

IRSTI: 28.23.15, 27.35.33

F.F. Lazko

*Moscow State Technological University «STANKIN», Moscow, Russia
(E-mail: fedor.lazko@phystech.edu)*

Overview and Comparison of Python Image Processing Tools with Gabor Functions

Abstract: With the invention and development of digital photography technology, the number of images obtained for various purposes has dramatically increased. So the need arose for efficient methods of processing, transferring and storing them. It is obvious that the methods of working with images should be scientifically grounded and reflect the peculiarities of human visual perception. One of such methods is systems of Gabor functions, which are a basis in the space $\mathbb{L}^2(\mathbb{R})$. Their construction is based on the application of the wavelet theory and multiresolution analysis presented in this article.

The next step after building the necessary mathematical model of images is its efficient and convenient software implementation. Python is a great tool for doing this. The purpose of this article is to provide an overview and comparison of libraries containing ready-made implementations of these functions, both as simple linear filters and as whole wavelet-bases.

Keywords: image processing, multiresolution analysis, wavelet-transform, Gabor functions, Python.

DOI: <https://doi.org/10.32523/2616-7182/2020-132-3-25-30>

I. Introduction

Nowadays, a substantial part of the performed scientific and technical measurements is stored in the form of images. Examples include microscopy slides, quadcopter images of the earth surface, astronomical observations, satellite maps, magnetic resonance images, or computer tomography. Sophisticated software tools, allowing efficient pre-processing of images for subsequent study are required for consequent study of these abundant data sources. The essence of this processing may be, for example, the signal notation as a set of its successive approximations. Sometimes it is necessary for transmission of images, when there is need to transfer their rough version first, and then an improved one. It is possible as well to remove insignificant small-scale details from images when compressing them with no loss in quality. These and many other problems are successfully solved by using a new wavelet theory for research and processing of one-dimensional and two-dimensional (image) signals.

Wavelets are special systems of functions and wavelet transformations, which started to develop rapidly in the mid-80s, early 90s of the XX century. In particular, the international Standard for Image Compression JPEG 2000 was developed based on them. The new wavelet theory was Fourier transformation alternative for signal analyses, processing and transmission. Wavelets and wavelet-transforms make it possible to determine the frequency-time characteristics of the signal, as opposed to Fourier transformation, which provides the spectral characteristic of signals only [1]. This weakness of Fourier transformation was partially eliminated by Gabor windows. Wavelets are commonly used in the theory and practice of signal processing of various nature (in processing and recognition of images, sound and seismic signals, in astronomy and medicine), as well as in mathematics and theoretical physics.

II. Gabor functions and wavelets

It is obvious that the methods of working with images should be scientifically grounded and reflect the peculiarities of human visual perception. Scientists Hubel and Wiesel demonstrated

in their experiments in 1959 that it can be modeled by linear filters, the impulse responses of which depend on the frequency and orientation of visual stimuli [2].

These impulse responses can be approximated with Gabor wavelets, obtained by multiplication of Gaussian window $g(x, y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}$ by sinusoidal wave [3]. Gabor wavelets are special system of functions, allowing to do with the minimum number of expansion terms with sufficiently accurate approximation of the description, and model well the way the visual cortex represents the image.

Wavelet theory made it possible to find and build the basis set out of such function systems. This construction is based on the multiresolution analysis, introduced by Mallat and Meyer [4]. The signal resolution concept is widely used herein, allowing to process just the required parts of the specific tasks.

Coherency of closed sub spaces $\mathbb{V}_j \subset \mathbb{L}^2(\mathbb{R})$, $j \in \mathbb{Z}$ is called the multiresolution analysis in $\mathbb{L}^2(\mathbb{R})$, subject the below is true:

1. $\dots \subset \mathbb{V}_{-1} \subset \mathbb{V}_0 \subset \mathbb{V}_1 \subset \dots \subset \mathbb{L}^2(\mathbb{R})$, $\forall j \in \mathbb{Z}$;
2. $\bigcup_{j=-\infty}^{+\infty} \mathbb{V}_j = \mathbb{L}^2(\mathbb{R})$;
3. $\bigcap_{j=-\infty}^{+\infty} \mathbb{V}_j = \{0\}$;
4. $f(t) \in \mathbb{V}_j \Leftrightarrow f(2t) \in \mathbb{V}_{j+1}$, $\forall j \in \mathbb{Z}$;
5. $f(t) \in \mathbb{V}_j \Leftrightarrow f(t-2^{-j}n) \in \mathbb{V}_j$, $\forall j, n \in \mathbb{Z}$;
6. There is a function (called a refinable function) $\phi \in \mathbb{L}^2(\mathbb{R}_+)$ with which the function system $\{\phi(t-k|k \in \mathbb{Z})\}$ generates an orthonormal basis in \mathbb{V}_0 .

As $\{\phi_{j,n} = \sqrt{2^j} \phi(2^j t - n)\}_{n=-\infty}^{\infty}$ – an orthonormal basis in \mathbb{V}_j (in particular, $\{\phi_{0,n}(t) = \phi(t-n)\}_{n=-\infty}^{\infty}$ – an orthonormal basis in \mathbb{V}_0), then $\phi_{-1,0}(t) = \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) \in \mathbb{V}_{-1} \subset \mathbb{V}_0$ and we obtain the below decomposition

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \phi(t-n). \quad (1)$$

Wavelet space \mathbb{W}_0 is defined as an orthogonal complement to \mathbb{V}_0 in space \mathbb{V}_1 , that is $\mathbb{V}_1 = \mathbb{V}_0 \oplus \mathbb{W}_0$. In space \mathbb{W}_0 the orthonormal basis generates shifts $\{\psi_{0,n}(t) = \psi(t-n)\}_{n=-\infty}^{\infty}$.

From (1) it follows that

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} h[n] \phi(t-n), \quad (2)$$

where $h[n] = \frac{1}{\sqrt{2}} \langle \phi\left(\frac{t}{2}\right), \phi(t-n) \rangle$. Coherency $h[n]$ is interpreted as discrete filter. Analogously, we can demonstrate that

$$\frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} g[n] \psi(t-n), \quad (3)$$

where $g[n] = \frac{1}{\sqrt{2}} \langle \psi\left(\frac{t}{2}\right), \psi(t-n) \rangle$. It is evident, that an orthonormal basis in \mathbb{W}_j can be built by scaling and shifting wavelet ψ .

III. Review and comparison of various implementations of Gabor wavelet-transforms

The next step after building the necessary mathematical model of images is its efficient and convenient software implementation. Python is a great tool for solving the alike problems. Plenty of first-class image processing tools are written on its basis, in particular, those, containing Gabor functions. The purpose of this article is to provide an overview and comparison of libraries, containing ready-made implementations of these functions, both as simple linear filters and as whole wavelet-bases.

One of the most popular and widely used libraries of image pre-processing applications is «OpenCV» (Open Source Computer Vision Library) [1]. This library was launched for the first time in Intel company in 1999 by Gary Bradski, and the first version came out in 2000. Vadim

Pisarevsky joined Gary Bradski to manage the Intel software «OpenCV» development team in Russia. This paper used Python version of its interface for preferred setting and deployment. Its source code written based on C/C++ ensures fast operation and is perfect for high-load computer vision programs.

However, it is worth noting that while reading the images «OpenCV» with function `cv2.imread(<filename>)` flips JPEG-files, but ignores the alpha-channel of the PNG-files (in charge of transparency) herein. The matter is that this library is still designed to work with verified sources. This is, for example, an outdoor surveillance camera, removing images once every second, and doing it for 5 years in the same format and the same resolution. In this case, there is no need for variability in the I / O issue.

The basic principle of image processing using Gabor functions, implemented in this library, is to apply filters with a certain set of parameters to them. Each of these filters can be assigned a certain null space (or kernel) - a matrix, used to calculate the correlation or convolution of a preset image and filter. The kernel itself is a two-dimensional array of "weights" of the neighboring pixels. Moving through the input matrix, it examines all points of the original image inside the neighborhood, specified by the filter mask. For each of the neighborhood points, the element by element multiplication of this point by the relevant weight value is performed. Then the results of the products are added up and the obtained result is set as a new value of the initial point. Thus, we see the original image, taking into account the contribution of the surrounding elements.

Gabor kernel can be obtained by function call `getGaborKernel` from module `cv2`: `kernel = cv2.getGaborKernel((ksize, ksize), sigma, theta, lambda, gamma, phi, ktype = cv2.CV_32F)`. In order to get the result of the linear filter effect on the image shown in Figure 1, we need to use the function call `filter2D` from module `cv2`: `dst = cv2.filter2D(src, cv2.CV_8UC3, kernel)`.

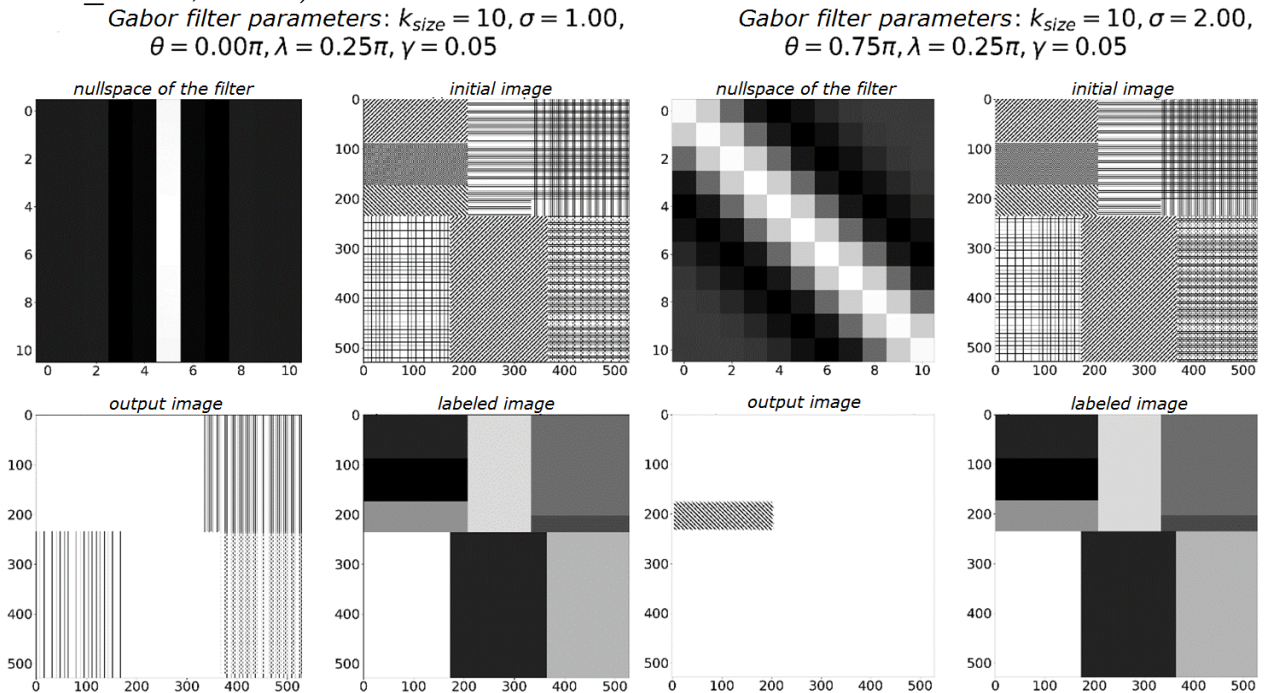


Fig. 1. Examples of Gabor filters with various values of defaults

It is worth noting that function `filter2D` actually calculates correlation with filter, but not the convolution:

$$\text{dst}(x, y) = \sum_{\substack{0 \leq x' < \text{kernel.cols} \\ 0 \leq y' < \text{kernel.rows}}} \text{kernel}(x', y') * \text{src}(x + x' - \text{anchor.x}, y + y' - \text{anchor.y})$$

As is clear from the formula, the null-space is not depicted around the anchor point (anchor). If there is need for convolution specifically (in particular, it is necessary for the implementation of S. Mallat cascade algorithm of fast wavelet transform), one should flip the kernel, using the function `cv2.flip()` (Method `cv2.flip` takes two input parameters – source image and reflation axis. The following numbers can be used as axes: 0 – vertically, 1 – horizontally, (-1) – vertically and horizontally.) and put the new anchor at point (`kernel.cols – anchor.x – 1`, `kernel.rows – anchor.y – 1`).

The next image processing library, worth mentioning, is «*Scikit-image*». Founded by Stefan van der Walt in August 2009, it is a collection of image automated processing algorithms with open and freeware source program code in C, Python or Cython (combination of C and Python languages with improved productivity). Welcome page of the site [2] encourages all future authors, without exception, to pull their fair weight to image processing. It also offers ample opportunities for review, conducted by an active community of volunteers. Despite of plenty of algorithms for segmentation, geometric transformations, filtering, morphology, and feature detection, this library is less specialized than others (such as «*OpenCV*» for computer vision). This library is an active participant to GSoC (The Google Summer of Code) program – annual award of scholarships to students who have made a significant contribution to the development of free and open source software.

In this case, Gabor filter kernel can be obtained by function call `gabor_kernel(frequency, theta=theta, sigma_x=sigma, sigma_y=sigma)` from the library `skimage.filters`. Note that, unlike the previous task, it is not necessary to specify the size of this kernel herein, since its best value will be selected by default: 3 standard deviations (in pixels). Probably, as suggested herein, a more convenient parameter for setting the Gabor filter is the frequency rather than the wavelength, as indicated earlier. Thereafter it is proposed to make convolution of the source image and filter, using `ndimage.convolve(image, kernel, mode='wrap')` from module `scipy`.

«*Bob*» – is also a free, freeware signal processing and machine learning tool, developed by the bio-metric data and privacy protection team together with a team of development engineers at the Idiap Research Institute in Martigny in the canton of Valais in Switzerland. This institute is specialized particularly in research in the field of speech processing, computer vision, information retrieval, bio-metric authentication. «*Bob*» is written in C++ and Python combination and consists of a huge number of packages, implementing various approaches to processing of audio and video signals as well as images. Nevertheless, some packages are not Windows - supported unfortunately.

One of such packages is «*bob.ip.gabor*», containing implementation of multiple functions, required for performance of Gabor wavelet-transform. The user-friendly interface facilitates interpretation of results, obtained in the form of one-dimensional or two-dimensional arrays, comparing them according to different norms. According to developers [2], it proved to be highly efficient in detection, recognition and classification of faces, conducted in the framework of scientific work in the laboratory of Neuroinformatics and cognitive robotics of the Technical University of Ilmenau in Thuringia (Ilmenau University of Technology) in Germany in 2011, and can be used for other applications, related to retrieval of texture information from images. To process the image $I(\vec{x})$ the discrete family of Gabor wavelets $\psi_{\vec{k}_j}$ is used. Gabor transform image results from convolution of the source one with the relevant Gabor wavelet [5]

$$\vec{k}_j(\vec{t}) = \sum_{\vec{x}} \overline{\psi_{\vec{k}_j}(\vec{x} - \vec{t})} I(\vec{x}), \quad (4)$$

where

$$\psi_{\vec{k}_j}(\vec{x}) = \frac{\vec{k}_j^2}{\sigma^2} \exp\left(-\frac{\vec{k}_j^2 \vec{x}^2}{2\sigma^2}\right) \left[\exp\left(i \vec{k}_j^T \vec{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right) \right].$$

It consists of J layers $J_{\vec{k}_j}(\vec{t})$, each of which corresponds to specific frequencies \vec{k}_j with due regard of all possible angles of rotation.

When Gabor wavelet transform occurs in the spatial domain, the resolutions of the image and the wavelet itself can vary a lot, thus summarising in the convolution will be performed by the lowest of these two resolutions. Since in common practice, when a given transformation is performed, Gabor wavelets fade fast at a distance of $3 - 5 \sigma_{eff}$ from the center, we can say that their resolution is limited by this value, which is often less than the source image dimensions. Additional complexity is summing the products of the source image $I(\vec{x})$ and wavelet shifts $\psi_{\vec{k}_j}(\vec{x})$ along all possible directions \vec{t} .

It turns out that the simplest way to get the Gabor image $J_{\vec{k}_j}(\vec{t})$ is to perform transformation in frequency domain. Discretization of Fourier transform is quite well studied, thus it is not a problem to get the Fourier image of the source image. Since the Fourier image of the convolution of two functions is nothing more than the product of images of these functions, the Gabor transform can be obtained by simple multiplication of $F[\psi_{\vec{k}_j}(\vec{\omega})]$ and $F[I(\vec{\omega})]$ pixel by pixel. There will be no resolution problems here, since the function images to be multiplied shall be of the same size. The Gabor transform image, obtained from Fourier transform $F[J_{\vec{k}_j}(\vec{t})]$ can be transferred back to the spatial domain.

Certainly, the information contained in Gabor images, is extremely redundant. From each present value of the pixel we get J complex numbers. To unify the selection procedure for position \vec{t} and its response to the relevant wavelet $\psi_{\vec{k}_j}(\vec{x})$ in [5], the Gabor jet concept was introduced. Each jet is a vector of J complex-valued responses in each position \vec{t} of the source image. The texture information, contained in these vectors, is widely used in the field of pattern recognition. One can compare jets, related to different images, by selecting different methods to calculate the distance, while performing normalization.

Thus, we have reviewed all known Python image pre-processing tools, using Gabor functions. Each of them has a number of advantages and disadvantages. For example, despite the apparent complexity, «OpenCV» opens up possibilities for filter parameters selection. В «Scikit-imageB» – completes definition of the unspecified values with default values, where it can be done, which makes it quite convenient for the first study. «Bob.ip.gabor» is a whole program implementation of Gabor wavelet-transform in frequency domain, which surely makes it most suitable for high-level use (for example, in image recognition tasks).

The author is deeply indebted to his academic adviser, Doctor of Physical and Mathematical Sciences, Professor N. N. Kholshcheynikova for the attention to the work and the comments made.

References

- 1 Добеши И. Десять лекций по вейвлетам. -Москва: Регулярная и хаотическая динамика, 2001. — 464 с
- 2 Hubel D., Wiesel T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex // J. of Physiology. - 1962. - P. 160.
- 3 Daugmann J. G. Two-dimensional spectral analysis of cortical receptive field profile // Vision Research. – 1980. – Vol.20. – P.847–856.
- 4 Малла С. Вейвлеты в обработке сигналов. -Москва: Мир, 2005. -371с.
- 5 Gunther M. Statistical gabor graph based techniques for the detection, recognition, classification, and visualization of human faces: Ph.D. dissertation, Fakultat fur Informatik und Automatisierung, Technischen Universitat Ilmenau. – Ilmenau, Deutschland. – 2011. – [Online]. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.3165rep=rep1type=pdf>.

Ф.Ф. Лазко

"Станкин" Мәскеу мемлекеттік университеті, Мәскеу, Ресей

Габор функцияларын қолдана отырып, Python тілінде кескінді алдын-ала өңдеу құралдарына шолу және оларды салыстыру

Аннотация: Сандық фотосурет технологиясының пайда болу және дамуымен белгілі бір мақсаттарда қолданылатын суреттердің саны күрт артты. Сол себепті оларды өңдеу, тасымалдау және сақтаудың тиімді әдістерін табу қажеттілігі туындады. Суреттермен жұмыс жасау әдістері ғылыми негізделген болып, адамның визуалды қабылдау ерекшеліктерін көрсетуі қажет. Осындай әдістердің бірі $\mathbb{L}^2(\mathbb{R})$ кеңістігінің базисын құрайтын Габор

функциялар жүйесі болып табылады. Олардың құрылысы осы мақалада келтірілген вейвлет пен көпөлшемді анализ аппараттарын қолдануға негізделген.

Суреттің қажетті математикалық моделін құрғаннан кейінгі қадам оның тиімді және ыңғайлы жүзеге асырылуы. Python мұндай есептерді шешудің тиімді құралы болып табылады. Мақаланың мақсаты бұл функциялардың қарапайым сызықты фильтрлер, сонымен қатар бүтін вейвлет-базис ретінде қолданылуларын қамтитын әдебиеттерді салыстырып, шолу жасау болып табылады.

Түйін сөздер: суреттерді өңдеу, көпөлшемді анализ, всплеск-түрлендіру, Габор функциялары, Python.

Ф.Ф. Лазко

Московский государственный технологический университет "Станкин", Москва, Россия

Обзор и сравнение средств предварительной обработки изображений на языке Python с помощью функций Габора

Аннотация: С появлением и развитием технологии цифровой фотографии резко выросло число изображений, получаемых для тех или иных целей. Так возникла потребность в эффективных способах их обработки, передачи и хранения. Очевидно, что методы работы с изображениями должны быть научно обоснованными и отражать особенности зрительного восприятия человека. Одним из таких методов являются системы функций Габора, представляющие собой базис в пространстве $L^2(\mathbb{R})$. Их построение основано на применении аппарата всплесков и кратномасштабном анализе, представленных в этой статье.

Следующим шагом после построения необходимой математической модели изображений является ее эффективная и удобная программная реализация. Python является отличным средством для решения подобных задач. Целью данной статьи является представление обзора и сравнение библиотек, содержащих готовые реализации этих функций как в качестве простых линейных фильтров, так и в качестве целых всплеск-базисов.

Ключевые слова: обработка изображений, кратномасштабный анализ, всплеск-преобразование, функции Габора, Python.

References

- 1 Daubechies I. Ten lectures on wavelets. Society for Industrial and Applied Mathematics, 1992. Mathematics. 357 p.
- 2 Hubel D., Wiesel T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. of Physiology. 1962. P. 160.
- 3 Daugmann J. G. Two-dimensional spectral analysis of cortical receptive field profile, Vision Research, 20, 847-856(1980).
- 4 Mallat S. A Wavelet tour of signal processing. San Diego, CA: Academic Press, 1999. 637 p.
- 5 Gunther M. Statistical gabor graph based techniques for the detection, recognition, classification, and visualization of human faces: Ph.D. dissertation, Fakultät für Informatik und Automatisierung, Technischen Universität Ilmenau. Ilmenau, Deutschland. 2011. [Online]. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.3165rep=rep1type=pdf>.

Сведения об авторах:

Лазко Ф.Ф. – аспирант, Московский государственный технологический университет "Станкин", Москва, Россия.

Лазко F.F. – post graduated student of Moscow State Technological University «STANKIN», Moscow, Russia.

Поступила в редакцию 15.09.2020