Frequency Analysis of Wood Textures

Encoding of the grain pattern’s orientation distribution for classification, comparison and search queries

Mathias Bernhard
Chair for Computer Aided Architectural Design CAAD, Institute for Technology in Architecture ITA, ETH Zurich, Switzerland
http://www.caad.arch.ethz.ch
bernhard@arch.ethz.ch

Abstract. “Frequency analysis of wood textures” presents the application of Fourier analysis to translate images of wood textures to the frequency domain. With this encoding, a lot more details can be captured by the same amount of data points than with other descriptions in the spatial domain. A small set of overlapping waves with different frequencies, magnitudes and phase angles allows to characterize the main features of the wood’s grain texture and to quantify and classify different samples. The sample’s color information is thereby enhanced with an array of direction vectors, describing the local orientation distribution.

Keywords. Wood; Fourier analysis; pattern recognition; information theory.

INTRODUCTION

Even though this paper deals with the analysis and synthesis of image textures, its primary goal is not to provide new algorithms for seamless texture synthesis used by rendering engines. However a lot of the involved knowledge is gained from the relevant works (e.g. (Szeliski, 2011)) in the field of computer graphics (CG), where these topics are profoundly described.

The goal is rather, to investigate in tools that can help architects in the design process. Architects and designers always used to work with references. With examining different answers to similar questions.

How can the computer learn and help us learn from these references, available in a tremendous abundance and growing.

The task is to enhance this flood of raw data with information that allows to query these collections, to compare them and measure differences and similarities. It assumes, that the sample date never is completely random, but contains some sort of redundancy that allows for quantifications and eventually predictions. The signal “01010101” presents such redundancy and one could also write 5x“01” without a loss of information. Terms for such processes are numerous and range from data mining to knowledge discovery.

WHY WOOD

Textural richness

Wood textures present a huge variety of colors, from bright yellow over fiery red and royal purple to cof-
fee brown. They distinguish from each other in an enormous richness of patterns and structures, like fiddle back flames, burls, bird’s eyes, quilts, masur and curly waves (Figure 1).

At the same time, they share a lot of common features. They all have regions of different densities, denser darker areas and less dense brighter areas. The textures are neither completely random (noise) nor deterministic (grid) but somewhere in between (stochastic). The periodicity of the occurrence of annual rings makes it an ideal topic for investigation with the presented process.

**Natural wood**

To do the experiments presented here with wood also has a provocative side. Wood in the public perception counts as warm, natural, pure, honest, the least processed building material. It is supposed to grow in a wild forest, to be cut, planed and directly nailed on the floor.

Most of construction materials are processed - doped - to improve their fitness to meet the requirements, not only synthetic but also "natural" ones.

Works of Christoph Schindler (Schindler and Salmerón Espinosa, 2011) or Hironori Yoshida (Digitized Grain, scan to production, (Yoshida, 2012)) are good examples of wood being tailored and customized to individual needs.

**GRAIN DIRECTION RECOGNITION**

**Computer vision - pattern recognition**

Computer vision is a field of computer since with an incredibly broad range of applications such as the automated detection of objects, faces, finger prints, gestures, characters or textures in both images and videos. Search engine companies apply techniques like the one discussed here in combination with machine learning (e.g. support vector machines SVM) to determine (or rather make an educated guess) what an image represents. Images are thereby enriched with this additional layer of information and can be searched and retrieved like the text on a web page.

Most of the literature discusses the application of Gabor filters for efficient edge detection and
pattern recognition (Daugman, 1988). With Gabor transforms, the image is tested against all of the samples in a so called filter bank. This bank consists of a series of basic wave patterns with different wavelengths and different phase angles. Thanks to the waves’ nature of cancelling each other out or on the opposite, amplifying each other, the instance of the bank with the strongest response tells about angle and wavelength of the input image.

**Fourier analysis, point cloud and linear regression**

The here discussed procedure applies an adapted method. The core idea however – the translation of the signal, the images color values, from the spatial to the frequency domain – is the same as with Gabor filters.

The first step is to subdivide the original image (Figure 2) into a reasonable number of sample images. Reasonable in this case is to be defined by the variety of local changes that should be taken into account. Smaller samples result in a higher resolution but also in a higher dimensional vector that has to be handled. For the presented study, a resolution of 16 x 9 has been chosen, which results in 144 sample images of 56 x 56 pixels.

The second step of the procedure performs a Fourier analysis of the matrix of luminosity values of each sample image. The idea of the procedure called Fourier Analysis (named after Jean Baptiste Joseph Fourier, 1768-1830) is, that any signal (of any dimension) can be decomposed into a series of sine waves of different frequencies, magnitudes and phase angles. Fourier transform makes a translation from the time domain (or in the case of two dimensional images spatial domain) into the frequency domain. The signals can be sound, heart pulses, stock market prizes or raster images, where the intensities of each row and each column is computed.

The short introduction above is meant to give the reader unfamiliar with Fourier analysis some necessary basic knowledge. The mathematics involved are not described in further detail here. One is referred to specialist literature amply available. Implementations are also available with linear algebra libraries for most programming languages. The
author employs the Java library *Parallel Colt* (Wendykier and Nagy, 2010). More details about the data acquisition part of the process are described in a paper submitted for acceptance (Bernhard, 2013).

Figure 3 shows one of these sample images on the left and the real part of the corresponding Fourier analysis on the right. The images are magnified so that the individual cells are well visible. The Fourier analysis displays the absolute values - therefore black means 0 - in a logarithmical scale, so that differences are perceivable also among very small values. The values are shifted by half of the number of rows and half of the number of columns, so that R0:C0 is in the center of the image.

A 8x8 section of the numbers behind the rendered visualization in Figure 3 are shown in Table 1. The values are the real part of a matrix of complex number. Except for cell R0:C0, the values occur twice, point symmetrically distributed around the origin (R2:C2 = R6:C6 = -0.42). Due to this symmetry, a bright line in Figure 3 (right) is visible, approximately perpendicular to the grain pattern on the left. The process avails itself of this redundancy for the next step.

For a small region around the center (R0:C0, shifted matrix), a number of points is calculated on this basis by scaling the unit vector of each of these cells coordinates with the value of that correspond-

<table>
<thead>
<tr>
<th></th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>R4</td>
<td>0.76</td>
<td>-0.04</td>
<td>-0.41</td>
<td>0.22</td>
<td>0.12</td>
<td>0.22</td>
<td>-0.41</td>
<td>-0.04</td>
</tr>
<tr>
<td>R5</td>
<td>0.51</td>
<td>3.52</td>
<td>-1.77</td>
<td>0.35</td>
<td>0.43</td>
<td>0.50</td>
<td>0.11</td>
<td>0.15</td>
</tr>
<tr>
<td>R6</td>
<td>-0.57</td>
<td>-1.37</td>
<td>-0.42</td>
<td>0.77</td>
<td>0.27</td>
<td>-0.09</td>
<td>-0.81</td>
<td>-0.90</td>
</tr>
<tr>
<td>R7</td>
<td>0.71</td>
<td>0.41</td>
<td>0.56</td>
<td>2.63</td>
<td>0.75</td>
<td>1.24</td>
<td>0.17</td>
<td>0.44</td>
</tr>
<tr>
<td>R0</td>
<td>0.12</td>
<td>0.04</td>
<td>0.03</td>
<td>1.08</td>
<td>42.75</td>
<td>1.08</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>R1</td>
<td>0.71</td>
<td>0.44</td>
<td>0.17</td>
<td>1.24</td>
<td>0.75</td>
<td>2.63</td>
<td>0.56</td>
<td>0.41</td>
</tr>
<tr>
<td>R2</td>
<td>-0.57</td>
<td>-0.90</td>
<td>-0.81</td>
<td>-0.09</td>
<td>0.27</td>
<td>0.77</td>
<td>-0.42</td>
<td>-1.37</td>
</tr>
<tr>
<td>R3</td>
<td>0.51</td>
<td>0.15</td>
<td>0.11</td>
<td>0.50</td>
<td>0.43</td>
<td>0.35</td>
<td>-1.77</td>
<td>3.52</td>
</tr>
</tbody>
</table>

*Table 1* Real part matrix of Fourier transform, shifted by n/2 so that the cell R0:C0 is in the center.
ing cell. For the matrix in Table 1, for example the cell R1:C1, unit vector 0.707/0.707 multiplied with 2.63 results in the point \( P \) in Figure 4 (left).

The dashed line across the point cloud represents the least squares linear regression of that set of data. That is a line who's summed up distance to all of the points is minimized. The line is described as in equation (1).

\[
y = m x + n \tag{1}
\]

where \( m \) is the slope of the line and \( n \) is the intersection with the y-axis. Since all the points of that scatterplot are symmetrically distributed around the origin, \( n \) should always be 0. The formula returns very small values above or below 0 due to rounding errors and they are therefore neglected. To compute the values \( m \) and \( n \), first the arithmetic mean of both the \( x \) and the \( y \) values is calculated as in equation (2).

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{and} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{2}
\]

Slightly different formulas exist for computing linear regression, the one applied here applies a least squares algorithm and is taken from (Sedgewick and Wayne, 2008)

\[
m = \frac{N \sum_{i}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i}(x_i - \bar{x})^2} \quad \text{and} \quad n = \bar{y} - m \bar{x} \tag{3}
\]

where \( N \) is the number of samples. The hereby generated line is one fitting best (with the least deviation) through the point cloud created by the weighted vectors. The main direction of the image texture – the wood grain in the present example – is perpendicular to that line. In addition to the direction of the grain, the predominant wavelengths in the frequency domain can also tell something about the distance of the annual rings and therefore the wood’s hardness – high frequency meaning slow growth resulting in harder wood (see also Figure 8).

Instead of the original array of 896x504 grey values, the image can now be described as a series of 16x9 angles ranging from 0 to \( \pi \). This set adds a new layer of information to the image. It can be queried, compared, measured and categorized. A small variance in this data for example means the grain direction is everywhere approximately the same, the wood has grown straight and there is no branch. Big variance on the other hand describes a lot of knots or burls, therefore the orientations change from one spot of the image to the other.

In computational object recognition, such a set of direction vectors is used to determine whether or not an instance is present in an image or not. This is done with a histogram of oriented gradients (HOG). Even though Figure 5 shows nothing but a couple of white lines on a black background, a bicycle can clearly be detected. Computers do that by being trained with a lot of images containing bikes and then extracting common metrics similar in all of those samples.

**Application**

Wood has always been used as a construction material. Industrialization and logistics gave birth to the need for and production of more and more stand-
ardized, uniform and homogeneous artifacts. These are confected by disassembling the natural and therefore irregular wood to chips and then reassembling it to e.g. sheets of flake board again.

On the design side, tools to create structures beyond the grid (Hovestadt, 2010) have been developed for a while. They demanded for ways to produce all the one-off-a-kind, non-standard parts with the same ease as an industrial manufacturing process. Having irregular, non-standard parts on the design side and irregular, non-standard building material (grown wood) on the other, why not try to bring them together?

The idea has already been treated in a workshop (Schindler, 2012) before. The here presented method is developed for potential application in the construction of a small structure. The design describes the need for a specific piece of wood, the database of available raw material is searched with the desired set of direction vectors as a search term. The best matching unit returned is then compared with target and in a perpetual feedback loop, the design could be readjusted to minimize the misfit.

CONCENTRATION VS. REDUCTION
The strive for saying more with less has a long history called information theory. Pioneers like Morse, Hartley or – maybe most prominently – Claude E. Shannon thought and wrote about how to encode as much information in as little data as possible extensively in the last two centuries (Gleick, 2011).

Not only for transmission and storage of static information, but also for the sake of flexibility in dynamic information, some sort of compression is crucial. With dynamic information, for example the computational description of a building design is meant. A building information model (BIM) that would contain the data about each and every part of an edifice can’t be handled at all because it’s a combinatorial explosion. The limitation of steering knobs does not have to be attained by a reduction of complexity but rather by a concentration on the distinctive most productive handles.

For the case of wood, a translation from the spatial to the frequency domain proves to be particularly efficient. With very little dimensions, the grain texture with spring and summer wood in specific directions can be approximated closely. In Figure 7, each of the 56x56 pixels cells is resampled by applying a 7x7 low pass filter. This corresponds to a reduction of data by more than 95%, but the image’s main features are clearly visible. Not contained anymore are small scale features such as noise, the grain of the paper or dust particles of the pencil.

A very similar technique is used in JPEG image compression. The image is thereby cut up in a grid of small cells of 8 by 8 pixel in size. Depending on the desired compression level, each cell is then represented as the overlapping of more or less different frequencies and angles. Any method of compression (or more generally formulated: abstraction) is only a certain view on things. And because it is a view, it poses a certain filter to the data and in turn, produces its own artifacts. Therefore, any method qualifies best for some uses while being unserviceable for others.
CONCLUSION
A language had to be found, which would allow for a concise description of irregular wood patterns. The pictures are to be enhanced with a layer of information encoding the semi-structured orientation distribution of the wood’s grain.

The presented method gives satisfactory results on images of planed or sanded wood, where the annual rings are responsible for the most dominant differences in brightness.

If the grain of the wood sample is obstructed with something richer in contrast like the parallel traces of a band saw, the results of the analysis are cluttered and partially imprecise.

A more adequately adapted lighting condition in the acquisition of the sample images could eventually reduce the influence of these relief features.

REFERENCES
Bernhard, M 2013, ‘Frequencies of Wood - designing in abstract domains’ in Design Modelling Symposium, Berlin.
