Understanding Collections of Images

COS 521 Final Project Report

Steven Englehardt, Maciej Halber, Elena Sizikova

January 13, 2014

Contents

Chapter 1

Introduction

1.1 Abstract

This report explores a variety of image properties that make it possible to understand vast collections of images. In particular, we look at how image color, saturation, sharpness, and detail can be extracted and compared between images using methods such as Fast Fourier Transform (FFT) and Singular Value Decomposition (SVD). We seek to understand how the theoretical underpinnings of these two algorithms affect the way the images are created in the first place. Ultimately, we provide a way of decomposing the image into mathematical notation (a descriptor) that differentiates well between a collection of images.

1.2 Background Work

There are many possible situations in which we would need to understand and compare image structure. For example, one might like to search for a location in which a photograph was taken, by looking at all the other available images, and finding the image closest to the search image. Alternatively, one may want to cluster images based on their content, and see what categories the image collection can be decomposed to. Both of these would be easy problems to solve, if the images were annotated with words: textual search is a well-solved problem. However, when the images are not labelled (this is known as unsupervised learning), and the image collection is extremely large, it is impractical to label the images by hand. For such problems, it is important to analyze image content automatically.

Existing methods of image search by analyzing content of the image include Google Goggles and Google Image Search, both are based on similar technology [1], which checks for distinctive points, analyzes lines and textures and finally creates a mathematical model of the image. While the exact implementation is not available, Google Image Search does not provide clustering capabilities of analyzing existing input datasets. A more closer work is that of Oliva and Torralba [2] which create

a GIST descriptor (cite Freedmans work!!!), and use perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) to classify natural images, for example, pictures of coasts, mountains or cities. The authors work with a low dimensional representation of the scene which is collectively known as the *Spatial Envelope*. The properties of the spatial envelope are estimated by means of Discrete Fourier Transform (DFT), Windowed Fourier transform (WFT), as well as spectral properties of the image are estimates (!!!! Expand).

Having completed a graduate course in algorithms, our goal was to understand the results that were obtained by such a study, and specifically answer the question: why can we estimate proporties of the spatial envelope the way [2] does?

Chapter 2

Methods

Three directions of image analysis were analyzed and tested. We first started with a collection of color images that were analyzed by the method of Color Histograms. We also analyzed the properties of the FFT method and the SVD method on the corresponding greyscale images. The resulting descriptors were then compared against each other and subsequently combined into a joint descriptor, that used the information from all three methods to describe images.

2.1 Data

We tested the descriptors on two data-sets ?????

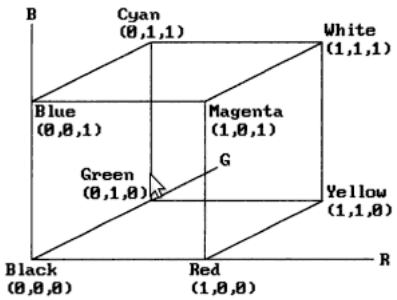
2.2 Implementation

When factoring images using SVD or working with the FFT of an image, we chose to work in grayscale. The choice to do so was motivated by the desire to explore the physical meaning of the decomposition or transformation in the context of images. Though it is entirely possible to separate the red, green, and blue channels and work with each separately, it is difficult to determine whether relative differences in colors or deeper properties of the decomposition/transformation are leading to the observed descriptor performance. To do the conversion we used matlab's built-in *rgb2gray* function, which removes hue and saturation information but preserves luminance. [[I originally was going to discuss this in the SVD section, but I think it fits well here.]]

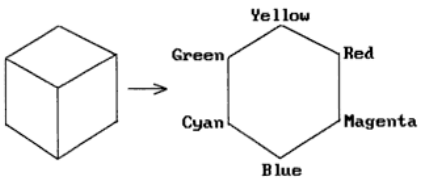
2.3 Color Analysis

We began the exporation of image properties by considering the color analysis method. In particular, the range of image colors that can be seen on a computer monitor are known as *gamut*, see [6]. As we only considered computer images in .jpg and .png formats, we were mainly concerned with the models that represent the gamut. One of the most common such models is known as the RGB model, in which color at every pixel is a combination of three intensity values, of the Red, Green, and Blue colors.

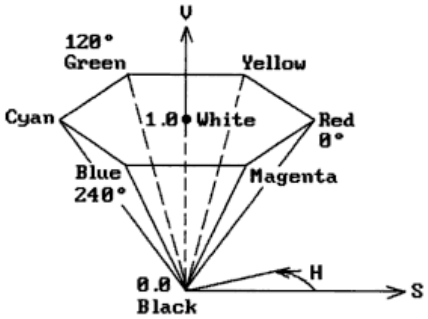
Figure 2.1: RGB Cube Model



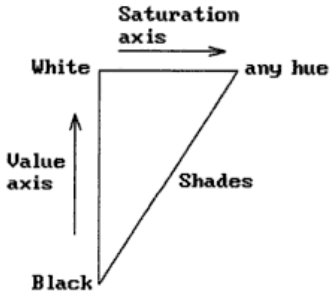
The Hue-Saturation-Value (HSV) Hexcone model is a fairly straightforward transformation from the RGB model, as shown in 2.2. In particular, the hue varies from 0° to 360°, and represents the color. The hue and value are numbers in



(a) The RGB cube viewed along Black-White diagonal



(b) Recoordination



(c) Hue-Saturation Triangle

Figure 2.2: Transformation of the RGB cube to HSV color space

the range $[0, 1]$ which represent how far is the hue from white and black, respectively. As noted further in [6], this model is representative of how artists select colors. We therefore also used the HSV model in our analysis as a first comparison to the RGB model.

[LAB color model intro !!!!!!!!!]

2.4 Fast Fourier Transform (FFT)

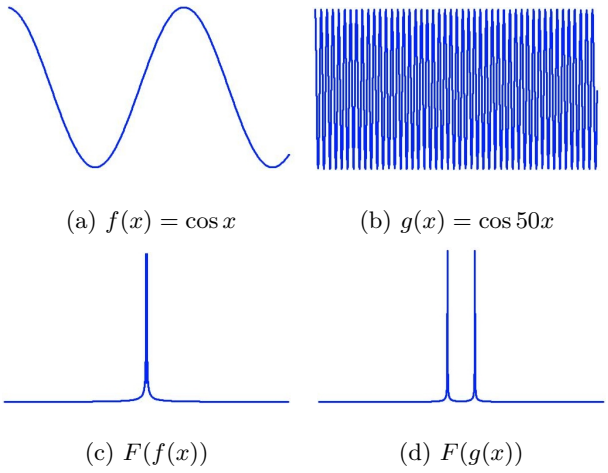
(????? what is the 2D FFT in terms of FFT) A 2-dimensional FFT of a (grayscale) image is a transformation from the spatial domain to the frequency domain. In the spatial domain, an image is represented by function $f(x, y)$ on all relevant points (x, y) in $\mathbb{R}^2$. In the frequncy domain, the image is represented by a function $F(u, v)$ where u and v are frequency values. It follows that in the frequency decomposition, an image is represented by a matrix of complex values F, where $F(u, v)$ encodes the amplitude and the phase of the frequencies u and v. Mathematically, the 2-D FFT is defined as:

$$F(u, v) = \int \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dxdy \tag{2.1}$$

For n points, we can compute the 1-D DFT efficiently in $O(n \log n)$ operations. The Matlab implemetation of the 2-D FFT is extremely fast, and we had no computational time issues with basing descriptors on these computations.

2.4.1 Localization

To understand the properties of the 2-D FFT, we started by analyzying the 1-D FFT first. Consider the following functions, and their representations in the Fourier domain (2.5). A plot of $f(x) = \cos x$ is not localized in the spatial domain, but is localized in the Fourier domain: it is represented by a single peak. Conversely, $g(x) = \cos 50x$ is localized in the spatial domain, as it consists of oscillations around $x = 0$. In the Fourier domain, the same function is no longer localized. This example is representative of the behavior of FFT, and suggested us a way of how to tackle images in the Fourier domain.



(a) $f(x) = \cos x$ (b) $g(x) = \cos 50x$

(c) $F(f(x))$ (d) $F(g(x))$

Figure 2.3: Example of localization properties of different functions in the spatial and Fourier domains. Images localized in the spatial domain are not localized in the Fourier domain, and vice versa.

The given analysis yielded an easy descriptor, in which we sorted the images according the increasing contribution of low frequencies (alternatively, the decreasing contribution of high frequencies) 2.4 (!!!!!! Maciej, how exactly did you generate this?? Is this based on amplitude?). The images should be read as one line that wraps from right to left. One can see that the

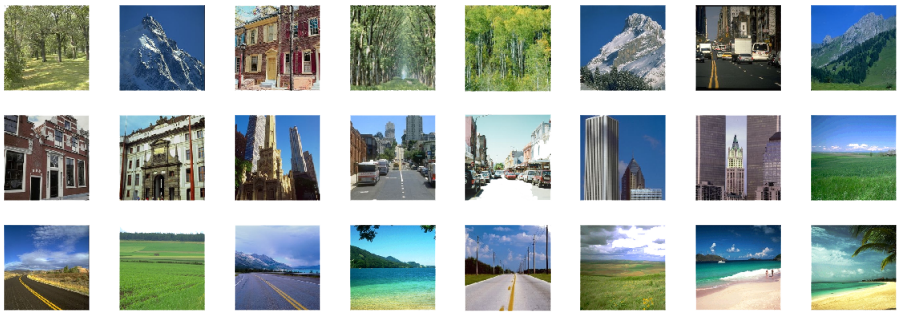
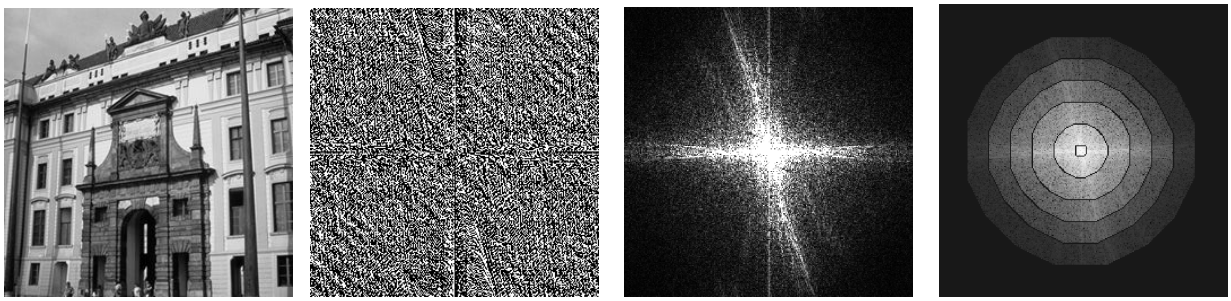


Figure 2.4: Ordering of a Subset of the ???? Dataset by Increasing Contribution of Low Frequencies

images on the left side of the spectrum, with a lot of contribution from high frequencies and not so much from low frequencies have many small details: they show leaves, rock incisions on the mountain, and fine building facade. In comparison, the images at the right side of the spectrum are images of open country, roads, and beaches. These images are simple, in the sense that they have a dominant horizon line and not so much small detail. It follows that these images are described mostly by low frequencies, and not by high frequencies. Notice that this analysis discards a lot of information about the distribution of frequency contribution. We therefore proceeded into analyzing both phase and amplitude footprints of the image in greater detail.

2.4.2 Amplitude and Phase Analysis

Consider the decomposition of an image of a building in a city from the (???????????) GIST dataset into its frequency and amplitude footprints.

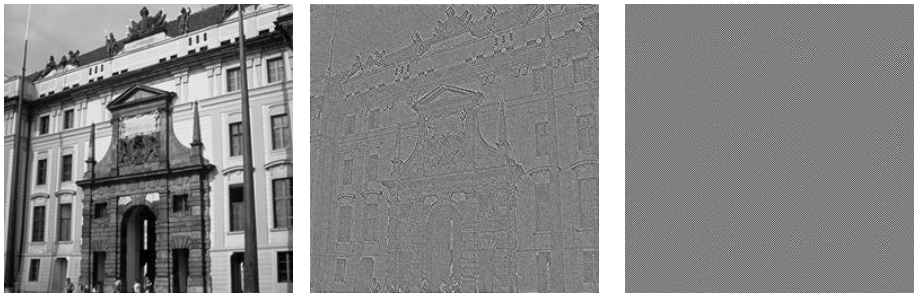


(a) Original image (b) Phase (c) Amplitude (d) Frequency Bins

Figure 2.5: Example of localization properties of different functions in the spatial and Fourier domains. Images localized in the spatial domain are not localized in the Fourier domain, and vice versa.

As Oliva writes in [2], the phase image represents local properties of the image. It contains information relative to the form and the position of image components, while the amplitude talks about orientation, smoothness, length and width of

the contours in the given image. This can be further understood by taking the true phase values of the image, and setting all the amplitude values to 1 (effectively taking out all amplitude variation and flattening the image), or taking true amplitude values and randomizing the phase:



(a) Original image (b) Flat amplitude (c) Randomized phase

Figure 2.6: Analysis of Contributions of both Phase and Amplitude to an Image

Note that a reconstructed image in which the amplitude information was not preserved retains the information about the edges and outlines in the original image. Conversely, the amplitude-preserved image is meaningless to out eyes: it shows the distribution and concentration of color.

2.5 Singular Value Decomposition

A deeper understanding of Singular Value Decomposition (SVD) in the context of images allows the creation of a descriptor that captures overall image complexity in a relatively small descriptor length. SVD is a factorization of any real or complex 2-dimensional matrix. Since images will always be represented by real matrices, we ignore the complex case in our analysis.

Consider an $m \times n$ matrix A. The singular values of A correspond to the non-zero square roots of the eigenvalues from AA^T and A^TA. The matrix AA^T is spanned by the row space of A, and the matrix A^TA is spanned by the column space of A [3]. The row space and column space being the set of all linear combinations of row vectors and column vectors of A, respectively.

SVD conveniently decomposes A, separating out singular values, row space vectors, and column space vectors into three different matrices. The SVD of A is defined as:

$$A_{m\times n} = U_{m\times m} S_{m\times n} V^T_{n\times n}$$

where S is a diagonal $m \times n$ matrix with the singular values of A on the main diagonal (in decreasing order). The columns of U are the eigenvectors of AA^T and the columns of V are the eigenvectors of A^TA [3]. U and V are known as the left and right singular vectors of A, respectively.

The rank of a matrix is informally defined as a measure of the "nondegenerateness" of system of linear equations encoded by that matrix, or more formally is equal to the size of the row space or column space of the matrix. Linearly independent singular vectors correspond to non-degenerate singular values. The rank of a matrix is thus equal to the number of non-degenerate singular values of the matrix, and is also equal to the number of linearly independent vectors in the row space and column space. If all singular values are non-degenerate, A is a full rank matrix and the SVD of A is unique.

2.5.1 SVD and Compression

It is useful to think of each singular value in S as scaling the row and column singular vectors from U and V to generate an 'eigenimage' [4], or the contribution of that specific singular value to the overall image. This construction is what enables image compression through a low-rank matrix approximation. Let $\tilde{A}$ represent the k-approximation of A, where $rank(\tilde{A}) = k$. Thus:

$$\tilde{A} = U(:, 1:k)\tilde{S}V(:, 1:k)^T$$

(a) Original image

(b) Rank 1 to 4 from top left to bottom right

Figure 2.7: Rank k-approximations of an image using SVD

The performance of SVD in image compression shows that there is a significant amount of visual information in low rank approximations of images. Our desire is to capture this low rank information in a concise descriptor for use in image retrieval. An examination of both the singular values and the singular vectors follows.

2.5.2 Retrieval using Singular Values

Image retrieval using all singular values of a matrix has been shown to perform better than a simple distance metric, such as the Mahalanobis distance, Manhattan distance, or Euclidean distance [5]. Several factors contribute to this performance, but relate to the fact that the distribution of singular values in S is connected to the rank of A, the image matrix.

The alignment of an images' strongest color contours to either the horizontal or vertical axis allow it to be fully captured in the left or right singular vectors of an image and thus requires few singular values. The means the image's matrix representation is of lower rank and thus loses less information

Chapter 3

Analysis

[Steve] I suppose that here we should give a brief summary of k-means clustering and multidimensional scaling, and then show the results of the descriptor (with and without windowing) and an analysis of that? I was planning to investigate the properties of SVD and show some basic performance results in the SVD section....but that presents the problem that I evaluate my results with k-means and k-means won't be introduced yet..

E: I think we can introduce K-means, embedding, search by query, and any other similar things we used to test descriptors with in the Methods section, all under one section, say testing framework?

Chapter 4

Suggestions for Further Work

Bibliography

[1] Johanna Wright, *Search by Text, Voice, or Image*. Inside Search: Official Google Search Blog, 2011.

[2] Aude Oliva, Antonio Torralba, *Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope*. International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.

[3] Ientilucci, Emmett J, *Using the singular value decomposition*. Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, 2003.

[4] Andrews, Harry C. and Patterson, C., III, *Singular Value Decomposition (SVD) Image Coding*. IEEE Transactions on Communications, Vol. 24(4): 425-432, 1976.

[5] Jie-xian Zeng; Dong-ge Bi; Xiang Fu, *A Matching Method Based on SVD for Image Retrieval*. Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference on, Vol. 1: 396-398, 11-12 April 2009

[6] Agoston, Max K. *Computer Graphics and Geometric Modelling: Mathematics*. Springer; 2005 edition, ISBN:1852338172