TRACKING OF MULTI-SKIN COLOURED OBJECTS UNDER OCCLUSION

Jayanthi Sai Muralidhar (130108014)
Peddakota Vikash (130108024)

ABSTRACT:

In this project, we aim to track multi skin coloured objects when they are partially occluded. Skin-colored objects are detected with a Bayesian classifier which is bootstrapped with a small set of training data. Then, an off-line iterative training procedure is employed to refine the classifier using additional training images. On-line adaptation of skin-color probabilities is used to enable the classifier to cope with illumination changes. Tracking over time is realized through a novel technique which can handle multiple skin-colored objects. Such objects may move in complex trajectories and occlude each other in the field of view of a possibly moving camera. Moreover, the number of tracked objects may vary in time. A prototype implementation of the developed system operates on 720x1080 pixel video with a frame rate of 30 per second.

ALGORITHM OVERVIEW

The method adopted for tracking multiple skin-colored objects operates as follows. At each time instance, the camera acquires an image on which skin-colored blobs are detected. The method also maintains a set of object hypotheses that have been tracked up to this instance in time. The detected blobs, together with the object hypotheses are then associated in time. The goal of this association is (a) to assign a new, unique label to each new object that enters the camera's field of view for the first time, and (b) to propagate in time the labels of already detected objects. Details about the steps involved are described in the sections below.

Skin colour modelling and detection

This step involves the following processes:

1. Training a GMM model iteratively with the training data of pixel values tagged as skin coloured.
2. During the detection phase, estimating the probability of a pixel being a skin coloured pixel.
3. Thresholding the obtained probabilities to classify if the given pixel is skin pixel or not.
4. Performing a connected component analysis to yield all the blobs and computing their statistical information.

A 'Gaussian Mixture Model' is used to model the probability of a pixel being a skin pixel. The class conditional density is given by

$$p(x|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i \in [0,k)} \pi_i N(x, \mu_i, \sigma_i)$$

Where

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

The model is now trained using the standard *Expectation-Maximization* algorithm.
The E step: This is the expectation part. Using the current mean and standard deviation guess, we calculate probabilities. We calculate these values for each Gaussian. This helps us predict which Gaussian is responsible for which datapoint (called the responsibility).
The M step: This is the maximization part. Using the responsibilities in the previous step, we update the mean and standard deviations. These calculations are somewhat analogous to their single Gaussian equivalents.
The process is iterated until convergence is achieved.

Gaussian Mixture Models (GMMs) are very popular in broad area of applications because its performance and its simplicity. However, it is still an open problem on how to determine the number of Gaussian components in a GMM. One simple solution to this problem is to use Bayesian Information Criteria (BIC) to penalize the complexity of the GMM. That is, the cost function of BIC-GMM is composed of 2 parts: 1) log-likelihood and 2) complexity penalty term. Consequently, the final GMM would be a model that can fit the data well, but not "overfitting" the model in BIC sense.

BIC uses the optimal log-likelihood function value and penalizes for more complex models, i.e., models with additional parameters. The penalty of BIC is a function of the sample size, and so is typically more severe than that of Akaike Information Criteria(AIC). Please refer [2] for detailed information.
The formula for BIC is:

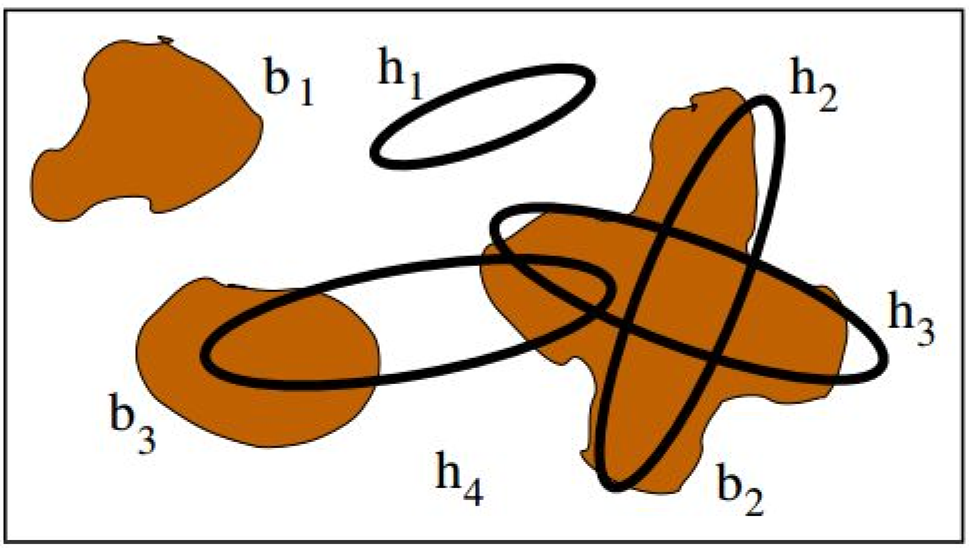
$$\text{BIC} = -2*(\log L) + \text{numParam}*\log(\text{numObs})$$

After training, posteriors are obtained which are used to classify if a pixel is skin pixel or not using a threshold. All the skin pixels in the image are then segmented into different blobs using connected components approach.

Tracking Multiple Objects over Time

Given a frame at time instant *t,* we assume that M blobs have been detected as described earlier. Let the blobs be: $M=\{b_j\}$ $1 \le j \le M$. The correspondence among blobs and objects is not necessarily one-to-one all the time. As an example, consider two crossing hands are two different skin-colored objects that appear as one blob at the time and one occludes the other. We assume that an object may correspond to either one blob or part of a blob. Similarly, one blob may correspond to one or many objects. We also assume that the spatial distribution of the pixels depicting a skin colored object can be coarsely approximated by an ellipse. This assumption is valid for skin-colored objects like hand palms and faces.

Let N be the number of skin-colored objects present in the viewed scene at time *t* and o_i $1\le i \le N$, be the set of skin pixels that image the i-th object. Let $h_i = h_i(c\_x_i, c\_y_i, \alpha_i, \beta_i, \theta_i)$ the ellipse model of this object where (c_x_i, c_y_i) is its centroid, α_i, β_i are respectively the lengths of its major and minor axis, and θ_i is its orientation on the image plane. Tracking amounts to determining the relation between object models (hi) and observations (bj) in time.



In this particular example there are three blobs (b1, b2 and b3) while there are four object hypotheses (h1, h2, h3 and h4) from the previous frame.

The later phases of the project involved tracking the skin objects detected in the previous phase. This phase comprised of three major steps:

1. Hypothesis Generation
2. Hypothesis Tracking
3. Hypothesis Removal

Hypothesis Generation

We define the distance metric $D(P, h_i)$ of a point $P(x,y)$ with ellipse h_i as follows:

$$D(p, h) = \sqrt{\vec{v} \cdot \vec{v}}$$

$$\vec{v} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \left(\frac{x - x_c}{\alpha}, \frac{y - y_c}{\beta} \right)$$

As a trivial observation, the value of this metric is less than 1.0, equal to 1.0 or greater than 1.0 depending on whether point p is inside, on, or outside ellipse h_i, respectively. Now, consider a blob b such that:

$$\forall p \in b, \ \min_{h \in H}\{D(p, h)\} > 1.0.$$

This means that all the pixels belonging to the blob b have an empty intersection with all the existing hypothesis. For such blobs, a new hypothesis is generated as follows:

$$\alpha = \sqrt{\lambda_1}, \quad \beta = \sqrt{\lambda_2}, \quad \theta = \tan^{-1}\left(\frac{-\sigma_{xy}}{\lambda_1 - \sigma_{yy}} \right)$$

where $\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Lambda}{2}$, $\lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Lambda}{2}$, and $\Lambda = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 - 4{\sigma_{xy}}^2}$

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$$

All such blobs for which a new hypothesis is generated are excluded from further consideration in the subsequent steps of object tracking.

Hypothesis Tracking

Upon generating new hypothesis, all the *remaining blobs* must support the existence of past object hypotheses. The main task of the tracking algorithm amounts to associating blob pixels to object hypotheses. There are two rules governing this association:

Rule 1: If a skin-colored pixel of a blob is located within the ellipse of some object hypothesis (i.e. supports the existence of the hypothesis) then this pixel is considered as belonging to this hypothesis.
Rule 2: If a skin-colored pixel is outside all ellipses corresponding to the object hypotheses, then it is assigned to the object hypothesis that is closer to it, using the distance metric D(P,h).

An interesting case is that of a hypothesis that is supported by more than one blobs. Such cases may arise when, for example, two objects are connected at the time they first appear in the scene and later split. To cope with situations where a hypothesis h receives support from several blobs, the following strategy is adopted. If there exists only one blob b that is predicted by h and, at the same time, not predicted by any other hypothesis, then h is assigned to b. Otherwise, h is assigned to the blob with which it shares the largest number of skin-colored points. After having assigned skin pixels to object hypotheses, the parameters of the object hypotheses h_i are re-estimated based on the statistics of pixels o_i that have been assigned to them.

Hypothesis Removal

An object hypothesis should be removed either when the object moves out of the camera's field of view, or when the object is occluded by another (non-skin colored) object in the scene. Thus, an object hypothesis h should be removed from further consideration whenever

$$\forall p \in B,\ D(p,h) > 1.0.$$

Equation essentially describes hypotheses that are not supported by any skin-colored image points.

Experimental Results

REFERENCES

[1] "Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera"-
Antonis A. Argyros and Manolis I.A. Lourakis
[2] https://in.mathworks.com/help/econ/aicbic.html#outputarg_bic