Texture Synthesis for Material Recognition

Master's Thesis in Articial Intelligence — Intelligent Systems

Jasper van Turnhout
Student no. 0312649
jturnhou@science.uva.nl

November 15, 2011

Task Description

What is the goal of this thesis?

Orientation estimation — direction in which a person is looking — using a *single ceiling-mounted camera*.

Why do we want to estimate orientations?

Useful for *analyzing social interaction* or as a part of a *surveillance system*, or even for *targeted advertisement*.

Why is this difficult?

The large *variance* in data due to the positioning of the camera, the changes in the *illumination*, the *low quality* of the images.

People-detection System & Camera Calibration

- *Backproject* the feet location from *2D* to *3D*.
- Build a *3D* bounding box in real-world considering the average human height.
- *Project* the *3D* bounding box in the image plane.

Data Preprocessing

We extract the foreground by thresholding the *background mask* or using the ground-truth annotations. We rotate the foreground area and the target angles with θ':

$$\theta' = \frac{\pi}{2} + \theta \quad \text{where} \quad \theta = \arctan\left(\frac{y_{head} - y_{feet}}{x_{head} - x_{feet}}\right)$$

Feature Extraction

1. We can extend the data with the *horizontally flipped* versions of the rotated images.
2. We add the *distance* between the projection of the camera coordinates and *the person* to each feature vector.
3. The *motion direction* can also be added to the feature vector.
4. *Multiple features* can be concatenated and *PCA* employed.
5. The feature vectors are normalized to have *zero mean and identity covariance*.

Gabor responses

We convolve the input image with a set of symmetrical, small-scale *Gabor filters* with the orientations: $\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}$. The resulted *Gabor responses* are horizontally concatenated.

Template Matching

8 head-templates are created for the orientations: $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$.

- They are resized to the head size given by the detection-templates.
- Are matched against overlapping regions in the input image using the *OpenCV* template-matching function.

Raw pixel values

For this case multiple situations have been tested:

- Using the grayscale image.
- Using the *saturation channel* of the *HSV* colorspace
- Using the concatenated color channels of the *BGR* colorspace.

Grayscale image, saturation channel & concatenated color channels

Other features

Other implemented features are:

- *Grid of interest points* — uses *Harris corner* detector and *MSERs (maximally-stable extremal regions)*
- *Mask of skin pixels*
- *Edges and contours* — use *Canny edge detector*
- *SURF (speeded up robust features)* descriptors — rely on *Haar-wavelet* responses
- *SIFT (scale invariant feature transform)* descriptors and the *codebook* method
- *HoG (histograms of oriented gradients)* descriptors

Classification vs. Regression

Classification
: Assigns each input vector to one of a finite number of *discrete classes*.

Regression
: Learns to predict one or more *continuous variables*.

The *longitudinal orientation* is a continuous variable in *[0, 360)*.

Due to the discontinuity between *360* degrees and *0* degrees, we learn on the *sine* and *cosine* of the target angles.

Learners for classification and regression

Implemented learners for classification:

- *K-nearest neighbor*, for the experiments k is set to *3*
- *Eigen-orientations*

Implemented learners for regression:

- *Multi-layer perceptron* — 2 layers of hidden units, 100 nodes in each layer
- *Gaussian process*

Regression — Gaussian Process

"The Gaussian process gives a prior probability to every possible function where higher probabilities are given to functions we consider more likely". [Rasmussen and Williams, 2006]

A *Gaussian process* established a distribution over functions evaluated at a set of points $\mathbf{X}_*$: $\mathbf{f}^* \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*))$.

A *Gaussian process* is completely specified by the:

- The *mean function*: $m(\mathbf{x})$ — usually taken to be 0.
- The *covariance function*: $k(\mathbf{x}_i, \mathbf{x}_j)$ — the elements on row i and column j of the covariance matrix $\mathbf{K}$.

Regression — Gaussian Process

We train 2 *Gaussian processes* — for the *sine* and the *cosine* of the prediction, α, and the optimal prediction is $\arctan \frac{sin\alpha}{cos\alpha}$.

The kernel function investigated are: *the squared-exponential*, *the exponential covariance (Matérn covariance with* $\nu = 0.5$*)*, *the Matérn covariance with* $\nu = 1.5$, *the Matérn covariance with* $\nu = 2.5$.

Different *kernel functions* can be employed provided that they correspond to a *positive definite* covariance matrix.

We weight the distance to the camera more than the rest of the feature by increasing the value of the *lengthscale*, in the covariance function, on the corresponding position in the feature vector.

Challenges of the Data

1. The data is *noisy* and of *low quality*.
2. Usually, the *faces of the people* can not be observed.
3. People can be *occluded by background* objects and this makes the detection-system fail in the position estimation.
4. A large variance in the data is generated by the positions of the people *with respect to the camera*.
5. It is difficult to correctly *align the foreground areas*.

Dataset 1

It contains images of *2 people* at 4-5 locations in the ground plane and having orientations in the range [0, 360).

A number of 1345 images are used in the experimental part

Dataset 2

- *14 people* and *8 directions*: $\{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$
- *72 images* annotated for each person — *9 different positions*
- The camera is positioned at at higher altitude, thus *the quality* of the images *is lower*
- Some of the people can be *very distinct* from the rest.

Dataset 3

Data used in *[Ozturk and Aizawa, 2009]* — was recorded in *an airport* with a single *elevated sideway camera*.

Usually, the people are *walking in a specific direction* throughout the whole sequence in which they appear.

Artificial dataset

Artificial data contains: *2* models of men and *2* models of women with 5 different appearances.

- *605* images close to camera's position
- *593* images farther from the camera's position
- *592* images far from the camera's position

Experimental Setup

The performance evaluation methods implemented are:

- the *RMS (root-mean-squared error)*: $\sqrt{\frac{\sum_i |\theta_i - \theta_i'|^2}{N}}$
- the *normalized RMS*: $\sqrt{\frac{\sum_i \frac{|\theta_i - \theta_i'|^2}{\pi^2}}{N}}$
- the *average difference* between the target angles and the predictions: $\frac{\sum_i |\theta_i - \theta_i'|}{N}$

If $\Delta_i = \mid \theta_i - \theta_i' \mid$ is larger than π, the quantity: $\Delta_i \leftarrow (2\pi - \Delta_i)$ is used instead.

These differences, Δ_i, are binned in 18 bins and plotted.

Results 1 — Comparison of learning methods

Dataset 2 (12-fold cross-validation), concatenated color channels on the upper half of the body

Learner	Experimental Settings	RMS Error (Normalized RMS)
NN	*1 network, 2 outputs*	**1.08** *(0.34) — 62 degrees*
NN	*2 networks, 1 output each*	**1.13** *(0.36) — 64 degrees*
NN	*1 network, 4 outputs*	**1.20** *(0.38) — 68 degrees*
k-NN	*2 separate classifiers*	**1.08** *(0.34) — 60 degrees*
GP	*2 separate learners*	**1.01** *(0.32) — 57 degrees*

Results 3 — Generalization over people and positions

Gaussian process on the *concatenated color channels* over the *predicted head area*

Generalization	**Dataset**	**Training/ Evaluation pts.**	**RMS Error** *(Normalized)*
Over people	*Dataset 2*	11/1 (12-folds)	**0.98** *(0.31)* *56 degrees*
Over positions	*Dataset 1*	1/1 (4 folds)	**0.84** *(0.26)* *48 degrees*
Over positions	*Dataset 2*	12/12	**0.82** *(0.26)* *47 degrees*

Results 5 — Results for different setups

Dataset 2 (12-fold cross-validation), learning on the *Gaussian process* over the *concatenated color channels* of the *head area*

Experimental setup	**RMS Error** *(Normalized RMS)*
Baseline	**0.98** *(0.31) — 56 degrees*
Artificial data is added	**1.03** *(0.32) — 59 degrees*
Horizontally flipped version of the images are added	**0.98** *(0.31) — 56 degrees*
The images & targets are not rotated wrt. camera	**1.28** *(0.40) — 73 degrees*

Results 6 — Generalization over orientations

- The training is done on *dataset 1*, the images are randomized and a *5-fold cross-validation* method is used.
- The *concatenated color channels* corresponding to the *head area* are used.
- Two *Gaussian processes* are used for learning the *sine*, respectively the *cosine*.

Training/ evaluation points	**RMS Error (Normalized RMS)**	**Standard** *deviation*
1/1 (5-fold cross-validation)	**0.59** *(0.18)* *34 degrees*	**0.11** *radians* *6 degrees*

Results 7 — Performance on *dataset 3*

Dataset 2 & *Dataset 3*, learning on the *Gaussian process* over the *concatenated color channels* of the *head area*

Training/evaluation people	**RMS Error** *(Normalized RMS)*
864/150 people	**1.07** *(0.34) — 61 degrees*

Conclusions & Future Work

1. The experiments prove that learning is possible and that the correct solution to this problem is a *regression method*.
2. For this problem the *upper half* of the body and the *head area* are more indicated for feature extraction.
3. *Better quality* of the images, *a larger number of people* present in the data and images corresponding to *more orientations*, would definitely improve the results.
4. The ability of training both the *cosine* and the *sine* on the same *Gaussian process* (and maybe the latitudinal angles also) could improve the results.
5. Having an automatic way for determining *specialized lengthscales* for different parts of the features is another extension that could improve the performance.

Demo

The Matérn covariance function

This function becomes simpler if the $\nu = 1/2 + p$ where $p \geq 0$.

$$k_{\nu=p+1/2}(r) = \exp\left(\frac{-\sqrt{2\nu}r}{l}\right)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}r}{l}\right)^{p-i}$$

where $r = \sqrt{\sum_i(\mathbf{x}_i - \mathbf{x}'_i)^2}$ and $\Gamma(n) = (n-1)!$.

- *Matérn covariance with* $\nu = 0.5$

  $\exp(-\frac{1}{l}\sqrt{\sum_i(\mathbf{x}_i - \mathbf{x}'_i)^2})$
- *Matérn covariance with* $\nu = 1.5$

  $(1 + \frac{\sqrt{3}\sqrt{\sum_i(\mathbf{x}_i-\mathbf{x}'_i)^2}}{l})\exp(-\frac{\sqrt{3}\sqrt{\sum_i(\mathbf{x}_i-\mathbf{x}'_i)^2}}{l})$
- *Matérn covariance with* $\nu = 2.5$

  $(1 + \frac{\sqrt{5}\sqrt{\sum_i(\mathbf{x}_i-\mathbf{x}'_i)^2}}{l} + \frac{5\sum_i(\mathbf{x}_i-\mathbf{x}'_i)^2}{3l^2})\exp(-\frac{\sqrt{5}\sqrt{\sum_i(\mathbf{x}_i-\mathbf{x}'_i)^2}}{l})$