Pit Pattern Classification of Zoom-Endoscopic Colon Images using DCT and FFT

Leonhard Brunauer    Hannes Payer    Robert Resch

Department of Computer Science
University of Salzburg

February 1, 2007
Outline

1. Introduction
2. Discrete Cosine Transformation (DCT)
   - Overview
   - Application
3. Fast Fourier Transformation (FFT)
   - Overview
   - Application
4. Pattern Classification
   - Statistical Pattern Classification
   - Feature Extraction
   - Performance Evaluation
5. Results
6. Experiments
   - Color Models
   - Optimization Problem
Goals

- Classify images retrieved from colonoscope.
- 6 class problem
  - Classify according to Pit Pattern classes.
  - Six cancer classes (I, II, III L, III S, IV, V)
- 2 class problem
  - Classify as either “operation required” or “operation not required”.
  - Type I and II need not be removed.
  - Type V cannot be removed.
  - Type III and IV should be removed.
Pit patterns

Type I

Type II

Type IV
Goals (cont’d)

- Previous work done at this university.
  - Wavelet-based approach.
  - Histogram-based approach.

- Work done in this project.
  - Discrete Cosine Transform (DCT) based approach.
  - Fast Fourier Transform (FFT) based approach.
Processing steps

Figure: Processing pipeline
Overview

DCT Overview

- decompose image in blocks of size $n \times n$
- compute 2D-FDCT on every single block

$$F_{x,y} = \frac{2 \cdot C(x) \cdot C(y)}{N} \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{i,j} \cdot \cos\left(\frac{(2i+1) \cdot x \cdot \pi}{2 \cdot N}\right) \cdot \cos\left(\frac{(2j+1) \cdot y \cdot \pi}{2 \cdot N}\right)$$

- $f_{i,j} :=$ pixel $i,j$ of the $n \times n$ input block
- $F_{x,y} :=$ the $x,y$ DCT coefficient of the $n \times n$ DCT coefficient matrix
- $C(x)$ and $C(y)$ are constants

$$C(n) \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n = 0 \\ 1 & \text{if } n \neq 0 \end{cases}$$
DCT Overview

- $F_{00}$ lowest frequency
- $F_{nn}$ highest frequency
DCT Overview

Figure: lowest frequency to highest frequency
DCT and Pitpat

- image size 256 $\times$ 256 pixels
- blocksize 8 $\times$ 8 pixels
  - $\Rightarrow$ 32 $\times$ 32 blocks
  - $\Rightarrow$ 64 DCT coefficients for a block
  - $\Rightarrow$ 65536 DCT coefficients altogether

calculate global information:

- calculate arithmetic mean over DCT matrices
  - $\Rightarrow$ 64 DCT coefficients
DCT Experiments

- different blocksizes \((2^n \times 2^n, n = 1, 2, \ldots)\)
- other statistic tools (standard deviation, variance, \ldots)
- color spaces:
  - YUV luminance channel
  - RGB red, green, blue channel
  - RGB all channels (3 result matrices)
Discrete Fourier Transformation (DFT)

Continuous case:

\[ \hat{f}(u) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} \, dx \]

\[ f(x) = \int_{-\infty}^{\infty} \hat{f}(u) e^{2\pi iux} \, du \]

Discretization:

\[ \hat{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-2\pi iux/N} \]

\[ f(x) = \sum_{u=0}^{N-1} \hat{f}(u) e^{2\pi iux/N} \]
Overview

Discrete Fourier Transformation (DFT)

2D case:

- \[ \hat{f}(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i (ux/M + vy/N)} \]

- \[ f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(u, v) e^{2\pi i (ux/M + vy/N)} \]

Separability:

- \[ \hat{f}(u, v) = \frac{1}{M} \sum_{x=0}^{M-1} \left( \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i vy/N} \right) e^{-2\pi iux/M} \]

Problem:

- High complexity \( O(N^2) \)

\( \Rightarrow \) FFT
**Overview**

**Fast Fourier Transformation (FFT)**

- $O(N \log N)$
- Cooley-Tukey (1965)
- Divide and conquer algorithm (Radix - 2)
- Divide the transform into two pieces of size $N/2$ at each step

\[
\hat{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)e^{-2\pi iux/N} = \\
\frac{1}{N} \sum_{x=0}^{N/2-1} \left( f(2x)e^{-2\pi iu \cdot 2x/N} + f(2x + 1)e^{-2\pi iu \cdot (2x+1)/N} \right)
\]

- Implementation: Technische Universität München, Fakultät für Informatik
Sample Images of 2D-FFT

**Figure:** Sample of Class I with the Fourier-Transformed
Feature Generation

Topologies

Figure: Partitioning of Fourier Spectrum (a) ring filter; (b) wedge filter
FFT Experiments

- Variable number of rings
- Variable width of rings
- Statistic tools (mean, standard deviation, ...)
- Color spaces: YUV, RGB, ...
  - YUV: luminance channel
  - RGB red, green, blue channel
  - RGB all channels
Pattern Classification

- A wide variety of classification approaches exists.
- Statistical pattern classification has been used in this project.
- Classifier must be trained before being ready to use.
- The classifier’s input is a feature vector extracted by FFT/DCT.
- Feature vectors are assigned to one of the classes provided during the training phase.
Statistical Pattern Classification

- For each class, use some probability density function.
- Assign a pattern to the class for which it yields the maximal density.
Statistical Pattern Classification (cont’d)

- Parametric approach
  - Select a statistical distribution (e.g. Gaussian).
  - Use the training set to adjust the distribution’s parameters (e.g. mean, covariance matrix).

- Non-Parametric approach
  - Use the training set to estimate a class’ density function.

- A parametric (Gaussian) approach has been used in this project.
Parameter Estimation - Maximum Likelihood Estimation

- Separate training set into corresponding classes.
- For every class $C$ calculate mean $\mu_C$ and covariance matrix $\Sigma_C$.

$$\mu_C = \frac{1}{|C|} \sum_{x \in C} x$$

$$\Sigma_C = \frac{1}{|C|} \sum_{x \in C} (x - \mu_C)(x - \mu_C)^T$$

- Assume $C$’s distribution is $N(\mu_C, \Sigma_C)$
Statistical Pattern Classification

Bayes Normal Classifier

- Assume that class affiliation is a Gaussian distribution.

Properties
- Good results for small training sets.
- Simple and fast classifier.

Linear decision boundaries
- Use the whole training set to estimate a single covariance matrix.
- This covariance matrix is used as a parameter for every class’ probability density function.

Quadratic decision boundaries
- For every class a separate covariance matrix is generated.
Statistical Pattern Classification

Figure: Linear decision boundary
Statistical Pattern Classification

Figure: Quadratic decision boundary

Leonhard Brunauer, Hannes Payer, Robert Resch

Pit Pattern Classification of Zoom-Endoscopic Colon Images using Discrete Cosine Transformation (DCT) and Fast Fourier Transformation (FFT)
Feature Extraction

Large feature vectors cause severe performance penalties.

A lot of data is irrelevant for classification.

Larger feature vectors may degrade the classifier’s performance.

“Peaking Phenomenon”
Peaking Phenomenon

- Intuitively, increasing the number of features should increase the classifier’s performance.

- Increasing the number of features often degrades the performance of parametric classifiers in practice.

- Parametric classifiers rely on accurate estimates of a class’ mean and covariance matrix.

- Increasing the size of feature vectors decreases the quality of these estimates.

- Number of training samples per class $n$ should be at least ten times the size of a feature vector $d$.

$$\frac{n}{d} > 10$$
Feature Selection (cont’d)

- Create a subset of relevant features.
- Do not transform feature space, but use original features.
- Optimize according to Fisher’s Criterion
  - Keep scatter within each class small.
  - Let scatter between different classes be high.
- For $C$ classes, the dimensionality must not be reduced below $C - 1$. 
Feature Extraction (cont’d)

- Branch-and-Bound search.
- Guaranteed to find optimal solution.
- Reasonable performance for dropping just a few features.
- Bad performance for selecting very small subsets.
- Exponential blowup in worst case.
Classifier Performance

- Separate pattern set into a *training set* and a *test set*.
- Use the training set to adjust the classification algorithm’s parameter.
- Use the test set to analyze the classifier’s performance (i.e. get the rate of patterns that have been classified correctly).
- Leave-one-out method has been used in this project.
  - For every pattern $x$ in the pattern set $P$, use $\{x\}$ as the test set and all other patterns $P \setminus \{x\}$ as the training set.
### Results - 2 classes

<table>
<thead>
<tr>
<th>Channel</th>
<th>Blocksize</th>
<th>Vectorsize</th>
<th>Correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>4</td>
<td>10</td>
<td>64.9%</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td></td>
<td>65.3%</td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td></td>
<td>64.5%</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td></td>
<td>60.6%</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>8</td>
<td>67.0%</td>
</tr>
<tr>
<td>RGB</td>
<td>2</td>
<td></td>
<td>70.4%</td>
</tr>
</tbody>
</table>
# Results - 2 classes

<table>
<thead>
<tr>
<th>Channel</th>
<th>Bands</th>
<th>Band widths</th>
<th>Correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>45</td>
<td></td>
<td>83,8%</td>
</tr>
<tr>
<td>Y</td>
<td>45 - 7</td>
<td></td>
<td>85,1%</td>
</tr>
<tr>
<td>R</td>
<td>40</td>
<td></td>
<td>83,4%</td>
</tr>
<tr>
<td>G</td>
<td>43</td>
<td></td>
<td>83,0%</td>
</tr>
<tr>
<td>B</td>
<td>41</td>
<td></td>
<td>84,2%</td>
</tr>
<tr>
<td>RGB</td>
<td>3 x 14</td>
<td>[1,1,1,2,2,2,8,11,11,9,6,6,9,6]</td>
<td>95,9%</td>
</tr>
</tbody>
</table>
### Results - 6 classes

<table>
<thead>
<tr>
<th>Channel</th>
<th>Bands</th>
<th>Band widths</th>
<th>Correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>3 x 5</td>
<td></td>
<td>58.5%</td>
</tr>
<tr>
<td>RGB</td>
<td>3 x 6</td>
<td></td>
<td>60.7%</td>
</tr>
<tr>
<td>RGB</td>
<td>3 x 7</td>
<td></td>
<td>14.9%</td>
</tr>
<tr>
<td>RGB</td>
<td>3 x 6</td>
<td>[1,1,7,10,5,1]</td>
<td>68.4%</td>
</tr>
<tr>
<td>RGB</td>
<td>3 x 6</td>
<td>[1,1,5,10,9,2]</td>
<td>80.4%</td>
</tr>
</tbody>
</table>
Change the Color Model

already used:

- YUV
- RGB

used for experiments:

- HSV
- HLS
HSV Color Model

- H - Hue $\in (0, 360)$
- S - Saturation $\in (0, 1)$
- V - Value $\in (0, 1)$ (brightness of the color)
HSL Color Model

- H - Hue $\in (0, 360)$
- S - Saturation $\in (0, 1)$
- L - Lightness $\in (0, 1)$
Optimization Problem

FFT Result to optimize

- dynamic amount of bands
- dynamic amount of coefficients in a band
- optimize amount of correct classified images (maximization problem)

⇒ use a genetic algorithm
Genetic Algorithm Design

- fitness function: amount of correct classified images
- chromosome encoding:
  - bit chromosome
  - fixed length (first bits determine amount of used bands, the following bits the amount of used coefficients in a band)
  - max 63 bands
  - max 63 coefficients in a band
  - \( \Rightarrow 6 \text{ bits header} + 6 \times 63 \text{ bits for coefficients in a band} \)
  - \( \Rightarrow 384 \text{ bits altogether} \)
- use tournament selection and 2-point crossover to evolve bit chromosome
- mutation rate: \( \frac{k}{\text{chromosomelength}} \) \( k = 1, 2, 3 \ldots \)
hold FFT coefficients in memory to speed up evolution process

- to calculate for every setting the FFT coefficients is very expensive
- hold the data in main memory is not possible - lack of memory :(
- use a database to handle information - too slow
- use distributed computing technology to distribute data