Face detection, validation and tracking

Océane Esposito, Grazina Laurinaviciute, Alexandre Majetniak

Agenda

- Motivation and examples
- Face detection
- Face validation
- Face tracking
- Conclusion

Motivation

- Goal: Allow an automatic system to sense the presence of people somewhere in a room
- It can be used in a very different way
- Assists people in their work and make it easily
- A very long process because of the amount of data in a video

Applications

- Surveillance
- Facial recognition systems
- digital cameras use face detection for autofocus
- Assists elderly people

Face Detection

Viola Jones's algorithm

Face detection

- A learning approach for visual object detection
- Robust and rapid object detection
- Capable of processing images rapidly
- Don’t use image differencing and skin color detection
- Use information present in a single grey scale image
- Using a Boosted cascade of simple features

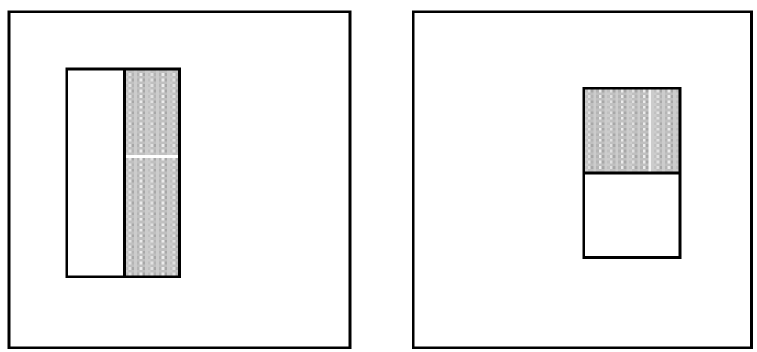
Face detection

1. Features
2. A new image representation called the "integral image"
3. Learning algorithm based on AdaBoost
4. Combining increasingly more complex classifiers in a "cascade"

Features

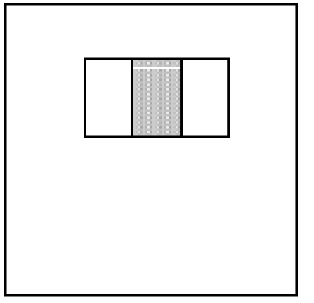
- Three different kinds of features:

1. The value of *2 rectangle feature:* difference between the sum of the pixels within two rectangular regions.

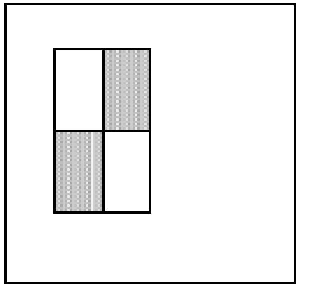


Features

- The value of *3 rectangle feature:* computes the sum within 2 outside rectangles subtracted from the sum in a center rectangle



- The value of *4 rectangle feature:* computes the difference between diagonal pairs of rectangles



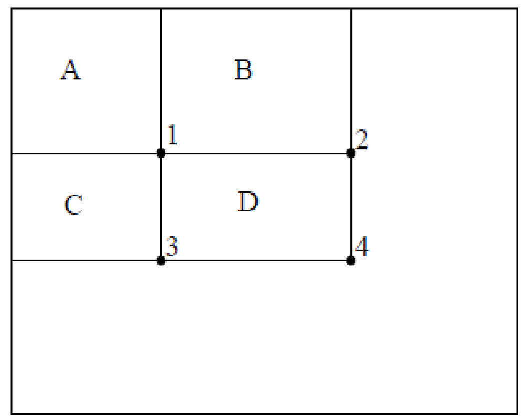
Integral image

- A few operations per pixel
- To compute the rectangle features we use an integral image:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

The integral image at location *x,y* contains the sum of the pixels above and to the left of *x,y*

Integral image



- The sum of pixel in rectangle D can be computed with four array references
- The sum within D can be computed as:

4+1-(2+3)

AdaBoost algorithm

- A learning process
- Boost the classification performance of a simple learning algorithm
- Select a small number of critical visual features from a larger set and made an efficient classification
- Train the classifier
- A small number of these features can be combined to form an effective classifier
- Challenge: to find the features

AdaBoost algorithm

1. Training set
   - A label: for N images xi. yi = 1, when an image is considered a face, yi = 0 otherwise.

2. combines T classifiers called "weak"
   - Combine "weak" classifiers to have a new classifier called "strong".
   - Each "weak" classifier consists of a single feature and an optimal threshold.
   - "Strong classifier": A combination of T weak classifiers.
   - It takes an image as input and produces a binary value (1 or 0) on output.

AdaBoost algorithm

3. A feature is computed for each image xi.
   - the image set {xi} can now be divided into two parts : negative and positive
   - The result can be compared with original labels {yi}
   - The threshold is chosen for each feature separately

AdaBoost algorithm

4. A round:
   - A round starts with a set of normalized weights associated with each image xi
   - in each round it is estimated how well every classifier separates positive and negative example images, in comparison to their actual classification {yi}
   - The classifier with a minimum classification error is chosen

5. After T rounds:
   - T selected "weak" classifiers (and their corresponding features)

Cascade classifiers

- Allows background regions of the image to be quickly discarded
- More computation on regions where they are objects
- More complex processing is reserved for these promising regions.

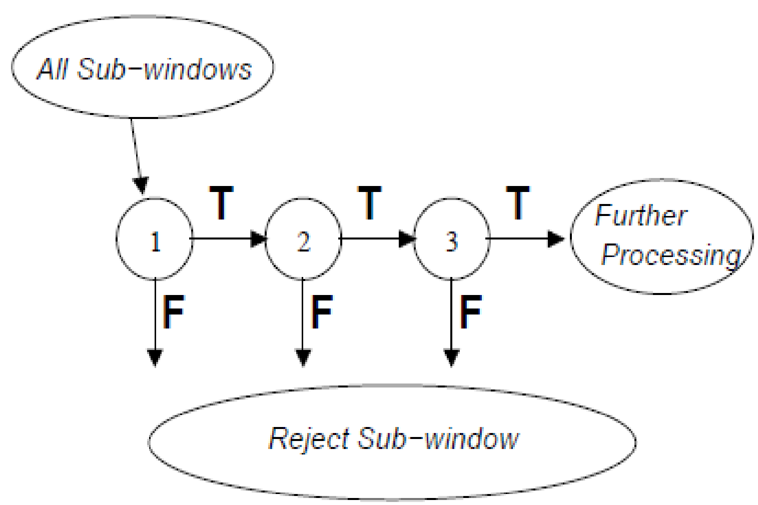
Cascade classifiers

- Combining increasingly more complex classifiers in a "cascade"
- Each stage in the cascade reduces the false positive and decreases the detection rate
- Complete face detection cascade have 38 stages over 6 000 features

Cascade classifiers

1. Simpler classifiers are use to reject the majority of sub-windows
2. More complex classifiers are called upon to achieve low false positive rate.

- Stages in the cascade are constructed by training classifiers AdaBoost and then adjusting the threshold to minimize false negative.



Face validation

Verifying, if a face detection returned by the face detector, is indeed a face

(otherwise, it is just a false positive)

Face validation: outline

- Motivation
- Types of face validation:
  - 2D image validation
  - 3D position validation
  - Color validation
  - Pattern validation

Face validation: outline

- **Motivation**
- Types of face validation:
  - 2D image validation
  - 3D position validation
  - Color validation
  - Pattern validation

Face detection: motivation

- Need for detecting resolution-limited faces of varying pose, expression and illumination
- Solutions:
  - Multiple situation-specific detectors and fuse their decisions: time consuming
  - **Single generic less strict detector**: a tradeoff between **hits** and **false positives**

Face detection: motivation

- The tradeoff of hits and false positives can be addressed by:
  - **increasing the number of stages** in Viola & Jones detector cascade;
  - using **face validation** as a post-processing step

Face validation: outline

- Motivation
- **Types of face validation:**
  - **2D image validation**
  - 3D position validation
  - Color validation
  - Pattern validation

Face validation: 2D validation

- Discarding one of two detections that are overlapping on the image plane.

- Measure of overlap – **common area ratio**:

$$CAR = A[BB1 \cap BB2] / \min(A[BB1], A[BB2])$$

Face validation: 2D validation

- 2D spatial validation seldom removes actual faces, since partially occluded faces are not detected at all

Face validation: outline

- Motivation
- Types of face validation:
  - 2D image validation
  - **3D position validation**
  - Color validation
  - Pattern validation

Face validation: 3D validation

- Idea:
  - estimate 3D position of a face using a single calibrated camera
  - discard faces which have an estimated 3D position outside the room
- Underlying assumption:
  - human face is approximately 15 cm wide

Face validation: 3D validation

- Idea:
  - **estimate 3D position of a face using a single calibrated camera**
  - discard faces which have an estimated 3D position outside the room
- Underlying assumption:
  - human face is approximately 15 cm wide

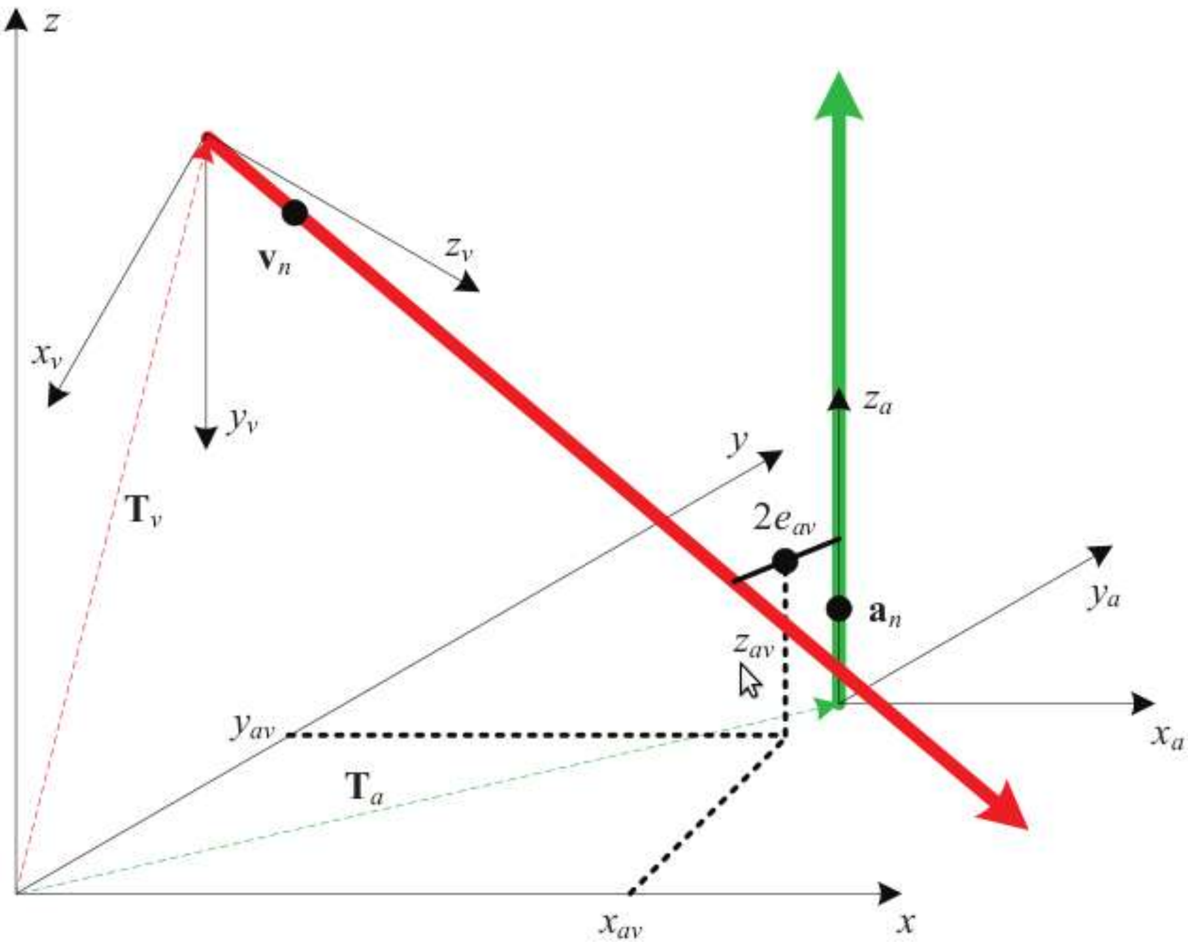
Face validation: 3D validation

- Face 3D position estimation consists of:
  - **Definining a mapping between an image pixel and a 3D line in camera coordinate system**
  - Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - Obtaining face width Wn in this plane
  - Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D
  - Using z coordinate to obtain x and y
  - Mapping {x,y,z} from camera to world coordinate system

Face validation: 3D validation

- Definining a mapping between an image pixel and a 3D line in camera coordinate system:
  - Pixel {xp , yp } --> 3D line {xn · zc , yn · zc , zc}
  - {xn, yn} - depth-normalized camera coordinates
  - Depth-normalized means z = 1
  - Mapping is non-linear
  - Knowledge of intrinsic camera parameters needed

Face validation: 3D validation



Face validation: 3D validation

- Face 3D position estimation consists of:
  - Definining a mapping between an image pixel and a 3D line in camera coordinate system
  - **Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1**
  - Obtaining face width Wn in this plane
  - Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D
  - Using z coordinate to obtain x and y
  - Mapping {x,y,z} from camera to world coordinate system

Face validation: 3D validation

- Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - Detected faces are represented as: {xf , yf , wf , hf }
  - {xf, yf} – top left corner of a bounding box
  - wf – width
  - hf - height

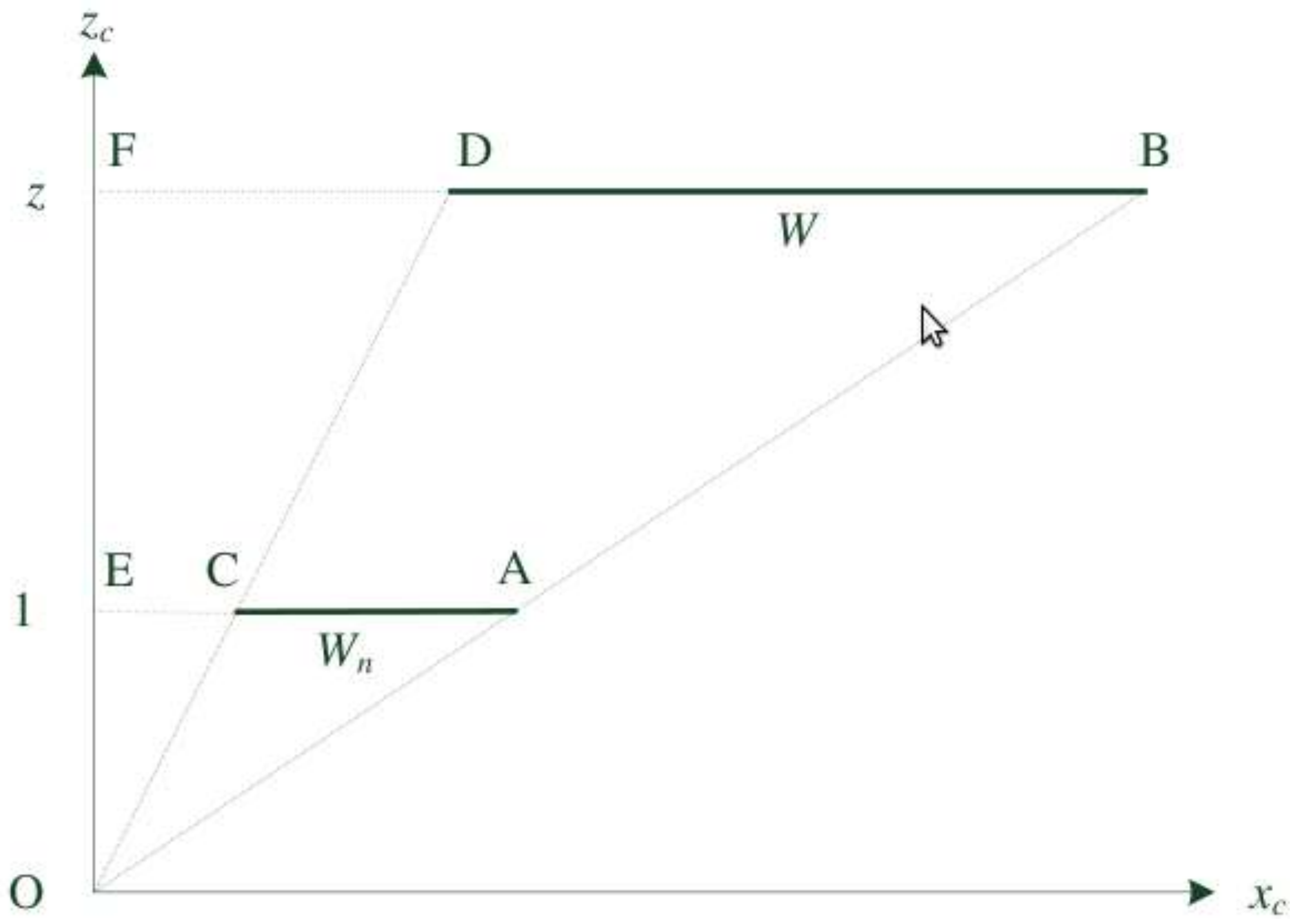
Face validation: 3D validation

- Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - **Left** border mid-point: {xf , yf + hf /2}
  - Its projection: {xpl , yp}

  - **Right** border mid-point: {xf + wf , yf + hf /2}
  - Its projection: {xpr , yp}

- Note: as z = 1, it is enough to calculate x and y

Face validation: 3D validation



Face validation: 3D validation

- Face 3D position estimation consists of:
  - Definining a mapping between an image pixel and a 3D line in camera coordinate system
  - Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - **Obtaining face width Wn in this plane**
  - Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D
  - Using z coordinate to obtain x and y
  - Mapping {x,y,z} from camera to world coordinate system

Face validation: 3D validation

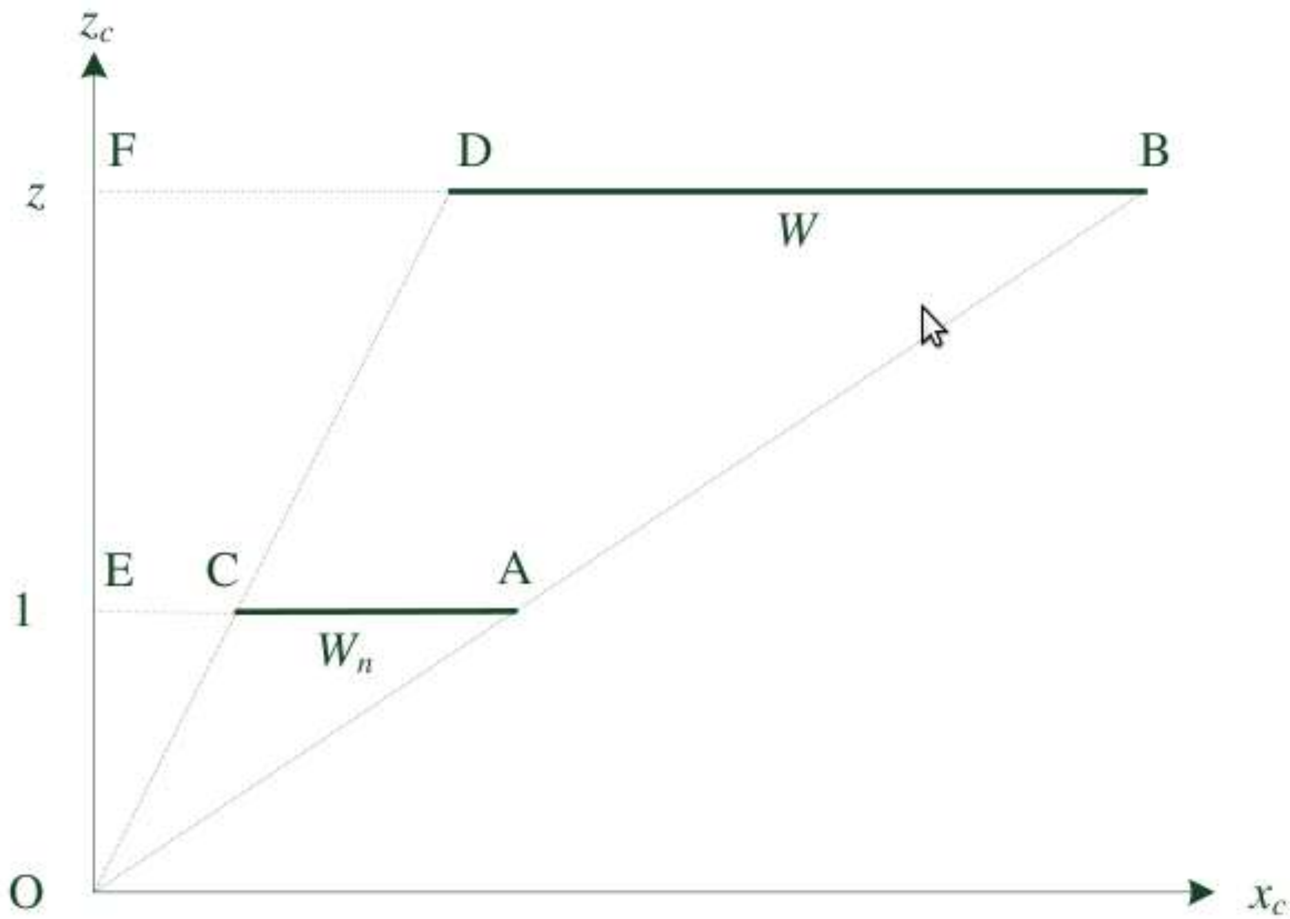
- Obtaining face width Wn in this plane:

$$W_n = |x_{nl} - x_{nr}|$$

Face validation: 3D validation

- Face 3D position estimation consists of:
  - Definining a mapping between an image pixel and a 3D line in camera coordinate system
  - Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - Obtaining face width Wn in this plane
  - **Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D**
  - Using z coordinate to obtain x and y
  - Mapping {x,y,z} from camera to world coordinate system

Face validation: 3D validation



Face validation: 3D validation

- Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D:
  - W / Wn = |OD| / |BD| = |OC| / |AC|
  - |OD| / |OC| = |OF| / |OE| = z / 1
  - z = W / Wn

Face validation: 3D validation

- Face 3D position estimation consists of:
  - Definining a mapping between an image pixel and a 3D line in camera coordinate system
  - Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - Obtaining face width Wn in this plane
  - Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D
  - **Using z coordinate to obtain x and y**
  - Mapping {x,y,z} from camera to world coordinate system

Face validation: 3D validation

- Using z coordinate to obtain x and y:
  - **Left** border mid-point in 3D:

{xnl · z, yn · z, z}

  - **Right** border mid-point in 3D:

{xnr · z, yn · z, z}

  - **Face centre** in 3D: their mid-point

Face validation: 3D validation

- Face 3D position estimation consists of:
  - Definining a mapping between an image pixel and a 3D line in camera coordinate system
  - Using the obtained mapping, projecting mid-points of left and right borders of the face bounding box into a plane, where z = 1
  - Obtaining face width Wn in this plane
  - Using the similarity of triangles and Wn, computing the actual z coordinate of the face in 3D
  - Using z coordinate to obtain x and y
  - **Mapping {x,y,z} from camera to world coordinate system**

Face validation: 3D validation

- Mapping from camera to world coordinate system
  - Produces {xw, yw, zw}
  - Mapping is linear
  - Camera extrinsic parameters needed

Face validation: 3D validation

- Idea:
  - estimate 3D position of a face using a single calibrated camera
  - **discard faces which have an estimated 3D position outside the room**
- Underlying assumption:
  - human face is approximately 15 cm wide

Face validation: 3D validationv

- Height and floorplan constraints:
  - $z_w \in [z_{min}, z_{max}]$
  - $x_w \in [x_{min}, x_{max}]$
  - $y_w \in [y_{min}, y_{max}]$

Face validation: outline

- Motivation
- Types of face validation:
  - 2D image validation
  - 3D position validation
  - **Color validation**
  - Pattern validation

Face validation: color validation

- Idea:

discarding faces based on the similarity between their color histogram and the skin color histogram of Jones and Rehg

Face validation: color validation

- Joint color histogram

$$c = \left\lfloor R\frac{N_h}{256} \right\rfloor + N_h \left\lfloor G\frac{N_h}{256} \right\rfloor + N_h^2 \left\lfloor B\frac{N_h}{256} \right\rfloor$$

  - $c \in [0, \ldots, Nh^3 - 1]$
  - Nh = 16 means 4096 bins in a histogram

Face validation: color validation

- Histogram similarity between detection and Jones and Rehg
  - Bhattacharyya coefficient:

$$BC(h_c, h_{JR}) = \sum_{n=1}^{N_h^3} \sqrt{h_c(n) h_{JR}(n)}$$

- High similarity means that the face detection is valid

Face validation: outline

- Motivation
- Types of face validation:
  - 2D image validation
  - 3D position validation
  - Color validation
  - **Pattern validation**

Face validation: pattern validation

- Idea:
  - Statistically measure the similarity between face detection and prototype frontal face
- Method used: **DFFS** (Distance From Face Space)

Face validation: pattern validation

- DFFS algorithm elements:
  - Training data set: non-occluded frontal faces
  - x_avg - the average face vector of the data set
  - W - a face subspace projection matrix, obtained from the data set using Principal Components Analysis
  - x - face detection represented as a vector (rearranging columns of grey levels)

Face validation: pattern validation

- DFFS algorithm:
  - Using matrix W, face vector x is projected to a a face subspace, where its difference d from x_avg is measured:

$$d_{proj} = \mathbf{x} - \left(W\left(W^T\left(\mathbf{x} - \overline{\mathbf{x}}_{AR}\right)\right) + \overline{\mathbf{x}}_{AR}\right)$$

  - d is used to find the measure of similarity:

$$DFFS = \sqrt{\frac{1}{N_{\mathbf{x}}} \sum_{n=1}^{N_{\mathbf{x}}} d_{proj}(n)^2}$$

Face validation: pattern validation

- High similarity (= low DFFS value) means a valid face detection

Face tracking

Outline

- The AIT tracking system
- Person tracking in enhanced cognitive care
- Joint Bayesian Tracking of Head Location

1. The AIT tracking system

- Tracking is widely used in surveillance
- Two approaches for face tracking: Stochastic and deterministic

Tracking system

- 2D face localization constrained in the body areas
- CAM-Shift tracker: Used when no association is made. Tracks similarly colored regions
- for all active tracks: check for duplicates

Body tracking module

- Provides the regions where to apply face detection.
- Dynamic foreground segmentation : adaptative background modeling, learning rates
- Kalman filter
- Benefit: solves the problem of targets fading into the background.
- Applications: tracking people in a meeting

Face tracking module

- Assigning detected faces to tracked targets (Munkres algorithm)
- Association conflict (occlusion, rotation..)
- CAM-Shift algorithm:
  - pending status
  - Trained color histogram
  - No update: uses only detections validated by GMM

Summary

- Overall performance improved. CAM-shift tracker complements the three face detectors
- Misses and false positives should be reduced:
  - Better detectors
  - Replacing CAM-Shift tracker by Kalman or particle filter tracker.

2. Person tracking in enhanced cognitive care

- Audiovisual system detecting movement in a cluttered and reverberant environment:
  - One camera
  - 3 microphones
- Audio will not be discussed further in the actual context

Visual tracker: prelude

- Face detection measurement
- Color model matching

- The visual tracker benefits from both the precise localization of the face detection and the presence of color model.

3. Joint Bayesian Tracking of head location

- Tracking of people location supports proxemic studies

Appearance likelihood for pose tracking

- Target are described in terms of a low-dimensional representation: color and shape
  - Shapes identify image patch (head, torso)
  - Color describe the appearance of the body parts
- Histogram: offers robustness
- Part based definition: exploits appearance Independence
- State of the target: position + horizontal reference plane/torso orientation, and its head plan and tilt angles (3D space).

Summary

- New approaches
  - Particle filtering approach
  - Joint Bayesian tracking

References

References

- Rapid Object Detection using a Boosted Cascade of Simple Features, *Paul Viola & Michael Jones.*
- A synergistic detection- and track-level system for monitoring people in smart spaces, *Panagiotis Papageorgiou, Nikos Katsarakis, Andreas Stergiou, Aristodemos Pnevmatikakis*
- Face Validation Using 3D Informtion from Single Calibrated Camera, *Nikos Katsarakis*
- Audio-Visual Person Tracking: A Practical Approach, *Aristodemos Pnevmatikakis*
- The AIT 2D Face Detection and Tracking System for CLEAR 2007, *Andreas Stergiou & Lazaros Karame*
- Joint Bayesian Tracking of Head Location and Pose from Low-Resolution Video, *Aristodemos Pnevmatikakis & Fotios Talantzis*

Questions ☺