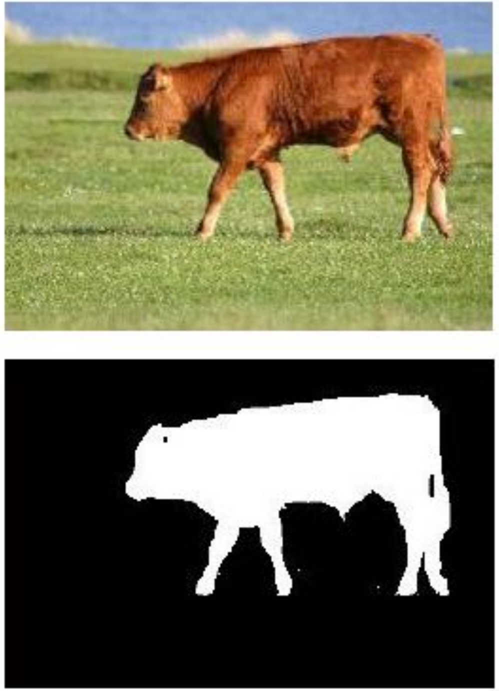
CMSC426
Project 3: Rotobrush

Zhiyuan Hua

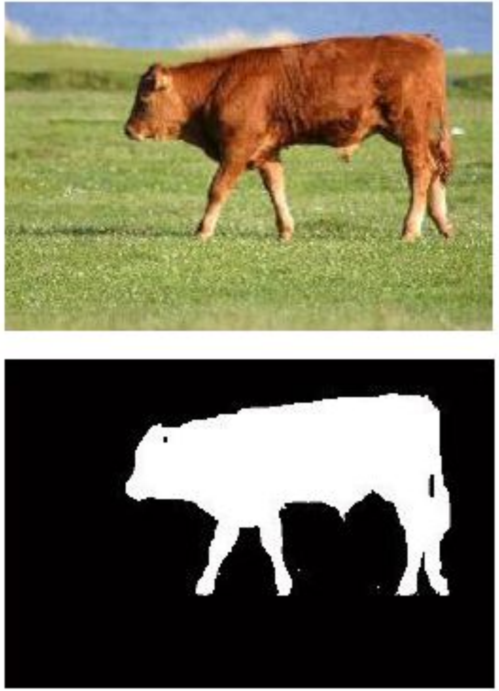
What is segmentation?

- Given a point/pixel $x_{i,j}$ in the image
- $x \in$ object 1 or 2 or 3 or .. We call this semantic/instance segmentation
- $\boldsymbol{x \in \mathcal{F}\ or\ \mathcal{B}}$, we call this foreground segmentation/background subtraction



Why do we need segmentation?

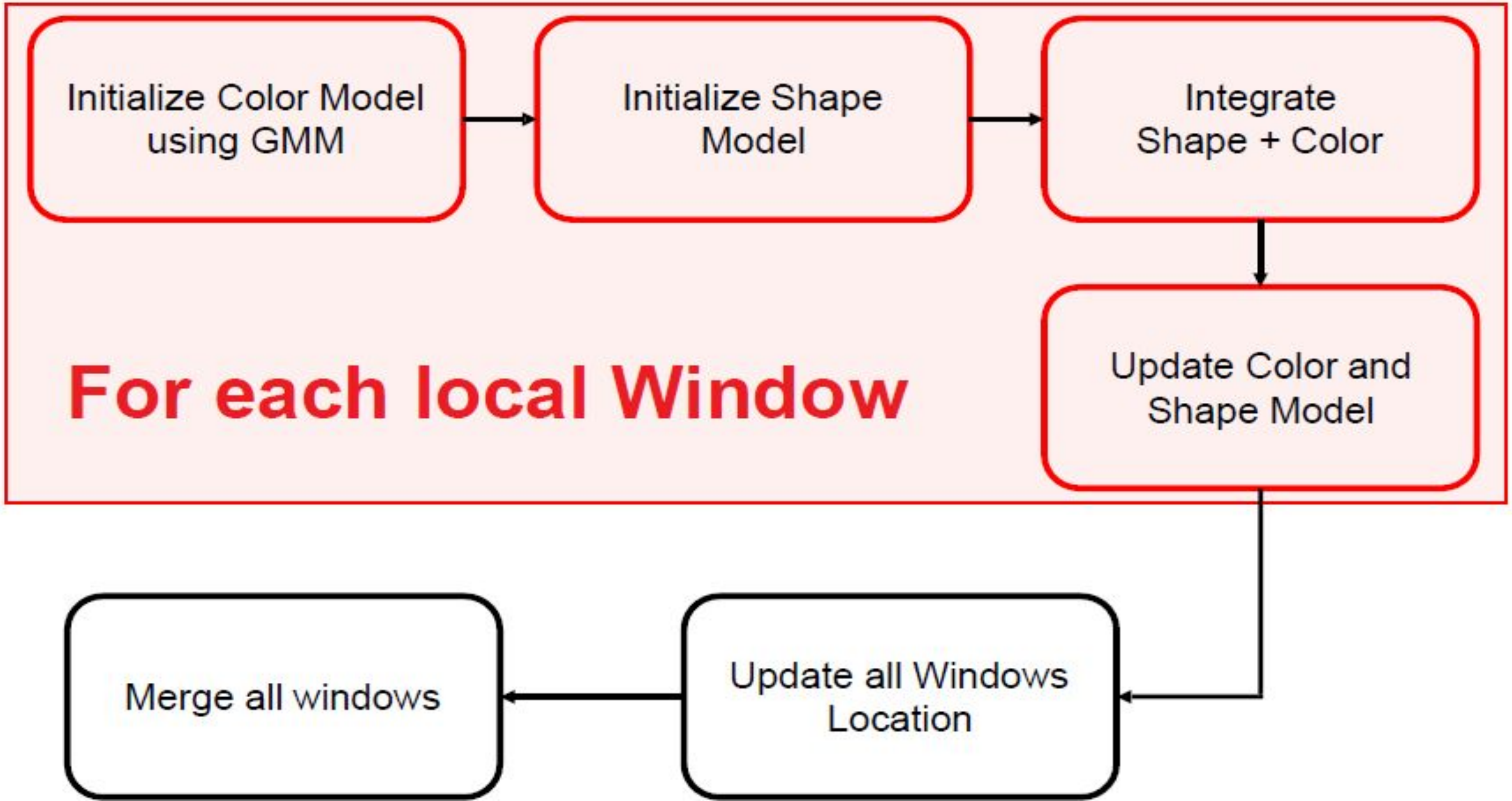
- Medical Imaging
- Face Detection
- Pedestrian Detection
- Traffic sign detection
- For recognition tasks
- Video Surveillance
- Action localization
- And much, much more…



Video:
https://www.youtube.com/watch?v=XSXRcXrPyIM

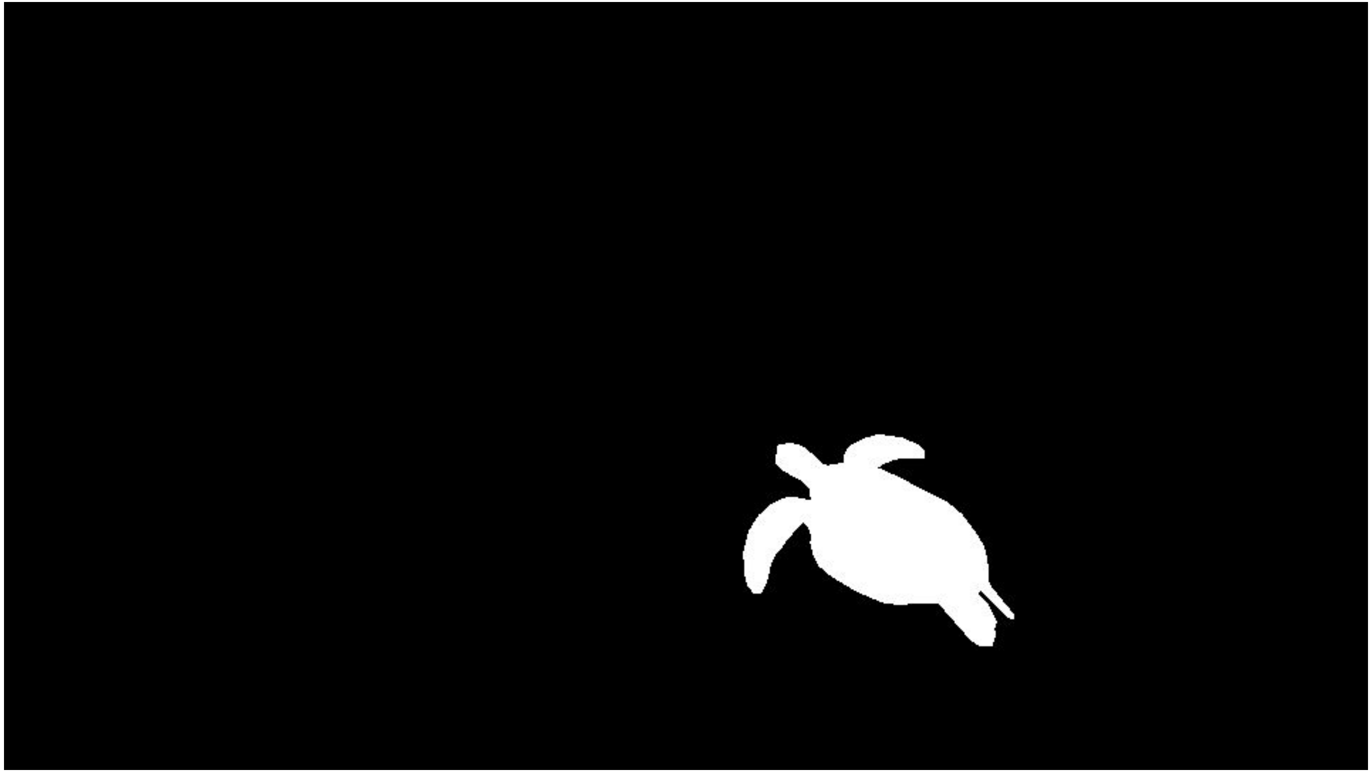
Paper:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.450.5794&rep=rep1&type=pdf

An Overview

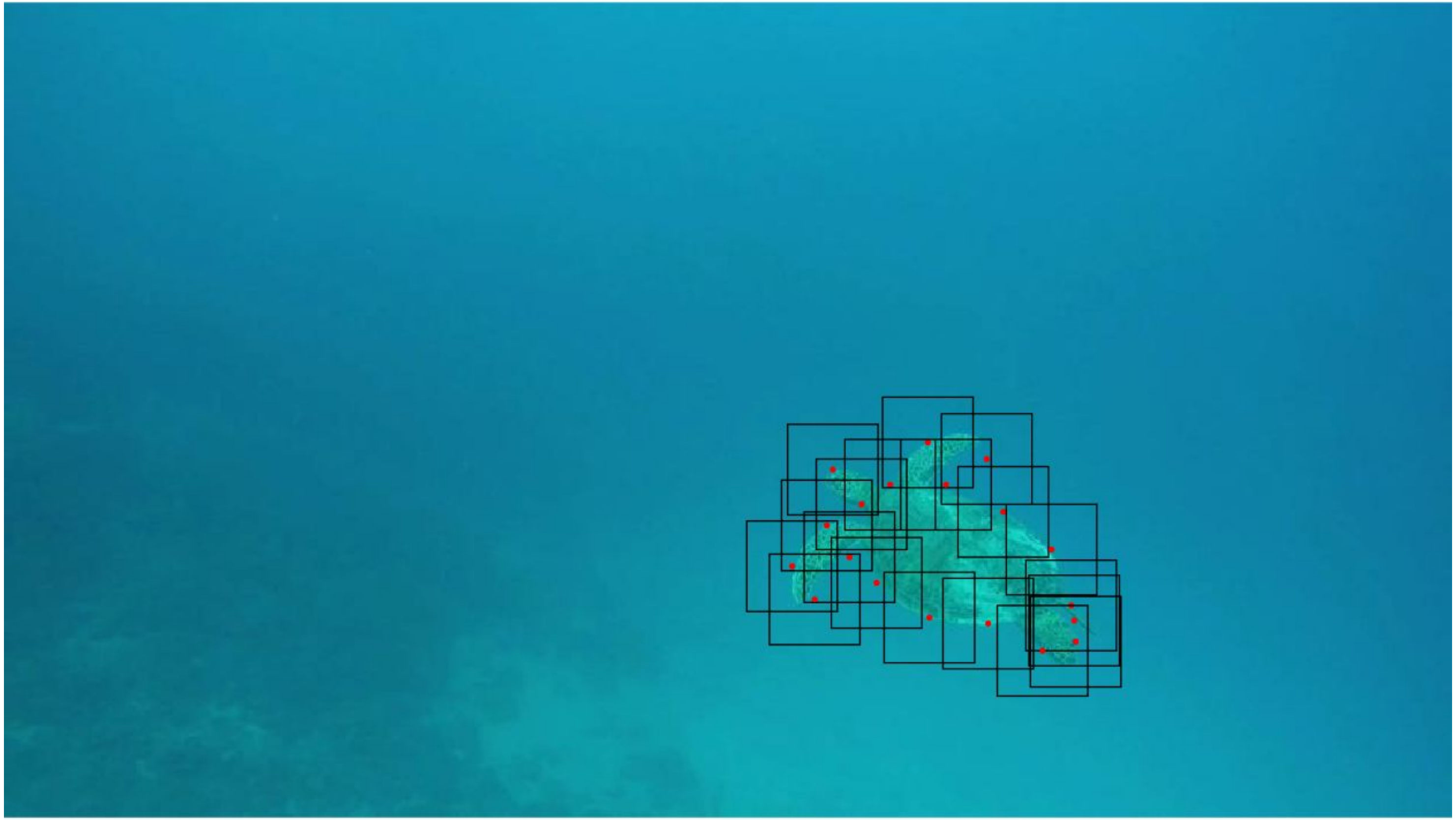


Initialize Color Model using GMM
Initialize Shape Model
Integrate Shape + Color
Update Color and Shape Model
For each local Window
Update all Windows Location
Merge all windows

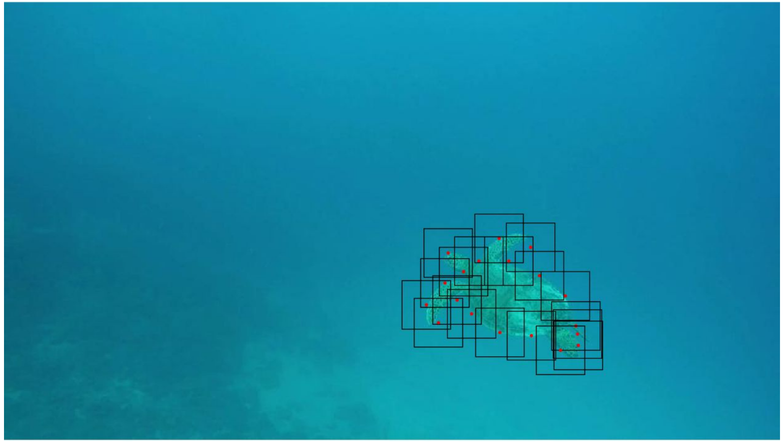
Create Mask using Roipoly for only the First frame



Create Local Window (initLocalWindows.m)

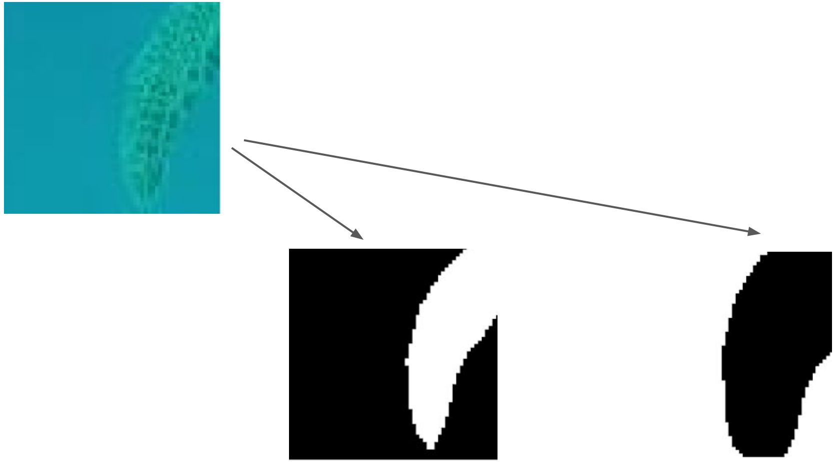


Create Local Window (initLocalWindows.m)



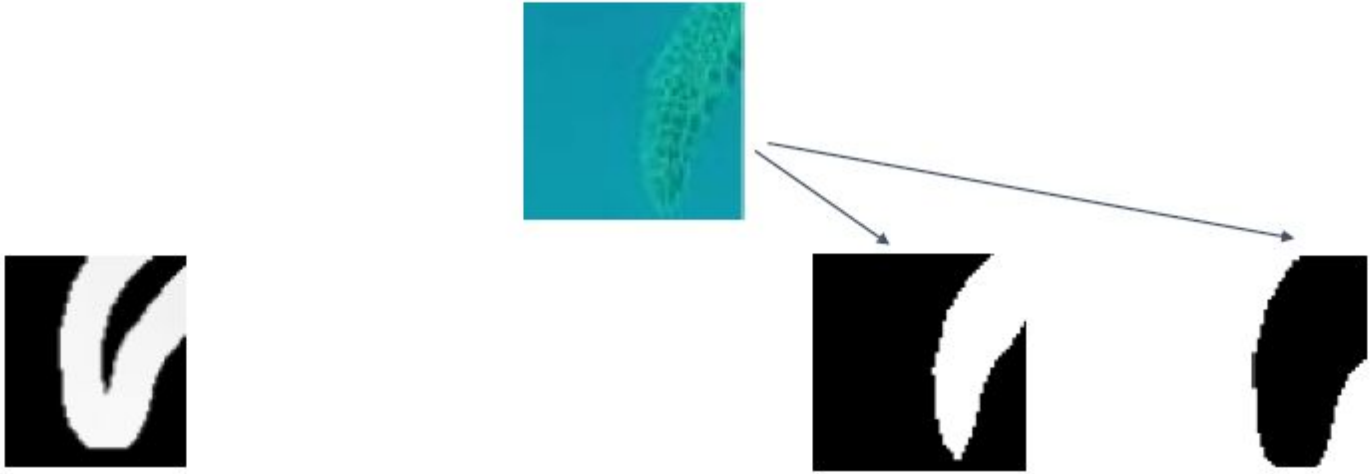
- Windows' centers stick to object boundary
- Window density and size is empirically chosen
- Window's size should remain the same within each set, but could be different with different sets.
- Window size is usually 30x30 to 80x80
- Too many windows/Too large windows will make your code SLOW.
- The boundary should be fully covered by overlapping windows

GMM Color model (initColorModels.m)



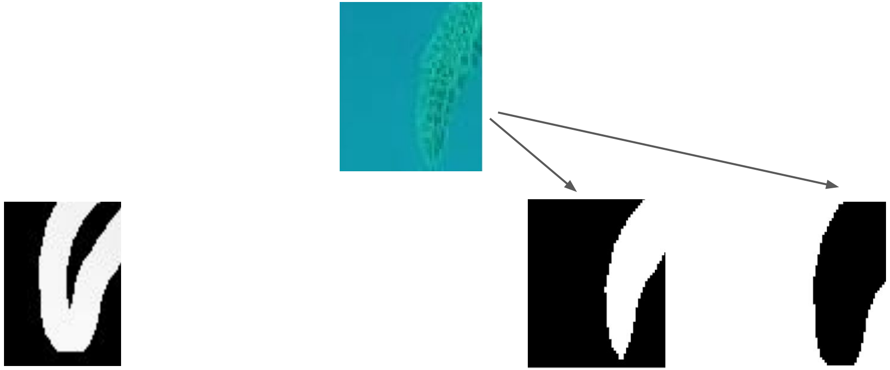
$$p_c(x) = p_c(x|\mathcal{F}) \ / \ (p_c(x|\mathcal{F}) + p_c(x|\mathcal{B}))$$

GMM Color model (initColorModels.m)



$$p_c(x) = p_c(x|\mathcal{F}) \;/\; (p_c(x|\boxed{\mathcal{F}}) + p_c(x|\boxed{\mathcal{B}}))$$

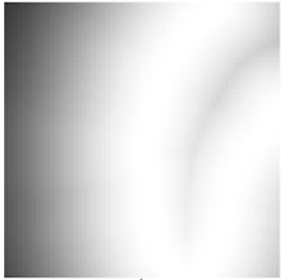
GMM Color model (initColorModels.m)



$$p_c(x) = p_c(x|\mathcal{F}) \;/\; (p_c(x|\mathcal{F}) + p_c(x|\mathcal{B}))$$

- GMM model is a window-level local classifier.
- F is foreground, B is Backgorund
- This GMM is the same as our project 1
- You are ALLOWED to use built in functions: fitgmdist, and gmdistribution.
- Two GMM are built for F and B separately.
- Lab color space
- 5 pixels (threshold) from the segmented boundary to train

Color Confidence model (initColorModels.m)



$$\omega_c(x) = exp(-d^2(x)\ /\ \sigma_c^2$$

$$f_c = 1 - \frac{\int_{W_k} |L^t(x) - p_c(x)| \cdot \omega_c(x)dx}{\int_{W_k} \omega_c(x)dx}$$

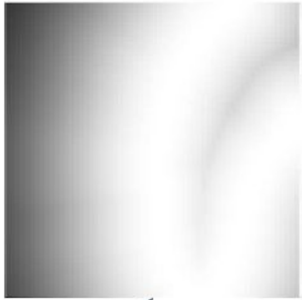
Color Confidence model (initColorModels.m)

L_t is the known segmentation label

p_c is foreground probability

$d(x)$ is the spatial distance between x and the foreground boundary

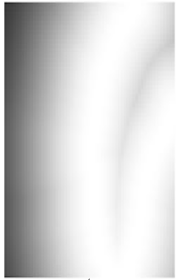
σ_c is fixed as half of the window size



$$\omega_c(x) = exp(-\boxed{d^2(x)} / \sigma_c^2$$

$$f_c = 1 - \frac{\int_{W_k} \boxed{|L^t(x) - p_c(x)|} \cdot \boxed{\omega_c(x)} dx}{\int_{W_k} \omega_c(x) dx}$$

Color Confidence model (initColorModels.m)

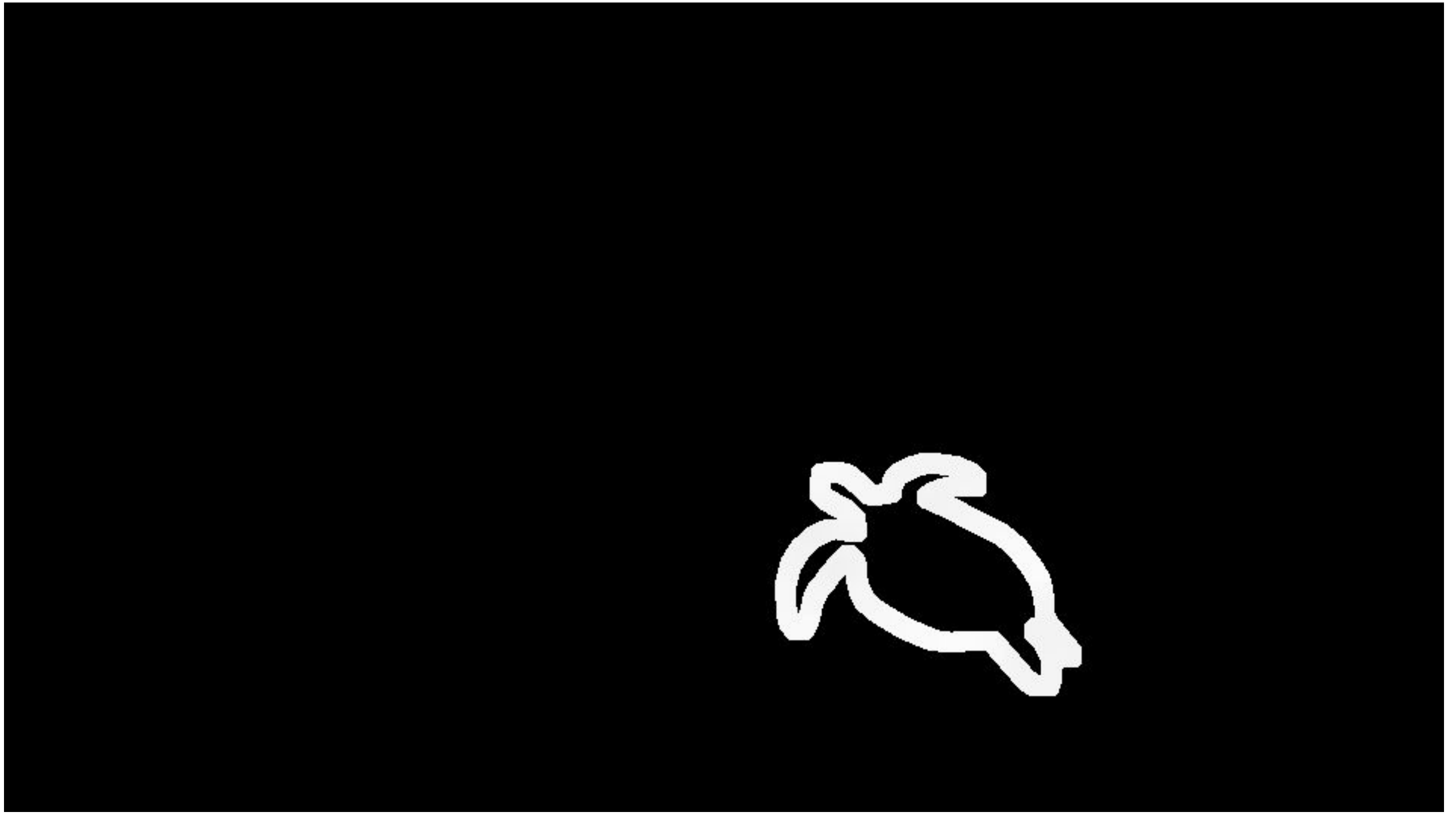


$$\omega_c(x) = exp(-d^2(x) / \sigma_c^2$$

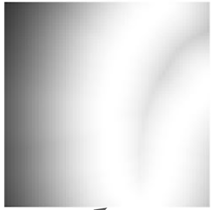
$$f_c = 1 - \frac{\int_{W_k} \left| L^t(x) - p_c(x) \right| \cdot \omega_c(x) dx}{\int_{W_k} \omega_c(x) dx}$$

- ***fc*** is local color model's confidence
- Describes how *separable* the local foreground is against the local background within each window
- Weighing function ***ωc(x)***
- ***d(x)*** is the spatial distance between x and the foreground boundary. Built-in bwdist()
- ***ωc(x)*** is higher when ***x*** is closer to the boundary
- ***σc*** here is fixed as half window size

Color model (initColorModels.m)



Shape model (initShapeConfidences.m)



$$f_s(x) = 1 - exp(-d^2(x) \;/\; \sigma_s^2)$$

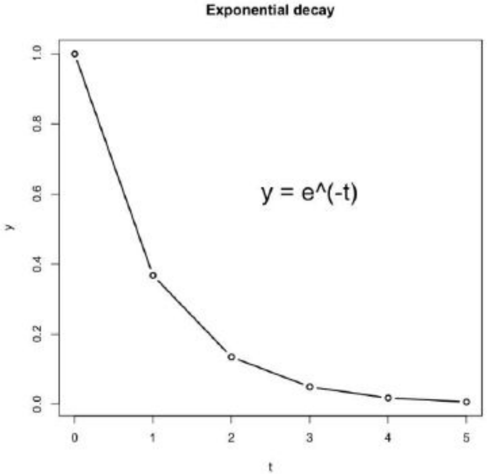
Shape model (initShapeConfidences.m)

$$f_s(x) = 1 - exp\left(\boxed{-d^2(x)} \,/\, \boxed{\sigma_s^2}\right)$$

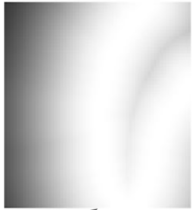
$\boldsymbol{f_s}$ is shape confidence mask

$\boldsymbol{d(x)}$ is the spatial distance between x and the foreground boundary

$\boldsymbol{\sigma_s}$ is a parameter which depends on $\boldsymbol{f_c}$



Shape model (initShapeConfidences.m)

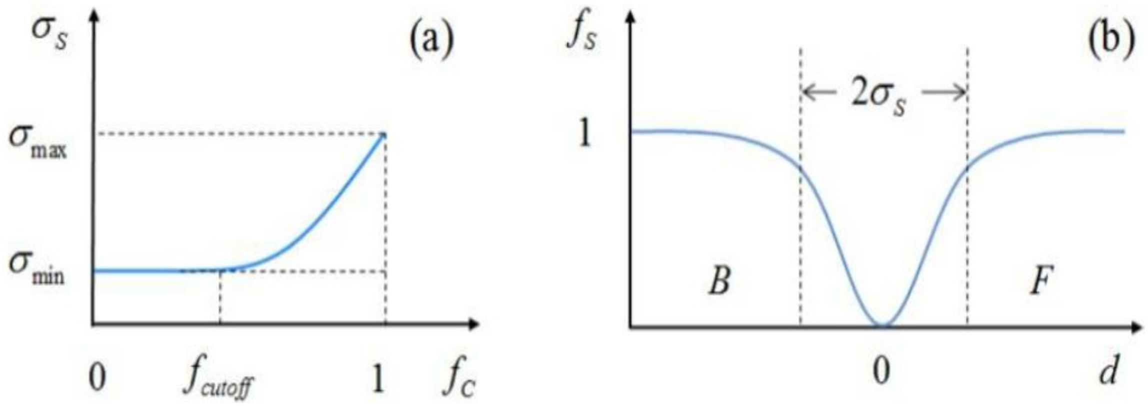


$$f_s(x) = 1 - exp(-d^2(x) \,/\, \sigma_s^2)$$

- ***σs*** here is not fixed. It is a very important parameter that can be adaptively and automatically adjusted.
- A larger ***σs*** means the shape confidence is low around the foreground boundary while a small ***σs*** means high confidence on the segmentation mask Lt(x)
- Please read paper 2.3 and 2.4 for one strategy to adapt ***σs***

Shape model (initShapeConfidences.m)

$$\sigma_s = \begin{cases} \sigma_{min} + a(f_c - f_{cutoff})^r & f_{cutoff} < f_c \leq 1, \\ \sigma_{min} & 0 \leq f_c \leq f_{cutoff}, \end{cases}$$



Shap

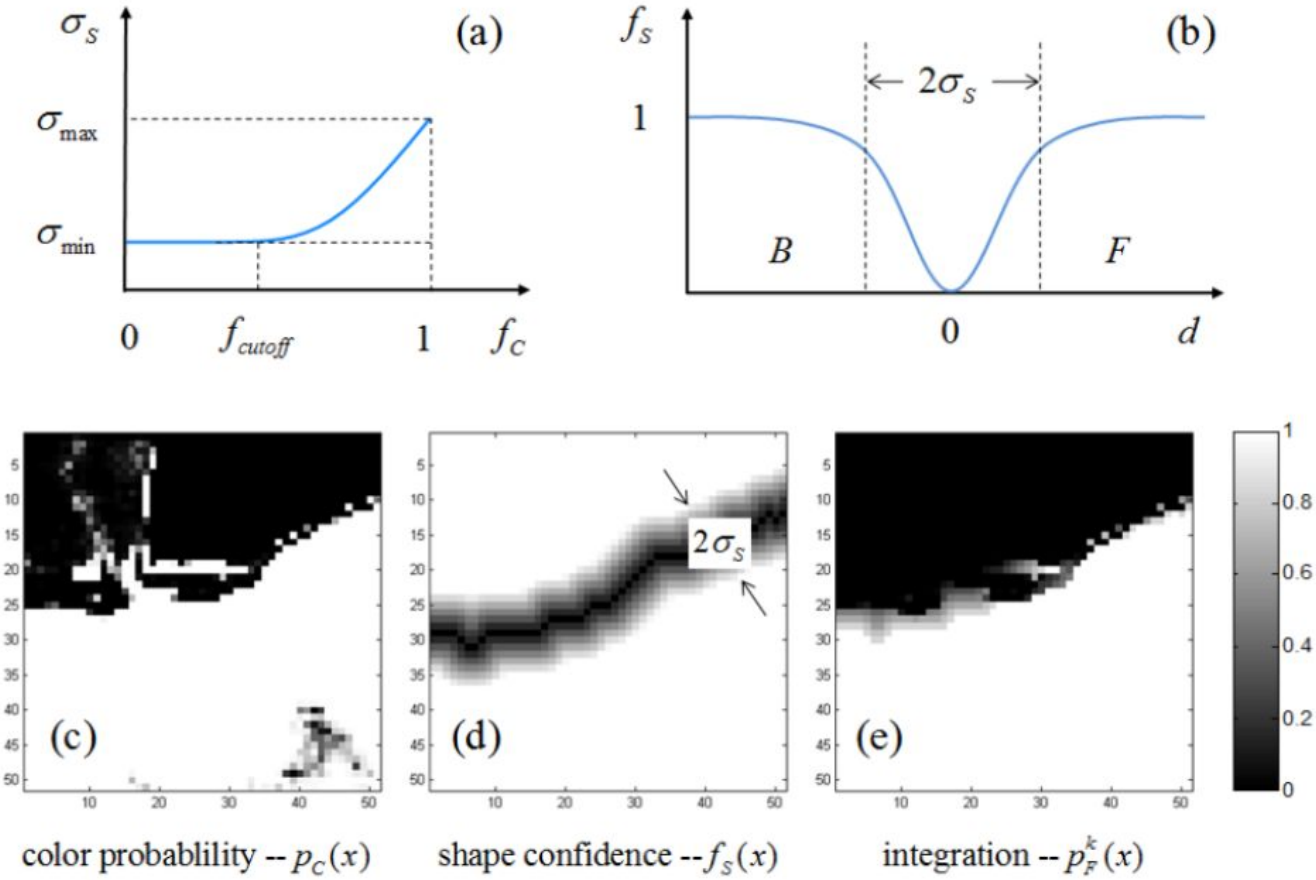
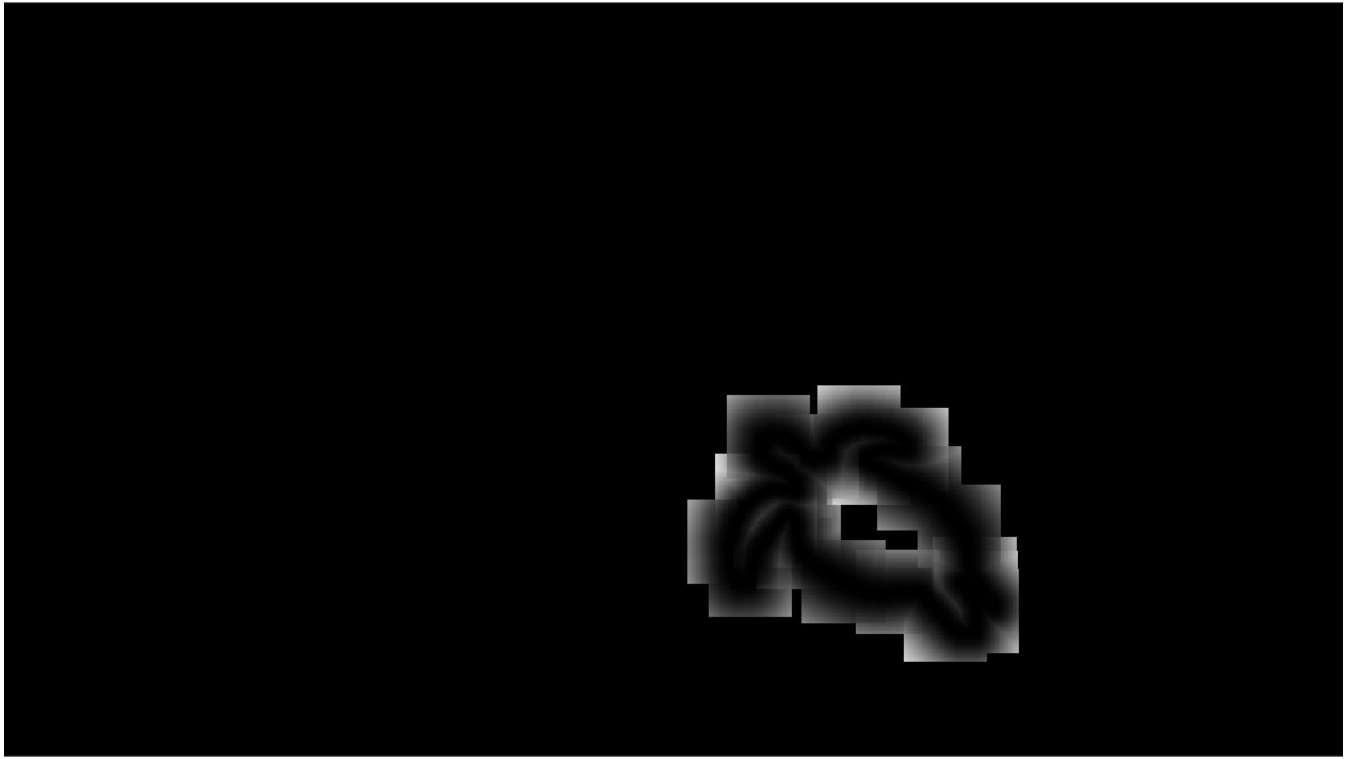
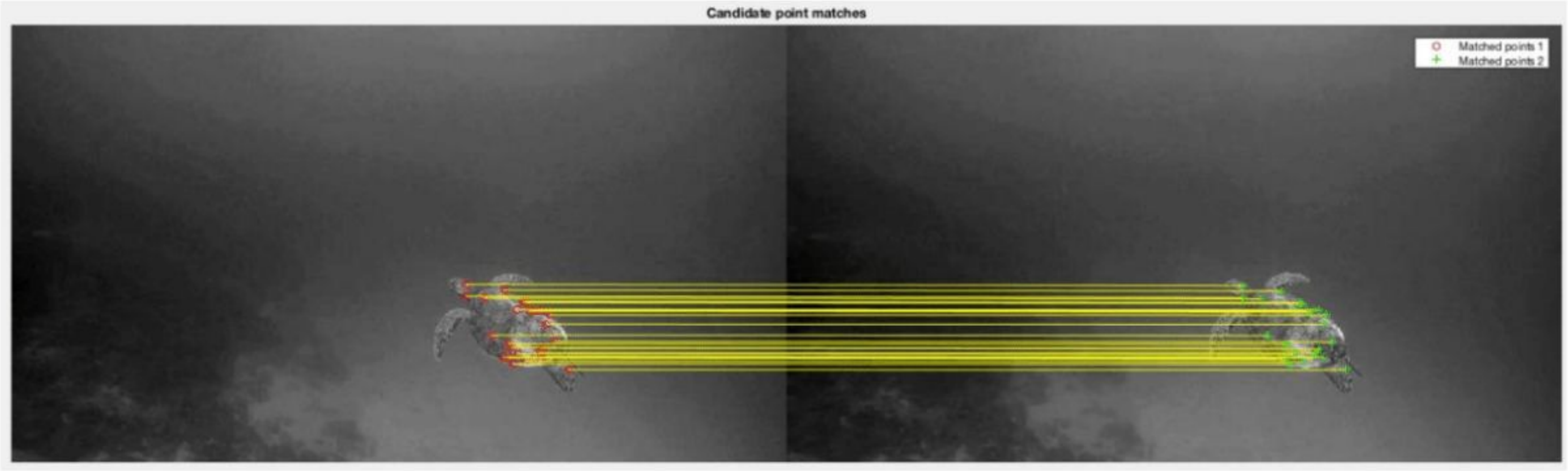


Fig. 5: (a) As color confidence f_c increases (more separable foreground and background color distributions), the value of σ_s increases, giving less weight to the shape prior. (b) Profile of the shape prior $f(x)$. An example of (c) color probability, (d) shape confidence with parameter σ, and (e) the integrated probability $p_k^k(x)$.

Shape model (initShapeConfidences.m)



Local Window Propagation (calculateGlobalAffine.m)



finds affine transform between two frames, and applies it to frame1, the mask, and local windows. Built-in Function `estimateGeometricTransform`

Optical Flow Wrapping (localFlowWrap.m)

- Calculate local window movement based on optical flow between frames.
- Find the average of the flow vectors inside the object's bounds and local windows bound, use that to calculate how to re-center windows.
- opticalFlowHS()

Updating the Shape and Color Models (updateModels.m)

$$p_{\mathcal{F}}^k(x) = f_s(x) L^{t+1}(x) + (1 - \boxed{f_s(x)})\, p_c(x)$$

$$p_{\mathcal{F}}(x) = \frac{\sum_k \boxed{p_{\mathcal{F}}^k(x)}\boxed{(|x - c_k|}+\epsilon)^{-1}}{\sum_k (|x - c_k| + \epsilon)^{-1}}$$

L^{t+1} is the warped segmentation label from previous frame

f_s is the shape confidence mask

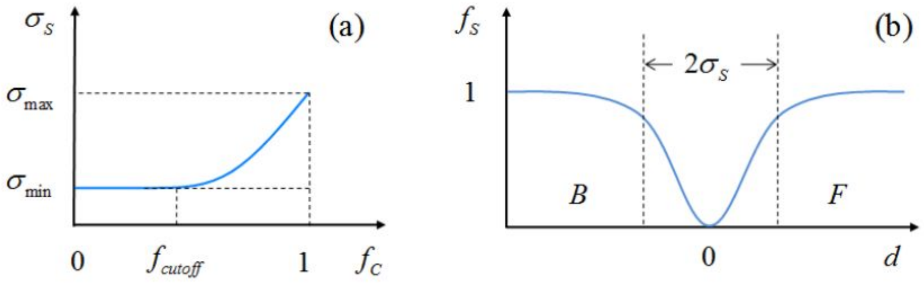
p_c is the foreground probability

c_k is the center of window

e is a small constant

Updating the Shape and Color Models (updateModels.m)

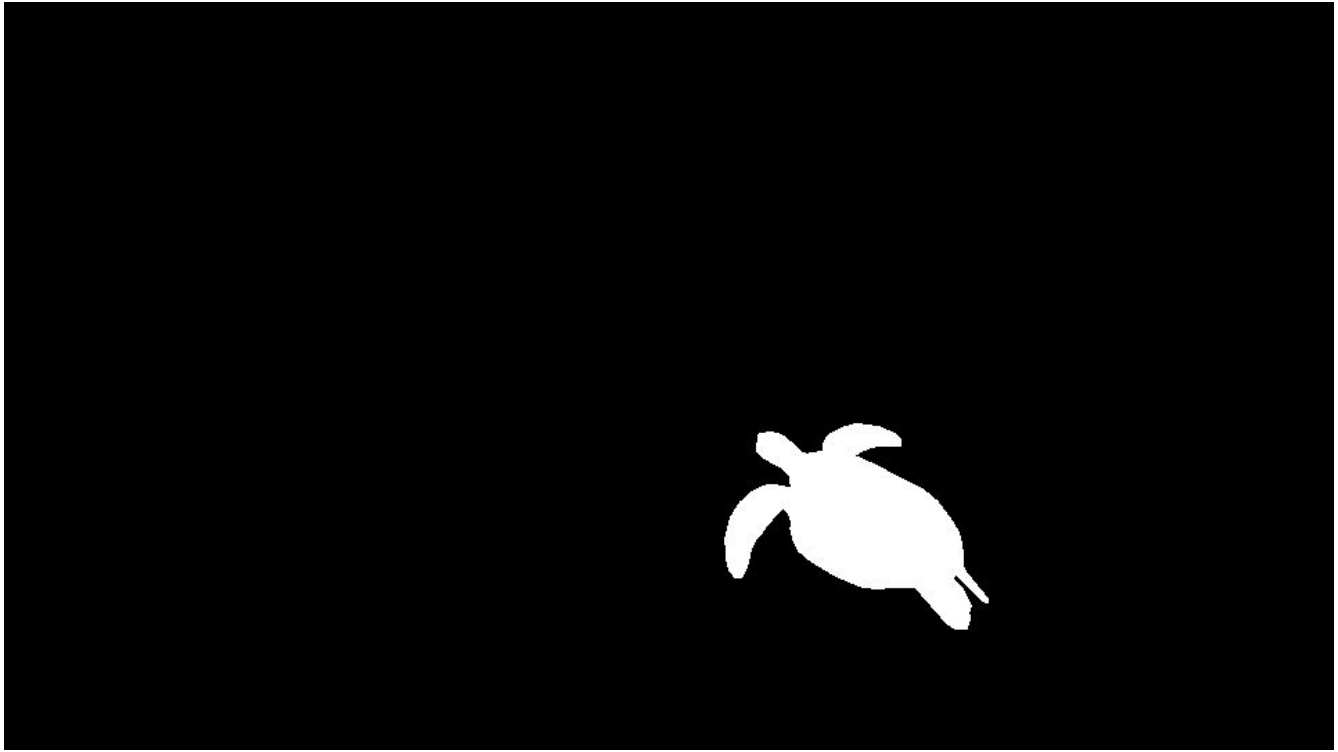
$$p_{\mathcal{F}}^{k}(x) = f_s(x)L^{t+1}(x) + (1 - f_s(x))\, p_c(x)$$



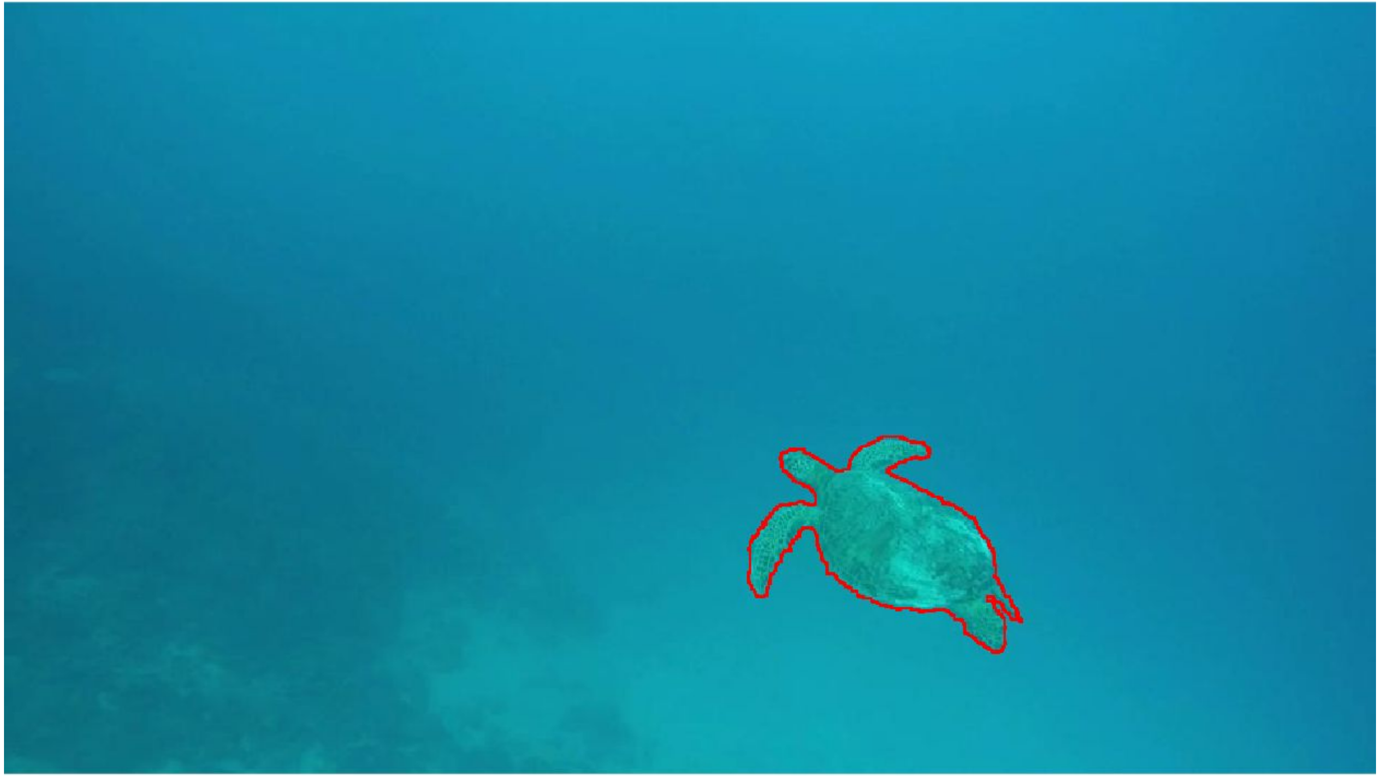
Merging local windows (updateModels.m)

$$p_{\mathcal{F}}(x) = \frac{\sum_k p_{\mathcal{F}}^k(x)(|x - c_k| + \epsilon)^{-1}}{\sum_k (|x - c_k| + \epsilon)^{-1}}$$

Updating the Shape and Color Models (updateModels.m)



Updating the Shape and Color Models (updateModels.m)



Pseudo-code (myRotobrush.m)

Algorithm 1 Rotobrush

```
procedure myRotobrush
    set parameters
    load images
    create mask
    initLocalWindows()                                   ▷ initialize local window
    initColorModels()                                    ▷ initialize Color model
    initShapeConfidences()                               ▷ initialize Shape model
    for every image do
        calculateGlobalAffine()     ▷ transform between previous and current frames
        localFlowWarp()                     ▷ local warping based on optical flow
        updateModels()                           ▷ update color and shape model
    end for
end procedure
```

- Setup Local Windows: 5 pts
- Initialize Color Models: 10pts
- Compute Color Model Confidence: 5 pts
- Initialize Shape Model: 10 pts
- Compute Shape confidence: 5 pts
- Estimate Entire-Object Motion: 5 pts
- Estimate Local Boundary Deformation: 10 pts
- Update Color Model (and color confidence): 15 pts
- Combine Shape and Color Models: 5 pts
- Merge Local Windows: 10 pts
- Extract final foreground mask: 20 pts