Face tracking with Constrained Local Models and 3D data

Mark Lyksborg og Anders Jensen

DTU Informatics, DTU

Project description

Following project deals with the problem of tracking faces in an image. The model we are going to use is the Constrained Local Model (CLM) suggested by D. Cristinacce and T. Cootes [Cristinacce and Cootes, 2006]. The strengths and weaknesses of CLM have been investigated.

Where face tracking traditionally is done on 2d images, we will investigate if inclusion of depth information will enhance performance of tracking. The depth information is acquired through a Time Of Flight (TOF) camera. In addition to using the TOF data in the optimization part of the tracking, we will look at the possibility of making a starting guess based on the TOF data.

TOF camera

The time of flight camera is used for supplying both a standard 2d image as well as real world coordinates of the object placed in the image, giving a x,y and z coordinate for each pixel. The principle for how the camera obtains these measurements is similar to how radar works. A signal is emitted and reflected of an object. When the reflected signal returns to the emitter, it processes the signal to extract information.

(a) (b)

Figure 1: *(a) Time of flight camera and (b) example of 3d image*

The TOF camera has some limitations. It is not good to handle hair and reflective items (like glasses). It is also sensitive to light conditions.

Facetracking

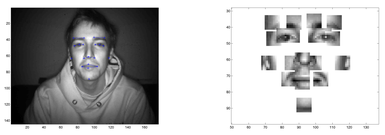
The purpose of face tracking is to find the face of a person in any given image, regardless of where the person is in the image, how the person looks and the pose of the face. Figure 2 shows an example of the tracked features.

Figure 2: *4 annotated images*

Face tracking can be used in various applications. One could be monitoring if the driver of a car is looking at the traffic or has fallen asleep. Another could be monitoring a meeting, seeing who is talking and automatically making a transcription of the meeting (with help of voice recognition).

Introduction to CLM

In CLM the location of the face is defined by a set of landmarks as shown in figure 2. The basic steps of the model is to train an appearance with shape, texture and distance data, initialize this model on an unknown image and use a search algorithm to deform it towards an optimization criterion.



(a) (b)

Figure 3: *(a) Annotated image (b) Sampled texture*

One special feature in CLM is that texture is sampled in square *patches* around shape landmark. This makes the sampling procedure more simple than other methods. Figure 3 shows an image and the sampled texture.

Appearance model

The appearance model is a parametric model that produce different shape coordinates and texture according to some parameterse. This is done using Principal Component Analysis (PCA) as suggested by [Cristinacce and Cootes, 2006]. This model consists only of a shape and texture part. We handle the 3d data (z-coordinates) as it was texture, giving the model a new third term.

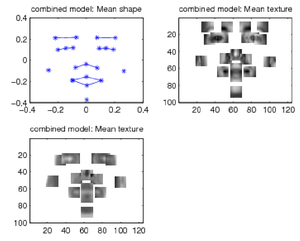


Figure 4: *Shows the mean shape, texture patches and depth patches for the PCA model*

The PCA models for shape x, texture g and z-coordinates z seperated are:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s \quad \mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g \quad \mathbf{z} = \bar{\mathbf{z}} + \mathbf{P}_z\mathbf{b}_z \quad (1)$$

The P's are the PCA Eigen vector matrices and the b's are model parameters. Variying these will change the result of the model. Finally a combined model that covers covariance is built:

$$\mathbf{b} = \mathbf{P_c} \cdot \mathbf{c} \quad (2)$$
$$\text{where} \quad \mathbf{b} = [w_s \cdot \mathbf{b_s} \quad w_g \cdot \mathbf{b_g} \quad w_z \cdot \mathbf{b_z}]$$

b is a vector with all the parameters and the w's are weight giving the parameters same scale.

The result is shown in figure 4 (the mean shape, texture and z-coordinates) and 5 (the first PCA mode). It is seen that the first mode is look up/down. The second mode (not shown) is looking left/right.

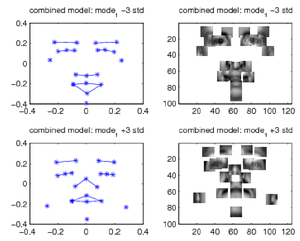


Figure 5: *Shows the combined extreme shape, texture for mode 1. Models looking up,down.*

CLM search algorithm

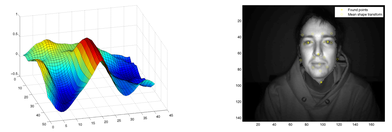
With the PCA model in place, the CLM search algorithm is used to find faces in new images. A objective function is minimized in an iterative manner. The objective function consists of a part for cross correlation with texture, one with z-coordinates and one for the shape:

$$f(\mathbf{p}) = \sum_{j=1}^{m} Ig_j(X_j, Y_j) + \sum_{j=1}^{m} Iz_j(X_j, Y_j) - K\sum_{j=1}^{n}\frac{b_j^2}{\lambda_j} \quad (3)$$

Ig_j and Iz_j denote the cross correlation values and K is a scalar balancing the correlations and shape terms. The cross correlation of a single patch is illustrated in figure 6. The algorithm is as follows:

1. Make initial guess
2. Loop
   (a) Project shape and texture onto joint model
   (b) Make texture templates from projection
   (c) Use Nelder-Mead to optimize objective function

The templates for cross correlation is fixed for each step of the loop, so it is possible to pre compute a correlation responce surface like shown in figure 6 (a). The shape parameters are changed in the Neadler-Mean optimization.



(a) (b)

Figure 6: *(a) Cross correlation for a single patch (b) [next section] Starting guess based on TOF data. The light area is the head region*

Creating starting guesses with TOF data

We made an algorithm for making a starting guess based on the TOF data. It uses tresholding on the distance coordinates to find the face followed by cross correlation to find the eyes and nose. The method is summarized here:

1. Sort all z-values and find 10% percentile to use as threshold value
2. Threshold and take out largest area
3. Use dilation to find the full head (stop at the edge, where the gradient of z-coordinates are high)
4. Use cross correlation to find eyes and nose
5. Project mean shape onto eye, nose and chin points (the chin beeing the lowest point)

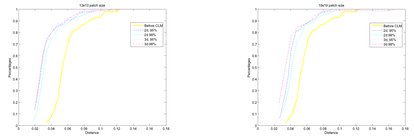
The process is illustrates in figure 6 (previous section). The light area in (a) is the face region. The points found with cross correlation and the mean shape transformated points are also shown.

Results and conclusion

Varoius experiments were conducted. On test with 2d data (the IMM face database) only we concluded:

- The starting guess should not be more than 10 pixels away from the true point.
- The scale and rotation of the starting guess does not influence much of the final result.

The result of the test with 3d is shown on figure 7. We concluded that a small patch size gave the best result. The 3d data only gave marginal improvements of the results at the cost of longer running times. However, since the data was limited, we cannot conclude what will be the case i general. Tests with more data is needed to draw final conclusions.



(a) (b)

Figure 7: *CDF for experiments with patch size of (a) 13 x 13 and (b) 19 x 19*

When making starting guesses with TOF data, the starting guess was always acceptable when the person was facing the camera. If the person looks in other direction it only sucseeds in 60% of the cases, so there are room for improvement.

References

[Cristinacce and Cootes, 2006] Cristinacce, D. and Cootes, T. F. (2006). Feature detection and tracking with constrained local models. In *Proc. British Machine Vision Conference*, volume Vol. 3, pages pp.929–938.