Kernel Density Estimation and Correntropy based Background Modeling and Camera Model Parameter Estimation for Underwater Video Object Detection

Susmita Panda (susmitapanda@soa.ac.in)

Siksha O Anusandhan University Institute of Technical Education and Research https://orcid.org/0000-0003-4123-0921

Pradipta Kumar Nanda

Siksha O Anusandhan University Institute of Technical Education and Research

Research Article

Keywords: kernel density estimation , correntropy , background modeling , camera parameter estimation

Posted Date: April 13th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-397880/v1

Noname manuscript No.
(will be inserted by the editor)

Kernel Density Estimation and Correntropy based Background Modeling and Camera Model Parameter Estimation for Underwater Video Object Detection

Susmita Panda · Pradipta Kumar Nanda

Received: date / Accepted: date

Abstract Underwater video object detection is challenging because of the complex background and the movement of the camera. In order to address this, we propose a novel scheme of simultaneously estimating the camera model parameters and detecting the object. The object detection phase includes background modeling and its learning. Background is modeled by the proposed Spatial Kernel Density Estimation (SKDE) model and the model learning happens in the SKDE feature space. Background modeling and its learning is pixel based approach. The model histograms learn the new pixel through its histogram representation. Our learning and classification strategy is different from the Heikkila et al. [17] in the context of similarity measure. We have proposed the correntropy based similarity measure that is used for model learning and pixel classification. The camera model parameters are estimated by 2D optimization method where we have used the corner features of an object at subpixel accuracy level. These subpixel level features are used in the proposed pipelining framework for model parameters estimation. The estimated model parameters are used to transform the input frame, which in turn is used for model learning and classification. The proposed scheme has been tested with underwater video frames from six data sets. The efficacy of the proposed scheme is compared with seven existing schemes and it is found that the proposed scheme exhibits improved performance as compared to the existing methods.

Keywords kernel density estimation · correntropy · background modeling · camera parameter estimation

Susmita Panda
Image and Video Analysis Lab
Dept. of Electronics and Communication Engineering
Institute of Technical Education and Research
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneshwar
India
E-mail: susmitapanda@soa.ac.in

Pradipta Kumar Nanda
Image and Video Analysis Lab
Dept. of Electronics and Communication Engineering
Institute of Technical Education and Research
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneshwar
India

1 Introduction

Moving object detection in video sequences is one of the fundamental tasks for a number of video processing applications such as recognition, tracking, understanding the behavior of the moving object. In order to detect the object of interest in a specific scene, the background of the scene has to be modeled. Because of the complex environment, the task of building and updating a background model has now become a major issue in the field of computer vision.

Many researchers [8, 16, 49, 56, 58] have been working on this challenging issue to detect the moving object in a scene having a dynamic background. The typical methods for moving object detection are foreground extraction and background subtraction. Foreground extraction techniques classify pixels according to the changes in the incoming frames, while background subtraction suppresses the background by comparing an incoming frame to the background template. Broadly, the background modeling techniques are classified as; (i) Parametric model, and (ii) Non-parametric model. In case of Parametric modeling, the background is mostly modeled using either a single Gaussian distribution or mix-

ture of Gaussian distributions [13, 23, 28, 49]. Some of the situations like the waves on water or trees shaken by the wind, where pixel intensities surrounding these objects tend to vary significantly over time and hence this poses a challenge. To overcome such issues, the probability distributions of the pixel intensity are estimated independently for each frame. Such type of schemes which involve estimation of pixel intensity distributions directly from the data is categorized as Non-parametric model. Here, the model [8, 39] adapts the fast changes in background process and detects a target with high sensitivity.

The background modeling task becomes challenging in case of complex scenes, both inland and underwater. Underwater object detection represents one of the most challenging scenarios due to the presence of dynamic entities like waves on the water surface, movement of the object with variable speed, waves created by boat and uneven illumination condition in the scene. Owing to the physical properties of the underwater environment such as light absorption, scattering, density and restrictive human access, visibility in underwater is greatly hampered. These conditions are the motivating factors for this research work to address the problem of underwater object detection. As the traditional methods [7,19,34,57] of moving object detection have proved not to be very effective for the underwater environment problems, Liu et al. [21] proposed an effective and reliable method to detect moving object from underwater video by combining the notion of background subtraction and three frame differences followed by morphological processing. Similarly, in case of Underwater environment, Prabowo et al. [36] have proposed a method to detect the object by subtracting the current frame from the pixels in the previous frame background model. The complexity of such type of environment is further compounded if the video sequence is captured by a moving camera. In case of moving camera, Stolkin et al. [50] have proposed Expectation Maximization (EM) based tracking algorithm for poor visibility condition. Besides, Panda et al. [30] have also addressed the incomplete data problem while simultaneously estimating the camera position and image labels using Expectation Maximization (EM) algorithm and Extended Markov Random Field (E-MRF). Further, a new Spatio temporal Markov Random Field based models have been proposed by Panda et al. [31] for simultaneously estimating the camera model parameters and detecting the object.

In this piece of work, a new scheme is proposed for simultaneously estimating the camera model parameters and background model learning. In the background modeling phase, we have estimated the probability density of every pixel of a video frame using the Kernel Density Estimation(KDE) technique. Spatial neighborhood pixels are used for the density estimation and hence this is known as the spatial KDE (SKDE) model. These estimated densities of pixels of frames are used as the features for background modeling and model learning. The histogram distribution of a window around a given pixel of the SKDE frame serves as the model for that pixel. For a given pixel, the histograms of the corresponding pixels in the temporal direction of SKDE frames serve as the model histograms of the given pixel. The principle of model learning is different from that of Heikkila et al. [17] in the context of proximity measure which is based on the proposed correntropy measure. These model histograms of a given pixel learn the histogram of the corresponding pixel of the incoming frame. After learning, depending upon the degree of proximity, each pixel is classified either to be a background pixel or a foreground pixel. In the above process, all the pixels of the incoming frame are classified as either background or foreground, thus the entire frame is classified to detect the object. These classified frames are used for the estimation of the intrinsic and extrinsic parameters of the moving camera in the underwater environment. The camera parameters are estimated based on the 2D optimization method using the proposed notion of pipelining. The estimated model parameters are used to transform the frames, which are subsequently used as the input frames for SKDE modeling and model learning. Thus in this process, the camera model parameter estimation and object detection are carried out simultaneously. The proposed scheme has successfully been tested on underwater video frames of six datasets. The results obtained by the proposed scheme are compared with seven existing techniques and the proposed algorithm exhibited improved performance in the context of different quantitative measures.

This paper is organized as follows. Section 2 deals with the related research works, while the proposed scheme is presented in Section 3. SKDE based modeling and model learning are provided in Section 4 and the camera model parameter estimation is provided in Section 5. Results and Discussions are presented in Section 6 while the concluding remarks are presented in Section 7.

2 Related work

Different classification schemes based on the notion of background subtraction (BS) are proposed in literature [6, 7, 57]. In the *pixel* based approach [13, 23, 28, 49, 55] the change of each pixel in the temporal direction is considered as an independent process. This

method is used for real time classification of moving objects. In literature, authors have also considered *region* based algorithms [15, 17, 29, 40, 46] where a frame is divided into blocks and the block based features are used to detect the foreground. In a region based algorithm, the histogram of that region is computed and edges are preserved while removing the noise. H.Liu et al. [21], Singla et al. [45] and Zhang [60] have developed *Per-frame* based algorithms that could detect global changes in the scene particularly in poor visibility conditions. Background modeling can also be grouped into *Multi-stage* category [18,53] where several steps are performed at different stages to improve the accuracy of the final result. Cheung and Kamat [41] proposed two types of BS methods namely the *Recursive* and *Non-Recursive*. In case of *Recursive* algorithm [28,49] a single background model is recursively updated on each new incoming frame. Here the researchers have used Gaussian mixtures to model a single pixel. In the case of *Non-Recursive* approach [9, 33], the authors maintain a buffer of previous frames and estimate the background model based on a statistical analysis of the available frames in buffer. BS methods are also broadly categorized as *Predictive* and *Non-predictive* by Mittal and Paragios [29]. Predictive algorithms [29] model the scene as a time series and develop a dynamic model to recover the current input based on past observation. The background model can also be predicted by Kalman filter [25], Wiener filter [53] and neural networks [10] where pixels of the incoming image which vary significantly from its predicted value are classified as foreground. Non-predictive methods [12, 49] neglect the order of input sequence and create a probabilistic representation of the observation at a particular level.

Further, in complex environment, the backgrounds are model using statistical approach to detect the foreground [7]. These statistical models can be categorised as Parametric and Non-parametric. In the case of the *Parametric approach*, the probability density function of the pixel process is represented parametrically using a prescribed statistical distribution. The parametric based appraoches [13, 23, 28, 49] have limitations in handling dynamic environments in an underwater environment. Alternatively, in *Non-parametric approach*, the density function can be directly obtained from the pixel without any assumptions about the underlying distribution. Though this approach [11, 14, 20, 27, 32, 38, 56] is able to construct statistical representation of foreground or background, but it is not able to learn all the changes of a dynamic background, especially the changes on the water surface. In order to handle such dynamic background, researchers have extended the temporal approach to develop the spatio-temporal models which is presented in [55]. Alvarez et al. [26] have proposed an adaptive background model within an adaptive learning framework considering the Spatio temporal relationships among pixels. Recently, authors [24] have attempted to minimize the effect of different video irregularities like dynamic background, change in illuminations, video noise. by Spatio-Temporal Region Persistence (STRP) descriptor and adaptive threshold. Further, to enhance the discriminative ability of the background model, Zhong et al. [62] have proposed a dual target nonparametric background model for classifying a pixel either as static object or dynamic background.

The maritime backgrounds are more complex than other dynamic backgrounds since waves on the water surface do not belong to the foreground despite being in motion. Additionally, the problem is more compounded due to poor illumination conditions. Srividhya et al. [48] have extracted different statistical features like autocorrelation and sum of entropy which are used by the learning algorithms to classify underwater objects. To detect underwater moving object, Liu et al. [21] have proposed a underwater object detection scheme which combines the notions of background subtraction and three frame differences under the assumption of a fixed camera position. Similarly, the authors in [36] have addressed an adaptive background modeling method to detect moving objects on an underwater video. Vasamsetti et al. [47] have proposed a new feature descriptor, a multi-frame triplet pattern for underwater moving object detection. With increasing demand of smart phone cameras, H. Sajid et al. [42] have proposed a hybrid method that combines motion and appearance in an online framework for foreground/background segmentation of videos. Further, researchers [52, 59] have also tried to detect moving objects in complex scenes by assuming a camera in motion.

3 Proposed Scheme

The proposed scheme is shown in Fig. 1. As observed from Fig. 1, for continious video object detection, the camera model parameters and the background model learning are alternated to simultaneously learn the background and detect the object. As observed from Fig. 1, at a given time $'t'$, the input frame is transformed by the previously estimated camera model parameters at (t-1). Thereafter, Spatial KDE (SKDE) of the transformed frame is found out. Background modeling and model learning are pixel based processes. For a given pixel of SKDE frame, the histogram of the window around the pixel contributes to the learning of the corresponding model histograms. In other words, the model his-

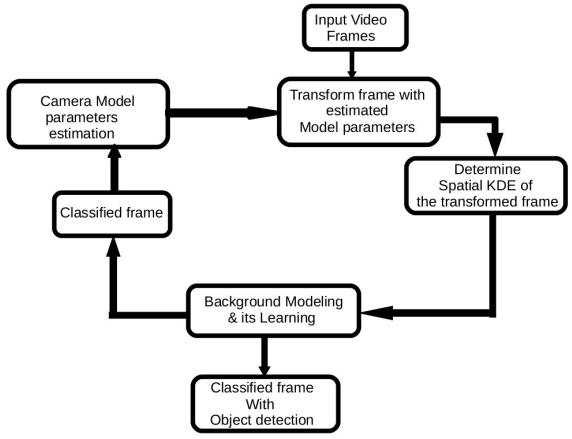

Fig. 1 Simultaneously model parameter estimation and object detection.

tograms of the corresponding pixel are updated. Subsequently, the pixel is classified as either a background pixel or foreground pixel. This process of learning and classification is repeated for all the pixels of a given frame. Thus the learning and classification takes place for the entire frame. The classified frame at t^{th} time instant is used with a few previously classified frames for parameter estimation.

Thus, as observed from Fig. 1, the object in a frame is detected by completion of all the processes once in the entire loop. This process is repeated for all the input video frames and hence object detection is carried out continiously in underwater environment. The details of the proposed scheme is presented by the block diagramatic representation of Fig. 2. Detail explainations of each block are provided below.

Model initialization Phase

Block 1 of Fig. 2 denotes the initialization phase where few model frames are selected. The model histograms are generated as follows. For any given pixel, a few

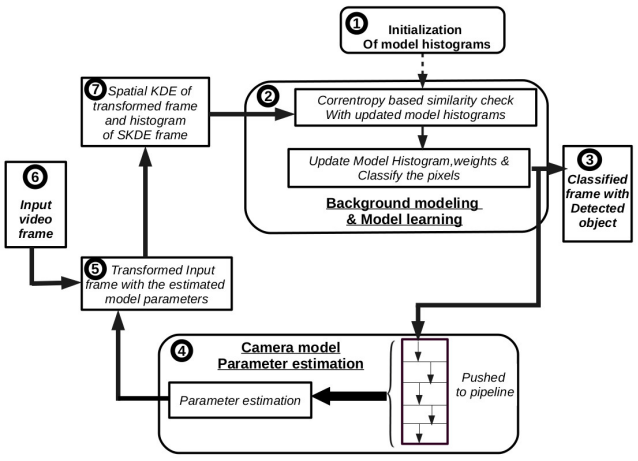

Fig. 2 Schematic representation of the proposed scheme.

SKDE frames in temporal direction are choosen. Windows of a given size are constructed around the pixels of the frames and the histograms of these windows serve as the model histograms for that pixel. For example, when we consider to have 3 model histograms for a given pixel of t^{th} SKDE frame, we consider the corresponding pixels of $(t-1)^{th}$, $(t-2)^{th}$ and $(t-3)^{th}$ frames and construct windows of a given size around them. The histograms of these windows of $(t-1)^{th}$, $(t-2)^{th}$ and $(t-3)^{th}$ serve as the background model histograms for the given pixel at t^{th} and these model histograms are updated to learn the information from the subsequent frames of $(t-1)^{th}$, $(t-2)^{th}$ and $(t-3)^{th}$.. . Similarly, model histograms are generated for every pixel of the SKDE frame.

Background Modeling and Model Learning

Model learning takes place in block 2 of Fig. 2. For learning of t^{th} frame, the entire frame is transformed by the camera model parameters estimated at $(t-1)^{th}$ frame as shown in block 7. Since learning is pixel based, learning of a given pixel of t^{th} transformed SKDE frame, a window of a given size is considered around the pixel of the t^{th} frame and the histogram of this window is considered for learning of the model histograms. For learning, the similarity check of this histogram with each of the model histograms is found out. The proposed correntropy measure between the histograms is considered as the similarity measure. The histogram of the pixel of the t^{th} frame is compared with each of the model histograms in the context of similarity measure. Based on the value of correntropy of each model histogram, a weight is assigned to the model histogram. The model histograms are now updated bin wise based on the adaption rule. Besides, the weights of the model histograms are also updated. This process of updation constitutes the learning phase of the background model for the given pixel. After learning of a given pixel, classification of the pixel takes place. This process is repeated for all the pixels of the t^{th} input frame to complete the model learning and the classification of the entire frame. This is shown in block 3 of Fig. 2.

Model Parameter Estimation

As seen in Fig. 2, the camera model parameter estimation phase follows the background learning phase. The classified t^{th} frame, thus obtained is fed to the camera model parameter estimation block which is shown in block 4 of Fig. 2. The classified frame is pushed to a pipeline which has earlier been filled up with the classified frames of $(t-1)^{th}$, $(t-2)^{th}$....,$(t-5)^{th}$ frames. As shown inside block 4, in the pipeline six classified frames are used for camera model parameter estimation based on 2D optimization method. Both the intrinsic and extrinsic parameters are estimated. The extrinsic param-

eters are used to transform the next input frame as shown in block 5. Thereafter, the SKDE of the frame is determined in block 7 and this SKDE frame is used for learning in block 2. This process of learning, classification and parameter estimation continues for subsequent frames.

4 KDE based Background Modeling and model learning

Since the background of the underwater is complex because of the poor visibility and dynamic conditions, we have adhered to SKDE based approach for background modeling. In this KDE based background modeling, Spatial KDE (SKDE) of the video frames are computed and the histogram of the window around a given pixel of the SKDE frame is considered as the model of the pixel. Background model histograms are found out by the histograms of the same pixel in the temporal domain. In the learning phase, Correntropy measure is used as the similarity measure between the incoming histogram and the model histograms. Hence, in the following we present the KDE estimation process in the spatial domain and correntropy measure.

4.1 Spatial Kernel Density Estimation (SKDE)

KDE aims to produce a smooth, continuous estimate of a univariate or multivariate probability density using a positive function kernel i.e $K_\sigma(X;\sigma)$ which is controlled by a bandwidth σ [11]. Given a sample S = $\{x_{i,j}\}_{i,j=1...N}$ consisting of pixel intensity, an estimate of probability density function $\hat{p}_c$ at a position c i.e center pixel within a group of pixels $X_i; i = 1...M$ can be calculated using,

$$\hat{p}_c = \frac{1}{M+1}\sum_{m=0}^{M-1} K_\sigma(X_c - X_m), \tag{1}$$

where X denotes an input frame and M denotes the total number of pixels in the neighborhood of center pixel c.
Here we choose the kernel function as a Gaussian function. The bandwidth acts as a smoothing parameter controlling the trade off between bias and variance in the result. High bias is obtained with a large value of bandwidth. Similarly, low bias or low variance is obtained with a small value of bandwidth. Hence the probability density estimated at the center point can be expressed as

$$\hat{p}_c = \frac{1}{M+1}\sum_{m=0}^{M-1} \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\frac{(X_c - X_m)^2}{\sigma^2}} \tag{2}$$

Since we have considered only the spatial neighborhood pixels of a given pixel at $'c'$ site, the KDE found at site $'c'$ i.e $\hat{p}_c$ is called as Spatial KDE (SKDE). Spatial KDE maps the pixel intensity with the probability density function. This estimation is expected to remove false detection due to the waves in the water surface or the random noise which occurs due to the uneven illumination. In this piece of work, the bandwidth of the Gaussian Kernel is assumed to be constant for all the pixels of a frame.

In our work, the SKDE of a pixel is computed as follows. A window of a given size is considered around the pixel. The KDE of a given pixel is computed using Eq. (2). While using Eq. (2), c denotes the center pixel and X_m denotes the neighborhood pixels of the given pixel. M denotes the number of neighborhood pixels in a window. This process is repeated for all the pixels of a given frame to obtain the SKDE of the frame. Similarly, SKDE of all the frames are computed and these SKDE frames are the feature frames used for background modeling and model learning.

4.2 Correntropy: A similarity measure

A nonlinear statistical measure of similarity between two random variables is named as Correntropy. It is a generalized correlation measure between two random variables induced by a kernel function. This is a single measure that includes time structure and the statistical distribution as stated in [22,43,44,61]. Liu et al. [22] defined Correntropy between two arbitrary random variables Y_1 and Y_2 as,

$$Cor(Y_1, Y_2) = E[< \Phi(Y_1), \Phi(Y_2) >] = E[k(Y_1 - Y_2)], \tag{3}$$

Where Y_1= Y_{t1} and Y_2=Y_{t2}. Thus correntropy is an extension of auto-correntropy between two random process. The name correntropy comes from the fact that its mean value is the argument of the log of quadratic Renyi's entropy of $Y_1 - Y_2$. It has a maximum value at the origin ($\frac{1}{\sqrt{2\pi\sigma}}$) and it is also a symmetric positive function. In our work, we have used correntropy for measuring the similarity between the new input histogram distribution and model histograms. The Correntropy between the n^{th} bins of the two histogram is expressed as,

$$Cor_\sigma(a_n, b_n) = E[k_\sigma(a_n - b_n)], \tag{4}$$

where $k_\sigma(.)$ is a positive definite kernel, with the kernel width determined by the parameter σ and a_n, b_n are the n^{th} bins of the corresponding histograms. The Correntropy values for all the bins of the histograms are

computed and the similarity measure between two histograms is the average of all the Correntropy values. As finite number of samples available, the following sample esitmators are used for the expectation operator as in [44].

$$\hat{Cor}_{N,\sigma}(a_n, b_n) = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(a_i - b_i). \quad (5)$$

Where N is the number of sample present within the kernel. We assume $k_\sigma(.)$ to be a normalized Gaussian kernel with variance σ. Hence Eq. (5) can be expressed as,

$$\hat{Cor}_{\sigma}(a_n, b_n) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}[e^{(\frac{-(a_i-b_i)^2}{2\sigma^2})}]. \quad (6)$$

Thus, Correntropy can be viewed as a generalized correlation function between two random variables, containing higher order moments of the error $(a_n - b_n)$ between them. It measures the similarity between two random variables, within a small neighborhood determined by the kernel width σ.

4.3 Background Modeling

In this work, background modeling is carried out in feature space instead of the raw data space. Few initial frames are chosen and the Spatial KDE (SKDE) of these frames are found out. The SKDE frames are considered to be the feature frames. The modeling and model learning are pixel based approaches. For modeling a given pixel of the SKDE frame, a window is constructed around the pixel and the histogram of the window serves as the model of the pixel. In order to obtain the model histograms of a pixel, the corresponding pixels in the temporal directions are considered and the histograms of these temporal pixels serve as the model histograms. The number of model histograms may vary, but for the sake of illustration, three such model histograms are shown in Fig. 3. Thus, these three histograms are considered as the model histograms for the pixel. In the learning phase, the corresponding model histograms are updated for each input pixel of the new input frame. This updation process of these model histograms together with the updation of the weights are known as **model learning**.

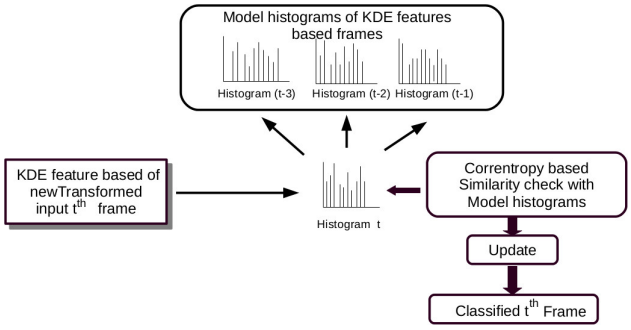

Fig. 3 Schematic diagram of the proposed spatiotemporal background modeling and model learning.

4.4 KDE based Model learning

The model learning of feature frames is a pixel based learning process. With the background model histograms of SKDE frames of Fig. 3, model learning takes place with every new input pixel of the new frame. For example, for background modeling of a pixel at t^{th} time instant, the histograms of the corresponding pixels of few past frames i.e $(t-1)^{th}$, $(t-2)^{th}$ and $(t-3)^{th}$ frames are considered as model histograms for background modeling which is shown in Fig. 3. The model histograms are also assigned initial weights such that $\sum_{k=1}^{K} w_k = 1$, where k denotes the number of model histograms. Let $H_o, H_1, H_2.....H_{k-1}$ denote these k model histograms. For a new SKDE frame, the histogram of the window around the corresponding pixel is considered as the new input histogram. Let H_n denotes the histogram of the new input pixel of the SKDE frame. The model histograms learn this new input histogram and the learning process is presented below.

Proximity of the new input histogram H_n is determined with each of the model histograms by correntropy measure. Correntropy values of H_n with respect to each of the model histograms $H_o, H_1, H_2.....H_{k-1}$ are computed. The highest Correntropy between the new input histogram and any of the model histograms corresponds to the best match between the new histogram and model histograms. If the Correntropy value of the input histogram with any of the model histogram is above a preselected threshold T_p then the pixel is considered as a background pixel or else foreground pixel. If the correntropy value is below the threshold T_p for all the model histograms, then the histogram with the lowest weight is replaced by H_n. This replaced H_n is assigned with lowest weight. Thereafter, the best match model histogram is updated with the new input histogram by the following bin updation procedure.

$$\hat{H}_{ok} = \alpha_1 H_n + (1-\alpha_1)H_{ok}, \quad (7)$$

where $\hat{H}_{ok}$ is the estimated model histogram, H_{ok} is the best match model histogram and H_n is the histogram of the new frame presented for learning and α_1 is the learning parameter.
The weights of the model histograms are updated as follow,

$$\hat{w}_k = \alpha_2 H_k + (1-\alpha_2)w_k, \quad (8)$$

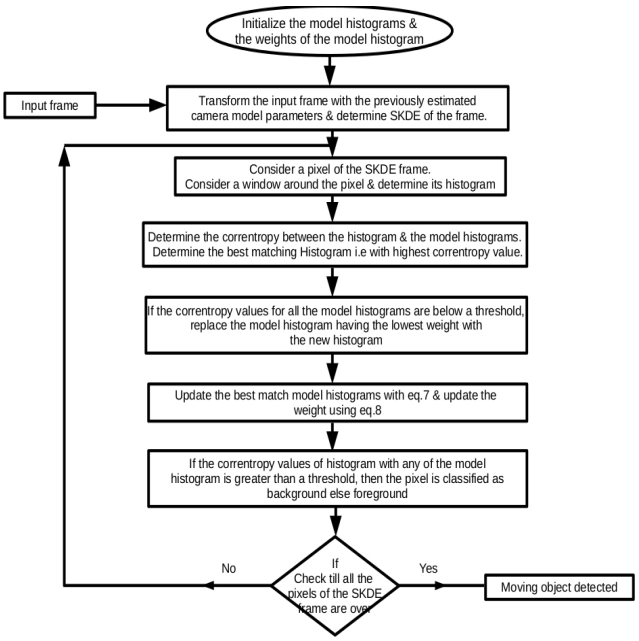

Fig. 4 Flowchart for model learning and classification.

where α_2 is user defined parameter and H_k is unity for best matching histogram and 0 for others.

Thereafter, the model histograms are sorted based on decreasing order of the weights and the first N histograms are selected as the background model histograms based on the following condition.

$$w_0 + ... + w_{N-1} > T_B \qquad T_B \epsilon (0,1) \tag{9}$$

Where T_B is a user defined parameter. The above process is repeated for all the pixels of the input frame and the pixels are classified. In this process, the model histograms for all the pixels of the new SKDE frame are updated.

4.5 Object Detection

Classification of a pixel of the input SKDE frame is carried out before updating the background model histograms for learning. H_{ni} denotes the histogram of i^{th} pixel of the SKDE frame. Let the corresponding model histograms be denoted as $H_{0ni}, H_{1ni}, H_{2ni}.....H_{(k-1)ni}$.

First the histogram H_{ni} is compared with the above model histograms for similarity check. Here the similarity measure is the correntropy measure. If the correntropy of the histogram H_{ni} with any of the model histograms is above a threshold T_p then the pixel is classified as background and otherwise the pixel is classified as foreground or object region. In otherwords, if the correntropy values H_{ni} with all the model histograms is less than the threshold T_p, then the pixel is classified as foreground or object. Thus, by classifying all the pixels of a given frame, the object in that frame is detected. The flowchart of model learning and classification algorithm is given in Fig. 4. The salient steps of the algorithm are enumerated below.

Algorithm 1 Model learning and classification algorithm

Input: A given pixel of SKDE frame with its histogram.
Output: Classified pixel as either background or foreground.
1 . Initialize the model histograms of every pixel of SKDE frame.
2 . Consider a given pixel of the transformed SKDE frame and determine the histogram of the pixel by considering a window around it.
3 . Determine the Correntropy of the histogram of the pixel with each of the model frames.
4 . If the correntropy value of the input histogram is below the threshold T_p for all the model histograms then classify the pixel as foreground or else background.
5 . Determine the best matching model histogram.
6 . Update model histogram according to Eq. (7).
7 . Update the weights of the histogram according to Eq. (8).
8 . Sort the histograms in accordance with the weights and select the background model histogram and foreground model histogram.
9 . Repeat the above process for every pixel of the new input SKDE frame.

5 Camera Model

Accurate estimation of camera model parameters leads to the detection of the video object accurately. We have estimated the camera parameters using the notion of pipeline shown in Fig. 5. After the classification of input frames, features are extracted from these frames. The accuracy of the estimation depends upon the proper choice of features of different views. In order to obtain the features corresponding to the shape of an underwater moving object, improved Harris corner detection algorithm [37] has been used to extract the features for parameter estimation.

We have used five stages in the pipeline to estimate the camera parameters. Hence, at a given time, features of five views of a given video have been used to obtain the estimates of the parameters. In the following, we present the process of parameter estimation. The pipeline consists of five stages and initially, all the stages are empty. As shown in Fig. 5, at $T = t - 4$ time slot, the corner features are input to the pipeline and the rest four stages are with null features. The available features in the pipeline would result in the inaccurate estimates of the parameters. Thereafter, the features corresponding to view 1 (first frame) are shifted to the next stage thus enabling the features of view 2 to occupy the first stage. This process is continued and at

time $T = t$, all the pipeline stages are filled up with the features of the respectives frames and the parameters are estimated using these features. As we progress further down the pipeline of Fig. 5, the pipeline stages are filled up with features of more views. Thereafter, the process of shifting and inputing of new feature is continued for estimating the parameters of different frames.

Camera model parameters are estimated based on the 2D optimization method proposed by Zhou et al. [63]. With parameter vector θ i.e $\theta = (f_x, f_y, u_0, v_0, R, t)^T$, the objective function is developed to minimize the distance between the estimated image point $\hat{i}_u$ and the distorted image point $\hat{i}_u^d$. Where $\hat{i}_u$ denotes the feature point mapped to the image coordinate system and $\hat{i}_u^d$ is the corresponding distorted image points. Let f_x, f_y denote the respective focal lengths and u_0, v_0 denote the initial positions of the camera coordinate. R and t denote the rotational matrix and the translation vector respectively. The objective function as proposed by Zhou. [63] is expressed as,

$$\hat{\theta} \;=\; \arg\min_{\theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \parallel i_u^d - \hat{i}_u^d \parallel^2. \tag{10}$$

In this research, the image frames are considered from six data sets. For the sake of illustration, Fig. 6 shows the Harris corner features of the whale in the frame. Firstly, these corner points are mapped into the camera coordinate plane and thereafter they are mapped into the image coordinate plane. In our work, distortion has not been taken into account and hence, the distance between the estimated image point in the image coordinate plane $\hat{i}_u$ and real image point i_u is minimized. Hence, the estimated point in the image coordinate $\hat{i}_u$ is a function of intrinsic parameters (f_x, f_y, u_0, v_0) and extrinsic parameters R and t i.e $\hat{i}_u = \mathrm{f}(f_x, f_y, u_0, v_0, R, t)$.

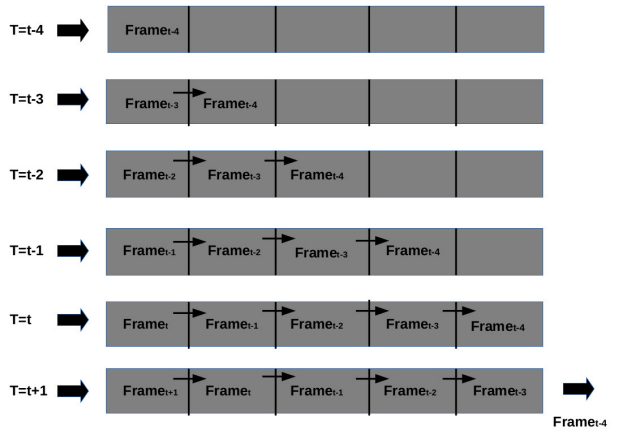

Fig. 5 Notion of pipelining for camera model parameter estimation.

The parameter vector $\theta = \arg\min(f_x, f_y, u_0, v_0, R, t)^T$. Therefore, in this case, the problem is reformulated as,

$$\hat{\theta} \;=\; \arg\min_{\theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \parallel i_u - \hat{i}_u \parallel^2. \tag{11}$$

Since the camera is in motion, the new input frame has been transformed by the estimated extrinsic parameter matrix consisting of rotational angle θ and translational parameters t_x, t_y, t_z. As the segmented frame and the estimated camera parameters are interrelated, hence the transformed input frame has been subjected to Spatial KDE. The block diagramatic represntation of the process of camera calibration is present in Fig. 7.

5.1 Camera Model parameter estimation

It is known that proper choice of feature points contributes predominantly for the accuarate estimation of parameters. Because of underwater environment, appropriate feature points may not be extracted. In order to ameliorate the issue, steerable pyramid filters with different angles are used for different frames. Steerable filters are used to obtain different features of a given

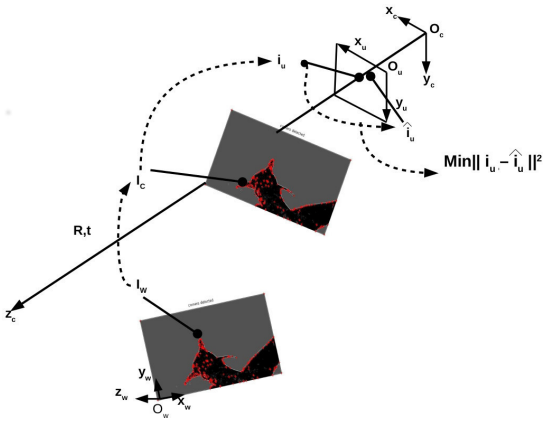

Fig. 6 This figure shows the step for the 2D optimization of whale. Where I_w = World coordinate plane, I_c = Camera coordinate plane, i_u = image plane, O_c = optical center of the camera and z_c = optical axis of the camera lens.

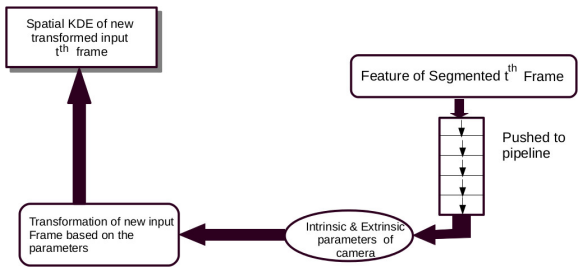

Fig. 7 This figure shows the different steps for estimating camera parameters.

frame with different orientation. This filter is recursive in nature and hence the k directional bandpass filter can be expressed as,

$$B_m(u,v) = HP(f_1, f_{\frac{N}{2}}, s)\cos^{k-1}(\theta - \frac{m\pi}{k}), \tag{12}$$

Where m=0,.....,k-1 and,

$$S = (u^2 + v^2)^{\frac{1}{2}}, \tag{13}$$

S is the radial variable in frequency space and $\theta = \tan^{-1}(\frac{v}{u})$ is the angular variable in frequency space. HP(a,b,f) is a high pass transfer function, raised to cosine.

$$HP(a,b,f) = \begin{cases} 0 & ; \quad f \leq a \\ \sqrt{\frac{1}{2}[1 - \cos[\pi(\frac{(f-a)}{(b-a)})]]} & ; \quad a < f < b \\ 1 & ; \quad f \geq b \end{cases} \tag{14}$$

The kernels at different angles have been applied to the considered frames for feature extraction. Further, the proper features points have been extracted by the use of Steerable pyramid filters by exposing different surfaces of the object. It has been reported in [37] that the corner points serve as the feature points in the checkerboard image. In our case, the underwater video objects are from six data sets. Harris corner detection algorithm has been the choice for detection of corner points. Though Harris corner detection operator can take care of image rotation, gray change and noise interference, but it has the limitation in detecting the corner points corresponding to the coordinate of pixels points only. From a practical standpoint, it may be conceivable that accurate corners may correspond to sub-pixel coordinate positions instead of pixel coordinates. Hence, in our case, the accurate corners of the underwater object may correspond to sub-pixel accuracy. This motivated us to adhere to the improved Harris corner detection algorithm with sub-pixel accuracy. We have used the improved Harris corner sub-pixel corner detection algorithm [37] in different video frames. For a given frame, the feature points are weighted to take care of the orientation and movement. We have assigned different weightage to different frame feature points intending to take care of the movements in different frames. These weighted features are mapped to the coordinate frame. Following are the salients steps of the camera parameter estimation as presented in Fig. 6:

Algorithm 2 Parameter estimation algorithm

Input: Feature points of segmented frame.
Output: Estimated Intrinsic and Extrinsic camera parameters.
1 . Transform the feature points I_w in the world coordinate system to the camera coordinate system using the extrinsic parameter R,t. These features points in camera coordinate systems are denoted as I_c.
2 . Project the camera coordinate point I_c using the intrinsic parameters (f_x, f_y, u_0, v_0) and this point is denoted as $\hat{i}_u$.
3 . Compute the distance between the $\hat{i}_u$ and i_u for every image point and minimize the following objective function.

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \parallel i_u - \hat{i}_u \parallel^2 \tag{15}$$

Use Levenberg Marquardt algorithm to solve the following optimization problem.

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{M}\sum_{j=1}^{N} \parallel i_u - \hat{i}_u \parallel^2 \tag{16}$$

4 . Obtain the estimated camera parameter vector by solving Eq. (16).

5.2 Combined algorithm for object detection and model parameters estimation

The object detection and camera model parameters estimation are carried out once in each epoch of the combined algorithm. One epoch of the combined algorithm detects object in one frame. Thus for continious object detection and parameter estimation, the combined algorithm corresponding to Fig. 1 and Fig. 2 is executed. The flowchart for the combined algorithm is presented in Fig. 8. Salient steps of the algorithm are enumerated below.

Algorithm 3 Combined algorithm

Input: Sequence of underwater video frames.
Output: Video frames with detected objects.
1 . Initiate the model histograms and the associated weights.
2 . Transform the input video frame with the camera model parameters. Compute the SKDE of the frame.
3 . Consider a pixel of the transformed SKDE frame and make the model histogram learn the histogram of the pixel according to the model learning algorithm of subsection 4.6.
4 . Classify the pixel and hence the frame.
5 . Use the classified frame with a few previously classified frames for parameter estimation according to the algorithm of subsection 5.1.
6 . Repeat the steps 2 to 5 till all the video frames are considered.

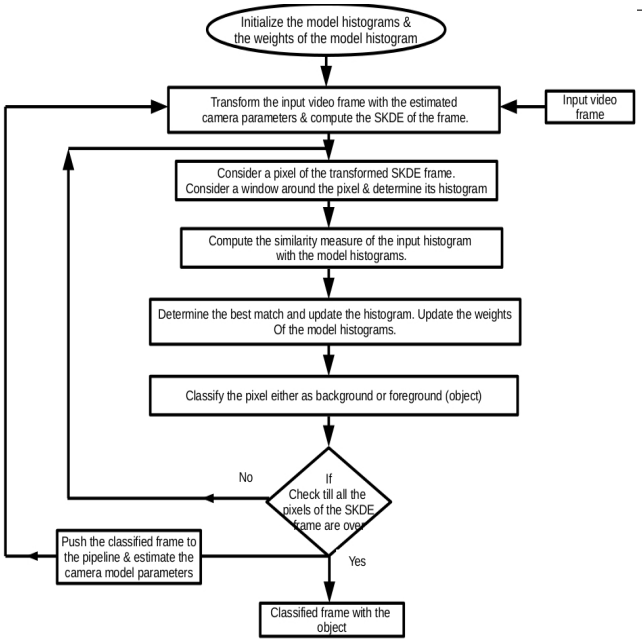

Fig. 8 Flowchart for the combined algorithm.

6 Results and Discussions

We have considered different views from 6 datasets namely Creepy chimara / Nautilus live video [5], Blainville's beaked whale dataset [3], Whalesharks in Philippines southern Leyte dataset [2], Montserrat - south - Nautilus dataset [4], Giant sea Turtle dataset [54] and Bluefin - 21 Unmanned Undersea Vehicle dataset [51]. These datasets [2–5,51,54] are the videos in which the moving underwater objects are captured by a nonstatic camera in a poor visibility dynamic environment. The proposed algorithm has been successfully tested on these datasets and the performance of the proposed algorithm is compared with that of Stolkin et al. [50], Prabowo et al. [36], Liu et al. [21], Elgammal et al. [11], Alvarez-Meza et al. [26], Zhong et al. [62] and Ahmed et al. [1].

The intrinsic and extrinsic camera model parameters are estimated using the proposed weighted corner features of the previously classified frames. These weighted features of five previous frames are used in the different pipeline stages to estimate the model parameters. There are five pipeline stages as shown in Fig. 5 and these stages are necessary for estimating the parameters with mimimum error. Hence, at a given point of time, features of five views of a given dataset have been used to obtain the estimated parameters. For example, in the case of dataset 1, features of classified frames with frame numbers 24, 25, 26, 27 and 28 are pushed into the pipeline to estimate the camera parameters of 28^{th} frame. Similarly, in order to estimate the camera parameters of frame number 31, features of the classified frames with frame numbers 27, 28, 29, 30 and 31 are pushed into the pipeline. In case of dataset 2, camera parameters of 16^{th} frame are estimated by pushing frames numbered 12, 13, 14, 15 and 16 into the pipeline. Similar process is performed to estimate the parameters of 18^{th} frame of the same dataset by pushing features of frames numbered 14, 15, 16, 17 and 18 into the pipeline. The same process is repeated for datasets 3, 4, 5 and 6.

Table 1 Camera intrinsic parameters (in mm) for different dataset using the proposed model

Intrinsic Parameters	dataset 1	dataset 2	dataset 3	dataset 4		dataset 5	dataset 6
				view1	view2		
f_x	57.9	29.41	38.5	43.39	42.4	62.1	34.9
f_y	53.67	29.97	38.8	41.59	41.02	67.8	41.2
u_0	336.39	321.01	200.1	321.3	308.02	406.7	226.5
v_0	452.68	432.68	70.8	442.81	446.19	312.8	106.1

Table 1 presents the estimated intrinsic parameters of different datasets. These parameters correspond to the optical centers and focal lengths of the moving camera. Since different cameras were used for different datasets, the intrinsic parameters differ from each other. In order to test the efficacy of the parameter estimation strategy, two different views of the same dataset (Dataset4) have been considered. As observed 4^{th} and 5^{th} column of Table 1, the estimated intrinsic parameters are close to each other, as the frames considered are from two views of the same data set. Camera calibration error is the difference between the actual image point and the estimated position.

Table 2 Camera calibration error (in pixels) for datasets 1, 2, 3 & 4 with our proposed model

	Dataset 1		Dataset 2		Dataset 3		Dataset 4			
Original Views	Frame 28	Frame 31	Frame 16	Frame 18	Frame 16	Frame 20	Frame 200	Frame 220	Frame 300	Frame 355
Camera calibration error	9.27	3.71	7.99	6.59	6.9	5.9	7.25	3.7	9.27	3.44

The accuracy of the estimated parameters depends upon the calibration error, the less the calibration error the better is the accuracy of estimation. The calibration errors for different datasets are provided in Table 2 and Table 3, where it may be observed that the calibration errors are of low values. Further, as observed from Table 2, in case of dataset 1 the calibration error for frame number 31 is less than that of frame number 28. In the case of the 2^{nd} dataset, the calibration error of 18^{th} frame is less than that of 16^{th} frame. Similar observations are also made for datasets 3, 4, 5 and 6 as presented in Table 2 and Table 3. Hence, the estimated parameters with low value of calibration errors are acceptable. Fig. 9 to Fig. 11 show the segmented results

Table 3 Camera calibration error (in pixels) for datasets 5 & 6 with our proposed model

	Dataset 5		Dataset 6	
Original Views	Frame 46	Frame 156	Frame 265	Frame 388
Camera calibration error	2.12	1.79	4.96	3.09

of different frames from different datasets. As observed from the original frames, there is a single moving object in a dynamic and unevenly lighted background. The results obtained by our proposed algorithm are compared with that of Stolkin et. al, Prabowo et. al, Liu et. al, Elgammal et. al, Alvarez-Meza et. al, and Zhong et. al's algorithms. Frames shown in Fig. 9(a), Fig. 10(a) and Fig. 11(a) correspond to the original frames while Fig. 9(b), Fig. 10(b) and Fig. 11(b) show the corresponding ground truth frames. Fig. 9(c) to Fig. 9(i), Fig. 10(c) to Fig. 10(i) and Fig. 11(c) to Fig. 11(i) show the results obtained by different algorithms while Fig. 9(j), Fig. 10(j) and Fig. 11(j) show the results obtained by the proposed algorithm. Visual inspection of the results of different frames reveals that there are different degree of misclassification in the object and background portions. In some cases, many object portions are not detected properly. But as observed from Fig. 9(j), Fig. 10(j) and Fig. 11(j), the proposed algorithm could detect the object with a minimum amount of misclassification error and also the background could be detected properly. As observed from the results obtained by the proposed algorithm, in some cases the shape of the object has been retained but because of some false positive cases the object appears to be a bit of different size as compared to the original object. This effect has been reflected on the quantitative measures such as *Precision* and *Recall*.

The segmentation accuracies of different frames have been measured by the five quantitative measures [35,64] i.e: (i) Percentage of Misclassification Error, (ii) Precision (iii) Recall (iv) Dice Coefficient and (v) F-measure as shown in Table 4 and 5. They are defined as follows.
*Percentage of Missclassification Error (**PME**),*

$$\mathbf{PME} = \frac{no.\ of\ missclassified\ pixels}{total number of pixels} \times 100.$$

The next quantitative measures considered are Precision and Recall and are defined as,

$$Precision(\boldsymbol{Pr}) = \frac{TP}{TP+FP},$$
$$Recall(\boldsymbol{Re}) = \frac{TP}{TP+FN},$$

where, TP is true positive, FP is false positive and FN is false negative.
The fourth quantitative measure considered is the Dice Coefficient which is defined as,

$$\mathbf{DC} = \frac{2\times|S_F \bigcap GT_F|}{|S|+|GT|},$$

where S denotes the segmented image, GT denotes the ground truth, FG and BG corresponds to the foreground and background respectively. The last quantitative measure is F-Measure which is defined as,

$$\mathbf{F\text{-}Measure} = (2 \times Precision \times Recall) \setminus (Precision + Recall).$$

For the 16^{th} and 18^{th} frames of dataset 2, it is observed that the values of Recall, Dice coefficient and F-measure for the proposed algorithm are highest values among all, whereas the precision value is more than two existing algorithms but less than the other four algorithms. This is attributed to the false positives in the object portions. As seen from Table 4, for the 16^{th} and 20^{th} frames of Dataset 3, the Recall, Dice coefficient and F-measure values are highest among all the algorithms considered. But in this case, the precision is 84.9% for the 16^{th} frame, which is higher than those of three algorithms and comparable to one algorithm and less than that of two algorithms. Hence, in this case, both Precision and Recall values are high thus indicating that the object has been detected. Similar observations are also made for the 20^{th} frame. Visual inspection of the results of Fig. 9 also reveals that there is almost no change in the size of the detected object.

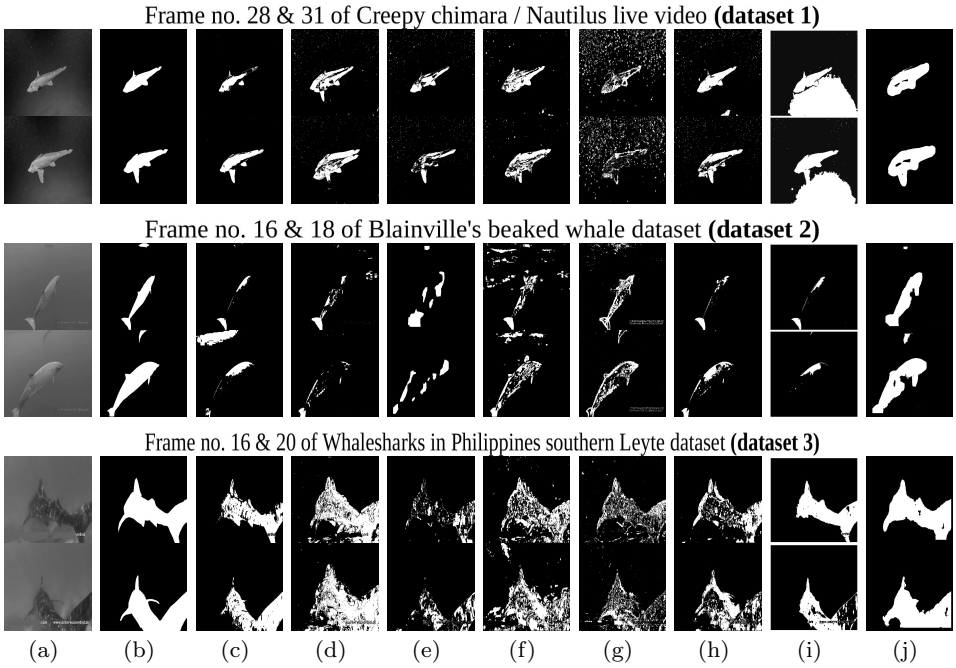

Fig. 9 (a) Original frame (b) Ground truth (c) Stolkin et al. [50] (d) M. R. Prabowo et al. [36] (e) H. Liu et al. [21] (f) A. Elgammal et al. [11] (g) A. M. Alvarez-Meza et al. [26] (h) Z. Zhong et al. [62] (i) S. Ahmed et al. [1] (j) Proposed algorithm.

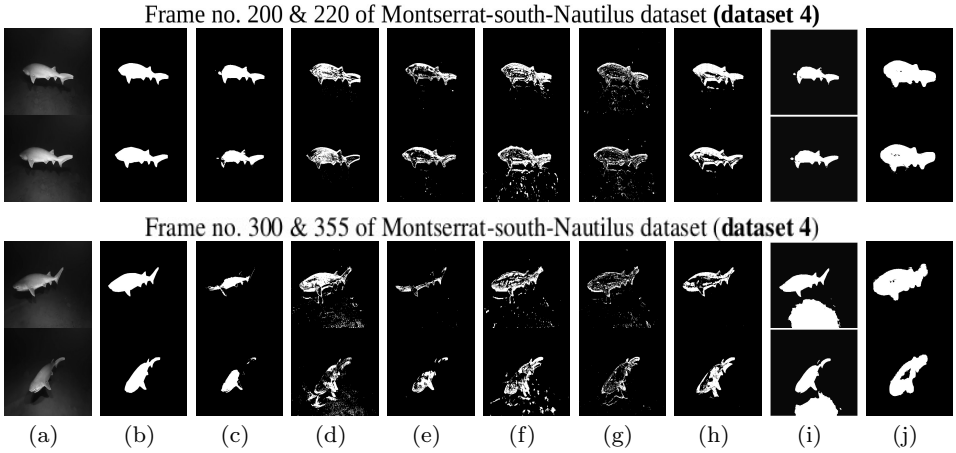

Fig. 10 (a) Original frame (b) Ground truth (c) Stolkin et al. [50] (d) M. R. Prabowo et al. [36] (e) H. Liu et al. [21] (f) A. Elgammal et al. [11] (g) A. M. Alvarez-Meza et al. [26] (h) Z. Zhong et al. [62] (i) S. Ahmed et al. [1] (j) Proposed algorithm.

For the four different frames of Dataset 4, as observed from Table 4, the Recall values are highest indicating that the object could be detected. The values of the Dice cooefficient and F-measure are also high and are comparable to others. But the Precision is less than that of other algorithms. This observation is similar to those of Dataset 1 and 2. Table 5 presents the quantitative measures for 5^{th} and 6^{th} datasets. As observed in all these four frames, the Recall values are highest amomg all the algorithms. In case of 46^{th} and 156^{th} frames of 5^{th} dataset, the Dice coefficient values are highest among all thus indicating the accuracy of the detected objects. The F-measure values for the proposed algorithm are highest for these frames. Similar observations are also made for 265 and 388^{th} frames of 6^{th} dataset. Further, the average quantitative measures for all the data sets are found out and are presented in Table 6. We have considered 15 frames or more in each data set to determine the average measures and the number of frames considered in each data set is presented in Table 6. As observed from Table 6, the recall values for the proposed algorithm are highest among all

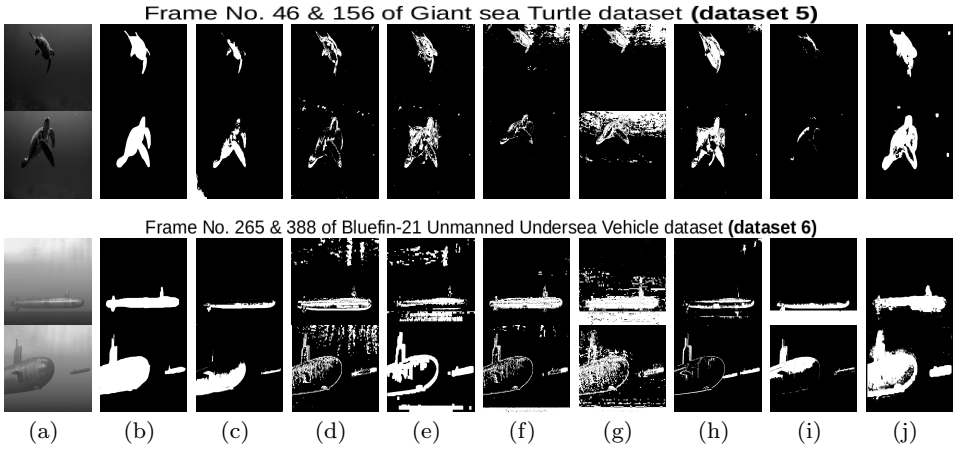

Fig. 11 (a) Original frame (b) Ground truth (c) Stolkin et al. [50] (d) M. R. Prabowo et al. [36] (e) H. Liu et al. [21] (f) A. Elgammal et al. [11] (g) A. M. Alvarez-Meza et al. [26] (h) Z. Zhong et al. [62] (i) S. Ahmed et al. [1] (j) Proposed algorithm.

Table 4 Quantitative measure of different datasets

Frame no.	Quantative measure	Stolkin's et. al	M. R. Prabowo et. al	H. Liu et. al	A. Elgammal et. al	A. M. Alvarez-Meza et. al	Z. Zhong et. al	S. Ahmed et. al	Proposed Model
Creepy chimaera/Nautilus live video dataset (dataset 1)									
28	Missclassification error (in %)	2.0	4.1	2.1	2.6	8.8	1.5	26.6	4.3
	Precision (in %)	99.7	56.3	87.0	67.7	26.2	78.6	14.4	52.8
	Recall (in %)	57.2	68.9	66.1	80.8	49.8	93.2	94.1	**95.1**
	Dice Coefficient (in %)	97.1	95.8	97.0	97.0	90.8	98.2	73.2	95.6
	F-measure (in %)	72.7	62.0	75.1	73.7	34.4	85.3	24.9	67.9
31	Missclassification error (in %)	1.3	4.5	3.6	3.5	11.0	3.1	19.7	5.7
	Precision (in %)	89.9	61.8	86.7	73.4	19.6	79.4	24.0	52.5
	Recall (in %)	80.3	73.2	50.9	68.8	24.2	70.0	92.4	**93.5**
	Dice Coefficient (in %)	98.6	95.4	96.3	96.1	88.6	96.6	79.9	94.8
	F-measure (in %)	84.8	67.1	64.1	71.1	21.7	74.4	38.1	67.3
Blainville's beaked whale dataset (dataset 2)									
16	Missclassification error (in %)	4.0	6.9	3.9	6.2	4.5	4.2	3.1	5.1
	Precision (in %)	98.5	34.8	77.6	43.8	58.8	91.7	99.3	52.3
	Recall (in %)	27.4	30.2	39.8	49.8	56.8	25.5	20.1	**82.8**
	Dice Coefficient (in %)	95.3	93.0	95.3	93.7	95.1	95.0	93.6	**95.8**
	F-measure (in %)	42.8	32.3	52.6	46.6	57.8	39.9	33.5	**64.1**
18	Missclassification error (in %)	12.9	12.2	9.5	8.6	7.0	7.7	8.9	6.4
	Precision (in %)	36.8	33.3	96.9	66.8	82.4	99.4	99.5	66.0
	Recall (in %)	24.0	11.3	14.1	43.2	45.6	30.1	8.1	**84.8**
	Dice Coefficient (in %)	87.0	87.7	90.4	91.3	92.6	92.2	88.5	**93.5**
	F-measure (in %)	29.1	16.8	24.7	52.5	58.7	46.2	15.1	**74.3**
Whalesharks in Philippines southern Leyte dataset (dataset 3)									
16	Missclassification error (in %)	3.6	6.3	9.0	8.2	8.6	6.9	3.1	**2.9**
	Precision (in %)	99.7	71.3	91.8	69.2	78.2	85.2	99.1	84.9
	Recall (in %)	70.0	78.2	26.1	55.9	37.9	50.0	64.8	**91.6**
	Dice Coefficient (in %)	96.2	93.4	90.7	85.4	84.5	87.5	90.1	**97.1**
	F-measure (in %)	82.2	74.6	40.6	61.9	51.1	63.1	78.3	**88.1**
20	Missclassification error (in %)	2.6	9.6	7.7	11.3	8.9	9.8	5.0	**3.7**
	Precision (in %)	99.7	55.9	96.9	50.7	72.1	60.1	99.2	82.1
	Recall (in %)	77.2	74.2	34.0	53.6	35.8	41.5	59.3	**86.3**
	Dice Coefficient (in %)	97.2	90.1	92.1	79.8	84.0	82.4	84.2	**96.3**
	F-measure (in %)	87.1	63.8	50.3	52.1	47.9	49.1	74.1	**84.1**
Montserrat-south-Nautilus dataset (dataset 4)									
200	Missclassification error (in %)	1.5	3.1	3.7	4.1	5.6	2.5	2.0	4.4
	Precision (in %)	99.8	84.6	95.5	74.7	67.5	82.8	91.1	62.01
	Recall (in %)	78.2	68.8	50.3	64.5	41.8	81.5	60.8	**99.75**
	Dice Coefficient (in %)	98.4	95.5	96.2	95.8	94.3	97.4	97.9	95.5
	F-measure (in %)	87.7	75.9	65.9	69.2	51.6	82.1	72.9	76.48
	Missclassification error (in %)	3.0	3.8	4.0	6.8	7.1	3.7	2.4	4.7
	Precision (in %)	90.9	85.9	89.4	53.3	53.9	80.5	91.4	60.8
	Recall (in %)	60.7	58.4	51.0	54.3	31.2	65.7	63.4	**99.2**
	Dice Coefficient (in %)	96.7	96.2	95.9	93.1	92.9	96.3	97.9	95.2
220	F-measure (in %)	72.7	69.5	64.9	53.8	39.5	72.3	73.4	**75.4**
Montserrat-south-Nautilus dataset (dataset 4)									
300	Missclassification error (in %)	5.0	4.6	6.3	5.2	7.2	4.1	16.1	6.2
	Precision (in %)	99.1	70.8	97.0	71.5	60.8	87.8	32.0	55.8
	Recall (in %)	39.6	70.9	20.9	57.1	28.0	55.2	91.8	**99.2**
	Dice Coefficient (in %)	95.1	95.3	93.6	94.7	92.8	95.8	83.8	93.7
	F-measure (in %)	56.9	70.9	34.5	63.5	38.4	67.8	47.5	**71.5**
355	Missclassification error (in %)	2.8	7.3	3.2	6.1	5.6	3.3	9.3	4.7
	Precision (in %)	99.9	40.4	99.0	49.9	58.3	76.1	39.1	57.3
	Recall (in %)	54.7	38.6	48.6	53.1	28.4	65.6	91.7	**88.8**
	Dice Coefficient (in %)	96.9	92.2	96.7	93.8	94.3	96.6	90.6	95.2
	F-measure (in %)	70.7	39.5	65.2	51.5	38.1	70.4	54.7	**70.0**

the algorithms. The F-measure and the Dice coefficient values are also highest one in five cases. Thus in all the cases, the proposed algorithm exhibited improved performance as compared to other algorithms. Hence, the proposed algorithm could detect the underwater object under poor visibility and dynamic background condition and with the movement of camera as well.

Table 5 Quantitative measure of few more datasets

Frame no.	Quantative measure	Stolkin's et. al	M. R. Prabowo et. al	H. Liu et. al	A. Elgammal et. al	A. M. Alvarez-Meza et. al	Z. Zhong et. al	S. Ahmed et. al	Proposed Model
				Giant sea Turtle dataset (dataset 5)					
46	Missclassification error (in %)	6.6	2.6	2.1	8.9	14.2	2.9	3.6	**1.6**
	Precision (in %)	99.7	86.6	88.2	31.7	22.3	67.7	99.6	75.8
	Recall (in %)	37.1	54.1	67.0	76.0	79.5	74.7	6.6	**98.1**
	Dice Coefficient (in %)	92.1	92.2	92.8	85.5	80.5	91.9	90.4	**93.2**
	F-measure (in %)	54.0	66.6	76.2	44.8	34.8	71.0	12.4	**85.5**
156	Missclassification error (in %)	8.3	6.7	6.5	7.2	28.3	5.3	8.7	**3.7**
	Precision (in %)	99.8	20.2	74.6	94.9	19.8	78.7	98.5	78.8
	Recall (in %)	36.3	13.0	53.7	29.2	60.8	66.3	2.9	**86.4**
	Dice Coefficient (in %)	90.2	87.5	88.9	87.5	67.6	90.1	85.8	**91.5**
	F-measure (in %)	53.2	15.8	62.4	44.7	29.9	72.0	5.7	**82.5**
				Bluefin -21 Unmanned Undersea Vehicle dataset (dataset 6)					
265	Missclassification error (in %)	8.1	10.4	17.6	2.6	21.7	9.1	26.3	**3.5**
	Precision (in %)	91.9	58.9	35.8	94.6	37.9	78.0	17.3	82.6
	Recall (in %)	34.7	67.9	46.4	84.5	90.9	41.1	34.9	**91.8**
	Dice Coefficient (in %)	67.6	66.5	61.9	70.0	56.5	67.6	54.6	**71.8**
	F-measure (in %)	50.4	63.1	40.5	89.3	54.3	53.8	23.2	**86.9**
388	Missclassification error (in %)	14.2	24.2	20.5	26.2	20.7	25.3	15.3	10.7
	Precision (in %)	4.6	61.6	65.5	54.0	63.5	74.4	99.5	76.3
	Recall (in %)	13.4	30.7	51.7	21.0	55.1	11.3	37.4	**86.4**
	Dice Coefficient (in %)	57.9	63.5	66.5	60.8	64.5	62.7	66.9	**73.9**
	F-measure (in %)	6.8	40.9	57.8	30.2	59.0	19.7	54.5	**81.0**

Table 6 Average Quantitative measure of different datasets.

Frame no.	Quantative measure	Stolkin's et. al	M. R. Prabowo et. al	H. Liu et. al	A. Elgammal et. al	A. M. Alvarez-Meza et. al	Z. Zhong et. al	S. Ahmed et. al	Proposed Model
				Creepy chimaera/Nautilus live video dataset (dataset 1)					
26-40	Avg. Missclassification error (in %)	2.8	4.5	2.7	2.9	9.2	1.9	24.9	4.7
	Avg. Precision (in %)	98.8	54.8	86.1	59.7	25.1	75.6	20.9	51.9
	Avg. Recall (in %)	55.2	67.1	65.3	78.8	49.1	92.2	90.5	**94.3**
	Avg. Dice Coefficient (in %)	96.5	94.8	96.1	95.0	90.2	97.3	78.2	94.6
	Avg. F-measure (in %)	70.8	60.3	74.3	67.9	33.2	83.1	36.1	66.9
				Blainville's beaked whale dataset (dataset 2)					
06-20	Avg. Missclassification error (in %)	5.6	8.2	4.8	7.8	6.2	5.9	6.9	6.9
	Avg. Precision (in %)	96.5	32.3	76.6	40.9	57.8	89.6	99.4	51.1
	Avg. Recall (in %)	25.4	28.4	38.1	47.4	55.8	23.7	23.1	**80.9**
	Avg. Dice Coefficient (in %)	94.1	91.7	94.3	92.9	93.9	93.2	91.9	**94.8**
	Avg. F-measure (in %)	40.2	30.2	50.8	43.9	56.8	37.5	31.9	**62.6**
				Whalesharks in Philippines southern Leyte dataset (dataset 3)					
08-22	Avg. Missclassification error (in %)	4.6	6.9	11.1	8.9	9.1	7.8	6.1	**3.4**
	Avg. Precision (in %)	99.3	69.3	90.1	67.8	76.2	84.2	99.1	83.1
	Avg. Recall (in %)	68.8	77.1	24.5	54.8	35.8	48.7	62.8	**90.2**
	Avg. Dice Coefficient (in %)	95.2	92.1	88.7	83.4	82.6	86.1	82.9	**96.1**
	Avg. F-measure (in %)	81.3	72.9	38.5	60.6	48.7	61.7	81.3	**86.5**
				Montserrat-south-Nautilus dataset (dataset 4)					
195-225	Avg. Missclassification error (in %)	2.3	3.1	4.7	6.2	6.8	3.7	2.9	5.3
	Avg. Precision (in %)	99.2	82.6	94.5	72.5	65.1	80.8	89.9	63.5
	Avg. Recall (in %)	76.2	67.3	51.9	62.1	38.8	79.8	61.9	**99.1**
	Avg. Dice Coefficient (in %)	97.1	93.8	95.2	94.2	93.3	95.8	98.4	94.1
	Avg. F-measure (in %)	86.2	74.2	67.0	66.9	48.6	80.3	70.9	77.4
				Montserrat-south-Nautilus dataset (dataset 5)					
295-360	Avg. Missclassification error (in %)	6.2	6.2	7.8	5.4	8.2	5.1	10.9	6.9
	Avg. Precision (in %)	98.2	71.5	95.3	70.1	61.6	85.8	31.7	57.2
	Avg. Recall (in %)	37.6	69.1	20.1	56.1	29.6	54.3	90.1	**99.1**
	Avg. Dice Coefficient (in %)	95.4	94.1	92.1	93.7	91.8	94.1	84.2	93.8
	Avg. F-measure (in %)	54.4	70.3	33.2	62.3	39.9	66.5	55.9	**72.5**
				Giant sea Turtle dataset (dataset 6)					
40-200	Avg. Missclassification error (in %)	8.4	5.8	7.2	9.2	20.5	6.2	8.2	**2.8**
	Avg. Precision (in %)	99.8	75.9	86.1	92.1	23.9	70.1	99.2	79.1
	Avg. Recall (in %)	39.1	40.2	58.9	56.9	72.1	75.9	5.1	**96.2**
	Avg. Dice Coefficient (in %)	92.5	88.9	93.1	67.9	65.3	78.5	89.2	**94.3**
	Avg. F-measure (in %)	56.1	52.6	69.9	70.3	35.9	72.9	9.7	**86.8**
				Bluefin -21 Unmanned Undersea Vehicle dataset (dataset 7)					
250-390	Avg. Missclassification error (in %)	13.1	23.5	22.9	23.8	22.1	17.9	25.5	**7.8**
	Avg. Precision (in %)	87.6	63.2	50.3	82.1	49.3	77.5	29.7	79.8
	Avg. Recall (in %)	33.9	52.7	48.3	80.1	69.7	31.9	36.2	**92.1**
	Avg. Dice Coefficient (in %)	59.6	65.4	62.8	67.7	62.5	70.9	56.9	**85.1**
	Avg. F-measure (in %)	48.9	57.5	49.3	81.1	57.7	45.2	32.6	**85.5**

7 Conclusions

In this work, attempts have been made to detect the moving underwater object when the camera is moving in the same environment. Our proposed scheme therefore estimated the camera model parameters which are subsequently used for object detection. The object detection phase is based on background modeling and its learning. Background modeling and model learning are carried in SKDE feature space to deal with the existing complexity of the background. We have computed the SKDE of frames which are used as the feature frames. Background modeling and model learning is pixel based approach where each pixel of SKDE frame is modeled by its histograms. In the learning phase, our Correntropy based similarity measure determines the proximity of the histogram of a pixel of the incoming frame with that of the model histograms. It is observed that the proposed modeling and model learning strategy could take care of the complex background of the underwater environment. The camera model parameters are estimated which are used to transform the input frame before presented for learning. Therefore, it has been observed that the learning and classification depends upon the accuracy of the estimates of the camera model parameters. It has also been observed that the accuracy of the estimated parameters depends on the proper choice of features. The proposed pipeling approach with improved Harris corner detection method resulted in good estimates of the parameters, thereby resulting in proper transformation of frames for learning. The accuracy of the parameters were based on minimizing the calibration error and it has been found that the estimated intrinsic parameters are close to the avail-

able camera parameters. The proposed scheme could be successfully tested on six underwater datasets and the results obtained are superior to the existing ones both qualitatively and quantitatively. Thus, the proposed scheme may be used in machine vision system for underwater object detection.

Compliance with Ethical Standards

Ethical approval
Not applicable in this research.
Funding details
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
Conflict of interest
There is no conflict of interest in this research.
Informed Consent
Not applicable in this research.

Authorship contributions

Susmita Panda(1^{st} author): Involved in formulation and development of algorithm, validating the algorithms with different data sets and manuscript preparation.
Pradipta Kumar Nanda(2^{nd} author): Conceptualization of the problem, problem formulation, validating results and manuscript.

References

1. Ahmed, S., Khan, M.F.R., Labib, M.F.A., Chowdhury, A.E.: An observation of vision based underwater object detection and tracking. In: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), pp. 117–122. IEEE (2020)
2. Anon: Whalesharks in philippines southern leyte. http://www.dvdunderwater.com (2010)
3. Anon: Blainville's beaked whales. http://www.youtube.com/watch?v=SGe93WbVZJM (2015)
4. Anon: Montserrat south/nautilus live. http://www.youtube.com/watch?v=sPMziH2mmKQ (2015)
5. Anon: Creepy chimaera/nautilus live. http://www.youtube.com/watch?v=jvArUKv9DvA (2016)
6. Bloisi, D.D., Pennisi, A., Iocchi, L.: Background modeling in the maritime domain. Machine Vision and Applications **25**(5), 1257–1269 (2014). DOI 10.1007/s00138-013-0554-5
7. Bouwmans, T.: Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review **11-12**, 31 – 66 (2014). DOI https://doi.org/10.1016/j.cosrev.2014.04.001
8. Chen, Z., Wang, R., Zhang, Z., Wang, H., Xu, L.: Background foreground interaction for moving object detection in dynamic scenes. Information Sciences **483**, 65 – 81 (2019). DOI https://doi.org/10.1016/j.ins.2018.12.047
9. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams. IEEE transactions on pattern analysis and machine intelligence **25**(10), 1337–1342 (2003)
10. Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B.: Neural network approach to background modeling for video object segmentation. IEEE Transactions on Neural Networks **18**(6), 1614–1627 (2007)
11. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE **90**(7), 1151–1163 (2002). DOI 10.1109/JPROC.2002.801448
12. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: European conference on computer vision, pp. 751–767. Springer (2000)
13. Ge, W., Guo, Z., Dong, Y., Chen, Y.: Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation. Pattern Recognition **59**, 112 – 125 (2016). DOI https://doi.org/10.1016/j.patcog.2016.01.031
14. Giordano, D., Palazzo, S., Spampinato, C.: Kernel density estimation using joint spatial-color-depth data for background modeling. In: 2014 22nd International Conference on Pattern Recognition, pp. 4388–4393 (2014). DOI 10.1109/ICPR.2014.751
15. Goyal, K., Singhai, J.: Texture-based self-adaptive moving object detection technique for complex scenes. Computers & Electrical Engineering **70**, 275–283 (2018)
16. Hao, J., Li, C., Kim, Z., Xiong, Z.: Spatio-temporal traffic scene modeling for object motion detection. IEEE Transactions on Intelligent Transportation Systems **14**(1), 295–302 (2013). DOI 10.1109/TITS.2012.2212432
17. Heikkila, M., Pietikainen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4), 657–662 (2006). DOI 10.1109/TPAMI.2006.68
18. H.Wang, D.S.: Background subtraction based on a robust consensus method. In: 18th International Conference on Pattern Recognition (ICPR'06), vol. 1, pp. 223–226 (2006). DOI 10.1109/ICPR.2006.312
19. Jaffe, J.S.: Underwater optical imaging: The past, the present, and the prospects. IEEE Journal of Oceanic Engineering **40**(3), 683–700 (2015). DOI 10.1109/JOE.2014.2350751
20. Kakizawa, Y.: Nonparametric density estimation for nonnegative data, using symmetrical-based inverse and reciprocal inverse gaussian kernels through dual transformation. Journal of Statistical Planning and Inference **193**, 117 – 135 (2018). DOI https://doi.org/10.1016/j.jspi.2017.08.008
21. Liu, H., Dai, J., Wang, R., Zheng, H., Zheng, B.: Combining background subtraction and three-frame difference to detect moving object from underwater video. In: OCEANS 2016 - Shanghai, pp. 1–5 (2016). DOI 10.1109/OCEANSAP.2016.7485613
22. Liu, W., Pokharel, P.P., Principe, J.C.: Correntropy: A localized similarity measure. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 4919–4924 (2006). DOI 10.1109/IJCNN.2006.247192

23. Liyuan Li Weimin Huang, I.Y.H.G.Q.T.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Transactions on Image Processing **13**(11), 1459–1472 (2004). DOI 10.1109/TIP.2004.836169
24. Maity, S., Chakrabarti, A., Bhattacharjee, D.: Background modeling and foreground extraction in video data using spatio-temporal region persistence features. Computers & Electrical Engineering **81**, 106520 (2020)
25. Messelodi, S., Modena, C.M., Segata, N., Zanin, M.: A kalman filter based background updating algorithm robust to sharp illumination changes. In: International Conference on Image Analysis and Processing, pp. 163–170. Springer (2005)
26. Álvarez Meza, A., Molina-Giraldo, S., Castellanos-Dominguez, G.: Background modeling using object-based selective updating and correntropy adaptation. Image and Vision Computing **45**, 22 – 36 (2016). DOI https://doi.org/10.1016/j.imavis.2015.11.006
27. Miao, X., Rahimi, A., Rao, R.P.: Complementary kernel density estimation. Pattern Recognition Letters **33**(10), 1381 – 1387 (2012). DOI https://doi.org/10.1016/j.patrec.2012.02.019
28. Migdal, J., Grimson, W.E.L.: Background subtraction using markov thresholds. In: Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on, vol. 2, pp. 58–65 (2005). DOI 10.1109/ACVMOT.2005.33
29. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, pp. II–II. Ieee (2004)
30. Panda, S., Nanda, P.K.: Segmentation of underwater video objects using extended markov random field model. In: 2015 IEEE Underwater Technology (UT), pp. 1–6. IEEE (2015)
31. Panda, S., Nanda, P.K.: Mrf model-based estimation of camera parameters and detection of underwater moving objects. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) **14**(4), 1–29 (2020)
32. Paragios, N., Tziritas, G.: Adaptive detection and localization of moving objects in image sequences. Signal Processing: Image Communication **14**(4), 277–296 (1999)
33. Peng, J., WeiDong, J.: Statistical background subtraction with adaptive threshold. In: 2012 5th International Congress on Image and Signal Processing, pp. 123–127 (2012). DOI 10.1109/CISP.2012.6469969
34. Piccardi, M.: Background subtraction techniques: a review. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), vol. 4, pp. 3099–3104 vol.4 (2004). DOI 10.1109/ICSMC.2004.1400815
35. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061 pp. 37–63 (2011)
36. Prabowo, M.R., Hudayani, N., Purwiyanti, S., Sulistiyanti, S.R., Setyawan, F.X.A.: A moving objects detection in underwater video using subtraction of the background model. In: 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 1–4 (2017). DOI 10.1109/EECSI.2017.8239148
37. Qiao, Y., Tang, Y., Li, J.: Improved harris sub-pixel corner detection algorithm for chessboard image. In: Measurement, Information and Control (ICMIC), 2013 International Conference on, vol. 2, pp. 1408–1411. IEEE (2013)
38. Qiao, Y., Xi, W.: A kernel density estimation model for moving object detection. In: Advanced Multimedia and Ubiquitous Engineering, pp. 386–392. Springer Singapore, Singapore (2017)
39. Rashid, M., Thomas, V.: A background foreground competitive model for background subtraction in dynamic background. Procedia Technology **25**, 536 – 543 (2016). DOI https://doi.org/10.1016/j.protcy.2016.08.142
40. Reddy, V., Sanderson, C., Lovell, B.C.: Robust foreground object segmentation via adaptive region-based background modelling. In: 2010 20th International Conference on Pattern Recognition, pp. 3939–3942 (2010). DOI 10.1109/ICPR.2010.958
41. Sen-ching S. Cheung, C.K.: Robust techniques for background subtraction in urban traffic video (2004). DOI 10.1117/12.526886
42. Sajid, H., Cheung, S.C.S., Jacobs, N.: Motion and appearance based background subtraction for freely moving cameras. Signal Processing: Image Communication **75**, 11–21 (2019)
43. Santamaria, I., Pokharel, P.P., Principe, J.C.: Generalized correlation function: definition, properties, and application to blind equalization. IEEE Transactions on Signal Processing **54**(6), 2187–2197 (2006). DOI 10.1109/TSP.2006.872524
44. Singh, A., Principe, J.C.: Using correntropy as a cost function in linear adaptive filters. In: 2009 International Joint Conference on Neural Networks, pp. 2950–2955 (2009). DOI 10.1109/IJCNN.2009.5178823
45. Singla, N.: Motion detection based on frame difference method. International Journal of Information & Computation Technology **4**(15), 1559–1565 (2014)
46. Spampinato, C., Palazzo, S., Kavasidis, I.: A texton-based kernel density estimation approach for background modeling under extreme conditions. Computer Vision and Image Understanding **122**, 74 – 83 (2014). DOI https://doi.org/10.1016/j.cviu.2013.12.003
47. Srikanth Vasamsetti Supriya Setia, N.M.H.K.S..G.B.: Automatic underwater moving object detection using multi-feature integration framework in complex backgrounds. IET Computer Vision (2018)
48. Srividhya, K., Ramya, M.M.: Accurate object recognition in the underwater images using learning algorithms and texture features. Multimedia Tools and Applications **76**(24), 25679–25695 (2017). DOI 10.1007/s11042-017-4459-6
49. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 2, p. 252 Vol. 2 (1999). DOI 10.1109/CVPR.1999.784637
50. Stolkin, R., Greig, A., Hodgetts, M., Gilby, J.: An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility. Image and Vision Computing **26**(4), 480–495 (2008)
51. Systems, G.D.M.: Bluefin sandshark micro-auvs conduct simulated missions with a bluefin-21 uuv. https://www.youtube.com/watch?v=qIKmfOWcpzk (2016)
52. Szolgay, D., Benois-Pineau, J., Megret, R., Gaestel, Y., Dartigues, J.F.: Detection of moving foreground objects in videos with strong camera motion. Pattern Analysis and Applications **14**(3), 311–328 (2011). DOI 10.1007/s10044-011-0221-2

53. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proceedings of the seventh IEEE international conference on computer vision, vol. 1, pp. 255–261. IEEE (1999)
54. Trumpet, C.: Giant sea turtles . amazing coral reef fish. https://www.youtube.com/watch?v=riFyKUyGb4k (2017)
55. Vemulapalli, R., Aravind, R.: Spatio-temporal nonparametric background modeling and subtraction. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pp. 1145–1152 (2009). DOI 10.1109/ICCVW.2009.5457574
56. Wintenby, J., Svensson, D.: Adaptive kernel background intensity estimation based on local 2d orientation. In: 2015 18th International Conference on Information Fusion (Fusion), pp. 1786–1793 (2015)
57. Xu, Y., Dong, J., Zhang, B., Xu, D.: Background modeling methods in video analysis: A review and comparative evaluation. CAAI Transactions on Intelligence Technology pp. 43 – 60 (2016). DOI https://doi.org/10.1016/j.trit.2016.03.005
58. Yang, Y., Liu, Y.: An improved background and foreground modeling using kernel density estimation in moving object detection. In: Proceedings of 2011 International Conference on Computer Science and Network Technology, vol. 2, pp. 1050–1054 (2011). DOI 10.1109/ICCSNT.2011.6182141
59. Zamalieva, D., Yilmaz, A.: Background subtraction for the moving camera: A geometric approach. Computer Vision and Image Understanding **127**, 73–85 (2014)
60. Zhang, Y., Wang, X., Qu, B.: Three-frame difference algorithm research based on mathematical morphology. Procedia Engineering **29**, 2705–2709 (2012)
61. Zhao, S., Chen, B., Príncipe, J.C.: Kernel adaptive filtering with maximum correntropy criterion. In: The 2011 International Joint Conference on Neural Networks, pp. 2012–2017 (2011). DOI 10.1109/IJCNN.2011.6033473
62. Zhong, Z., Wen, J., Zhang, B., Xu, Y.: A general moving detection method using dual-target nonparametric background model. Knowledge-Based Systems **164**, 85–95 (2019)
63. Zhou, F., Cui, Y., Peng, B., Wang, Y.: A novel optimization method of camera parameters used for vision measurement. Optics & Laser Technology **44**(6), 1840–1849 (2012)
64. Zou, K.H., Warfield, S.K., Bharatha, A., Tempany, C.M., Kaus, M.R., Haker, S.J., Wells III, W.M., Jolesz, F.A., Kikinis, R.: Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. Academic radiology **11**(2), 178–189 (2004)

Figures

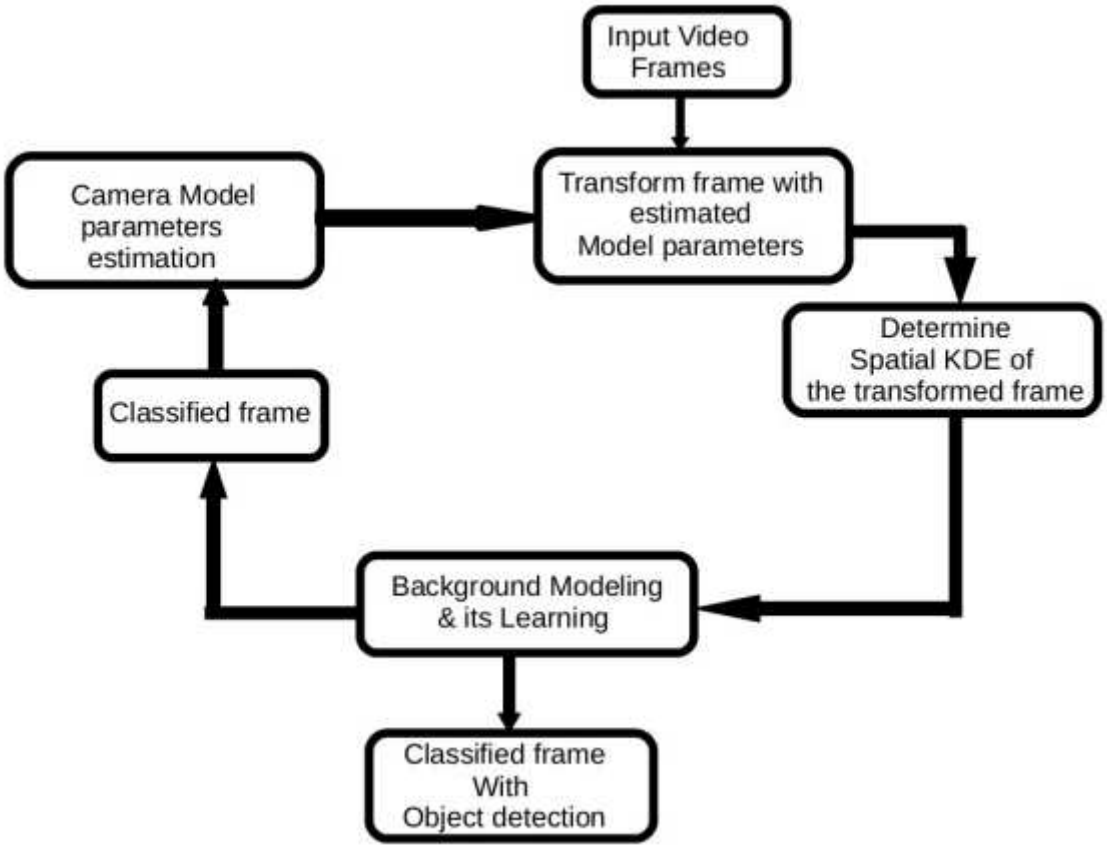

Figure 1

Please see the Manuscript PDF file for the complete figure caption

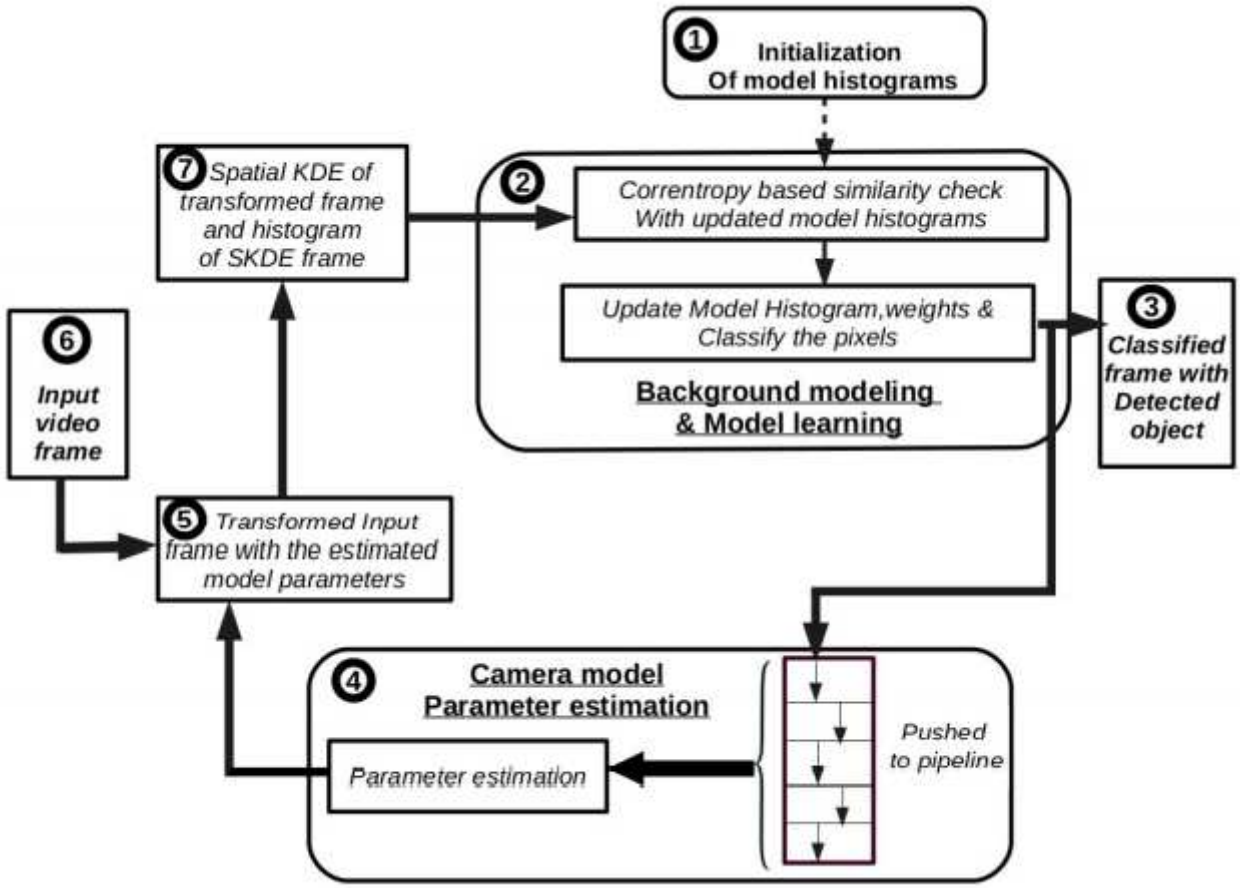

Figure 2

Please see the Manuscript PDF file for the complete figure caption

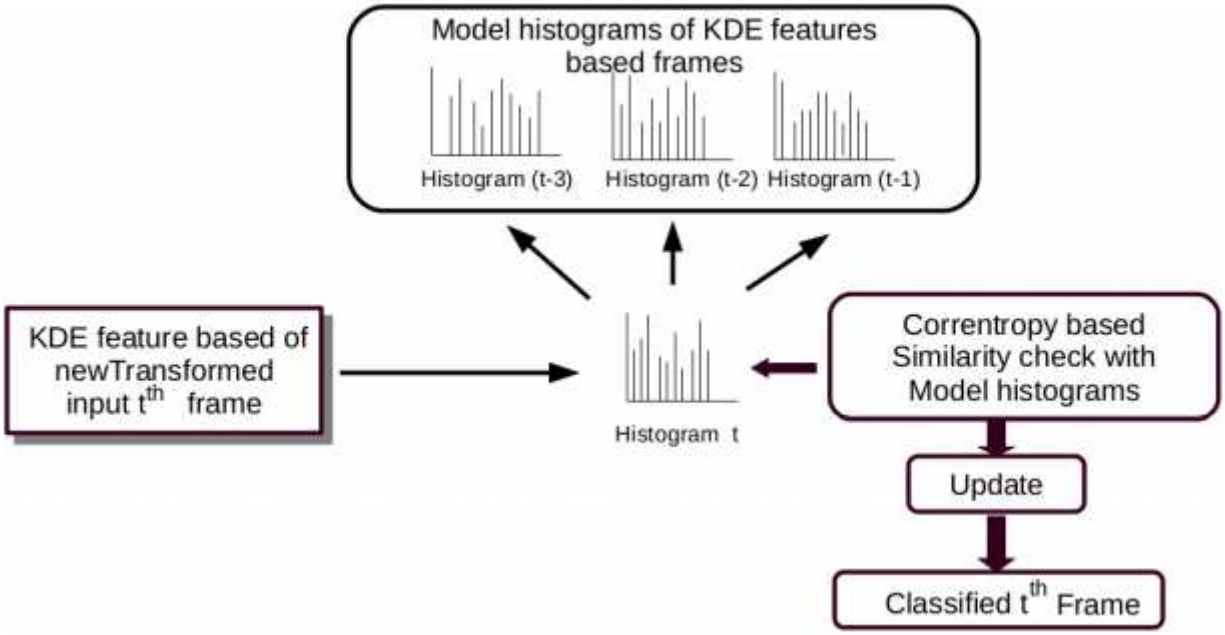

Figure 3

Please see the Manuscript PDF file for the complete figure caption

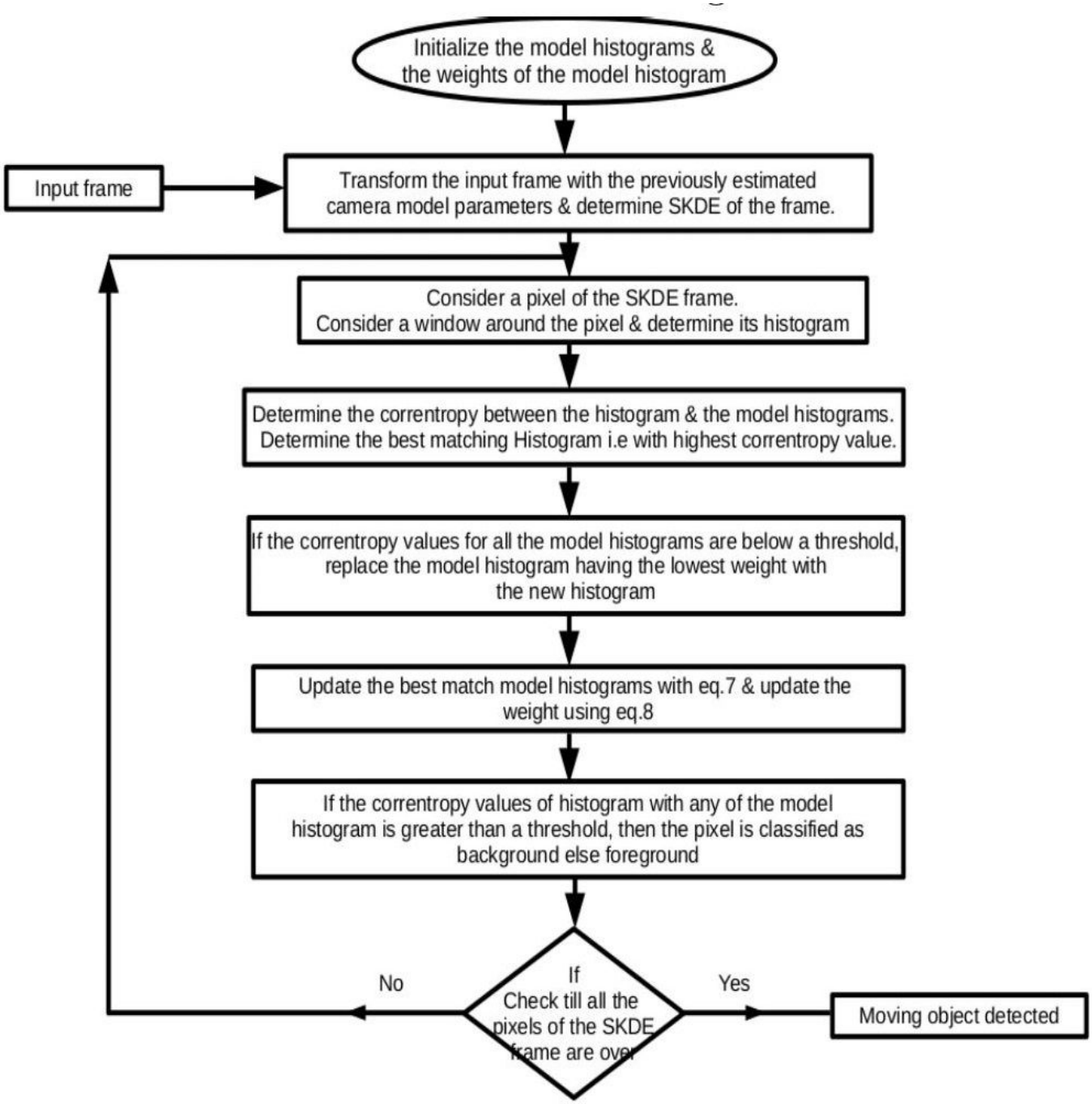

Figure 4

Please see the Manuscript PDF file for the complete figure caption

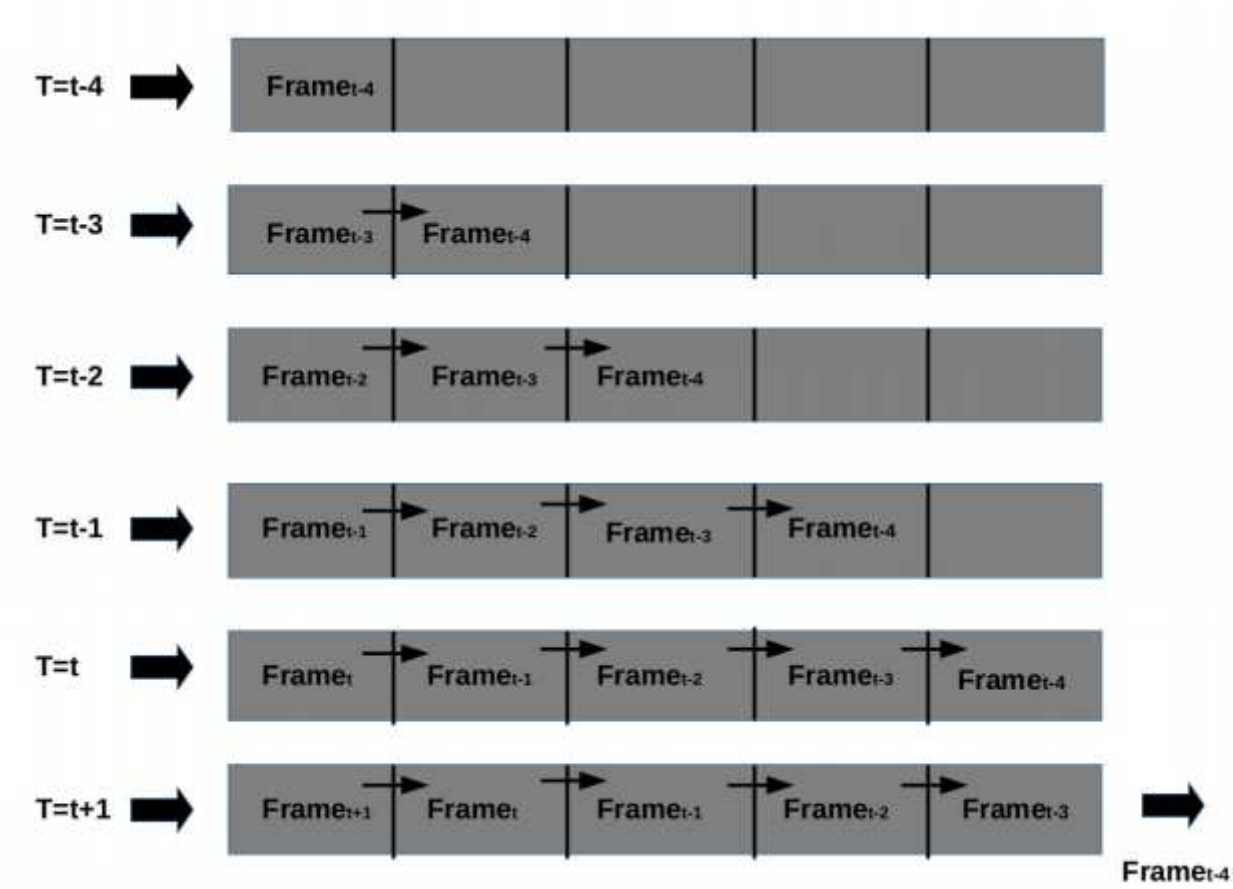

Figure 5

Please see the Manuscript PDF file for the complete figure caption

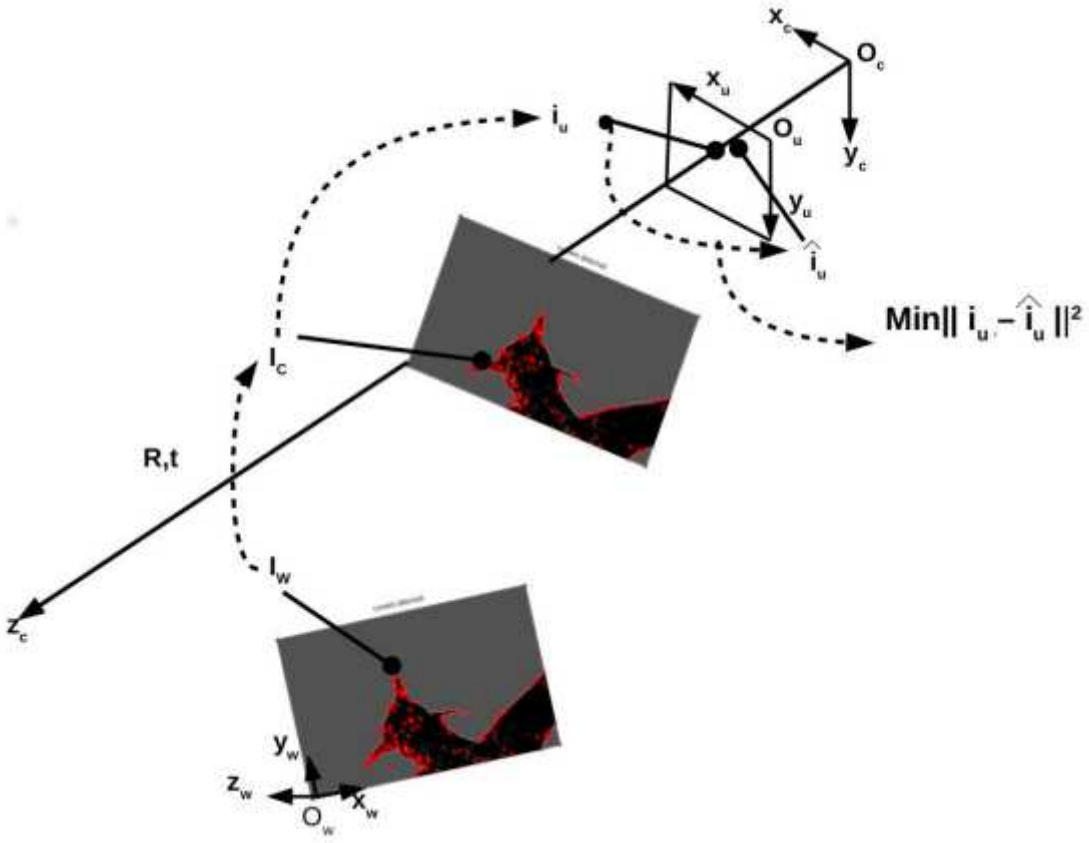

Figure 6

Please see the Manuscript PDF file for the complete figure caption

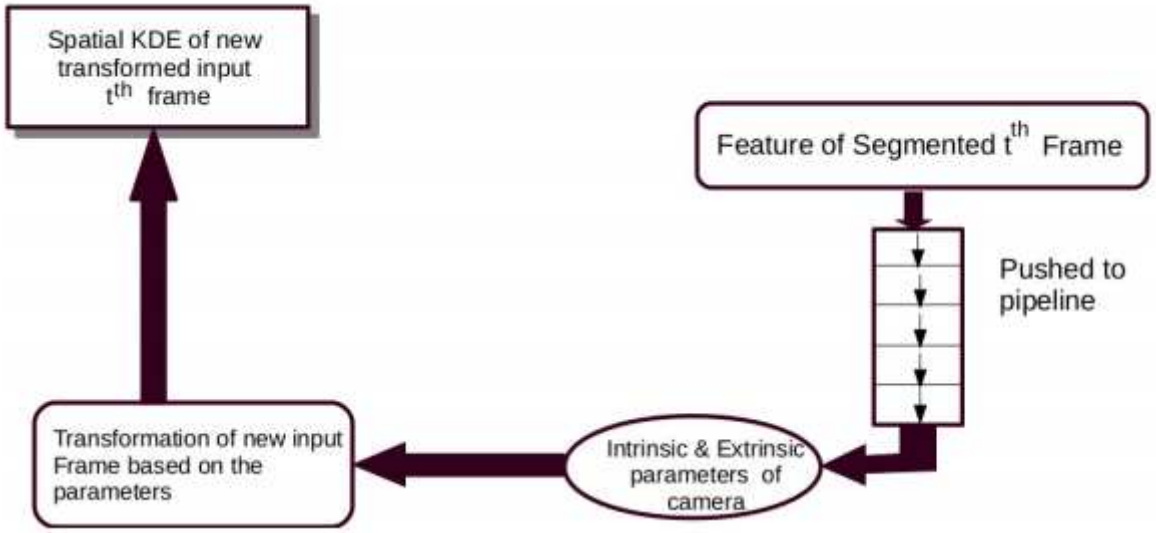

Figure 7

Please see the Manuscript PDF file for the complete figure caption

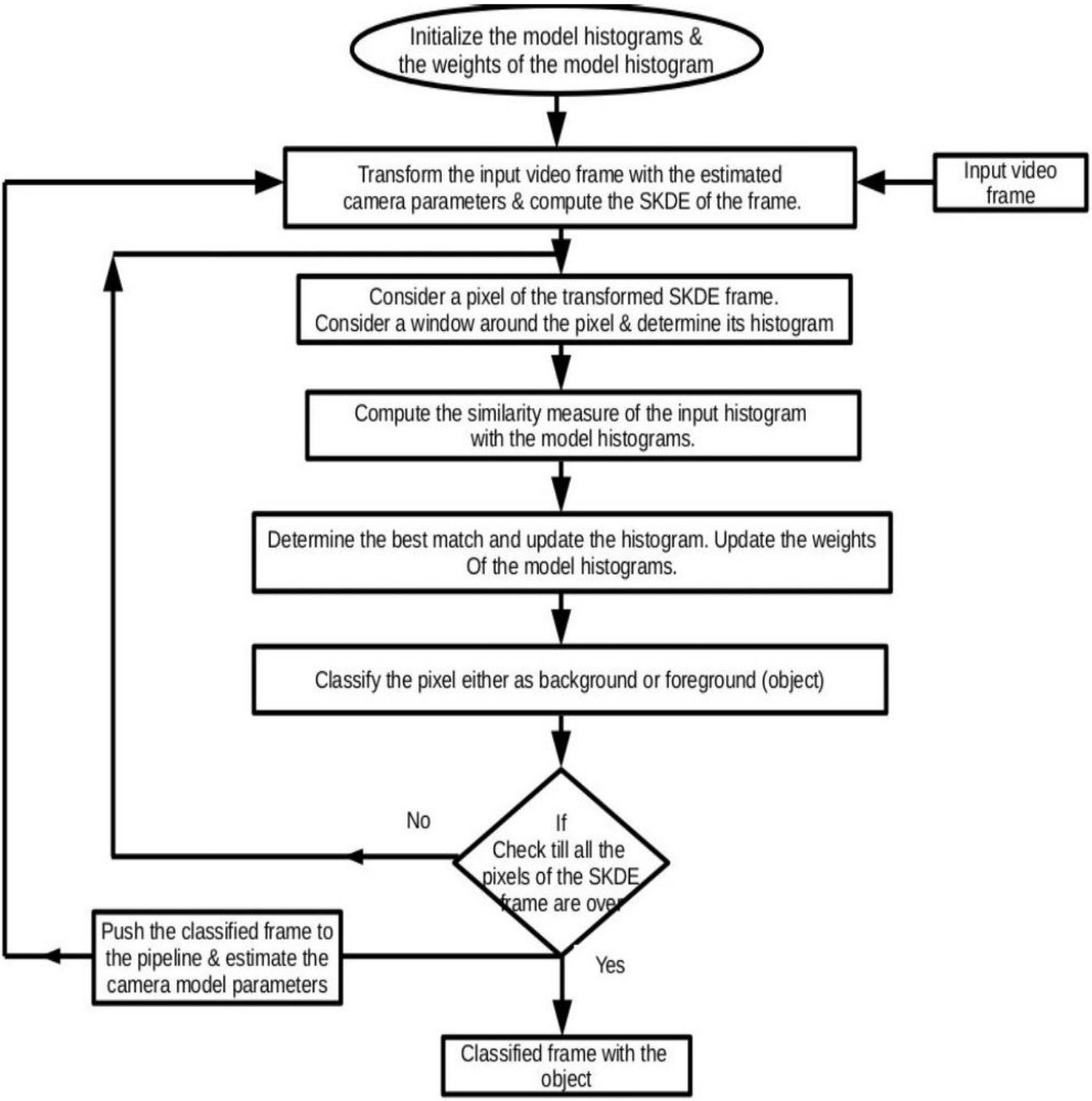

Figure 8

Please see the Manuscript PDF file for the complete figure caption

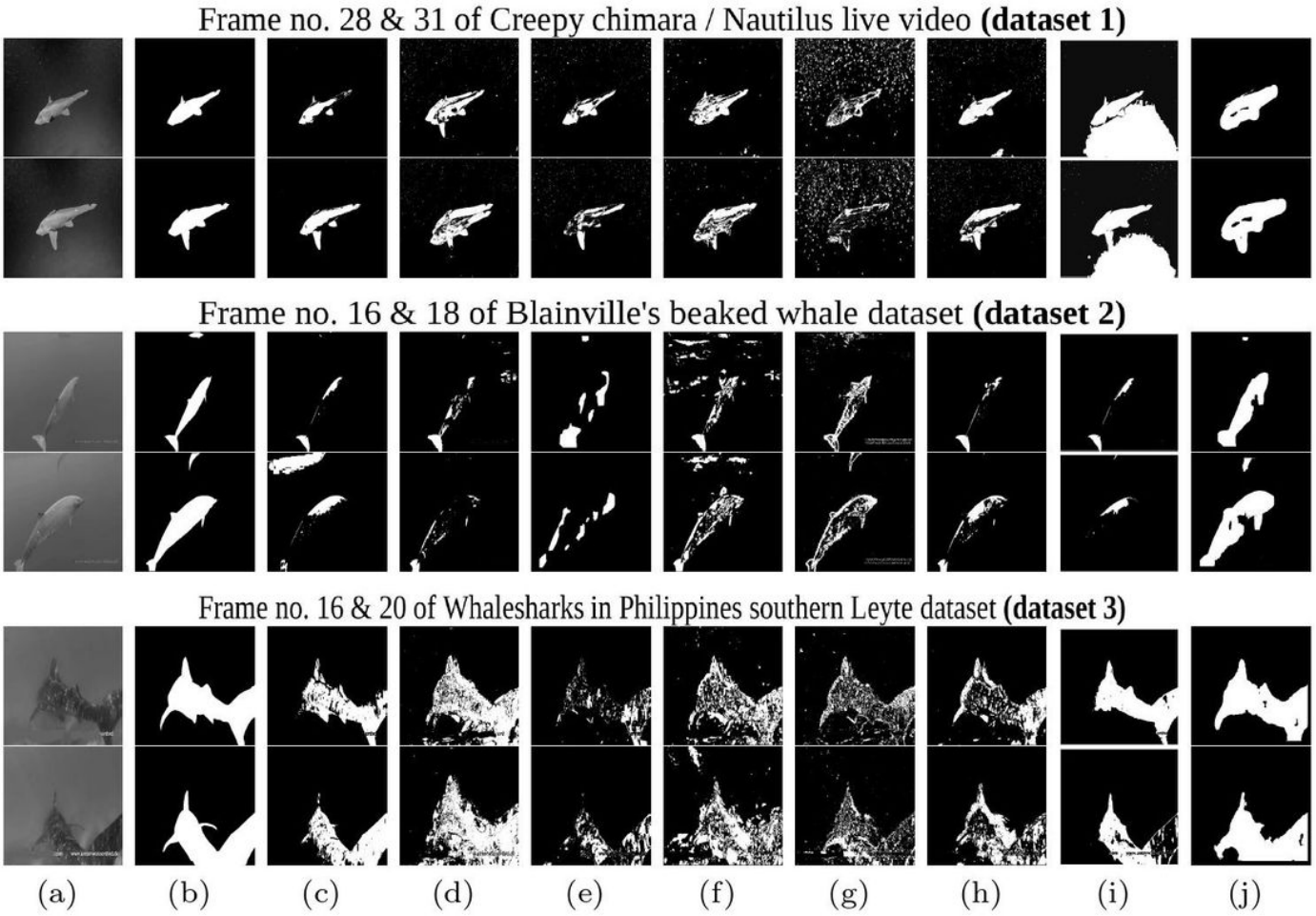

Figure 9

Please see the Manuscript PDF file for the complete figure caption

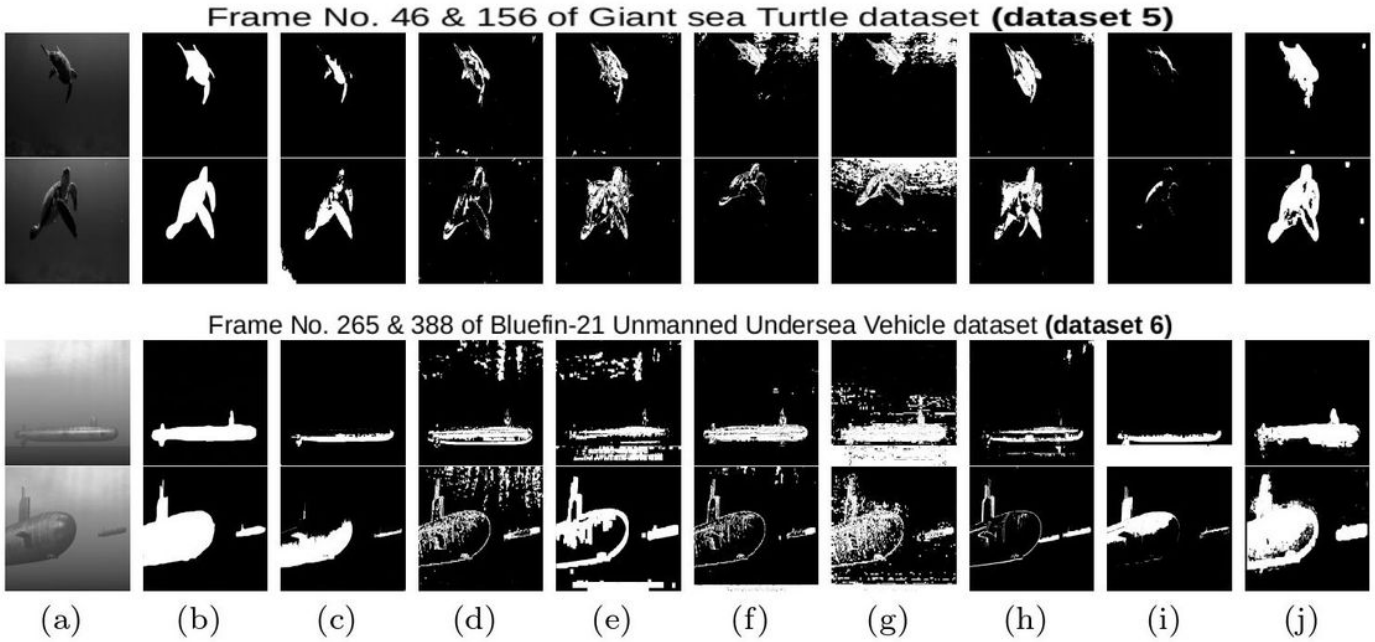

Figure 10

Please see the Manuscript PDF file for the complete figure caption

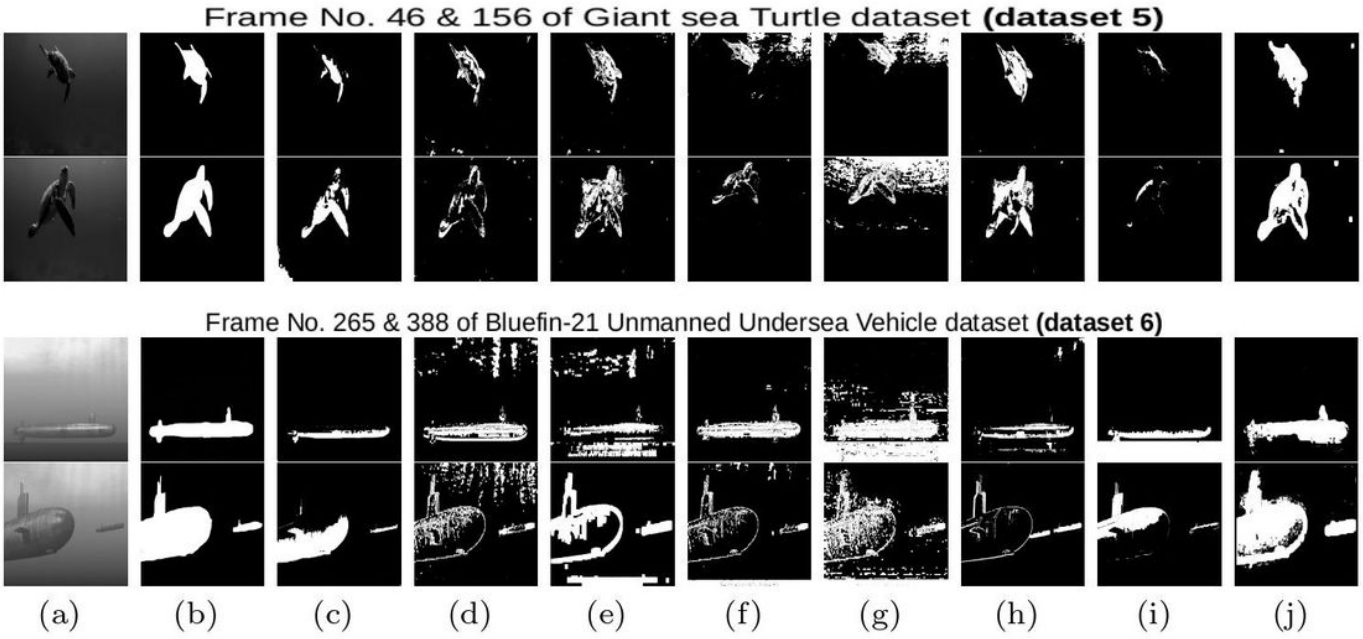

Figure 11

Please see the Manuscript PDF file for the complete figure caption