints. In this context, achieving reasonable solution for such a visual tracking problem, mostly requires employing and combining various types of cues and also shrinking solution space in a sort of way. Recently, particle filtering, one of the stochastic approach methods, has been increasingly considered among the most favorable methods to address the online object tracking problem. Particle filtering methods have inherent capability of eliminating the curse of dimensionality based-on probabilistic importance sampling. Besides, it is generally more uncomplicated to fuse information obtained from various sensors or data sources in a particle filter framework in comparison with the confocal approaches. The most significant requisite for a successful particle filter-based method is having a proper observation model involving a convenient information extraction mechanism from one or more data sources.

In visual object tracking, information extraction process is usually applied on camera images. The easy availability of different types of cameras today, gives the opportunity to the researchers to employ various cameras in object tracking applications. Microsoft's Kinect is a certain example of those cameras that has been increasingly used in such applications. A number of recent works regarding online visual object tracking based on particle filtering and/or use of Kinect RGB-D camera are examined in this section. In object tracking, the challenge is building an adaptive model for the changes in object appearance during tracking process. Hu et al. [6] proposed a long-term object tracking method addressing occlusion, scale variation and out-of-view cases. Tran et al. [7] proposed a scalable, reliable and on-line system for human fall detection basedon multimodal features obtained from Kinect sensor. Li et al. [8] proposed a multiview method for object tracking which fuses various features and selects more discriminative attributes. Zhou et al. [9] proposed a particle filter-based tracking method in the presence of illumination changes and occlusions integrating measurements from color and spatio-temporal motion energy components. Bueno et al. [10] constructed a metrological system for comparing the KinectFusion system against standalone Kinect sensor and they showed that the KinectFusion gave better results than Microsoft Kinect. Caruso et al. [11] used Kinect V2 as a vision device for detecting position, shape and dimensions of the object in manufacturing plant in textile sector. Boutellaa et al. [12] used Microsoft Kinect for face detection to investigate the gender and ethnicity according to images taken.

Hbali et al. [13] purposed an approach for handling the spatial-temporal aspects of human activity sequences for monitoring elderly people in health care sector. Anton et al. [14] proposed a Kinect-based telerehabilitation framework supporting real-time transmission of video, audio and depth data for remote physical therapy sessions. Cabrera et al. [15] developed a software (KiSens Numeros) to show that Kinect can be used for monitoring disabled patients without the capability of controlling their movements completely. Yavsan and Uçar [16] presented a framework using an humanoid robot equipped with an Xbox 360 Kinect sensor to catch upper-body human motions. Napoli et al. [17] presented a study for evaluating Kinect's performances and limitations in tracking human motion in the context of biomechanics. Henseler et al. [18] constructed a Kinect Recording system for both three and four-dimensional breast evaluation and demonstrated that their system can be used for women along with plastic and breast surgeons. Hayat et al. [19] proposed a face recognition method with low computational cost based on image classification using low quality depth and texture information captured from RGB-D sensor. Riahi et al. [20] proposed a system involving target appearance model and data association strategy to select the best candidate for each target. Wang et al. [21] used a part-based structure in their study for obtaining precise position by combining color and depth statistics. Alongside the visual object tracking methods and applications relying on Kinect or similar RGB-D sensors in the field literature, a number of recent researches have been also presented covering particle filter usage with some forms of improvement addressing accurate, robust and online tracking. In the methods reviewed, fusion of multiple cues has seen as the most adopted way of achieving those addressed capabilities. Jiang et al. [5] proposed an online object tracking method based on discriminative model exploiting fused multiple features such as intensity, histogram of gradient and color naming features. Guipeng et al. [22] in their study, employed a correlation particle filter within an adaptive feature fusion framework to improve tracking performance on various occlusion-enabled cluttered scenes including faces and objects. Dong et al.[23] proposed a method using particle filter and mean-shift tracking algorithm that combines robust global color information with local texture information for achieving stable target tracking. In another study presented by Lei et al.[24], based-on CS-LBP features, the authors developed an improved particle filter method providing accurate face tracking while ensuring high real-time performance. Kumar et al.[25] proposed an adaptive multicue particle filter based real-time visual tracking framework using color histogram, LBP and pyramid of histogram of gradient features. Walia et al.[26], on the other hand, presented a multicue particle filter method based on fusion of color and texture information and a novel resampling method based upon crow search optimization to overcome low performing particles detected as the outlier. An extensive review of the recent trends in multicue based visual tracking can be seen in the work of Kumar et al. [4].

The literature on object tracking covers a wide range of different methods and applications. The use of particle filtering and/or Kinect, constitutes a relatively small portion of the literature in this field, the majority of which are evaluated in this paper. It is clear from those works that a fair amount of object tracking applications exist that are based on either particle filtering or Kinect usage. However, the practical use of the Kinect device is not found in any investigated work in conjunction with the structure that allows particle filtering to effectively reduce the curse of dimensionality and to simply fuse data from different sensors in order to achieve better results. In this study, a sensor-fusion based particle filtering framework is designed with a special observation model for combining object tracking results obtained from Kinect device with color and depth information, resulting in better results especially in the case of occlusions.

3. Proposed system

The particular problem addressed in this study is to enhance real-time motion tracking process performed by Kinect in case of self-occlusion. A unique data-fusion based particle filtering approach is adopted in which Kinect acquired images are utilized through a bilateral observation model in order to detect more precise coordinates by extracting color and depth-based cues. Obtained coordinates via color (C) and depth (D) based observation models with Kinect's built-in output (K) are then separately included into particle filter framework as distinct sensor inputs. Finally, the expected locations of coordinates are estimated through particle filter algorithm where ground-truth data is regarded as the average of coordinates detected manually by different human operators. In a similar way followed by the applications to vast majority of real-world problems, a sequence of assumptions, such as fixed-camera position, tight clothing and smooth movement regarding environment, appearance and motion are also accepted in solution process of the problem covered in this study.

3.1. Information flow

The information flow of the proposed system for a single time step is depicted in Figure 1. These operations are repeated for each time step until a proper solution is met.

At the beginning, a smaller window containing Kinect's output position for object's region of interest is identified in order to form a basis for performing further operations within a quite smaller solution space rather than whole stage image window Kinect-acquired RGB images are converted to 8-bit grayscale and Depth

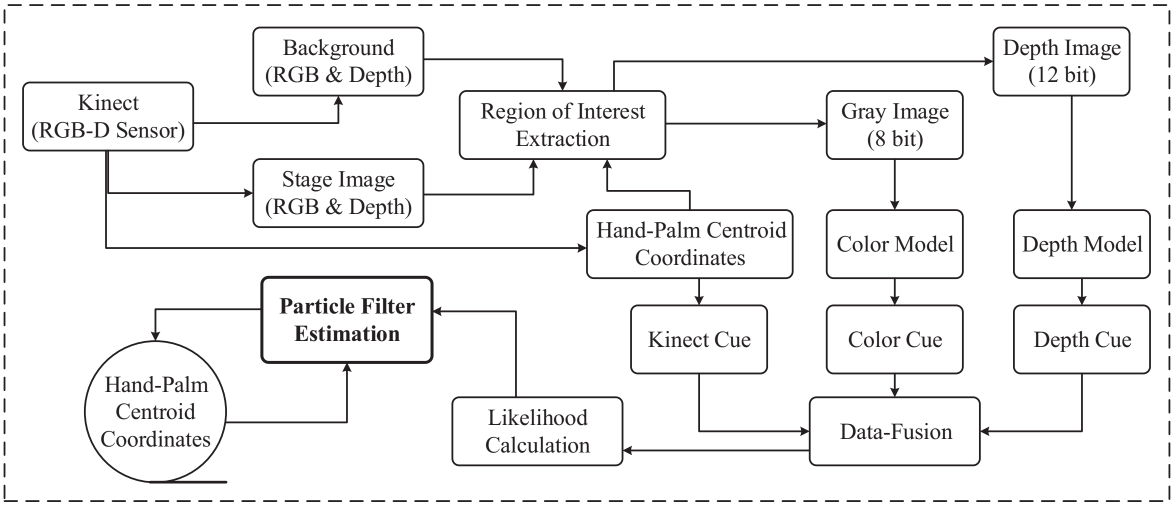

Figure 1. Information flow of the proposed system.

images are quantized into 8-bit images in order to decrease computational load. Active window images are then subtracted from the corresponding background images to obtain moving only regions as indicated in the network of Shehzad et al. [27]. In the next step, a separate cue which means an extraction of requested object coordinates is achieved by using the RGB and depth moving only active window images while Kinect's output is directly taken as a distinct cue in this phase. The existing cues are associated, and a joint likelihood is calculated using particle filter prediction through a dynamic model. This joint likelihood is involved in the particle filtering framework in order to make an estimation. The particle filter estimation and Kinect's output are finally compared with the operators' detections to evaluate performances.

3.2. Particle filter

In object tracking problem, objects may become different objects or separate into smaller regions due to occlusion, image noise, or change in light and shadow. Accordingly, analyzing current stage using available past information and extracting confident information for the purpose of accurate estimation is one of the principal tasks in object tracking problem. Estimation allows shrinking solution space, thereby helping to reduce computational processing load which is one of the main goals for tracking problems with real-time constraint.

Efficiency of particle filter primarily depends on dynamic motion and observation models convenient to the specific tracking problem tackled. Dynamic motion model is actually an assumption for the motion of subject in which speed or acceleration is assumed to be constant. Generally, a few types of motion models are used in object tracking applications such as constant velocity, constant acceleration, random walk, Brownian, angular and bearing only motion models. Observation model is a problem specific algorithm providing a kind of mechanism for extracting useful information through noisy measurements. The operation steps of color and depth-based observation models are explained with details in subsubsections 3.5.1 and 3.5.2, respectively.

Particle filter is based upon the idea of sampling from a proposal distribution rather than the target distribution which may not be convenient to sample from [28]. A particle is essentially a candidate solution to the problem addressed containing the parameters to be estimated. Particle filter is one of the powerful methods providing a suboptimal solution by approximating the posterior pdf by a sequence of weighted samples given by (1).

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{\mathbb{N}} \tilde{w}_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}), \tag{1}$$

where $p(x_{0:k}|z_{1:k})$ is the joint conditional probability of all system states given all available measurements, $\delta(.)$ is the Dirac delta function, $x_{0:k}^{(i)}$ is the i^{th} particle and $\tilde{w}_k^{(i)}$ is the normalized weight of that particle and calculated by (2).

$$\tilde{w}_k^{(i)} = \frac{w(x_{0:k}^{(i)})}{\sum_{j=1}^{\mathbb{N}} w(x_{0:k}^{(j)})} \tag{2}$$

A proportional relation between each consecutive time step pair is obtained by (3) which is specified by the likelihood of measurements, $p(z_k|x_k)$, calculated using information extracted from sensor data. A detailed explanation on the derivation of particle filter can be found in [29].

$$\tilde{w}(x_k^{(i)}) \propto \tilde{w}(x_{k-1}^{(i)}) \times p(z_k|x_k) \tag{3}$$

3.3. Particle filter algorithm

The solution to the problem observed in this study is provided by five main phases within particle filter framework. The operation steps of the algorithm are given in Figure 2 in which the updated parts are indicated by grey color.

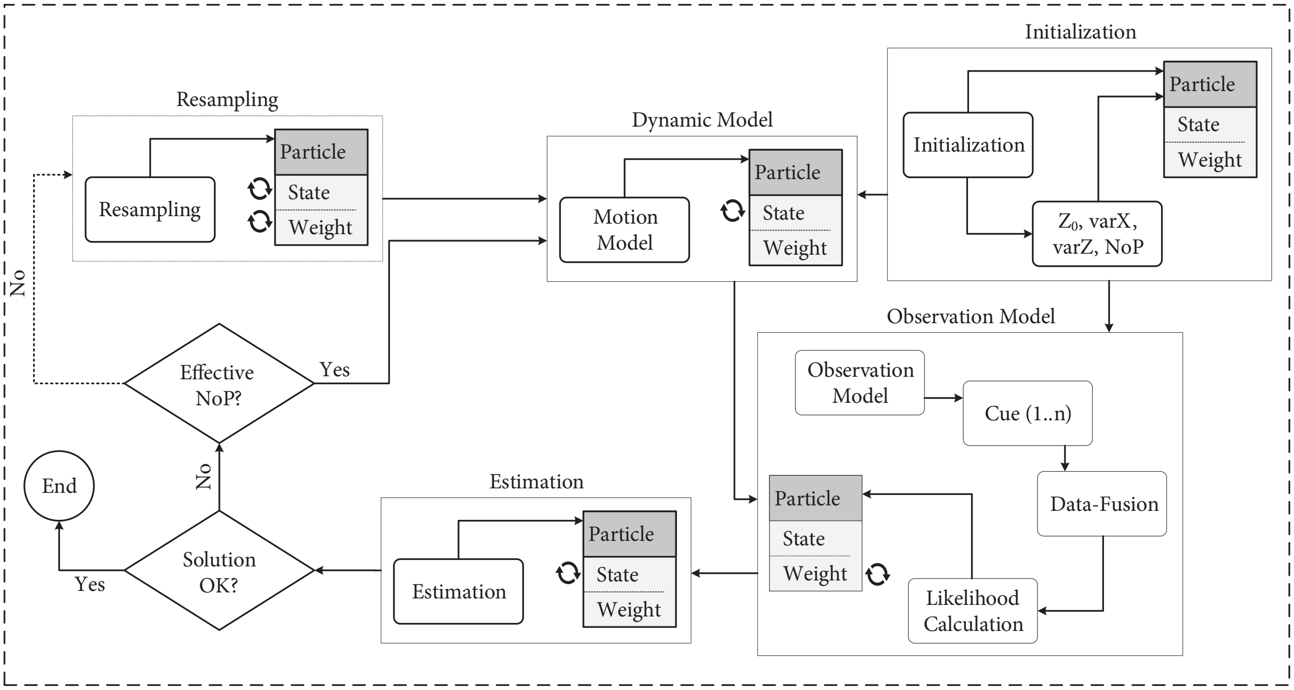

Figure 2. Particle filter algorithm.

The first phase is initialization. In the initialization phase, values of some variables and parameters including the number of particles, initial positions of particles and process and observation noise are identified based on initial beliefs and experience as well as similar implementations. In the end of this phase, each particle is weighted with $1/N$ where N is the number of total particles. The dynamic model phase is the second phase in which the particles are moved within the solution space according to the predefined motion model. In this study, constant velocity motion model is employed. At the end of this phase, the weights of the particles remain same while the positions of particles referring to a candidate solution are updated. The third phase is the observation model phase. In the observation model phase, the new weight of each particle is calculated using the extracted information based on the sensor data. In this study, a single physical sensor based three data sources are taken into consideration. The first one is the Kinect's output for subject's hand-palm and the second and third are the extracted coordinates via color and depth-based observation models, respectively. Three cues are obtained from these three data sources in this phase.

The distance between cue-based computation and each particle is calculated for each data source and particle. Thus, a distance matrix is formed for each data source. The distance matrices are then converted to likelihood matrices using a likelihood function. The likelihood function produces a large value for a closer particle and a small value for an unlikely particle. A joint likelihood matrix is calculated by element-wise production of three likelihood matrices. These likelihood values in this matrix are converted to weights by normalizing into interval (0,1). At the end of observation model phase, the positions of the particles remain same while the weights of particles are updated.

The forth phase is the estimation phase. In the estimation phase, the best-weighted particle is chosen using some techniques such as particle with largest weight, average weight of the particles or robust mean of a certain percent of the largest weighted of particles. In this study, the robust mean of 10% of the largest-weighted particles are used as the particle filter estimation. At the end of estimation phase, both the position and weight values of chosen particle can be different from the remaining particles of that time step. The estimated value is checked whether it is a convenient solution or not. If it is a suitable solution than the algorithm is ended. If not, an optional phase is started.

The optional and last phase is the resampling phase. In the resampling phase, firstly the number of effective particles (having meaningful weights) are calculated. If the number of effective particles is more than 2/3 of the total particles than resampling is bypassed. If not, the particles are resampled by using one of the resampling schemes such as multi-nominal, stratified, residual or systematic. In this study, residual resampling is used. At the end of resampling phase, the number of large-weighted particles are increased, while the particles with small weights are decreased by some rate and the weights of the resampled particles are made equal once again. Thus, the algorithm becomes ready as in the initialization phase while preserving the previous information. Through such an operation, a better approximation to the optimal solution is obtained for each new time step depending on the usage of proper dynamic and observation models.

3.4. Dynamic motion model

There exist several well-known dynamic models including random walk, Brownian, angular, constant velocity and constant acceleration. It is likely to hit any of these models in the particle filter-based tracking literature. Specification of the dynamic model in accordance with the problem tackled in particle filtering is a considerable issue affecting the performance of the solution. In the proposed system, constant velocity motion model is employed which is given by (4), in order to detect 2D centroid coordinates of subject's palm at current time step by adding constant values and white Gaussian noise to its horizontal and vertical positions at previous time step. The initial positions of the particles are specified prior to the first time step and this dynamic model is executed for the prediction phase of each subsequent time step before obtaining any measurements.

$$x^{(i)} = M \times x^{(i-1)} + \gamma \otimes \sigma_x, \tag{4}$$

where $x^{(i)}$ and $x^{(i-1)}$ are the particles at current and previous time steps respectively, I_2 is 2×2 unit matrix, 0_2 is 2×2 zero matrix and $M = \left(\begin{smallmatrix} I_2 & I_2 \\ 0_2 & I_2 \end{smallmatrix}\right)$ is the coefficient matrix of constant velocity motion model, γ is a trial-error coefficient, σ_x is a random matrix produced by standard normal distribution representing a random location for any particle and $\otimes$ is the element-wise multiplication operator.

3.5. Observation model

Observation model considered as the main factor affecting particle filter performance is based on extracting useful information from sensor data. It is mostly impossible to get accurate results in case of using an improper observation model. A single physical Kinect data source with two sensors is used in the proposed observation model. The first sensor of Kinect is for acquiring RGB images and the second one is for depth images. The output of Kinect is real-time 3D skeletal data, obtainable from within specific software programs through the implementation of interfaces provided. In the proposed model, three types of inputs are employed. First one is the output of Kinect, the remaining two are RGB color and depth images, respectively, calculated using the RGB and depth images obtained. The output of Kinect is regarded directly as an observation cue without further processing. On the other hand, a separate observation cue is obtained for each of the remaining inputs, by applying a particular sequence of image manipulation techniques that are clarified in the following subsections.

3.5.1. Color-based observation model

The entire process in the color-based observation model is depicted in Figure 3. A difference image (I_C2) for region of interest is obtained at first, by background subtraction.

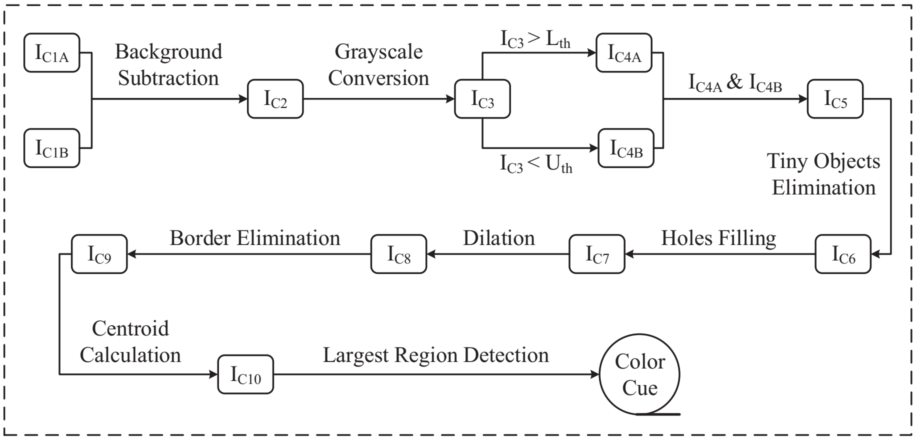

Figure 3. Color-based observation model.

Then, this difference image is converted to grayscale for reducing computational load. The obtained grayscale difference image (I_C3) is thresholded for both lower and upper intensity values in the next step.

While, the background pixels are cleaned by lower-level thresholding, the pixels around the subject's outline are eliminated by upper-level thresholding. Two binary images ($Ic_4A\&Ic_4B$) are obtained at the end of this phase. Rest of the image manipulation operations are performed on those binary images which provides a significant computational gain that is essential for real-time object tracking applications. A logical AND operation is applied in the next step on the existing binary, resulting a coarse prediction of the region (I_C5), which contains requested centroid point of subject's hand-palm. This information becomes available, due to the fact that the pixel intensities inside and outside of the obtained region are distinguishable by virtue of subject's clothing and of stable illumination. After this phase, several morphological operations are applied on the ready image (I_C5) successively in order to obtain a fine prediction of requested region. Firstly, the unwanted small regions are eliminated. Then, the regions which supposed to be hand with higher probability are highlighted (I_C6) with filling the tiny holes (I_C7). Filling operation is mostly improper for merging possible unconnected parts of a single region. Therefore, a dilation operation is followed (I_C8) in order to combine the parts fitting this case. Lastly, the unwanted pixels arising from the successively applied morphological operations around the borders of the region of interest are eliminated or at least minimized. At the end of these operations, a convenient region of interest is obtained (I_C9) that is ready for detecting the requested centroid of subject's hand-palm. In this phase, a simple algorithm is executed for detecting the requested point based on the assumption in which the largest region is accepted as the hand region. The algorithm has three steps: (i) Find all regions in the window, (ii) Calculate areas of each region. (iii) Get the centroid point of the largest region. The coordinates of obtained centroid point are used as color cue.

3.5.2. Depth-based observation model

In the depth-based observation, 12-bit reference background and subject's depth images are used as sources for each time step as shown in Figure 4.

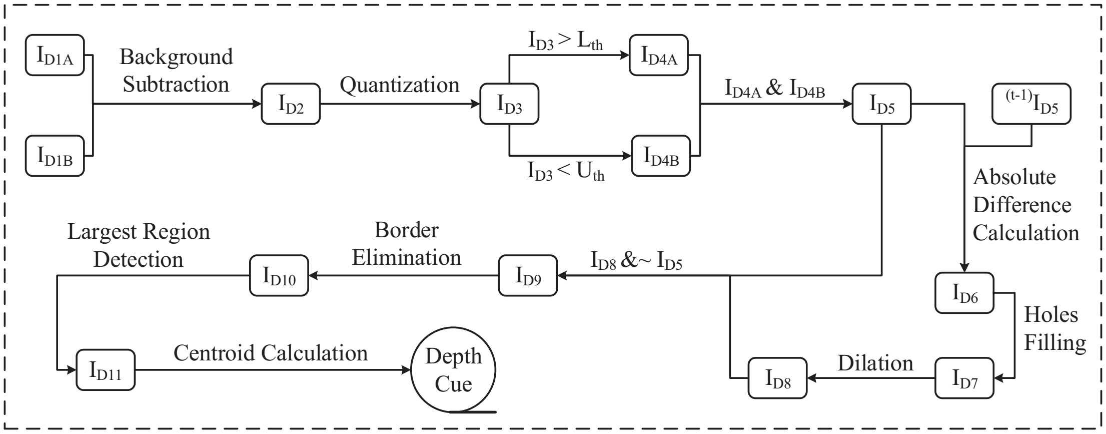

Figure 4. Depth-based observation model.

Similar to the color-based model, a difference image (I_d2) is obtained via background subtraction in the beginning. Then, this 12-bit difference image is quantized into 8-bit grayscale. In the next phase, the grayscale difference image (I_d3) is thresholded with predefined lower and upper density values as in the color-based model. Two binary images ($Id_4A\&Id_4B$) are obtained at the end of this phase. Next, a logical AND operation is applied on the existing binary images which produces a coarse prediction of the desired region (I_d5). In this phase, the previous time step image obtained by the same operations is used as reference for a subtraction operation. A new difference image (I_d6) is obtained by this subtraction. After this phase, morphological filling (I_d7) and dilation (I_d8) operations are applied successively. A logical AND-NOT operation is applied afterwards in order to get the holes as regions in image (I_d9) window containing the requested subject's hand-palm. The last two steps in the depth-based observation model, border cleaning operation (I_d10) followed by the largest region detection, are the same with the color-based model producing the requested centroid point of subject's hand-palm.

3.6. Data-fusion

The main difference of the method implemented in this study from the classical particle filtering applications for enhancing motion tracking process is the additional data association step. Actually, data association is potentially applicable for providing solution of various real-world problems. It has been theoretically proven and practically applied in many recent studies that more precise and accurate results can be achieved by fusing information extracted from multiple sensor data in accordance with the given problem within the Bayesian framework. In this context, in order to decrease the real-time error in tracking the centroid point coordinates of hand-palm of moving human subject via Kinect's built-in algorithms, two additional information extraction processes are executed in this stage on the Kinect-acquired RGB and depth images. The output of Kinect and the computations obtained using those new processes are utilized in this study for calculating joint likelihood and substantially producing particle filter estimation. It is depicted that the results are better in comparison to the results of solo applied Kinect's built-in algorithms. The implementation steps of the data association process are given below:

Step 0. Define Kinect output as K, color-based cue as C, depth cue as D, operator detection as O

Step 1. The Euclidean distance between O and K, C and D are calculated by (5)

$$L_{\varphi}^{(t)} = (p_{x_i}^{(t)} - \varphi_x^{(t)})^2 + (p_{y_i}^{(t)} - \varphi_y^{(t)})^2, i = 1..N \tag{5}$$

where N is the number of particles, φ is the data source of cue (Kinect, color, depth), $L_{\varphi}^{(t)}$ is the distance matrix at time step t, $p_{x_i}^{(t)}$ and $p_{y_i}^{(t)}$,$\varphi_x^{(t)}$ and $\varphi_y^{(t)}$ are the horizantal and vertical positions for i^{th} particle and the related cue, respectively.

Step 2. A separate likelihood matrix is produced for each data source using distance matrices. Kinect, color and depth likelihood matrices are calculated by (6).

$$Lh_{\varphi}^{(t)} = \left\{ e^{\frac{L_{\varphi}^{(t)}}{2\rho\varphi}} \right\}, t = 1..T \tag{6}$$

where ρ_{φ} is the noise covariance matrix of data source and $Lh_{\varphi}^{(t)}$ is the likelihood matrix of the data source at time step t. An inverse proportionality is formed between the distance and the likelihood when calculating likelihood values. Thus, the particles close to the operator-detected points have higher likelihood while the other particles have lower likelihood values.

Step 3. The likelihood values for each data source are normalized to interval $(0, 1)$.

Step 4. The three likelihood matrices are associated in order to obtain a joint likelihood (see (7)).

$$Lh^{(t)} = \left\{ \tilde{Lh}_K^{(t)} \otimes \tilde{Lh}_C^{(t)} \otimes \tilde{Lh}_D^{(t)} \right\}, t = 1..T \tag{7}$$

where $Lh^{(t)}$ is the joint likelihood matrix at time step t, $Lh_K^{(t)}$, $Lh_C^{(t)}$ and $Lh_D^{(t)}$ are normalized likelihood matrices of K, C and D sensors, respectively.

The combined likelihood value at the time step t, $Lh^{(t)}$, are multiplied by the weights of particles at the previous step in order to calculate current time step weights. Those calculated weights are used in estimation process.

4. Findings and evaluation

The problem addressed in this work is the improvement of Microsoft Kinect based real-time motion tracking process by using data-fusion based particle filter. In this section, we shared the findings and evaluated the performance of our method in terms of accuracy, robustness and latency.

4.1. Method Validation

We verified the performance of our method by comparing to the results produced by human operators. In the testing process, a 111-frame video clip including subject's motion has been set for the use of human operators through a web interface. In the study, 7 different human operators manually tracked video for 111 frames.

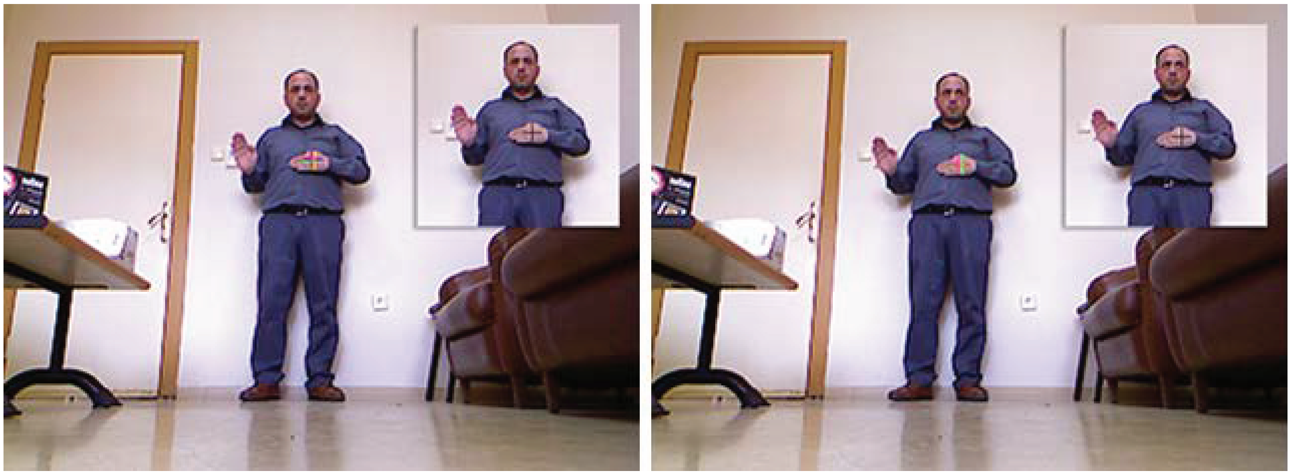

Figure 5. Reference values; left: frame 1, right: frame 81.

The center coordinates of the hand-palm of the moving human subject under tracking has been recorded for each frame, and the average of the coordinates has taken as a reference for the corresponding frame. Figure 5 demonstrates the selections of different human operators (large images) and their averages (thumbnails) in frames 1 and 81, respectively. In each frame (time step), the coordinates computed by the use of color and depth cues and Kinect outputs were combined under the particle filter framework, and the results were compared with the average coordinate values obtained by the human operators.

4.2. Color cue results

The methods used to obtain the color cue for solving the problem of improving the motion tracking process with data-fusion based particle filter are described in subsubsection 3.5.1. The result images obtained by the process at each time step are given Figure 6.

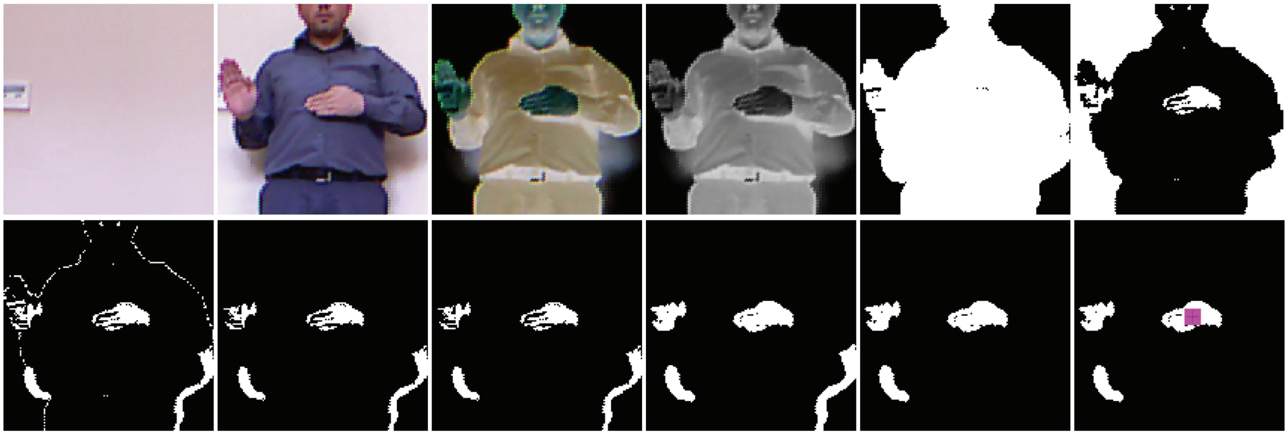

Figure 6. Color cue results referred in subsubsection 3.5.1

In the process of obtaining the color clue, blank and full RGB scene images of size 141 × 141 of 8 bits produced by Kinect were used as the source. Since a real-time motion tracking process is targeted, morphological operations with logical operators are performed predominantly on 1-bit images. Due to the accepted assumptions, the targeted coordinates have been obtained with great accuracy. However, since these assumptions are not valid in real world problems, a Gaussian noise is added to the coordinate values in order to show the effectiveness of the filter.

4.3. Depth cue results

The methods used to derive the depth cue to solve the problem of improving the motion tracking process with data-fusion based particle filter are described in subsubsection 3.5.2. The result images obtained by the process at each time step are given by Figure 7 in which the first four images depicted with histogram equalization operation so as to uncover extremely dark regions.

In the process of obtaining the depth clue, 12-bit blank and full 141×141 size images produced by Kinect were used as the source. As the color cue is obtained, it is preferred to apply morphological operations with the logical operators on the images due to the real-time constraint in obtaining the depth cue. Since the accepted assumptions for the depth cue are less restrictive than the accepted assumptions for the color cue, the targeted coordinates are obtained with a lower accuracy than the color-based model. Similar to the color-based model, in the depth-based model, the Gaussian noise is added to the coordinate values for showing the effectiveness of the filter.

4.4. DF-PF method performance results

The performance of data-fusion based particle filter method proposed in this study is compared with the results achieved by Kinect built-in algorithms. The results of color and depth-based cue implementations are also given in order to reveal the method's efficiency. Results are obtained for the set $N = \{5, 10, 25, 50, 75, 100, 125, 150, 200,$

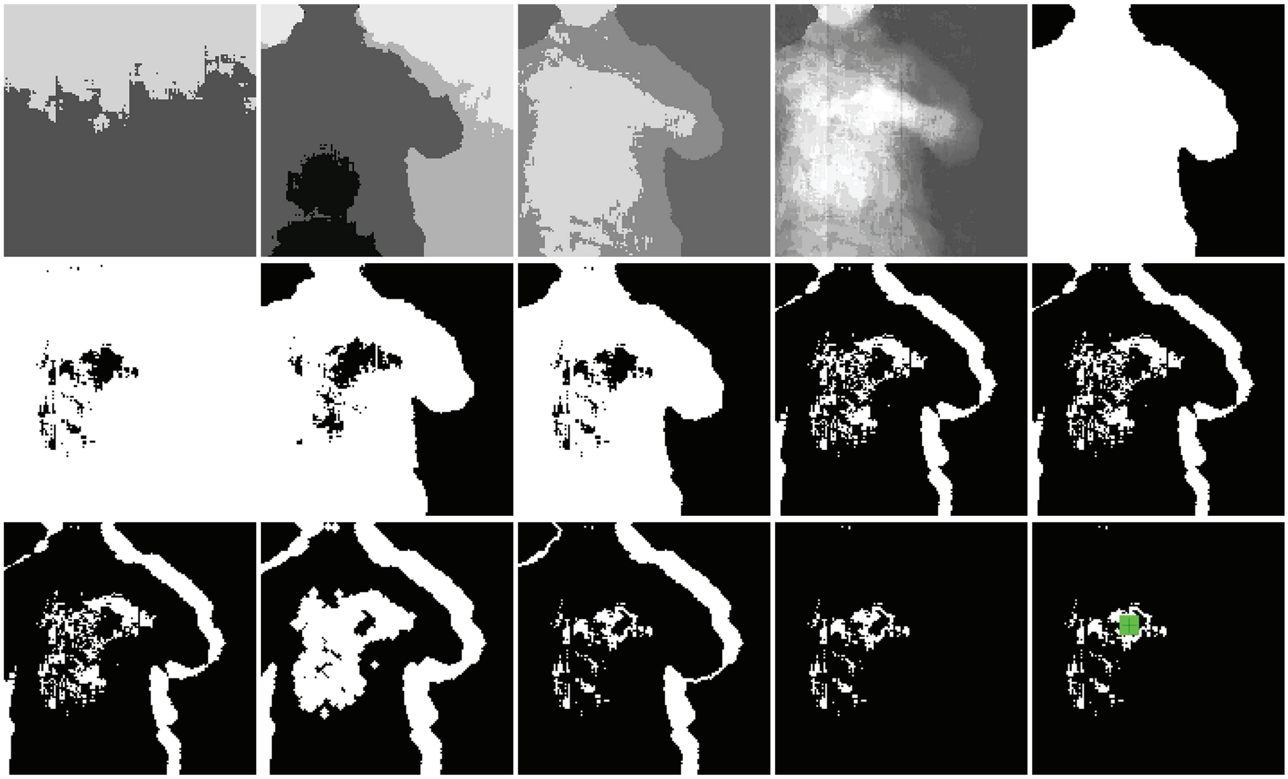

Figure 7. Depth cue results referred in subsubsection 3.5.2

250} in which the elements represent the number of particles simulated. The algorithm is executed once for each selection of the number of particles over 111-time steps. For each time step, 4 RMSE values are obtained for color, depth, Kinect and proposed DF-PF based implementations. The smallest RMSE value among these four values is accepted as the best solution and one point is given for the associated implementation. At the end of the time steps, a total score, BestMax, for each implementation is obtained by summing up their points. The results including BestMax scores are depicted in Figure 8 as line chart.

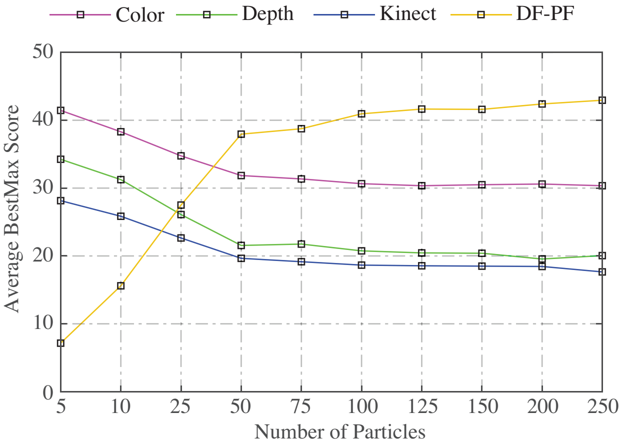

Figure 8. BestMax scores comparison results.

Our method is produced greater BestMax scores than other implementations as the number of particles increases, particularly after 50. Looking at the results in detail, it is obvious that the second best BestMax scores are obtained via color-based implementation. Although, the results obtained with the depth and Kinect-based implementations are close to each other the figures demonstrate that Kinect-based implementation is the worst among all. In addition, we can also extract another information from these figures. The BestMax scores produced by our method do not increase remarkably after the number of particles exceeds beyond 100.

4.5. DF-PF method multiexecution performance

Multiexecution is one of the processes showing the robustness of the method used in problem solving. In this subsection, the process explained in subsection 4.4 is repeated for 20 times and the average of BestMax scores is calculated for each implementation. The results are obtained for the set $N = \{5, 10, 25, 50, 75, 100, 125, 150, 200, 250\}$ in which the elements represent the number of particles simulated. However, only one part of the results is depicted in Figure 9 due to the fact that they are adequate to state the purpose. It is seen from the Figure 9 that multiexecution results are compatible with the results obtained at once. For the particle counts 5 and 25, the best average is produced by color-based implementation while the proposed DF-PF method surpasses as the number of particles approximates to 100. It is also understood from the figure that the results obtained with the depth-based implementation are in some cases worse than Kinect-based implementation.

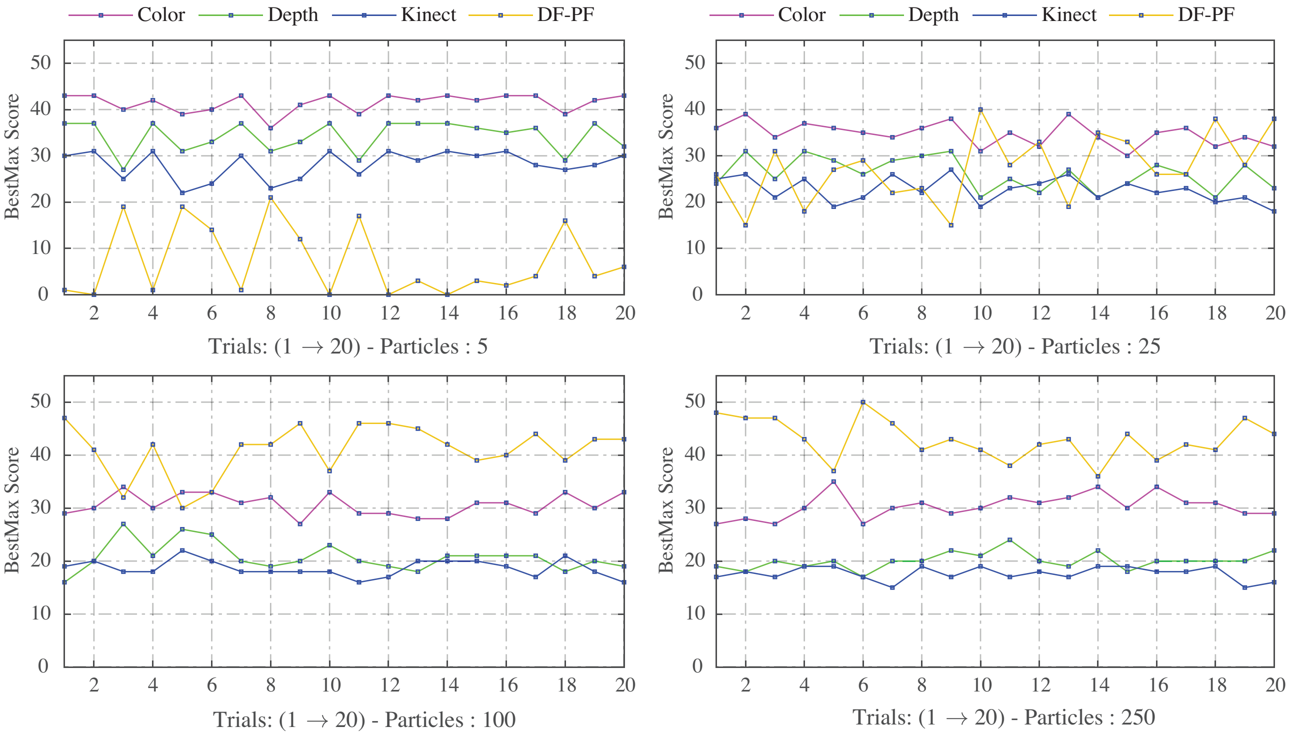

Figure 9. Multiexecution BestMax scores.

4.6. Latency results

It is claimed in this study that the real-time motion tracking process proceeded by Kinect's built-in algorithms is enhanced by using a data-fusion based particle filter method. Therefore, the latency arising from the interim processes applied in the observation model of the proposed method must be at a certain level. In this subsection, the delay times occurred by the color and depth-based observation models are evaluated. The latencies occurred in each time step during the operations performed in color and depth-based observation models are given by Figure 10. It can be understood from this figure that the latencies occurring in the early time steps are pretty much greater than the remaining time steps. Also, the latencies created by depth-based observation model are ten times greater than color-based observation model. However, total latencies occurred in each time step is not surpassing 0.01 s which does not violate the constraints of a real-time motion tracking process.

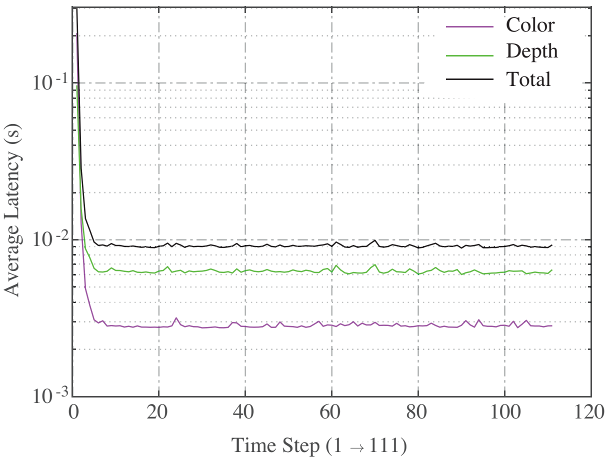

Figure 10. Latency results.

5. Conclusion

Nonlinear state estimation is a common problem addressed in diverse fields of science including almost all engineering disciplines, nature-life and even social sciences. Particle filtering has become an increasingly popular and frequently used method in recent years, as it has the potential to provide reasonable solutions for such problems requiring real-time or online state estimation. The solutions obtained by using particle filter are reasonably approximate but not optimal in the case of using problem specific, appropriate dynamic and observation models. However, particle filtering and also similar Bayesian approaches are already applied in case of nonachievable analytic as well as optimal solution due to some compelling and inherent issues of real-world problems such as curse of dimensionality, process and measurement noises and latency sensitivity. The specific problem handled in this study is enhancing the real-time process of tracking nonstationary hand-palm of a moving human subject performed by Kinect device, a product of Microsoft, using a sensor-fusion based particle filtering algorithm. It is observed that Kinect's built-in algorithms are not efficient enough and produce relatively large error rates particularly in case of hand-body occlusion. However, it is also evaluated that the produced tracking outputs may be utilized as base and the whole tracking process can be improved using a more intelligent algorithm. To this end, a novel algorithm is employed fusing Kinect's outputs together with fresh information yet extracted from Kinect-acquired RGB and depth images within a particle filter framework. The joint coordinates of moving human subject are located using Kinect's built-in motion tracking algorithms. A constant velocity based dynamic model is adopted for predicating subject's motion and an observation model including extraction of desired point coordinates of hand-palm based on both color and depth images. All this frame-based information is fused together in order to make a particle filter estimation which is getting better in the end of every new time step. Consequently, the best available solution is obtained throughout a survival of the fittest process.

The proposed method is compared with Kinect's built-in tracking system under the guidance of ground-truth information recorded manually by human operators. We demonstrated finally that under real-time constraints, our method is superior to Kinect's built-in tracking system both for a single run and multiexecutions when the number of particles is between 50 and 100. The experimental results are visualized revealing various aspects of the algorithm including effects of the selected number of particles, comparison of RMSE values produced by different implementations, multiexecution results and latency related issues.

A number of improvements to the proposed method are envisaged as future work including refinement of algorithm parameters such as initial belief, process and measurement noise variances, dynamic model equation, decreasing the number of operations in both color and depth-based observation models as well as increasing their efficiency in terms of accuracy and speed, and employing novel particle resampling implementations in order to get more accurate and robust solutions.

References

[1] Moeslund TB, Granum E. A survey of computer vision-based human motion capture. Computer Vision and Image Understanding 2001; 81 (3): 231-268.

[2] Moeslund TB, Hilton A, Krüger V. A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 2006; 104 (2-3): 90-126.

[3] Poppe R. Vision-based human motion analysis: an overview. Computer Vision and Image Understanding 2007; 108 (1-2): 4–18.

[4] Kumar A, Walia GS, Sharma K. Recent trends in multicue based visual tracking: a review. Expert Systems with Applications 2020; (162): 113711.

[5] Jiang H, Li J, Wang D, Lu H. Multi-feature tracking via adaptive weights. Neurocomputing 2016; 207: 189-201.

[6] Hu M, Liu Z, Zhang J, Zhang G. Robust object tracking via multi-cue fusion. Signal Processing 2017; 139: 86-95.

[7] Tran TH, Le TL, Hoang VN, Vu H. Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. Computer Methods and Programs in Biomedicine 2017; 146: 151–165.

[8] Li X, Liu Q, He Z, Wang H, Zhang C, Chen WS. A multi-view model for visual tracking via correlation filters. Knowledge-Based Systems 2016; 113: 88–99.

[9] Z H, Fei M, Sadka A, Zhang Y, Li X. Adaptive fusion of particle filtering and spatio-temporal motion energy for human tracking. Pattern Recognition 2014; 47 (11): 3552–3567.

[10] Bueno M, Díaz-Vilariño L, Martínez-Sánchez J, González-Jorge H, Lorenzo H et al. Metrological evaluation of KinectFusion and its comparison with Microsoft Kinect sensor. Measurement 2015; 73: 137-145.

[11] Caruso L, Russo R, Savino S. Microsoft Kinect V2 vision system in a manufacturing application. Robotics and Computer-Integrated Manufacturing 2017; 48: 174–181.

[12] Boutellaa E, Hadid A, Bengherabi M, Ait-Aoudia S. On the use of Kinect depth data for identity, gender and ethnicity classification from facial images. Pattern Recognition Letters 2015; 68: 270-277.

[13] Hbali Y, Hbali S, Ballihi L, Sadgal M. Skeleton-based human activity recognition for elderly monitoring systems. IET Computer Vision 2017; 12 (1): 16-26.

[14] Antón D, Kurillo G, Goñi A, Illarramendi A, Bajcsy R. Real-time communication for Kinect-based telerehabilitation. Future Generation Computer Systems 2017; 75: 72-81.

[15] Cabrera R, Molina A, Gómez I, García-Heras J. Kinect as an access device for people with cerebral palsy: A preliminary study. International Journal of Human-Computer Studies 2017; 108: 62-69.

[16] Yavşan E, Uçar A. Gesture imitation and recognition using Kinect sensor and extreme learning machines. Measurement 2016; (94): 852–861.

[17] Napoli A, Glass S, Ward C, Tucker C, Obeid I. Performance analysis of a generalized motion capture system using microsoft kinect 2.0. Biomedical Signal Processing and Control 2017; 38: 265-280.

[18] Henseler H, Bonkat SK, Vogt PM, Rosenhahn B. The Kinect recording system for objective three-and four-dimensional breast assessment with image overlays. Journal of Plastic, Reconstructive & Aesthetic Surgery 2016; 69 (2): e27–e34. doi: 10.1016/j.bjps.2015.10.021

[19] Hayat M, Bennamoun M, El-Sallam AA. An RGB–D based image set classification for robust face recognition from Kinect data. Neurocomputing 2016; 171: 889-900.

[20] Riahi D, Bilodeau GA. Online multi-object tracking by detection based on generative appearance models. Computer Vision and Image Understanding 2016; 152: 88-102.

[21] Wang Q, Fang J, Yuan Y. Multi-cue based tracking. Neurocomputing 2014; 131: 227-236.

[22] Guipeng D, Gang T, Chunqiao P, Xiaofeng W. A Correlation Particle Filter Target Tracking Algorithm Based on Adaptive Feature Fusion. In: IEEE 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI); Chengdu, China; 2020. pp. 419-423.

[23] Dong J, Xu Y, Liu C. Multi-feature Fusion Target Tracking Algorithm Based on Global and Local Consistency. In: IEEE 2020 International Conference on Artificial Intelligence and Computer Applications (ICAICA); Dalian, China; 2020. pp. 1280-1284.

[24] Lei Q, Li Z, Wang M, Feng J, Zhang R. Research on multi-feature adaptive fusion face tracking algorithm. Journal of Physics: Conference Series 2020; 1518 (1).

[25] Kumar A, Walia GS, Sharma K. Real-time visual tracking via multi-cue based adaptive particle filter framework. Multimedia Tools and Applications 2020; 79 (29): 20639-20663.

[26] Walia GS, Kumar A, Saxena A, Sharma K, Singh K. Robust object tracking with crow search optimized multi-cue particle filter. Pattern Analysis and Applications 2020; 23 (3): 1439-1455.

[27] Shehzad MI, Karam FW, Azmat S. Low-cost multiple object tracking for embedded vision applications. Turkish Journal of Electrical Engineering and Computer Science 2019; 27 (3): 1737-1751.

[28] Chen Z. Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics 2003; 182 (1): 1–69.

[29] Taşcı T, Öz C. A closer look to probabilistic state estimation – case: particle filtering. Optoelectronics and Advanced Materials – Rapid Communications 2014; 8 (5-6): 521-534.