Audio-Visual Automatic Speech Recognition

Helge Reikeras

June 30, 2010

SciPy 2010: Python for Scientific Computing Conference

Introduction 1/2

- What?
  - Integration of audio and visual speech modalities with the purpose of enhancing speech recognition performance.
- Why?
  - McGurk effect (e.g. visual /ga/ combined with an audio /ba/ is heard as /da/)
  - Performance increase in noisy environments
  - Progress in speech recognition seems to be stagnating

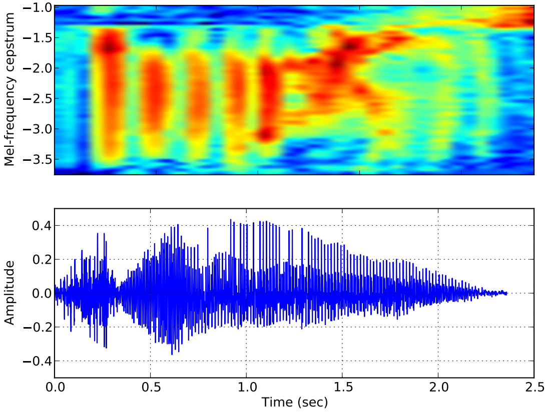
Introduction 2/2

- Example: YouTube automatic captions

Acoustic speech: MFCCs (1/2)

- Mel-frequency cepstrum coefficients (MFCCs).
- Cosine transform of the logarithm of the short-term energy spectrum of a signal, expressed on the mel-frequency scale.
- The result is a set of coefficients that approximates the way the human auditory system perceives sound.

Acoustic speech: MFCCs (2/2)

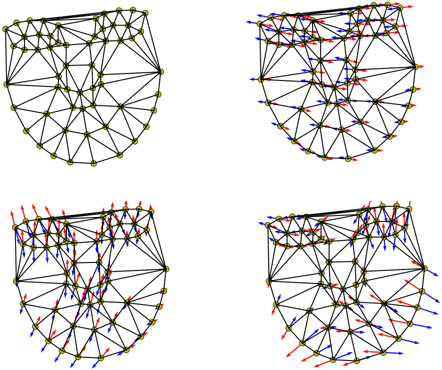


Visual speech: Active appearance models (1/3)

- Visual speech information mainly contained in the motion of visible articulators such as lips, tongue and jaw.

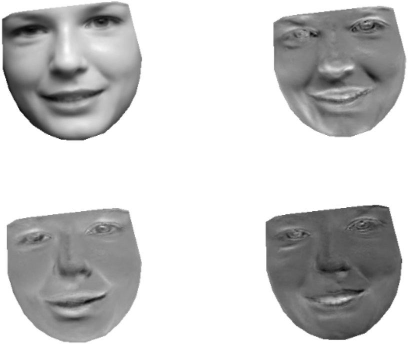
Active appearance models (shape) (2/3)

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{N} p_i \mathbf{s}_i. \qquad \text{(PCA)}$$



Active appearance models (appearance) (3/3)

$$A(\mathbf{x}) = A_0 + \sum_{i=1}^{M} \lambda_i A_i(\mathbf{x}), \qquad \mathbf{x} \in \mathbf{s}_0. \qquad \text{(PCA)}$$



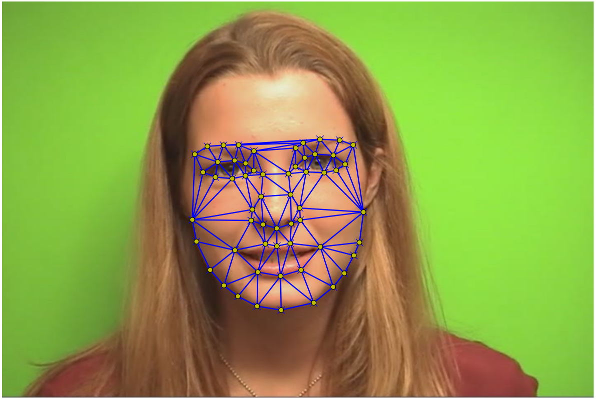
Facial feature tracking (1/2)

- Minimize difference between AAM and input image (warped onto the base shape $\mathbf{s}_0$).
- Warp is a piecewise affine transformation (triangulated base shape).
- Nonlinear least squares problem

$$\operatorname*{argmin}_{\boldsymbol{\lambda},\mathbf{p}} \sum_{\mathbf{x}\in\mathbf{s}_0} \left[A_0(\mathbf{x}) + \sum_{i=1}^{M} \lambda_i A_i(\mathbf{X}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2$$

- Solve using non-linear numerical optimization methods.

Facial feature tracking (2/2)



Modeling: Gaussian mixture models

- Gaussian Mixture Models (GMMs) provide a powerful method for modeling data distributions.
- Weighted linear combination of Gaussian distributions.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Data: $\mathbf{x}$
- Model parameters:
  - Weights $\boldsymbol{\pi}$
  - Means $\boldsymbol{\mu}$
  - Covariances $\boldsymbol{\Sigma}$

Expectation maximization (EM) (1/2)

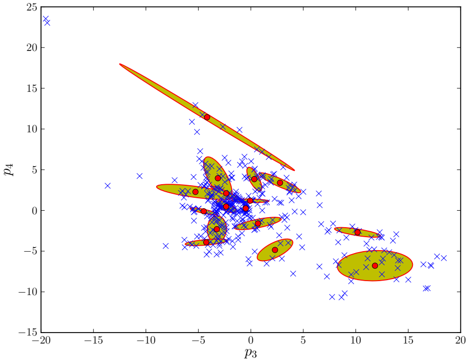
- Log likelihood function gives the likelihood of the data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ given GMM model parameters

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- EM is an iterative algorithm for maximizing the log likelihood function w.r.t. GMM parameters.

Expectation maximization (EM) (2/2)

- Visual EM-GMM (16 mixture components)



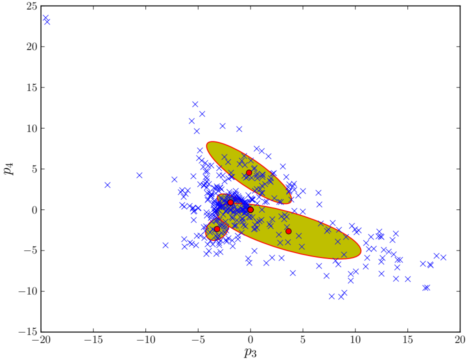
(Note that in practice we use more than 2 dimensional feature vectors)

Variational Bayesian (VB) inference (1/2)

- How do we choose the number of Gaussian mixture components?
- VB differs from EM in that parameters are modeled as random variables.
- Suitable conjugate priors for GMM parameters are:
  - Weights; Dirichlet
  - Means: Gaussian
  - Covariances (precision): Wishart
- Avoids overfitting, singular solutions (when a Gaussian collapses onto a single data point) and leads to automatic model complexity selection.

Variational Bayesian (VB) inference (2/2)

- Visual VB-GMM (16 mixture components)

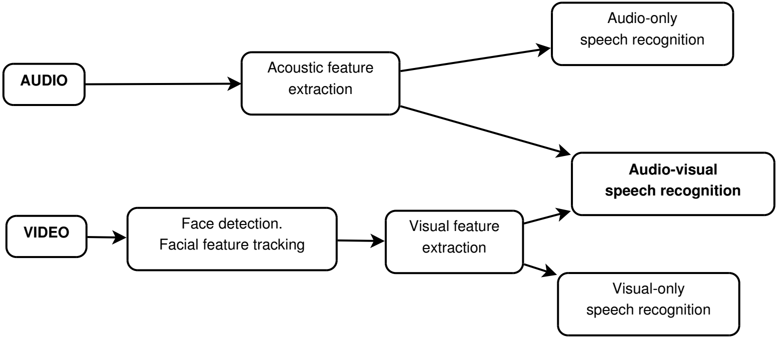


- Remaining components have converged to their prior distributions and been assigned zero weights.

Audio-visual fusion

- Acoustic GMM: $p(\mathbf{x}_A|c)$
- Visual GMM: $p(\mathbf{x}_V|c)$
- Classification (e.g. words or phonemes)
- Stream exponents λ_A, λ_V
- $\text{Score}(\mathbf{x}_{AV}|c) = p(\mathbf{x}_A|c)^{\lambda_A} p(\mathbf{x}_V|c)^{\lambda_V}$
- $0 \leq \lambda_A, \lambda_V \leq 1$
- $\lambda_A + \lambda_V = 1$
- Learn stream weights **discriminatively**.
- Minimize misclassification rate on development set.

Summary


AUDIO
Acoustic feature extraction
Audio-only speech recognition
Audio-visual speech recognition
VIDEO
Face detection. Facial feature tracking
Visual feature extraction
Visual-only speech recognition

Python Implementation

- Implemented in Python using SciPy (open source scientific computing Python library).
- Signal processing, computer vision and machine learning are active areas of development in the SciPy community.
- SciPy modules used:
  - scikits.talkbox.features.mfcc (MFCCs)
  - scikits.image (image processing)
  - scipy.optimize.fmin_ncg (facial feature tracking)
  - scipy.learn.em (EM)
- New modules developed as part of this research:
  - vb (VB inference)
  - aam (AAMs)

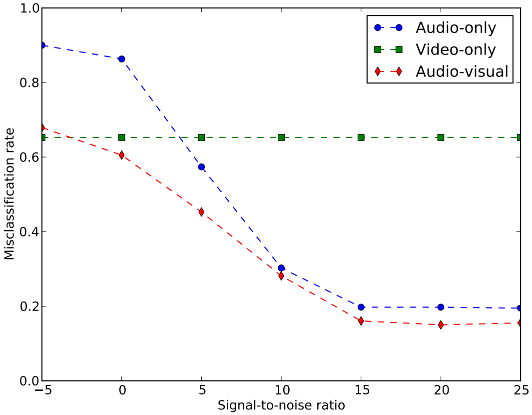
Experimental results (1/3)

- Using the Clemson University audio-visual experiments (CUAVE) database.
- Contains video of 36 speakers, 19 male and 17 female, uttering isolated and connected digits in frontal, profile and while moving.

Experimental results (2/3)

- Use separate training, development and test data sets (1/3, 1/3, 1/3).
- Add acoustic noise ranging from -5dB to 25 dB.
- Test audio-only, visual-only and audio-visual classifiers for different levels of acoustic noise.
- Evaluate performance based on misclassification rate.

Experimental results (3/3)



Conclusion

- Visual speech in itself does not contain sufficient information for speech recognition...
- ...but by combining visual and audio speech features we are able to achieve better performance than what is possible with audio-only ASR.

Future work

- Speech features are not i.i.d. (hidden Markov models) (sprint)
- Audio and visual speech is asynchronous (dynamic Bayesian networks) (GrMPy)
- Adaptive stream weighting
- ...

The end

Thank you!

Any questions?