Computer Vision
Head pose estimation

Jonas Vanthornhout
University of Leuven

June 18, 2012

Outline

Possible approaches [5]

Possible Approaches
- Appearance template methods
- Detector arrays
- Nonlinear regression methods
- Manifold embedding methods
- Flexible models
- Geometric methods
- Tracking methods
- Hybrid methods

Appearance template methods

1. Match head with set of heads
2. Best match determines pose

Appearance template methods

1. Match head with set of heads
2. Best match determines pose

Advantages	Disadvantages
No features needed	Only discrete poses
Can be easily expanded	Head region must be known
	Performance degradation possible
	Pairwise similarity

Detector arrays

1. Similar to appearance template methods
2. Detector for each pose is made

   $\Rightarrow$ only variation corresponding to pose change is learned

   $\Rightarrow$ pairwise similarity problem is solved

Detector arrays

1. Similar to appearance template methods
2. Detector for each pose is made
   $\Rightarrow$ only variation corresponding to pose change is learned
   $\Rightarrow$ pairwise similarity problem is solved

Advantages	Disadvantages
No head localization needed	More difficult training phase
Learns the correct appearance variation	Performance degradation possible

Nonlinear regression methods

1. Calculate from each image a value
   Possible to use features to calculate this value
2. Use nonlinear regression on these values to estimate the pose

Remark: mostly used in combination with a neural network

Nonlinear regression methods

1. Calculate from each image a value
   Possible to use features to calculate this value
2. Use nonlinear regression on these values to estimate the pose

Remark: mostly used in combination with a neural network

Advantages	Disadvantages
Fast	Nontrivial regression
Accurate	Head locator needed

Manifold embedding methods

1. Reduce the dimensionality of the image
2. Uses these dimensions to estimate pose

Manifold embedding methods

1. Reduce the dimensionality of the image
2. Uses these dimensions to estimate pose

Advantages	Disadvantages
Uses correct dimension	Difficult to reduce dimensionality

Flexible models

1. Map a known model to an image
2. Compare this mapping to other mappings
3. Best match determines the head pose

Flexible models

1. Map a known model to an image
2. Compare this mapping to other mappings
3. Best match determines the head pose

Advantages	Disadvantages
Robust to deformations	Feature detector needed
	Computationally expensive
	Frontal position needed

Geometric methods

1. Based on psychophysical experiments
2. Use this knownledge to determine head pose

Remark: a lot of implementations possible

- Features
- Gradients

Geometric methods

1. Based on psychophysical experiments
2. Use this knownledge to determine head pose

Remark: a lot of implementations possible

- Features
- Gradients

Advantages	Disadvantages
Simplicity	Feature/... detector needed
Uses information known to humans	Accuracy limited by humans

Tracking methods

1. Use information from consecutive images
2. "Normal" head pose estimator for first image
3. Next images can use previous estimation and the image

Remark: pose estimation of next images

- Feature tracking
- Texture mapping

Tracking methods

1. Use information from consecutive images
2. "Normal" head pose estimator for first image
3. Next images can use previous estimation and the image

Remark: pose estimation of next images

- Feature tracking
- Texture mapping

Advantages	Disadvantages
Robust	Consecutive images needed

Hybrid methods

- Combine results to get an average head pose
- Use weights to get best head pose

Hybrid methods

- Combine results to get an average head pose
- Use weights to get best head pose

Advantages	Disadvantages
Robust	Multiple estimators needed
	Slower

Geometric method [3]

- Based on results of psychophysical experiments
- Uses features
- Perception of head pose

Chosen approach

Geometric method

Advantages	Disadvantages
Intuitive	Noise sensitive
Simplicity	Features needed
Fast	Head locator needed

Yaw detection: training

1. Detect eyes

   Cascade classifier [4]

   Unable to detect eye $\rightarrow$ position $= (0, 0)$

Yaw detection: training

1. Detect eyes
   Cascade classifier [4]
   Unable to detect eye $\rightarrow$ position $= (0, 0)$
2. Get relative horizontal positions of eyes

Yaw detection: testing

1. Detect eyes (e_1 & e_2)
2. Calculate average position ($p_1 = e_1 + e_2$)
3. Get score of each image
   Calculate average position (p_2)
   Check if amount of eyes matches, otherwise score $= 1,000,000$.
   No eyes, score $= 0$.
   One eye, different eye, score $= 1,000,000$
   One eye, same eye, score $= |p_1 - p_2|$
   Two eyes, score $= |p_1 - p_2|$
4. Best score determines the yaw

Yaw detection: results

method	correct	accuracy	avg. abs. err.	wrong direction	wrong frontal
detect (sorted)	209	86.0%	6.3	4	9
landmarks (sorted)	214	88.1%	8.4	15	0
detect	211	86.8%	5.6	2	9
landmarks	228	93.8%	0.84	0	0
hybrid	183	75.3%	10.3	4	13
landmarks (logical)	212	87.2%	3.4	0	0
hybrid (logical)	191	78.6%	9.3	4	13

Table 1: 20-fold cross validation on the Bosphorus database [6] with 243 samples

Yaw detection: implementation

Eyes
- resize image
- raise minimum neighbours till maximum two eyes are found
- two eyes must be at the same height, otherwise delete lowest
- two eyes overlap $\rightarrow$ remove biggest eye
- if no eyes are found, redo with bigger image

Yaw detection: futher improvements

- Make the estimator more robust to inconsistencies of the head locator
- Use a more accurate feature detector
- Use a continuous pose instead of a discrete pose
- Use a eye detector that knows which is the left and right eye

Yaw detection: futher improvements

- Make the estimator more robust to inconsistencies of the head locator
- Use a more accurate feature detector
- Use a continuous pose instead of a discrete pose
- Use a eye detector that knows which is the left and right eye

Yaw detection: futher improvements

- Make the estimator more robust to inconsistencies of the head locator
- Use a more accurate feature detector
- Use a continuous pose instead of a discrete pose
- Use a eye detector that knows which is the left and right eye

Yaw detection: futher improvements

- Make the estimator more robust to inconsistencies of the head locator
- Use a more accurate feature detector
- Use a continuous pose instead of a discrete pose
- Use a eye detector that knows which is the left and right eye

Pitch detection: training

1. Detect eyes [4], mouth and nose [2]

Pitch detection: training

1. Detect eyes [4], mouth and nose [2]
2. calculate $d1 = d(eyes, nose)$ and $d2 = d(nose, mouth)$

Pitch detection: training

1. Detect eyes [4], mouth and nose [2]
2. calculate $d1 = d(eyes, nose)$ and $d2 = d(nose, mouth)$
3. calculate $r = \frac{d1}{d2}$

Pitch detection: testing

1. Calculate ratio r_{test}

Pitch detection: testing

1. Calculate ratio r_{test}
2. Let every image vote for its pitch
   Image may vote if $|r_{train} - r_{test}| < threshold$

Pitch detection: testing

1. Calculate ratio r_{test}
2. Let every image vote for its pitch
   Image may vote if $|r_{train} - r_{test}| < threshold$
3. Pitch with most votes wins

Pitch detection: testing

Fine tuning: threshold

1. good initial value?
2. decrease threshold when tie $\rightarrow$ more specific wins
3. increase threshold when tie $\rightarrow$ more general wins

Pitch detection: results

method	correct	accuracy	avg. abs. err.	wrong direction	wrong neutral
detect	71	68.3%	0.35	0	20
landmarks	71	68.3%	0.35	1	16
facial normal [1]	51	49.0%	1.1	27	22
facial normal (corr.) [1]	74	71.2%	0.29	0	14

Table 2: The obtained results of the pitch detector

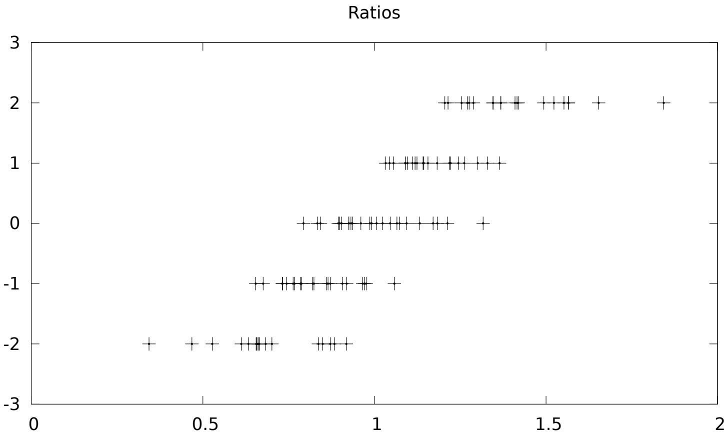

Figure 1: The ratios (x-axis) for the pitches (y-axis) using the cascade classifier.

Pitch detection: ratios

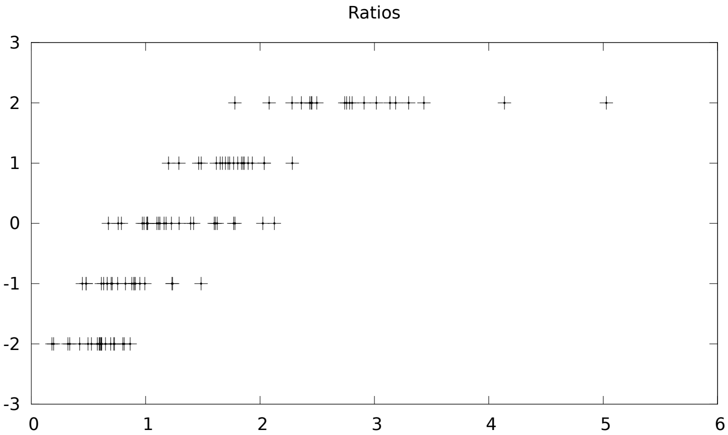

Figure 2: The ratios (x-axis) for the pitches (y-axis) using the landmarks.

Pitch detection: ratios

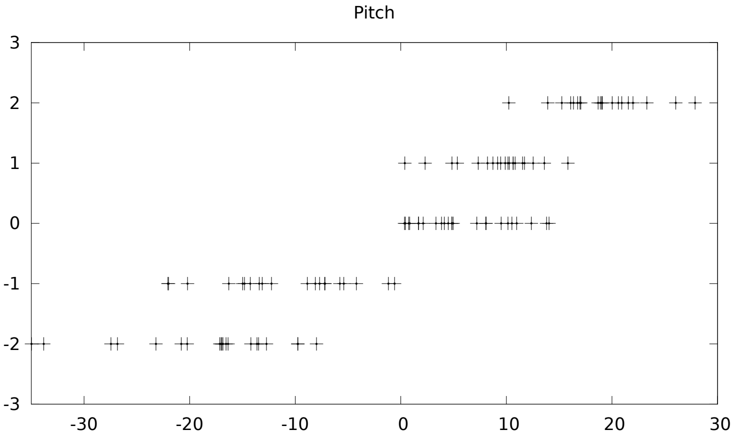

Figure 3: The ratios (x-axis) for the pitches (y-axis) using the facial normal.

Pitch detection: implementation

Eyes	see yaw detection
Nose	resize image raise minimum neighbours till maximum one nose is found if no nose is found, redo with bigger image
Mouth	resize image delete every mouth that doesn't have a point in the lower quarter two mouths overlap $\rightarrow$ delete smallest too many eyes $\rightarrow$ increase minimum neighbours, redo too few eyes $\rightarrow$, redo with bigger image

Pitch detection: futher improvements

- Normalize the ratios $\rightarrow$ deviation head length needed
- We can conclude we've reached the limits of the method $\rightarrow$ new method

Pitch detection: futher improvements

- Normalize the ratios $\rightarrow$ deviation head length needed
- We can conclude we've reached the limits of the method $\rightarrow$ new method

Conclusion

- Yaw detection is far mor easier than pitch detection
- Yaw detector
  better feature detector $\rightarrow$ better results
- Pitch detector: inherently difficult

Bibliography I

[1] Oljira Dejene Boru.
Head pose estimation using opencv.
http://mmlab.disi.unitn.it/wiki/index.php/Head_Pose_Estimation_using_OpenCV.

[2] M. Castrillón Santana, O. Déniz Suárez, M. Hernández Tejera, and C. Guerra Artal.
Encara2: Real-time detection of multiple faces at different resolutions in video streams.
Journal of Visual Communication and Image Representation, pages 130–140, April 2007.

[3] A. H. Gee and R. Cipolla.
Determining the gaze of faces in images.
Image and Vision Computing, 12:639–647, 1994.

Bibliography II

[4] Shameem Hameed.
`http://www-personal.umich.edu/~shameem/haarcascade_eye.html`.

[5] E. Murphy-Chutorian and M. Trivedi.
Head pose estimation in computer vision: A survey.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31 Issue 4(Pages 607-626), 2009.

[6] A. Savran, N. Alyz, H. Dibekliolu, O. eliktutan, B. Gkberk, B. Sankur, and L. Akarun.
Bosphorus database for 3d face analysis.
In *The First COST 2101 Workshop on Biometrics and Identity Management (BIOID 2008)*, May 2008.