Ecole Polytechnique, CMAP

PhD thesis

Rigid-Motion Scattering For Image Classification

Author:
Laurent Sifre

Supervisor:
Prof. Stéphane Mallat

Defended October 6th, 2014

Abstract

Image classification is the problem of assigning a label that best describes the content of unknown images, given a set of training images with known labels. This thesis introduces image classification algorithms based on the scattering transform, studies their properties and describes extensive classification experiments on challenging texture and object image datasets.

Images are high dimensional signals for which generic machine learning algorithms fail when applied directly on the raw pixel space. Therefore, most successful approaches involve building a specific low dimensional representation on which the classification is performed. Traditionally, the representation was engineered to reduce the dimensionality of images by building invariance to geometric transformations while retaining discriminative features. More recently, deep convolutional networks have achieved state-of-the-art results on most image classification tasks. Such networks progressively build more invariant representations through a hierarchy of convolutional layers where all the weights are learned.

This thesis proposes several scattering representations. Those scattering representations have a structure similar to convolutional networks, but the weights of scattering are designed to provide mathematical guaranty of invariance to geometric transformations, stability to deformations and energy preservation. In this thesis, we focus on affine and more specifically on rigid-motion transformations, which consist in translations and rotations, and which are common in real world images.

Translation scattering is a cascade of two dimensional wavelet modulus operators which builds translation invariance. We propose a first separable rigid-motion separable scattering, which applies a first scattering along the position variable to build translation invariance, followed by a second scattering transform along the rotational orbits of the first scattering, to build invariance to rotations.

As any separable representation, separable scattering has the advantage of simplicity but also loses some information about the joint distribution of positions and orientations in the intermediate layers of the representation. We define a joint rigid-motion scattering which does retain this information. The joint scattering consists in a cascade of wavelet modulus applied directly on the joint rigid-motion group. We introduce convolutions, wavelets, a wavelet transform and scattering on the rigid-motion group and propose fast implementations. Both separable and joint scattering are applied to texture image classification with state-of-the-art results on most available texture datasets.

Finally, we demonstrate the applicability of joint scattering and group convolutions on generic object image datasets. It is shown that convolutional networks performances are enhanced through the use of separable convolutions, similar to the rigid-motion convolutions. Also, a non-invariant version of the rigid-motion scattering is demonstrated to achieve results similar to those obtained by the first layers of convolutional networks.

Résumé

La classification d'image consiste à assigner un label à une image inconnue, étant donné un ensemble d'images d'entraînement avec des labels connus. Cette thèse introduit des algorithmes de classification basés sur la transformée en scattering, étudie leurs propriétés et décrit des expériences de classification sur des bases de données de texture de d'images.

Les images sont des signaux de haute dimension pour lesquels les algorithmes d'apprentissage échouent lorsque appliqué directement sur l'espace des pixels. La plupart des approches qui fonctionnent construisent une représentation de basse dimension sur laquelle la classification est effectuée. Traditionnellement, cette représentation est conçue pour construire de l'invariance aux transformations géométriques tout en retenant le plus d'information discriminante. Plus récemment, les réseaux convolutionnels ont supplantés ces représentations sur la plupart des taches de classification d'image. Les réseaux convolutionnels construisent des représentations progressivement de plus en plus invariantes à travers une hiérarchie de couches où les poids sont appris.

Cette thèse propose plusieurs transformées en scattering. Ces représentations ont une structure similaire à celles des réseaux convolutionnels, mais les poids sont conçus pour fournir une garantie mathématique d'invariance aux transformations géométriques, une stabilité aux déformations, et une conservation d'énergie. Cette thèse se concentre sur les transformations affines et plus particulièrement sur les transformations rigides, très courantes dans les images.

Le scattering en translation est une cascade de transformée en ondelette et module, qui construit de l'invariance par translation. Nous proposons un premier scattering séparable, qui applique un premier scattering en translation suivi d'un second scattering le long des orbites de rotation du premier scattering, pour construire l'invariance par rotation.

Comme toute représentation séparable, le scattering séparable a l'avantage de la simplicité mais perd aussi l'information de la distribution jointe de positions et d'orientations dans les couches intermédiaires de la représentation. Nous proposons donc un scattering joint qui consiste en une cascade de transformées en ondelettes module appliquées directement sur le groupe joint des transformations rigides. Nous présentons les convolutions, ondelettes, transformées en ondelettes et scattering sur ce groupe joint de transformations rigides. Les deux scattering, séparable et joint, sont appliqués à la reconnaissance de textures et fournissent des résultats comparables voire supérieurs à ceux de l'état de l'art sur la plupart des bases de données de textures disponibles.

Enfin, nous démontrons l'applicabilité du scattering joint et des convolutions de groupes aux problèmes de classification d'objet génériques. Il est montré que les performances des réseaux convolutionnels sont améliorées par l'utilisation de convolutions séparables, similaires à celles que nous utilisons dans le scattering joint. Par ailleurs, une version non-invariante du scattering joint permet d'attendre des résultats comparables à ceux obtenus avec les premières couches de réseaux convolutionnels.

Acknowledgements

Stéphane, travailler avec toi m'a apporté énormément, merci pour le temps et l'énergie que tu m'as consacré. Tes qualités de scientifique continueront de m'inspirer bien au-delà de ces années de thèse.

Je remercie le DI de l'ENS, et notamment les membres de l'équipe data, Joan, Joakim, Irène, Edouard, Vincent, Mia, Xiu-Yuan, Guy, Matthew et Gilles. Merci aussi aux visiteurs notamment Remco, Charles, Y-lan, Mike pour les discussions intéressantes. Merci aussi aux membres du CMAP et aux autres doctorants et post-docs. Je remercie aussi Bertrand Thirion, son équipe de Neurospin et toute l'équipe de Google Brain, notamment Vincent, Mathieu, Benoît, Rajat, Ilya.

Merci à Claire pour m'avoir soutenue. Merci aussi a mes parents, ma famille, tous mes proches, amis et colocataires qui ont positivement ou négativement impacté la probabilité de l'existence de ce document.

Contents

Chapter 1

Introduction

1.1 Image Classification

Image classification requires to build representations that discard irrelevant information and retain discriminative properties. Irrelevant information correspond to image variability that commonly occurs in all classes of images and therefore do not constitute a clue in the task of determining the image content. Due to the physics of the imaging process, geometric transformations is a good example of such irrelevant variability. Geometric transformations include translations, rotations, dilations, shears and more complex phenomena such as elastic deformations or occlusions. Geometric transformations are often present in comparable range in all classes of images and thus only distracts the classifier. Providing a representation that is invariant to some of these transformations therefore generally increases the classifier performance especially for small number of training examples.

This thesis focuses on building image representations that are invariant to translation, rotation and dilations, while being stable to elastic deformations, and retaining sufficient information to be discriminative. Taken independently, those three problems are easy and can be solved by common tools such as registration, averaging, modulus of Fourier transform. Considered together, they constitute a hard and not very well posed problem.

Scattering operators, introduced by Mallat in [Mal12] tackle this problem by cascading wavelet modulus operators. Each wavelet transform extracts invariant coefficients through averaging and covariant coefficients through convolutions with wavelets followed by non-linearities. All the operations involved in the scattering are stable to deformation and covariant to translation, which makes the whole cascade also stable to deformation and covariant to translation. The averaging component of the wavelet modulus builds the actual invariance to translation. The high-pass components of the wavelet modulus systematically recover the information that is lost through averaging. The recovery provides mathematical guaranties that the energy of the input signal is preserved. Translation scattering of two dimensional images is reviewed in Chapter 2. It has been previously applied to digits and

texture recognition in [BM13].

While the original scattering transform mainly builds invariance to translation, this thesis focuses on extending its architecture to more complex geometric transformation groups, with an emphasis on the rigid-motion group that consists of translations and rotations. Invariance to more general transformations is an important issue and can dramatically improves classification performances on datasets of images containing complex geometric variability. This thesis introduces two possible strategies, the separable and the joint scattering. We discusse their advantages and limitations, study their properties and demonstrate their effectiveness on a wide range of texture and generic object classification datasets.

Chapter 3 introduces a separable scattering, that sequentially builds the translation and rotation invariance. A two dimensional scattering with oriented filters is applied, which is invariant to translation and covariant to rotation. Covariance means that the scattering of a rotated image is equal to the scattering of the original image up to a permutation of scattering coefficients. Separable scattering then builds rotation invariance by applying a second scattering along the rotation variable, which is efficient due to the covariance property.

By building the translation and rotation invariance sequentially, the separable scattering loses information about the joint distribution of positions and orientations within its internal layers. This problem is addressed in Chapter 4 by the joint scattering where the internal layers are considered as multidimensional functions of the joint translation and orientation variable. To build this representation, Section 4.3 reviews multiresolution analysis of the rigid-motion group, and introduces an efficient wavelet transform on this group. Section 4.4 presents the joint rigid-motion scattering, which cascades rigid-motion wavelet modulus operators. The joint scattering is invariant to translation and rotation while retaining the joint information contained in its internal layers.

Texture classification is a fundamental but relatively well posed problem of computer vision for which geometric variability plays a significant role. There have been a large body of work focusing on the design of invariant but informative texture image representations. Chapter 5 presents texture classification results obtained with the separable and joint scattering which are compared with other state-of-the-art methods.

Chapter 6 shows that joint scattering can also be used to tackle more complex and less well-posed problems such as generic object recognition. Section 6.2 adapts the efficient rigid-motion convolution of section 4.3 to the context of deep learning for large scale image classification. We show that a deep network with such separable convolution is faster to process data and requires less data to achieve similar to slightly better final accuracy. Section 6.3 shows that a non-invariant version of the joint scattering performs similarly to the first layers of deep network.

The rest of this introduction briefly reviews the literature on image classification of images of texture or objects, and the construction of geometric invariants. The principal contributions of this thesis are outlined.

1.2 Filters Response Representations

This section briefly reviews filter-based image representations and their invariance and stability properties.

1.2.1 Fourier and Registration Invariants

Building translation invariant representations can be done with common tools such as registration or modulus of Fourier transform or averaging. For a given image $x(u)$, let us denote $\mathcal{L}_v x(u) = x(u-v)$ a translation of x by a vector v. A representation Φx is invariant to translations if a translated image has the same representation as the original

$$\Phi x = \Phi \mathcal{L}_v x \tag{1.1}$$

This property of global translation invariance can be achieved with simple techniques. For example, one can compute an anchor point $a(x)$ as the location of the maximum of x convolved with some filters h

$$a(x) = \arg\max_u x \star h(u) \tag{1.2}$$

where $\star$ denote the spatial convolution

$$x \star h(u) = \int x(v) h(u-v) dv. \tag{1.3}$$

A registration invariant consist in translating the signal with this anchor point

$$\Phi_a x = \mathcal{L}_{a(x)} x. \tag{1.4}$$

Since the convolution is covariant, the anchor point is covariant to translation

$$a(\mathcal{L}_v x) = a(x) + v \tag{1.5}$$

so that the registration is invariant to translation

$$\begin{aligned} \Phi_a \mathcal{L}_v x(u) &= x(u - a(\mathcal{L}_v x) - v) && (1.6)\\ &= x(u - a(x)) && (1.7)\\ &= \Phi_a x(u). && (1.8) \end{aligned}$$

Yet, registration invariants are unstable, which is not ideal for image classification. Indeed, depending on the image content, $x \star h$ may have several close maxima, which would make the anchor point detection very unstable. In addition, registration invariants essentially use the image itself as a representation, which is unstable to small deformations, as we shall later see.

Another possibility to build global translation invariance is to use complex modulus of Fourier transform. The Fourier transform is defined as

$$\mathcal{F}x(\omega) = \int x(u)e^{-i\omega.u}du. \tag{1.9}$$

If an image is translated, its Fourier transform is multiplied by a complex phase

$$\mathcal{F}\mathcal{L}_v x(\omega) = e^{-i\omega.v}\mathcal{F}x(\omega) \tag{1.10}$$

so that its amplitude is translation invariant

$$|\mathcal{F}x_v| = |\mathcal{F}x|. \tag{1.11}$$

Yet, small deformations induce non-negligible distortions of high frequencies. If the Fourier transform is well localized, this may cause the Euclidian distance between the Fourier modulus of an image and its deformed version to be large, even for a small deformation. This is illustrated in Figure 2.1, where a flat texture exhibits periodic patterns which corresponds to very localized bump in the Fourier plane. When the texture is deformed, the bumps spread on the entire Fourier plane which makes the two modulus of Fourier transform difficult to match.

1.2.2 Averaged Filter Responses for Global Translation Invariance

Inspired by Julesz textons [Jul81], many texture representations [FS89, LM01, VZ05] build local and stable non-linear descriptors of image patches, and aggregate those descriptors into a global histogram. The local description of a patch often consists in a vector of non linear filter responses

$$y_\lambda(u) = f(x \star \psi_\lambda)(u) \tag{1.12}$$

where ψ_λ is a local filter and f is a non linearity. For a given patch position u, $\{y_\lambda(u)\}_\lambda$ is a vector indexed by λ which describes the content of the image in the neighborhood of u. The filters ψ_λ are typically smooth oriented filters. Figure 2.3 shows the Morlet filters that will be used throughout this thesis. There are many experiments that suggest that the first visual area V1 of the brain performs similar computations. The smoothness and localization of the filters guaranties that the responses y_λ is stable to small deformations and translation covariant. Covariance means that if the image is translated by v, the filter response will also be translated by the same v

$$f(\mathcal{L}_v x \star \psi_\lambda)(u) = f(x \star \psi_\lambda)(u - v). \tag{1.13}$$

To actually make the responses invariant to translation, a first strategy would be to average them on all patch positions

$$\Phi_\lambda(x) = \sum_u y_\lambda(u). \tag{1.14}$$

This has been used [LW03] for texture classification but the averaging tends to lose too much information. Therefore, more sophisticated methods [BRB+11, ZYZH10] approaches intertwine a coding step between the patch description and the spatial averaging. There are often called bag-of-word methods. The coding step is interpreted as a decomposition of the local descriptor in a dictionary of visual words, and the averaging steps is interpreted as an histogram of the occurrences of the visual word. The coding step lifts the local descriptor into a higher dimensional space, with the aim of limiting the information loss during the averaging step.

1.2.3 Local Descriptors and Transformation Invariance

Describing the patch with the filter responses at only one position is limited and does not capture higher order statistics of the texture. Therefore, researchers sometime use descriptors that consist in the concatenation of the responses at difference offsets v around the patch center u. The concatenated descriptor

$$y_{\lambda,v}(u) = f(x \star \psi_\lambda)(u+v) \tag{1.15}$$

is then encoded and spatially averaged. This augmentation helps to capture more information but also introduces instability to deformation. For example, if the image is slightly dilated with $\tau(u) = (1+\epsilon)(u)$, the components of $y_{\lambda,v}$ that correspond to the patch center $v = 0$ will be translated by $(1+\epsilon)u$ but remain roughly the same

$$f(\mathcal{L}_\tau x \star \psi_\lambda(u)) \approx f(x \star \psi_\lambda((1+\epsilon)u)). \tag{1.16}$$

The components that corresponds to large offset v will be translated as well by $\tau(u)$ but will also moves within the patch at offset $(1+\epsilon)v$

$$f(\mathcal{L}_\tau x \star \psi_\lambda(u+v)) \approx f(x \star \psi_\lambda((1+\epsilon)u + (1+\epsilon)v))) = y_{\lambda,(1+\epsilon)v}(u). \tag{1.17}$$

For this reason, more recent descriptors such a Scale Invariant Feature Transform (SIFT) [Low04] or Histogram of Oriented Gradients (HOG) [DT05] therefore incorporates a local averaging of the response. This local averaging can be implemented as a convolution of responses with low pass filter ϕ

$$y_{\lambda,v}(u) = f(x \star \psi_\lambda) \star \phi(u+v). \tag{1.18}$$

Locally averaged filter responses are stable to the translations within the patch that are induced by small deformations. This explains the success SIFT and its wide use in handcrafted image recognition architectures.

The offsetted deformed position of the center of the patch is $u - \tau(u) + v$ while the deformed offsetted position is $u + v - \tau(u+v)$. The displacement within the patch at offset v is thus

$$(u - \tau(u) + v) - (u + v - \tau(u+v)) = \tau(u+v) - \tau(u) \tag{1.19}$$

which is bounded by $\|\nabla\tau\|_\infty|v|$. Thus, patch offset with larger eccentricity are more sensitive to deformation. Therefore, several descriptors average response proportionally to the patch eccentricity $|v|$. DAISY [TLF10], Gradient Location-Orientation Histogram (GLOH) [MS05] and Rotation Invariant Feature Transform (RIFT) [LSP05] both use a log-polar grid to average orientations on subregions within a patch. Shape context [BMP02] and spin images [LSP05] also use a log-polar grid but average pixel intensities instead of orientations.

All theses local descriptors share the same architecture of convolutions, non-linearity a averaging. While the averaging provides stability to local deformation, it also loses information.

1.3 Hierarchical Invariant Representations

The bag-of-word model has several limitations. First, it builds full translation invariance, which is not always desirable. Secondly, it completely discard the statistics of co-occurence of different words. To be stable to deformation, the descriptor have to be either very local (as in concatenation of point-wise filter response), in which case the final averaging will lose too much information, or the descriptor need to incorporate averaging (as in SIFT) in which case they themselves lose some information. This section reviews the different strategies that have been proposed to overcome those limitations.

1.3.1 Spatial Pyramid

For generic image recognition, full translation invariance is often not a desirable property since the position of sub-objects within an image may not be uniform. For example, the sky usually appears the upper regions while the ground appear in the bottom regions. Spatial pyramid matching [LSP06] therefore uses a hierarchy of dyadic spatial subregions of different size 2^J of the local descriptors. This can be seen as a convolution with several windows ϕ_J of different dyadic width 2^J

$$\Phi_{J,\lambda,u}x = y_\lambda \star \phi_J(u) \tag{1.20}$$

Spatial pyramid improves performance for scene or generic object recognition but is of little use for texture recognition where full translation invariance is a desirable property. Yet, global pooling loses lot of co-occurence information. Several methods have been proposed to replace histogram with higher order statistical quantities that involve multiple layers of operations.

1.3.2 Higher Order Statistics

Fractals are known to accurately describes some classes of natural texture images with repeating structure at different scales such as tree leaves or branchs. Multifractal spectrum

[XJF09] first calculates averaged local filter responses at different scales 2^j

$$y_j(u) = f(x \star \psi_j) \star \phi_j(u) \tag{1.21}$$

and then computes a density $\alpha(u)$ by linearly fitting the slope $\log(y_j)(u)$ versus j at each position u. For each quantized value, the density $\alpha_i(u)$ is a binary image of which the fractal dimension is computed. The multi fractal spectrum representation is the concatenation of the fractal dimension corresponding to all the quantized values. If we interpret the quantized density α as a code, this is similar to bag-of-word model for which the global averaging of each map $\alpha_i(u)$ would be replaced by the estimation of the fractal dimension, which is sensitive to the relative activation of the code at different positions. Wavelet Multifractal spectrum [XYLJ10] is similar but uses several oriented filters instead of one.

Log gaussian COX processes [HGFB11] begins with the computation of encoded descriptors $\alpha_i(u)$ but replaces the spatial histogram of α_i with an estimation of the multivariate covariance of the underlying process. Similarly, Basic Image Feature (BIF) [CG10] estimate the covariance of α_i along scales. They use a small sized dictionary with 6 words and computes the encoded response α_i at four different scales 2^j. Instead of computing one histogram per scale, [CG10] compute a joint histogram of occurrences of sequences of 4 words along scale at the same position, which is of dimension 6^4.

If those strategies can achieve state-of-the-art results on texture datasets [XJF09, HGFB11], they make assumptions on the underlying processes which may not generalize to other classification tasks.

1.3.3 Deep Convolutional Networks

Deep networks [LKF10, HS06, KSH12, DCM+12] are generic hierachical representations with little a priori information. A deep network consists in a cascade of linear operators $\mathcal{W}_1, \dots \mathcal{W}_M$ intertwined with non-linearities f. It typically computes successive layers $\Phi_m x$ with

$$\Phi_m x = f(\mathcal{W}_m f(\mathcal{W}_{m-1} \dots f(\mathcal{W}_1 x))). \tag{1.22}$$

The last layer $\Phi_M x$ should produce the desired output. A cost function $E(\mathcal{W}_1, \dots \mathcal{W}_M)$, quantifies how much the network fits to the desired output. In the context of classification, the desired output is the image label and the cost function is often implemented as a soft-max function. The weights of the linear operators $\mathcal{W}_1, \dots, \mathcal{W}_M$ are usually learned by minimizing the cost function E with a stochastic gradient descent. The derivatives $\partial E / \partial W_m$ of the cost function with respect to the weights are obtained by back propagating [LLB+98] the error from the last layer $\Phi_M x$ all the way down to the input.

When the input vector becomes large (e.g. of dimension 100,000, which is common for images), a vanilla deep network requires to learn a huge number of weights and becomes impractical. Convolutional networks [LBD+89, LKF10, KSH12, DCM+12] limit the number of weights by restricting the operators to local connectivity and by sharing the weights

across different spatial positions. Their internal layers $\Phi_m x(u, p_m)$ are indexed by the spatial position u and some other variable p sometime called the depth of the layer. The next linear operator computes

$$(\mathcal{W}_{m+1}\Phi_m x)(u, q) = \sum_{\substack{v \in \Omega \\ p}} w_{v,p,q}\ \Phi_m x(u + v, p). \tag{1.23}$$

Local connectivity means that the sum is limited to small offsets $v \in \Omega$. Such layers are also called convolutional because their weights $w_{v,p,q}$ do not depend upon the position u in the convolution. These two constraints dramatically reduce the number of weights, make the computations faster and parrallelizable, require less data to learn and prevent the network from overfitting.

Deep networks arguably do not have to be intrinsically invariant since they can learn invariance from data. In practice they often use a number of tricks that do enforce invariance. In particular, most successful networks for image classification [LKF10, HS06, KSH12, DCM+12] make intensive use of pooling non-linearity that explicitly build translation invariance and have practical advantages of limiting the number of nodes by downsampling spatial position.

Chapter 4 introduces convolutions and multiresolution analysis on larger groups than translation, in particular on the rigid-motion group consisting of translations and rotations. Inspired by those rigid-motion convolution, Section 6.2 introduces generic separable convolutional layers, which further reduce the number of parameters in a convolutional network. It is shown that a deep network with separable convolutions processes data faster and requires less data to achieve similar to slightly better accuracy on large scale object classification tasks, compared to the same network were convolutional layers are implemented as dense convolutions (1.23).

1.3.4 Translation Scattering

A translation scattering network is a particular instance of deep networks where the weights and the structure of the network are designed in such a way as to provide mathematical guaranties of translation invariance, energy conservation and stability to deformation. Scattering networks were introduced [Mal12] and first applied to image classification in [BM13]. They are reviewed in Chapter 2. A scattering network recursively separates a signal into its translation invariant part, which is output, and its covariant part, which is propagated to deeper layers. The input layer $U_0 x = x$ consists in the image itself. Its translation invariant part is obtained through convolution with a window

$$S_0 x(u) = x \star \phi(u). \tag{1.24}$$

$S_0 x$ is called scattering of order 0. It is invariant to translation up to the width of the window ϕ. The window can be arbitrary large, it can even be a fully delocalized constant

in which case the representation will be fully translation invariant. The covariant part of x is obtained through convolutions with complementary multi-scale and multi orientation complex high pass filters ψ_λ, also called wavelets, followed by a complex-modulus non-linearity

$$U_1 x(u, \lambda) = |x \star \psi_\lambda|(u). \tag{1.25}$$

Since $U_1 x$ is not translation invariant but covariant, its invariant part is obtained with a convolution with the window ϕ

$$S_1 x(u, \lambda) = |x \star \psi_\lambda| \star \phi(u). \tag{1.26}$$

Again, the covariant part of $U_1 x$ is captured by convolutions with wavelet-modulus

$$U_2 x(u, \lambda_1, \lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|. \tag{1.27}$$

Those definitions are extended by induction to any scattering order m

$$\begin{aligned} S_m x(u, \lambda_1, \ldots, \lambda_m) &= |\ldots|x \star \psi_{\lambda_1}| \ldots \star \psi_{\lambda_m}| \star \phi(u) && (1.28)\\ &= U_m x(., \lambda_1, \ldots, \lambda_m) \star \phi(u) && (1.29)\\ U_{m+1} x(u, \lambda_1, \ldots, \lambda_{m+1}) &= |\ldots|x \star \psi_{\lambda_1}| \ldots \star \psi_{\lambda_{m+1}}(u)| && (1.30)\\ &= |U_m x(u, \lambda_1, \ldots, \lambda_m) \star \psi_{\lambda_{m+1}}(u)|. && (1.31) \end{aligned}$$

The $m - th$ covariant layer $U_m x$ is decomposed into its invariant part $S_m x$ through averaging with window ϕ, and its covariant part $U_{m+1} x$ through modulus of convolutions with wavelets $\{\psi_{\lambda_{m+1}}\}_{\lambda_{m+1}}$. Covariant layers $U_m x$ are indexed with the spatial position u and a sequence $p = \lambda_1, \ldots, \lambda_m$ of wavelet indices which the signal x has been successively filtered with. The sequence p is called a scattering path. Invariant layers $S_m x$ are called scattering of order m, m being the number of wavelets the signal has been successively filtered with. The scattering representation consists in the concatenation of all invariant layers from order 0 to some maximum order M

$$Sx = \{S_0 x, \ldots, S_M x\} \tag{1.32}$$

All layers $S_m x$ of the translation scattering end with an windowing $\star\phi$, which makes the representation invariant to translations up to the window size. Each layer is decomposed into an invariant and covariant part so that there is no loss of energy. Unlike Fourier or registration invariant, the translation scattering only involves operations which are stable to deformation. As a result, translation scattering is stable to deformation.

Translation scattering has been previously applied classification problem to image classification problems where slight or total translation invariance with stability to deformation are important properties [BM13]. Yet, it is much less general than deep networks which can potentially learn arbitrary geometric invariants. This thesis focuses on extending the scattering representation to more complex geometric groups such as the rigid-motion group

consisting of translations and rotations. As it is the case for translation, those groups of transformations are common in natural images and some of them are simple enough so that it is possible to design an invariant representation with the same stability and energy preservation than translation scattering.

1.4 Invariance to Geometric Transformation

1.4.1 Elastic Deformation and the Affine Group

A deformation which maps u to $u - \tau(u)$ can be locally approximated by a translation of $\tau(u)$ and a linear transformation $\mathbb{1} - \nabla\tau(u)$. Indeed, if τ is sufficiently smooth, a Taylor extensions of τ shows that

$$u + v - \tau(u + v) \approx u - \tau(u) + (\mathbb{1} - \nabla\tau(u))v. \tag{1.33}$$

When the linear part $\nabla\tau$ of the deformation is small, translation invariance and stability to deformation are sufficient properties. When this is not the case, the representation should explicitly build invariance to the linear transformation $\mathbb{1} - \nabla\tau(u) \in GL(\mathbb{R}^2)$.

Any linear transformation can be uniquely decomposed [MY09] into a rotation r_θ, a dilation and a shear. Rotations and dilations constitutes groups, which means that the successive application of two rotations is equivalent to the application of a single rotation of angle the sum of the angle. The same result holds with the product of factor of dilations. Therefore, similar strategies as the one used to build translation invariance can be considered. Shear however do not constitute a group. To build invariance to shear, one should either consider the linear group $GL(\mathbb{R}^2)$ as a whole, which raises many complexity and parameterization issues, or use a more ad hoc strategy for shear such as registration [MY09, LSP05]. Therefore, this thesis focuses on rotation and dilation, with an emphasis on rotation which also has the advantage of being periodic.

1.4.2 Separable Invariants

The rigid motion group contains all possible pairs of translation and rotations $g = (v, r_\theta)$. A possible strategy to build invariances to this group is to build a first representation that is invariant to translations and covariant to rotations, and to append a second representation on the rotations orbit to build invariance to rotation. Such a strategy is called separable, in the sense that it processes variables v and θ separately. Separable invariants have the advantage of simplicity and have been very popular in the texture classification literature. A typical example is the Local Binary Pattern Histogram Fourier (LBP-HF) [ZAMP12] representation, which computes a first translation invariant histogram of binary patterns over the whole image. Each image patch is associated with the closest binary pattern in a dictionary. The dictionnary is covariant to rotations, which means that for every pattern in the dictionary, all its rotated versions are also in the dictionary. The rotation invariance

is build separately through a modulus of Fourier transforms performed over the angular difference that relates two rotated patterns. Other strategies involve registration, which can be done by detecting local dominant orientation as in SIFT [Low04], or averaging along the orientation as in WMFS [XYLJ10]. As translation invariants, rotational invariants in image representation often suffer from either instabilities or too large information loss.

1.4.3 Separable Rigid-Motion Scattering

There is a wide variety of possible choices for the first and second invariants when constructing separable invariants. We argue that the desired properties for the rotation invariant are the same as for translation invariant, i.e. information preservation and stability. We therefore propose a separable rigid-motion scattering $\mathring{S}$ that will apply a first translation scattering followed by a scattering along the orientation variable. The first and the second scattering operators are both stable and preserve marginal information so that the resulting separable operator $\mathring{S}$ is also stable and captures important features of the image.

For texture images, a translation scattering with fully delocalized window ϕ is already invariant to translations. If it is constructed with dilated and oriented wavelets

$$\psi_{\theta,j}(u) = 2^{-2j}\psi(2^{-j}r_{-\theta}u) \tag{1.34}$$

then the scattering of a rotated image $\mathcal{L}_\theta x(u) = x(r_{-\theta}u)$ is

$$S\mathcal{L}_\theta x(\theta_1, j_1, \dots, \theta_m, j_m) = Sx(\theta_1 - \theta, j_1, \dots, \theta_m - \theta, j_m) \tag{1.35}$$

A rotation of θ of the original image x corresponds to a translation of vector $\theta(1, 0, \dots, 1, 0)^T$ of the scattering Sx. The rotation invariance can thus be cast as a translation invariance problem along the direction $(1, 0, \dots, 1, 0)^T$. Therefore, the scattering path p is factorized into disjoints orbits indexed by $\bar{p} = (0, j_1, \bar{\theta}_2, j_2, \dots, \bar{\theta}_m, j_m)$ and position θ_1 within an orbit such that

$$p = \theta_1.\bar{p} = (\theta_1, j_1, \bar{\theta}_2 + \theta_1, j_2, \dots, \bar{\theta}_m + \theta_1, j_m) \tag{1.36}$$

When the image is rotated by θ, the path $\theta_1.\bar{p}$ is moved to $(\theta_1 - \theta).\bar{p}$

$$S\mathcal{L}_\theta x(\theta_1.\bar{p}) = Sx((\theta_1 - \theta).\bar{p}) \tag{1.37}$$

but the index $\bar{p}$ is not changed, and is therefore called an orbit. We denote $\theta_1.\bar{p}$ rather than $\theta_1, \bar{p}$ because . denotes the action of the group variable θ_1 on the path $\bar{p}$, not to be confused with , which denotes the concatenation of variables. The convolution of a function h of the path variable $p = \theta_1.\bar{p}$ with a filter $\mathring{\psi}_k(\theta)$ is defined as the regular periodic convolution $\circledast$ along the variable θ_1 with fixed orbit $\bar{p}$:

$$h \circledast \mathring{\psi}_k(\theta_1.\bar{p}) = \int_{[0,2\pi)} h((\theta_1 - \theta).\bar{p})\mathring{\psi}_k(\theta)d\theta \tag{1.38}$$

Separable rigid-motion scattering applies a second scattering to $S_m x(\theta_1.\bar{p})$ to build invariance to rotation. This yields another cascade of modulus wavelets $\mathring{\psi}_{k_1}, \dots \mathring{\psi}_{k_{\mathring{m}}}$ convolutions that ends with low pass $\mathring{\phi}$ along the rotation parameter

$$\begin{aligned}
\mathring{U}_{m,0}x(\theta_1.\bar{p}) &= S_m x(\theta_1.\bar{p}) && (1.39)\\
\mathring{S}_{m,\mathring{m}}x(\theta_1.\bar{p}, k_1, \dots, k_{\mathring{m}}) &= |\dots|S_m x \circledast \mathring{\psi}_{k_1}| \dots \circledast \mathring{\psi}_{k_{\mathring{m}}}| \circledast \mathring{\phi}(\theta_1.\bar{p}) && (1.40)\\
&= \mathring{U}_{m,\mathring{m}}x(., k_1, \dots, k_{\mathring{m}}) \star \mathring{\phi}(\theta_1.\bar{p}) && (1.41)\\
\mathring{U}_{m,\mathring{m}+1}x(\theta_1.\bar{p}, k_1, \dots, k_{\mathring{m}+1}) &= |\dots|S_m x \circledast \mathring{\psi}_{k_1}| \dots \circledast \mathring{\psi}_{k_{\mathring{m}}}(\theta_1.\bar{p})| && (1.42)\\
&= |\mathring{U}_{m,\mathring{m}}x(., k_1, \dots, k_{\mathring{m}}) \circledast \mathring{\psi}_{k_{\mathring{m}+1}}(\theta_1.\bar{p})|. && (1.43)
\end{aligned}$$

As the translation scattering (1.28-1.31), the separable rigid-motion scattering recursively separate covariant signal $\mathring{U}_{m,\mathring{m}}x$ into a rotation invariant part $\mathring{S}_{m,\mathring{m}}x$ through averaging with window $\mathring{\phi}$ and a rotation covariant part $\mathring{U}_{m,\mathring{m}+1}x$ through modulus of convolution with wavelets $\{\mathring{\psi}_{k_{\mathring{m}+1}}\}_{k_{\mathring{m}+1}}$. Both $\mathring{U}_{m,\mathring{m}}x$ and $\mathring{S}_{m,\mathring{m}}x$ are indexed by $(\theta_1.\bar{p}, \mathring{p})$ where θ_1 is the rotation variable, $\bar{p}$ is the rotational orbit index of the translation scattering, and $\mathring{p}$ is the scattering path along the rotation, i.e. the sequence of wavelet indices $k_1, \dots, k_{\mathring{m}}$ the signal has been filtered with along the rotation. Separable scattering ends with a low pass filtering with window $\mathring{\phi}$ which actually builds the rotation invariance, up to the width of the window. If the window is a fully delocalized constant, the separable rigid-motion scattering is fully rotation invariant and hence does not depends upon θ_1. Separable rigid-motion scattering has the same properties with respect to rotation as the scattering had with respect to translation. It is therefore invariant to rotation, stable to deformation and preserves the marginal distribution of all rotation orbits of the translation scattering. It does not suffer from the instabilities or the information loss of other invariants such as registration or averaging. Also, separable scattering is relatively simple to implement since we essentially apply twice the same operator in different directions and with different wavelets. This can be sufficient to obtain good results on relatively simple classification problems such as the Outex texture dataset [OMP+02]. Yet, we shall see in section 1.5 that separable scattering, as any separable operator, is in certain situations a too strong invariant that is not able to discriminate simple textures.

We introduced the separable scattering, originally denoted as "combined" scattering, in the paper [SM12] with state-of-the-art results on the Outex10 texture dataset [OMP+02]. Chapter 3 introduces it in more details and texture classification experiments are reviewed in Chapter 5.

1.5 Joint Invariants and Joint Rigid-Motion Scattering

Separable invariants have the advantage of simplicity since one can think of them as a cascade of independent black boxes, each of which builds invariance in different direction. Yet, separating variables produces too strong invariance and yields representations that are

not sufficiently discriminative. Section 4.2 gives an intuition of what kind of information is lost by separable invariants. Section 4.3 introduces multiresolution analysis on transformations, to be used in Section 4.4 in the construction the joint rigid-motion scattering $\widetilde{S}$. Joint scattering is a joint operator that process the different parameters of a group together rather than separately. Therefore, it is a tighter invariant that captures information to which separable invariants are blind.

1.5.1 Joint versus Separable Invariants

To understand the information that is lost by processing different variables separately, Section 4.2.1 first consider the group of two dimensional translations. Two-dimensional translation invariance can be build by a separable operators that consists in first a translation invariant operator Φ_1 which transforms $x(u_1, u_2)$ along u_1 for u_2 fixed. Then a second translation invariant operator Φ_2 is applied along u_2. The product $\Phi_2\Phi_1$ is thus a separable invariant to any two-dimensional translation. However, if $\mathcal{L}_v x(u_1, u_2) = x(u_1 - v(u_2), u_2)$ then $\Phi_1 x_v = \Phi_1 x$ for all $v(u_2)$, although $\mathcal{L}_v$ is not a translation because $v(u_2)$ is not constant. It results that $\Phi x = \Phi\mathcal{L}_v x$. This separable operator is invariant to a much larger set of operators than two-dimensional translations and can thus confuse two images which are not translations of one-another, as in Figure 4.1. To avoid this information loss, it is necessary to build a translation invariant operator which takes into account the structure of the two-dimensional group. This is why translation scattering in $\mathbb{R}^2$ is not computed as a product of scattering operators along horizontal and vertical variables, but rather as a joint invariant that processes both variables (u_1, u_2) together, through two-dimensional wavelet modulus operators.

Section 4.2.2 explains that the same phenomena appears with translation and rotation variables, also it is more subtle because the rotation variable does not parameterize images but is created in the first layer of deep networks. Consider a first image, which is the sum of arrays of oscillatory patterns along two orthogonal directions, with same locations. If the two arrays of oriented patterns are shifted as in Figure 4.3 we get a very different textures, which are not globally translated or rotated one relatively to the other. However, an operator Φ_1 which first separates different orientation components and computes a translation invariant representation independently for each component will output the same values for both images because it does not take into account the joint location and orientation structure of the image. This is the case of separable scattering transforms or any of the separable translation and rotation invariant in use in [OMP+02, XYLJ10, LSP05].

Several researchers [CS06, BDGR12, DB07] have proposed to take into account the joint structure of the rigid-motion group of translations and rotations in applications such as noise removal or image enhancement with directional diffusion operators [DF11]. In addition, convolutions in recent deep convolutional networks [KSH12, DCM+12] now consider internal layers as multidimensional functions, as indicated by the sum over the input depth p in convolution (1.23). Similarly, a joint rigid-motion scattering $\widetilde{S}$ is constructed directly

on the rigid-motion group in order to take into account the joint information between positions and orientations. This joint rigid-motion scattering will iterate on rigid-motion wavelet modulus operator. Section 1.5.2 introduces rigid-motion convolution and wavelet transform operators, which are detailed in Section 4.3. Section 1.5.3 describes the joint rigid-motion scattering which is detailed in Section 4.4. Joint rigid-motion scattering is invariant to rigid-motion and provides state-of-the-art results on texture classification with datasets containing significant affine transformations, as demonstrated in Section 5.5. It can also be adapted in context where the rigid-motion invariance is not a desirable property, such as generic object classification. For this kind of applications, Section 6.3.1 introduces a non-invariant version of the joint-scattering. Promising results on object classification datasets Caltech101, 256 are demonstrated in Section 6.3.2.

1.5.2 Rigid-Motion Wavelet Transform

The orientation variable is not intrinsic to the parameterization of images and thus needs to be created by a first layer. For this purpose, we use the same oriented and dilated wavelet transform as for translation or separable scattering, to obtain a first invariant part

$$S_0 x = x \star \phi \tag{1.44}$$

and covariant part

$$U_1 x(u, \theta, j) = |x \star \psi_{\theta,j}|(u) \tag{1.45}$$

The translation invariant $S_0 x$ happens to be also rotation invariant if we use a rotation invariant window ϕ such as the Gaussian window. We can see $U_1 x$ as a function of $g = (u, \theta)$ and an orbit variable j. The rigid-motion scattering processes independently each orbit and apply to them a wavelet transform $\widetilde{\mathcal{W}}$ on the group of rigid-motion consisting of rotation and translation. For a signal $\tilde{x}(g)$ parameterized by a rigid-motion, this wavelet transform consists in convolutions with multidimensional wavelets

$$\widetilde{\mathcal{W}}\tilde{x} = \{\tilde{x} \tilde{\star} \tilde{\phi},\ \tilde{x} \tilde{\star} \tilde{\psi}_\lambda\}_\lambda \tag{1.46}$$

Section 4.3 introduces the necessary tools to builds this wavelet transform. Section 4.3.1 defines a rigid-motion convolution $\tilde{\star}$ and describes a fast implementation for separable filters. Section 4.3.2 shows how to define a separable wavelet frame $\{\tilde{\phi}, \tilde{\psi}_\lambda\}_\lambda$ on the rigid-motion group and Section 4.3.3 describes a fast implementation of the associated rigid-motion wavelet transform $\widetilde{\mathcal{W}}$.

A rigid-motion $g = (v, \theta)$ can be applied to an image x

$$\mathcal{L}_g x(u) = x(g^{-1}u) \tag{1.47}$$

$$= x(r_{-\theta}(u - v)) \tag{1.48}$$

Given two rigid-motions $g = (v, \theta)$ and $g' = (v', \theta')$ their group product is

$$g'.g = (v' + r_{\theta'} v,\ \theta + \theta') \tag{1.49}$$

It is defined in such a way that successive applications of two rigid-motions corresponds to the application of a the product rigid-motion.

$$\mathcal{L}_{g'}\mathcal{L}_g = \mathcal{L}_{g'.g} \tag{1.50}$$

Due to the non-symmetric term $r_{\theta'}v$ in (1.49), the rigid-motion group $SE(\mathbb{R}^2)$ is non-commutative. One can observe that U_1x is covariant to the rigid-motion group in the sense that the first layer of $\mathcal{L}_{g'}x$ is equal to the first layer of x up to a displacement of g' within the rigid motion group

$$U_1\mathcal{L}_{g'}x(g,j) = U_1x(g'^{-1}g,j). \tag{1.51}$$

Invariance to rigid-motion is thus essentially a translation problem on the rigid-motion group. The construction of stable invariant can be achieved through scattering, which thus naturally requires the construction of convolutions, and wavelets on the rigid-motion group.

The group convolution $\tilde{\star}$ of two functions $\tilde{x}(g)$ and $\tilde{y}(g)$ of the rigid-motion group is defined in Section 4.3.1 as

$$\tilde{x}\tilde{\star}\tilde{y}(g) = \int_{SE(2)} \tilde{x}(g')\tilde{y}(g'^{-1}g)dg' \tag{1.52}$$

For separable filters $\tilde{y}(g) = y(v)\mathring{y}(\theta)$ we observe that (1.52) can be factorized as

$$\tilde{x}\tilde{\star}\tilde{y}(g) = \int_{\theta'\in[0,2\pi)} \left(\int_{v'\in\mathbb{R}^2} \tilde{x}(v',\theta')y(r_{-\theta'}(v-v'))dv' \right) \mathring{y}(\theta-\theta')d\theta' \tag{1.53}$$

The inner integral of (1.53) is a two dimensional convolution of $\tilde{x}(.,\theta')$ with rotated filter $y_{\theta'}(v) = y(r_{-\theta'}v)$ along v while the outer integral is a one dimensional periodic convolution of the inner integral with filter $\mathring{y}$ along θ. This factorization yields a fast semi-separable implementation of the rigid-motion convolution which is detailed in Section 4.3.1.

Defining a wavelet transform operator $\widetilde{\mathcal{W}}$ requires to build a wavelet frame $\{\tilde{\phi},\tilde{\psi}_\lambda\}_\lambda$ on the rigid-motion group. We propose a construction of such a wavelet frame in section 4.3.2. It is similar to the construction of separable two dimensional wavelets. Indeed, one can build two dimensional wavelets $\{\phi(u_1,u_2),\psi_l(u_1,u_2)\}_l$ from one dimensional wavelets $\{\phi(u_1),\psi(u_2)\}$ as a separable product of the two wavelets applied to both variables

$$\begin{aligned}
\phi(u_1,u_2) &= \phi(u_1)\phi(u_2) && (1.54)\\
\psi_1(u_1,u_2) &= \psi(u_1)\psi(u_2) && (1.55)\\
\psi_2(u_1,u_2) &= \psi(u_1)\phi(u_2) && (1.56)\\
\psi_3(u_1,u_2) &= \phi(u_1)\psi(u_2). && (1.57)
\end{aligned}$$

Similarly, we build a separable wavelet frame $\tilde{\phi},\tilde{\psi}_{l,j,k}$ on the rigid-motion group as a separable product of a two dimensional wavelet frame $\{\phi_J,\psi_{l,j}\}_{l,j<J}$ applied to the spatial

parameter v and a one dimensional wavelet frame $\{\mathring{\phi}_K, \mathring{\psi}_k\}_{k<K}$

$$\begin{aligned}
\tilde{\phi}_{J,K}(v,\theta) &= \phi_J(v)\mathring{\phi}(\theta) && (1.58)\\
\tilde{\psi}_{l,j,k}(v,\theta) &= \psi_{l,j}(v)\mathring{\psi}_k(\theta) && (1.59)\\
\tilde{\psi}_{l,j,K}(v,\theta) &= \psi_{l,j}(v)\mathring{\phi}_K(\theta) && (1.60)\\
\tilde{\psi}_{0,J,k}(v,\theta) &= \phi_J(v)\mathring{\psi}_k(\theta). && (1.61)
\end{aligned}$$

The corresponding rigid-motion wavelet transform operator $\widetilde{\mathcal{W}}$ maps a signal $\tilde{x}(v,\theta)$ to the set of signals

$$\widetilde{\mathcal{W}}\tilde{x} = \{\tilde{x}\tilde{\star}\tilde{\phi}_{J,K},\ \tilde{x}\tilde{\star}\tilde{\psi}_{l,j,k}\}_{l,j,k} \tag{1.62}$$

A family of function $\{\phi, \psi_\lambda\}_\lambda$ is an ϵ frame if the corresponding wavelet operator $\mathcal{W}$ preserves the norm up to ϵ, that is for all x

$$(1-\epsilon)\|x\| \leq \|\mathcal{W}x\| \leq \|x\| \tag{1.63}$$

Theorem 2 claims that if both $\{\phi_J, \psi_{l,j}\}_{l,j<J}$ is an ϵ frame and if $\{\mathring{\phi}_K, \mathring{\psi}_k\}_{k<K}$ is an $\mathring{\epsilon}$ frame then $\{\tilde{\phi}_{J,K}, \tilde{\psi}_{l,j,k}\}_{l,j,k}$ is an $1-(1-\epsilon)(1-\mathring{\epsilon})$ frame so that the associated wavelet transform $\widetilde{\mathcal{W}}$ almost preserves the norm of all functions $\tilde{x}(v,\theta)$ of the rigid-motion group

$$(1-\epsilon)(1-\mathring{\epsilon})\|\tilde{x}\| \leq \|\widetilde{\mathcal{W}}\tilde{x}\| \leq \|\tilde{x}\|. \tag{1.64}$$

Fast algorithms to compute this rigid-motion wavelet transform $\widetilde{\mathcal{W}}$ are presented in section 4.3.3. These algorithms take advantage of the semi-separability of the rigid-motion convolution (1.53) and the multi-scale nature of wavelets (1.58-1.61).

1.5.3 Joint Rigid-Motion Scattering

Having defined a wavelet transform operator $\widetilde{\mathcal{W}}$ on functions of the rigid-motion group, we can successively apply it with intertwined modulus non-linearity to build a scattering network on the rigid-motion group.

As previously mentioned, the image is naturally parameterized by the translation group, not by the rigid-motion group. We therefore use a first oriented and dilated wavelet modulus operator to create the rotation variable. $S_0 x = x \star \phi$ is our first rigid-motion invariant. Since ϕ is chosen to be a rotation invariant window, $S_0 x$ is fully rotation invariant, and invariant to translation up to the width of the window ϕ. Our first covariant layer is thus

$$U_1 x(g,\bar{p}) = |x \star \psi_{\theta,j}(v)| \quad \text{where } g = (v,\theta) \quad \text{and } \bar{p} = j \tag{1.65}$$

It is a function of the rigid-motion group g and some orbit path $\bar{p}$ which is here limited to the scale index j.

As opposed to the separable scattering, the joint scattering cascades a rigid-motion wavelet modulus operator to $U_1 x$ which therefore processes together the translation and rotation variables v and θ

$$\begin{aligned}
\widetilde{S}_m x(g, j_1, \lambda_2, \ldots, \lambda_m) &= |\ldots|U_1 x(., j_1) \tilde{\star} \tilde{\psi}_{\lambda_2}| \ldots \tilde{\star} \tilde{\psi}_{\lambda_m}| \tilde{\star} \tilde{\phi}(g) && (1.66)\\
&= U_m x(., \lambda_2, \ldots, \lambda_m) \tilde{\star} \tilde{\phi}(g) && (1.67)\\
\widetilde{U}_{m+1} x(g, j_1, \lambda_2, \ldots, \lambda_{m+1}) &= |\ldots|U_1 x(., j_1) \tilde{\star} \tilde{\psi}_{\lambda_2}| \ldots \tilde{\star} \tilde{\psi}_{\lambda_{m+1}}(g)| && (1.68)\\
&= |\widetilde{U}_m x(., \lambda_2, \ldots, \lambda_m) \tilde{\star} \tilde{\psi}_{\lambda_{m+1}}(g)|. && (1.69)
\end{aligned}$$

The joint scattering vector consists in the concatenation of all invariant coefficients of all orders:

$$\widetilde{S} x = \{S_0 x, \widetilde{S}_1 x, \widetilde{S}_2 x, \ldots, \widetilde{S}_M x\} \tag{1.70}$$

Since it involves only stable operators $|\mathcal{W}|$ and $|\widetilde{\mathcal{W}}|$, $\widetilde{S}$ is also stable. It is invariant to translation, up to the width of the translation window ϕ used in W and $\widetilde{\mathcal{W}}$. It is also invariant to rotation up to the width of the rotation window $\mathring{\phi}$ used in $\widetilde{\mathcal{W}}$ to construct $\tilde{\phi}$. Finally, it preserves the joint information of $U_1 x$ along translations and rotations. In particular, Joint scattering is able to discriminate the two images in Figure 4.3, because it is sensitive to the fact that the rotation distribution of $U_1 x$ has respectively one or two peaks for the bottom and top image. The joint rigid-motion scattering was introduced in our papers [SM13, SM14] where it was also called roto-translation scattering. In those papers we have applied it to a wide range of texture datasets with state-of-the art results on most available datasets. A non-invariant simplified version has also been applied to generic object recognition in [OMS14]. Those results are briefly summarized in section 1.6 and detailed in Sections 5.5 and 6.3.

1.6 Classification experiments

We have evaluated the effectiveness of both the separable and the joint rigid-motion scattering on image classification datasets. Chapter 5 presents texture classification experiments and analyses the impact of the different invariants. Joint scattering is demonstrated to achieve state-of-the-art performance on most available texture classification datasets. Chapter 6 applies joint convolution to generic object recognition.

1.6.1 Texture Classification

Texture images have some specificities that requires to adapt representations and classifiers. Chapter 5 introduce representations and classifiers specific to texture classification problem. Section 5.2 presents rigid motion separable and joint scattering representations of stationary processes. Separable rigid-motion scattering is applied on the Outex10 datasets in Section 5.3. For datasets with important intraclass and scale variability, Section 5.4

reviews a PCA generative classifier and a data augmentation algorithm which provides additional scale invariance. This PCA classifier is used with joint scattering to classify images from harder texture datasets such as UIUCTex in Section 5.5.

A texture can be modeled as a stationary, ergodic process X over $\mathbb{R}^2$. Scattering representations of stationary process is presented in Section 5.2. Section 5.2.1 reviews translation expected scattering $\bar{S}X$, which is a deterministic quantity which does not depend upon a particular realization of the process. Since expected scattering definition involves an expected value, expected scattering cannot be computed from a single finite realization, but it can be estimated by windowed scattering representation SX where the convolution with the spatial low pass filter ϕ is replaced with a spatial averaging over the whole image. Expected scattering is not invariant to geometric transformations of the process, such as rotations or dilations. Section 5.2.2 and 5.2.3 introduce rigid-motion separable and joint expected scattering, which adapt the separable and joint scattering of Chapters 3 and 4 to stationary processes. These representations are invariant to rigid-motions and can be estimated from the rigid-motion separable and joint windowed scattering of finite realizations.

In our paper [SM12], he have experimented the separable scattering on the Outex 10 [OMP+02] texture dataset. The training images of this datasets do not contains rotations, while the testing images contains uniform rotations. Therefore, it is a good test case for rotation invariant representations, since the classifier can not learn the invariance from data. Separable scattering obtains slightly better results than other state-of-the-art representations. To emphasize the stability to deformation of the separable scattering, we have done another experiment where we have modified the dataset by applying a slight shear in all the test images. This largely degrades the performance of other descriptors such as LBP-HF [ZAMP12], but has little effect on the separable scattering. The results of these experiments are presented in Section 5.3.

In our papers [SM13, SM14], we have experimented with more challenging texture datasets, in particular with the UIUCTex [LSP05] texture datasets which contains a wide range of uncontrolled full affine transformations and elastic deformations. To achieve good results on this datasets, we have had to extend the invariance to dilation. The group of dilations bring additional difficulties because a very limited range of dilation is available and because it is not periodic. Also texture images tends to have a power-law behavior with respect to scale, which limits the effectiveness of local averaging invariant. To overcome those difficulties, Section 5.4 introduces a specific augmented log PCA classifier. For each image, we compute the log scattering of several dilated version the image. The logarithm linearizes the power-law behavior with respect to scale. For each class, we estimate the affine space $\mu_c + \mathbf{V}_c$ on which lies the log scattering of all dilated training images in a class. The log scattering of all dilated test images is then averaged, which provides an additional scale invariance, projected into the affine space of each class, and classified according to the minimum projection distance.

Papers [SM13, SM14] applies the rigid-motion scattering of Chapter 4, Section 5.2.3

and the augmented log-PCA classifier of Section 5.4 to several challenging texture datasets. Section 5.5 reports those results and shows that each invariants included in the scattering and in the classifier significantly improve results, which eventually match or improve upon state-of-the-art representations on all studied datasets.

1.6.2 Object Classification

Because of the large variability contained in real world images, generic object classification is a much harder problem than texture and requires much more flexible architecture to achieve good results. Deep convolutional networks [LBD+89, LKF10, KSH12, DCM+12, ZF13] have demonstrated state-of-the-art results on these kind of problems. They consists in a cascade of linear, convolutional layers followed by non-linearties. During an internship at Google, we have had the opportunity to experiment with top-performing large scale implementations of the architecture described in[KSH12, DCM+12, ZF13]. As explained in Section 1.3.3, these networks contains dense convolutional layers (1.23). Section 6.2.1 reviews these dense convolutional layers in greater details, and shows that some of them learn highly redundant weights. In Section 6.2.2, we propose separable convolutional layers whose implementation has striking similarities with the fast implementation of rigid-motion convolution for separable filters described in Section 4.3.1. Section 6.2.3 presents early experiments that demonstrate the effectiveness of such separable convolutional layers. The same neural network, adapted from [ZF13], processes data faster and requires less data to reach similar to slightly better final accuracy when its first layers are implemented with separable convolutions, compared to dense convolutions.

This suggests that the first layers of neural networks captures basic geometric properties that do not necessarily have to be learned. A joint scattering network as a similar architecture and could therefore be a good candidate, but it imposes rotation invariance. Objects in real world human environment tend to appear in specific orientations, therefore rotation invariance may not be a desirable property for object classification. Therefore, Section 6.3.1 presents a non-invariant version $\overset{\times}{S}$ of the rigid-motion scattering which was introduced by Oyallon in [OMS14]. Experiments described in Section 6.3.2, show that non-invariant joint scattering network $\overset{\times}{S}$ of two layers can achieve results competitive with the first two layers of fully trained convolutional network similar to the one presented in [KSH12]. This is a promising results for better understanding convolutional neural networks.

Chapter 2

Translation Scattering and Convolutional networks

2.1 Introduction

In this chapter we review the properties of currently used image representations for classification, and present the translation scattering representation. A common property of most image representations is some form of invariance to geometric transformations, in particular to translations. Achieving translation invariance is fairly easy but doing so while enforcing stability to deformation and retaining sufficient information is a hard, not very well posed problem.

Section 2.2 introduces the properties of translation invariance and stability to deformations. It is shown that some image representations such as the modulus of Fourier transform, can build full translation invariance but are unstable to deformations. Section 2.3 reviews the wavelet transform, which builds slight local invariance to translations, preserves information but which is also stable to deformations. Oriented Morlet wavelets, which will be particularly adequate for building rotation invariance in Chapters 3 and 4, are introduced. Fast algorithms that implement the wavelet transform are presented.

Wavelet transform only build slight local translation invariance. To achieve broader translation invariance, while retaining information and maintaining stability to deformation, scattering cascades several wavelet modulus operators. Scattering was initially introduced in [Mal12] and is reviewed in Section 2.4. Scattering provides larger translation invariance than the wavelet transform, while preserving the norm of the input signal, and being stable to deformations.

Deep neural networks [HS06, LLB+98, LBD+89, LKF10, DCM+12, KSH12, ZF13] are data-driven generic representations for complex task which also cascade linear and non-linear operators. In convolutional neural networks (ConvNets) [LBD+89, DCM+12, KSH12, ZF13], the linear operators are restricted to local, convolutional operators which makes

their architecture very similar to scattering networks. These techniques are quickly reviewed in Section 2.5 which highlights the similarities and the differences with scattering.

2.2 Stability to Deformation

A representation Φx of an image x is said to be globally translation invariant if it is equal for all translated versions $\mathcal{L}_v x(u) = x(u - v)$ of x

$$\forall v \in \mathbb{R}^2,\ \Phi \mathcal{L}_v x = \Phi x \tag{2.1}$$

Full translation invariance is relatively easy to achieve while maintaining most of the image information. The two dimensional Fourier transform is an operator $\mathcal{F}$ that maps a square integrable two dimensional function $x \in \mathbf{L}^2(\mathbb{R}^2)$ to another square integrable function $\mathcal{F}x$

$$\mathcal{F}: \quad \mathbf{L}^2(\mathbb{R}^2) \ \to\ \mathbf{L}^2(\mathbb{R}^2) \tag{2.2}$$

$$x \ \mapsto\ \mathcal{F}x \tag{2.3}$$

with

$$\mathcal{F}x(\omega) = \int_{\mathbb{R}^2} x(u) e^{-i\omega.u} du. \tag{2.4}$$

The Plancherel theorem states that the Fourier transform is an operator that preserves the norm in the sense that

$$\int_{\mathbb{R}^2} |x(u)|^2 du = (2\pi)^{-2} \int_{\mathbb{R}^2} |\mathcal{F}x(\omega)|^2 d\omega. \tag{2.5}$$

The Fourier transform is invertible and its inverse can be obtained as

$$x(u) = \int_{\mathbb{R}^2} \mathcal{F}x(\omega) e^{i\omega.u} d\omega. \tag{2.6}$$

Formally, the translation by $v \in \mathbb{R}^2$ is also an operator

$$\mathcal{L}_v: \quad \mathbf{L}^2(\mathbb{R}^2) \ \to\ \mathbf{L}^2(\mathbb{R}^2) \tag{2.7}$$

$$x \ \mapsto\ \mathcal{L}_v x \tag{2.8}$$

with

$$\mathcal{L}_v x(u) = x(u - v). \tag{2.9}$$

The Fourier transform of a translated image is equal to the original Fourier transform, up to a phase multiplication

$$\mathcal{F}(\mathcal{L}_v x)(\omega) = e^{-i\omega.v} \mathcal{F}x(\omega). \tag{2.10}$$

Applying a complex modulus to the Fourier transform removes the phase and thus yields a global translation invariant

$$\forall v \in \mathbb{R}^2,\ |\mathcal{F}x| = |\mathcal{F}\mathcal{L}_v x|. \tag{2.11}$$

It has been proved [BN84] that, under some conditions, the values of $|\mathcal{F}x|$ everywhere on $\mathbb{R}^2$ and its phase on a small subset of $\mathbb{R}^2$ uniquely determines the values of x, which means that the modulus of the Fourier transform essentially loses no information other than a global translation.

Natural images rarely differ only with a global translation, they also contain elastic deformations due to the physics of imaging, which maps three dimensional surfaces to the two dimensional plane. A representation Φx used to define a metric of similarity between images should be stable to those deformations, which means that a slightly deformed image $\mathcal{L}_\tau x(u) = x(u-\tau(u))$ should have a representation $\Phi\mathcal{L}_\tau x$ that is close to the representation of the original image Φx, if the deformation τ is small. The amount of deformation can be quantified as the largest possible displacement induced by the deformation

$$\|\tau\|_\infty = \sup_{u\in\mathbb{R}^2} |\tau(u)|. \tag{2.12}$$

A deformation also changes the local behavior of x. Indeed, a Taylor extension of τ in the neighborhood of u shows that

$$\begin{aligned} u + v - \tau(u+v) &\approx u + v - \tau(u) - \nabla\tau(u)v & (2.13)\\ &= u - \tau(u) + (\mathbb{1} - \nabla\tau(u))v. & (2.14)\end{aligned}$$

In the neighborhood of u, the deformation τ induces a uniform translation of $\tau(u)$ and a linear warping by $\mathbb{1} - \nabla\tau(u)$ which differs from $\mathbb{1}$ by $\nabla\tau(u)$. This amount of warping can be quantified as

$$\|\nabla\tau\|_\infty = \sup_{u\in\mathbb{R}^2} \|\nabla\tau(u)\|. \tag{2.15}$$

A representation is stable to deformation when the difference induced on the representation by a deformation can be bounded by a weighted sum of the two above-mentioned metrics, that is there exist two reasonably small positive constants C_1 and C_2 such that for all $x \in \mathbf{L}^2(\mathbb{R}^2)$

$$\|\Phi\mathcal{L}_\tau x - \Phi x\| \leq (C_1\|\tau\|_\infty + C_2\|\nabla\tau\|_\infty)\,\|x\|. \tag{2.16}$$

If $C_1 = 0$ and some arbitrary C_2 verifies (2.16), the representation Φ is fully translation invariant. Indeed, for a pure translation $\|\tau\|_\infty = \|v\|$, and the linear transformation penalization vanishes $\|\nabla\tau\|_\infty = 0$. Yet, the fact that a representation is fully translation invariant does not imply that it is stable to deformations i.e. that there exists a pair C_1, C_2 that verifies (2.16) and for which C_2 is small. Fourier modulus is an example of representation that is unstable to deformation while being fully translation invariant. Indeed, the Fourier transform of a complex sine wave of frequency ω is a dirac located at ω. A small dilation $u - \tau(u) = u - \epsilon u$ maps a sine wave to another sine wave whose Fourier transform is a dirac located at $(1-\epsilon)^{-1}\omega$. Thus the difference $\Phi\mathcal{L}_\tau x - \Phi x$ has a constant $\mathbf{L}^1$ norm of 2, even when the linear transformation penilazition term $\|\nabla\tau\|_\infty = \epsilon$ is made arbitrarily small. The counter example is arguable since since waves or diracs do not belong to $\mathbf{L}^2$ and since

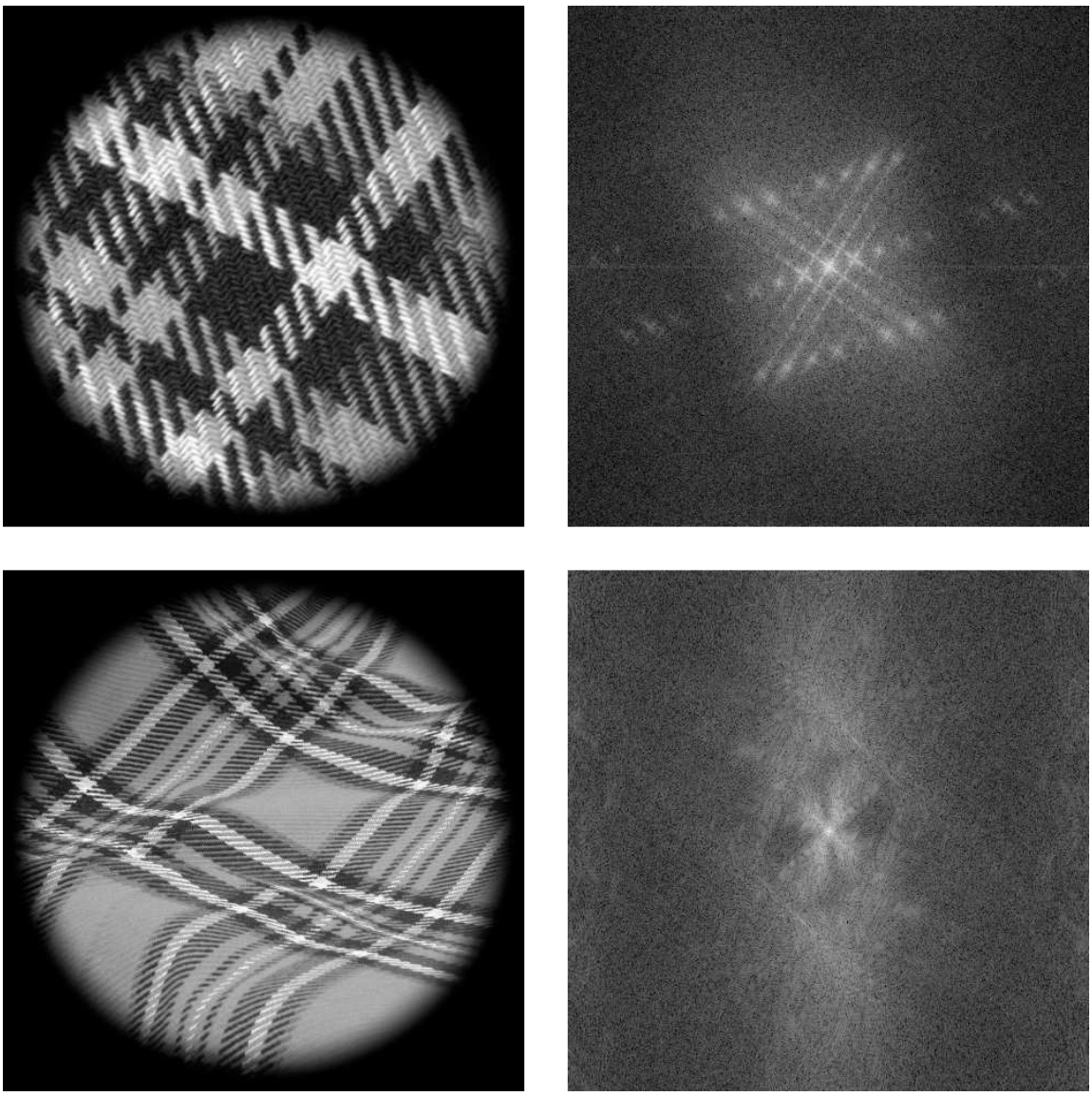


Figure 2.1: Left : two images of the same texture from the UIUCTex dataset [LSP05], multiplied by a radial plateau function to avoid discontinuities. The top left image is a a texture of fabric that lies on a flat surface. The bottom image is the same texture, but the surface is deformed. Right : the log of their modulus of Fourier transform. The periodic patterns of the texture corresponds to fine grained dots on the Fourier plane. When the texture is deformed, the dots spread on the Fourier plane, which illustrates the fact that modulus of Fourier transform is unstable to elastic deformations.

$\|\tau\|_\infty = \infty$ for a dilation, but a similar example with functions belonging to $\mathbf{L}^2(\mathbb{R}^2)$ can be obtained by localizing the sine wave and the dilation [Mal12]. A more concrete example is illustrated in Figure 2.1 which shows to images of the same texture, where one image is flat and the other is deformed. On the flat image, one can see that the periodic patterns of the texture correspond to very localized bumps in the Fourier domain. When the texture is deformed, the bumps spread over the entire Fourier plane, which makes it difficult to relate the Fourier transform of the original and deformed image.

2.3 Wavelet Transform

2.3.1 Wavelets, Window and the Wavelet Transform Operator

Stability to deformations can be obtained by grouping together close sine waves, which is equivalent to localizing them. This leads naturally to the wavelet representation.

A family of wavelets is obtained from a set of mother wavelets $\{\psi_l\}$ by dilating each mother wavelet with dyadic scales $1 \leq 2^j < 2^J$

$$\psi_{l,j}(u) = 2^{-2j}\psi_l(2^{-j}u). \tag{2.17}$$

The multiplicative constant 2^{-2j} is chosen so that the dilated wavelet verifies

$$\mathcal{F}\psi_{l,j}(\omega) = \mathcal{F}\psi_l(2^j\omega) \tag{2.18}$$

For compactness of notations, we will use $\lambda = (l, j)$ and denote the wavelets $\{\psi_\lambda\}_\lambda$. The wavelet coefficients of an image consist in all convolutions $\{x \star \psi_\lambda\}_\lambda$ of the image with the wavelets ψ_λ where the convolution is defined as

$$x \star \psi_\lambda(u) = \int_{v\in\mathbb{R}^2} x(v)\psi_\lambda(u-v)dv. \tag{2.19}$$

We usually assume that the wavelets have a zero average $\int \psi_l = 0$ and some regularity. Therefore, if the image also has some regularity, most of the wavelet coefficients are nearly zero, in which case the representation is said to be sparse. Sparsity is useful for compressing signals (one only needs to store the non-zero values, of which there are few), but also for classification. Indeed, sparsity allows to aggregate coefficients into invariant features, without losing too much of the distribution of coefficients.

The wavelet coefficients $\{x \star \psi_\lambda\}_\lambda$ retain the variations of the signal x at all scales smaller than 2^J. Complementary to these variations, a coarse approximation of the signal is captured through convolution with a single lowpass window, ϕ_J. The window ϕ_J is obtained by dilating a window ϕ at scale 2^J

$$\phi_J(u) = 2^{-2J}\phi(2^{-J}u) \tag{2.20}$$

where ϕ is typically a Gaussian window $\phi(u) = \exp(-|u|^2/(2\sigma^2))$. In this case, ϕ_J is a Gaussian window of width $2^J\sigma$.

The wavelet transform $\mathcal{W}x$ of a signal x consists in both the local average and the wavelet coefficients

$$\mathcal{W}x = \{x \star \phi_J,\ x \star \psi_\lambda\}_\lambda. \tag{2.21}$$

The norm of the wavelet transform is defined as

$$\|\mathcal{W}x\|^2 = \|x \star \phi_J\|^2 + \sum_\lambda \|x \star \psi_\lambda\|^2. \tag{2.22}$$

For appropriate wavelets and window, the wavelet transform is a complete representation, in the sense that it almost preserves the norm of signal. To quantify this, one can define the Littlewood-Paley function associated to the family $\{\phi_J, \psi_\lambda\}_\lambda$ which is the sum of squared of modulus of Fourier transform of all the functions of the family

$$A(\omega) = |\mathcal{F}\phi_J(\omega)|^2 + \sum_\lambda |\mathcal{F}\psi_\lambda(\omega)|^2. \tag{2.23}$$

The Littlewood-Paley function quantifies how tightly the family tiles the Fourier plane. If this tiling is α tight, i.e.

$$\forall \omega \in \mathbb{R}^2,\ 1 - \alpha \leq A(\omega) \leq 1, \tag{2.24}$$

then the wavelet transform operator almost preserves the norm of the signal

$$(1-\alpha)\|x\|^2 \leq \|\mathcal{W}x\|^2 \leq \|x\|^2 \tag{2.25}$$

in which case we the family $\{\phi_J, \psi_\lambda\}_\lambda$ is said to be an α frame.

If this is the case, the wavelet transform is contractive, that is for two signals $x, y \in \mathbf{L}^2(\mathbb{R}^2)$

$$\|\mathcal{W}x - \mathcal{W}y\| \leq \|x - y\|. \tag{2.26}$$

Also, if $\alpha = 0$, then the wavelet transform is a unitary operator

$$\|\mathcal{W}x\| = \|x\|. \tag{2.27}$$

If $0 \leq \alpha < 1$, then $\mathcal{W}$ is invertible and its inverse is obtained by filtering all its components with a dual family $\{\phi_J^{(-1)}, \psi_\lambda^{(-1)}\}_\lambda$ and summing the filtered components

$$x = (x \star \phi_J) \star \phi_J^{(-1)} + \sum_\lambda (x \star \psi_\lambda) \star \psi_\lambda^{(-1)}. \tag{2.28}$$

The dual wavelet family $\{\phi_J^{(-1)}, \psi_\lambda^{(-1)}\}_\lambda$ is defined by its Fourier transform as

$$\mathcal{F}\phi_J{}^{(-1)}(\omega) = A(\omega)^{-1}\mathcal{F}\phi_J(\omega) \tag{2.29}$$

$$\mathcal{F}\psi_\lambda^{(-1)}(\omega) = A(\omega)^{-1}\mathcal{F}\psi_\lambda(\omega). \tag{2.30}$$

One can see that α controls the stability of the inverse wavelet transform. Indeed, if $A(\omega)$ is close to zero for some ω, then the dual wavelets corresponding to λ for which $\mathcal{F}\psi_\lambda(\omega)$ is non-negligible will become very sensitive to the frequency ω.

2.3.2 Oriented Gabor and Morlet Wavelets

Among all possible sets of mother wavelets $\{\psi_l\}_l$ those that are oriented will be of particular interest for building rotation invariants. Such mother wavelets are indexed by an angular variable θ instead of the generic variable l. They are constructed as rotated versions of a single mother wavelet ψ.

$$\psi_\theta(u) = \psi(r_{-\theta}u) \tag{2.31}$$

so that the family of multiscale wavelets $\{\psi_\lambda\}_\lambda$ is indexed by $\lambda = (\theta, j)$ its elements are defined as rotated and dilated versions of a single mother wavelet ψ

$$\psi_{\theta,j}(u) = 2^{-2j}\psi(2^{-j}r_{-\theta}u). \tag{2.32}$$

A two dimensional wavelet is said to be analytic if its Fourier transform has a support which is contained in a half space of $\mathbb{R}^2$. Analytic wavelets have their complex modulus which tends to be much more regular than their real or imaginary part. A typical example of such wavelet is the Gabor wavelet

$$\psi_{\text{Gabor}}(u_1, u_2) = (2\pi\sigma^2)^{-1}\exp(-|u|^2/(2\sigma^2) + i\xi u_1). \tag{2.33}$$

A Gabor wavelet consists in a Gaussian envelope of width σ modulated by an horizontal complex sine wave of frequency ξ. Its Fourier transform is

$$\mathcal{F}\psi_{\text{Gabor}}(\omega) = \exp(-2\sigma^2|\omega - \xi(1,0)^T|^2). \tag{2.34}$$

It is a Gaussian of width $(2\sigma)^{-1}$ centered around $\xi(1,0)^T$. Therefore a Gabor wavelet is not exactly analytic because the support of a Gaussian is infinite but if σ/ξ is sufficiently small, then it is a good approximation to say that $\mathcal{F}\psi_{\text{Gabor}}(\omega) = 0$ for $\omega \leq 0$.

Gabor wavelets have been widely used in signal processing and neuroscience because they have many useful properties both theoretically and practically. They are the best theoretical trade off between spatial and frequency localization. Indeed, the Heisenberg inequality claims that for normalized wavelets $\|\psi\|_1 = \int |\psi| = 1$, the product of the spatial spread σ_ψ defined by

$$\sigma_\psi^2 = \int |\psi(u)| \left| u - \int |\psi(u)| u du \right|^2 du \tag{2.35}$$

and its Fourier spread $\sigma_{\mathcal{F}\psi}$ cannot be lower than $1/2$

$$\sigma_\psi \sigma_{\mathcal{F}\psi} \geq 1/2 \tag{2.36}$$

and the only functions for which the bound $1/2$ is reached are Gabor wavelets. Also, Gabor wavelets have an hermitian symmetry

$$\psi_{\text{Gabor}}(-u) = \psi_{\text{Gabor}}(u)^* \tag{2.37}$$

where $*$ denote the complex conjugate. Therefore, when computing convolutions of a real signal x with rotated Gabor wavelets, one can deduce the wavelet coefficients for $\theta \in [\pi, 2\pi)$ from those for which $\theta \in [0, \pi)$ since

$$x \star \psi_\theta(u) = (x \star \psi_{\theta+\pi}(u))^* \tag{2.38}$$

which will save roughly a factor of two of computational time. In addition, if we are only interested in the complex modulus, as it will be the case for scattering networks, we can discard all the orientations $\theta \in [\pi, 2\pi)$ since

$$|x \star \psi_\theta(u)| = |x \star \psi_{\theta+\pi}(u)| \tag{2.39}$$

which will save a factor of two of memory.

In all classification experiments, we use a slight variation of Gabor wavelets, namely elongated Morlet wavelet. One of the main disadvantages of Gabor wavelets is that they have a non-zero average which makes their responses non-sparse as shown in Figure 2.2. One way to fix this while preserving the smoothness and spatial localization of a Gabor wavelet is to subtract a Gaussian envelope multiplied by an appropriate constant K. The resulting wavelet is called a Morlet wavelet and is defined as

$$\psi_{\text{Morlet}}(u) = (2\pi\sigma^2)^{-1} \exp(-|u|^2/(2\sigma^2)) \left(\exp(i\xi u_1) - K\right) \tag{2.40}$$

The constant K is computed numerically so that $\int \psi_{\text{Morlet}} = 0$. Because of this subtraction of a low pass envelope, Morlet wavelets are slightly less analytic and less localized in frequency than Gabor wavelets, but they have an exact zero-mean.

Computing the wavelet transform with Morlet wavelets requires to discretize θ in $2C$ of orientations between $[0, 2\pi)$. When choosing a larger C, we want to modify the wavelets so as to increase their angular sensitivity, otherwise the extra orientations do not bring additional information. For this reason, we replace the circular envelope of the Gaussian envelope with an elliptical one whose horizontal and vertical semi axes are respectively σ and σ/s

$$\psi_{\text{Elongated Morlet}}(u_1, u_2) = (2\pi\sigma^2/s)^{-1} \exp(-(2\sigma^2)^{-1}(u_1^2 + u_2^2 s^2)) \left(\exp(i\xi u_1) - K\right) \tag{2.41}$$

We typically chose $s \propto C^{-1}$ where C is the number of orientations, so that the horizontal wavelet $\theta = 0$ has a growing vertical semi axes proportional to C and thus becomes more sensitive to the horizontal orientation when the number of orientations C increases. One can also compute several scales per octave Q instead of just one. In this case we will typically chose $\sigma/\xi \propto 2^Q$ so as to increase the scale sensitivity of wavelets when the number of scales per octave grows. Figure 2.3 shows two families of elongated Morlet wavelets with different number of scales J, orientations C and scales per octave Q, along with their associated Littlewood Paley function. One can see that in all configurations, the Littlewood Paley has reasonable bounds with $\alpha \approx 0.7$. Increasing the number of orientations and scales per

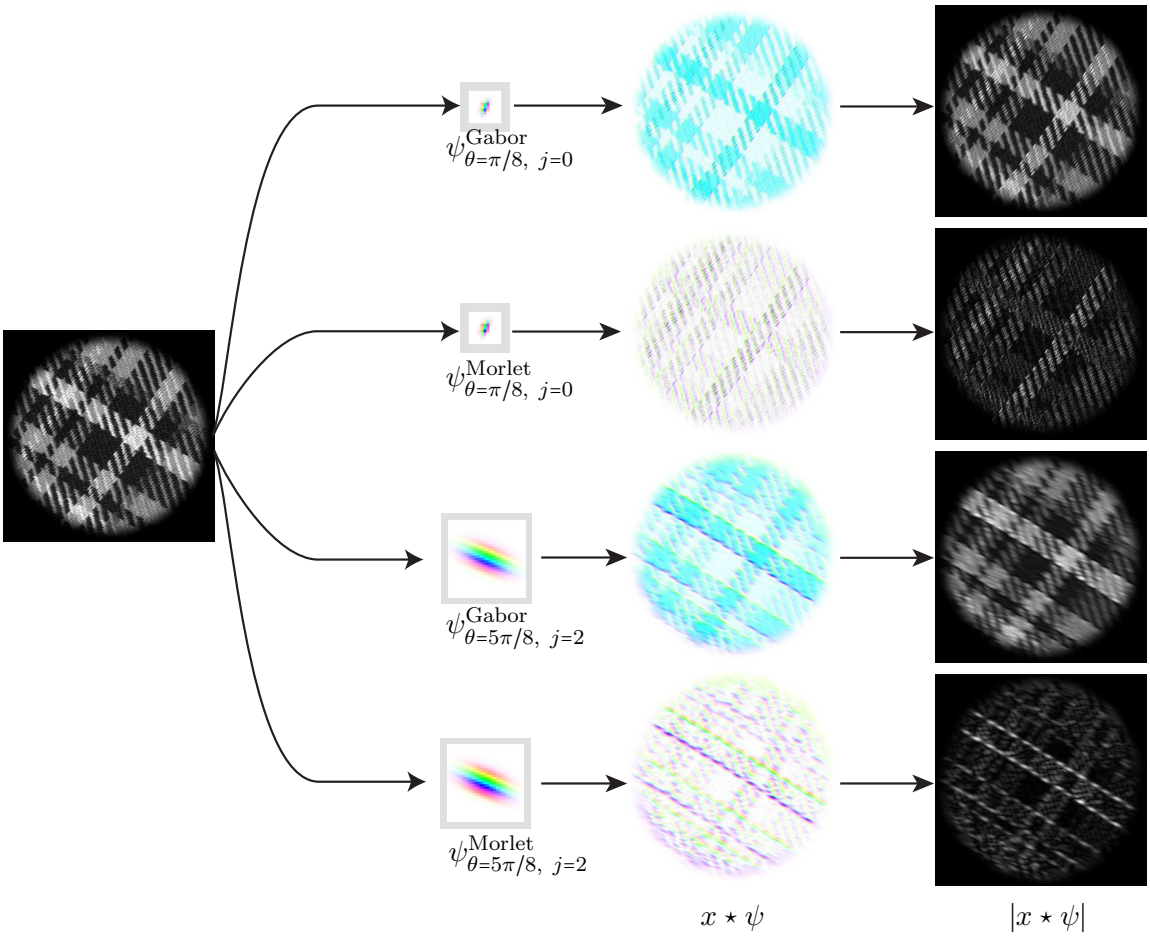


Figure 2.2: Comparison of elongated Gabor and Morlet two dimensional wavelets. The left image is filtered with Gabor and Morlet wavelets obtained with constant window spread $\sigma = 0.8$, a top frequency $\xi = 3\pi/4$ and a slant $s = 0.5$, at two pairs of orientation and scale. The image is filtered and which result in a complex signal $x \star \psi$ of which we compute the modulus $|x \star \psi|$. The Gabor wavelets have a non-zero DC component, which tends to dominate over the high frequencies.

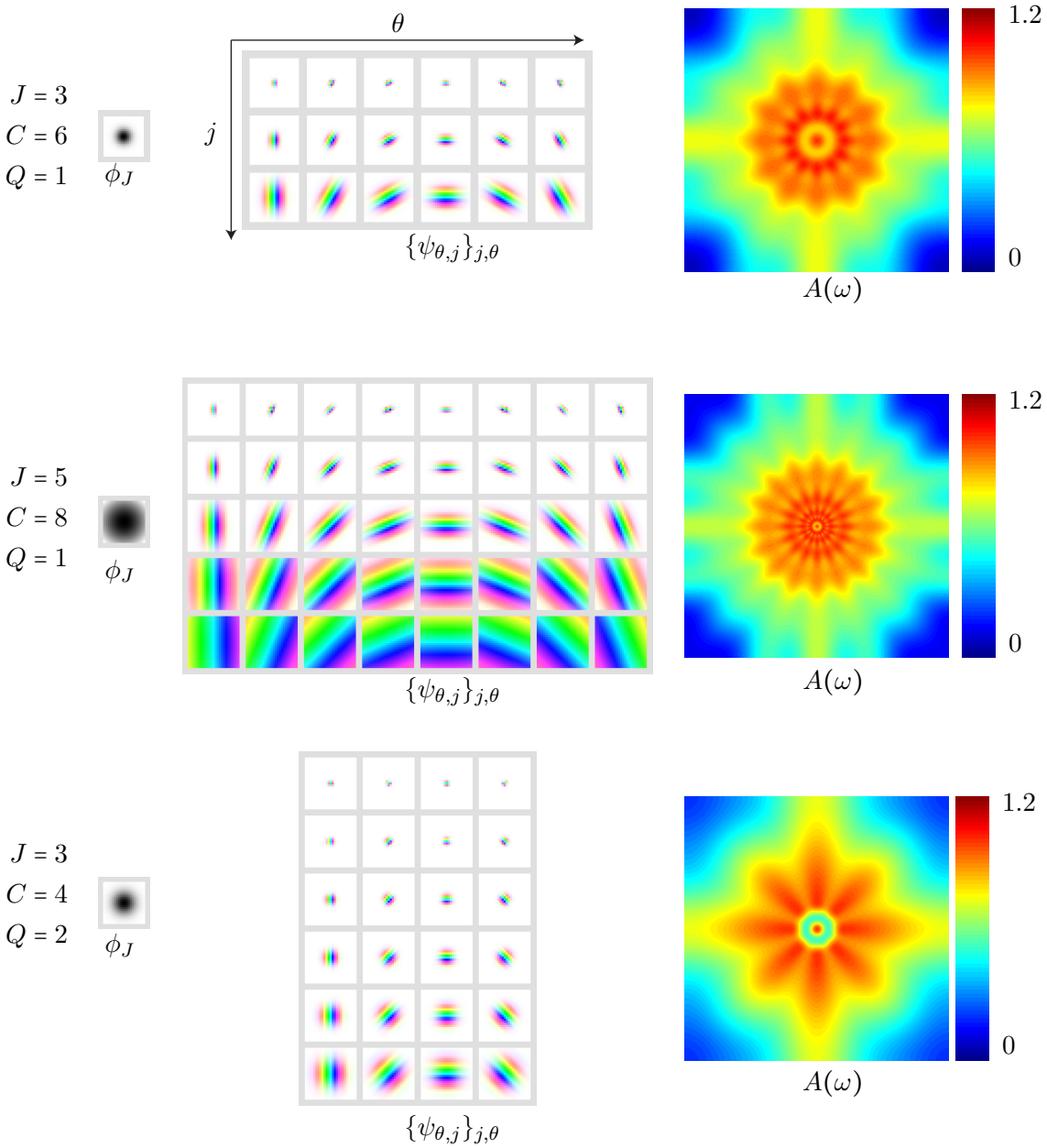


Figure 2.3: Three Morlet wavelet families with different sets of parameters. For each set of parameters, we show, from left to right, the gaussian window ϕ_J, all the Morlet wavelets $\psi_{\theta,j}$, and the associated Littlewood Paley sum $A(\omega)$. When the number of scales J increases, so does the width of the low pass wavelet ϕ_J. When the number of orientations C increases or when the number of scales per octave Q decreases, the Morlet wavelets become more elongated in the direction perpendicular to their orientation, and hence have an increased angular sensitivity.

octave of wavelets increases their localization in the Fourier plane, so that a particular wavelet $\psi_{\theta,j}$ is always slightly redundant with its immediate neighbors $\psi_{\theta\pm\pi/C,\ j\pm1/Q}$ and almost orthogonal to all other wavelets.

When increasing the number of orientations and scales per octave, elongated Morlet wavelets have better angular and scale sensitivity and therefore better separate the different frequency components of an image. This can positively impact classification, but it also has a number of disadvantages. The most obvious one is the increased number of convolutions to compute a wavelet transform. Since the wavelets are better localized in Fourier, they are also less well localized spatially. This will grow their spatial support and therefore increase the time to perform a single convolution, if computed in the spatial domain. As explained in section 2.2, better Fourier localization also means less stability to deformation, which can adversely affect classification at a certain point. Therefore, the choice of the number of orientations and scales per octave is a trade off between the better separability and worsen instability to deformations of the wavelet transform, and the resources we are willing to dedicate to its computation.

2.3.3 Fast Implementations

This section discusses different implementations of the wavelet transform. The Fourier implementation is more flexible. Its complexity does not depends upon the support of wavelets and it allows arbitrary downsampling. The filter bank implementation is faster but also more restricting. It supposes that the Fourier transform of the wavelets can be written as an infinite produce of Fourier transform of some dilated filters. Its running time is proportional to the support of the filters, and it does not easily allow arbitrary downsampling.

Fourier Implementation

A first possible strategy to implement the wavelet transform is to compute all the convolutions in Fourier domain. Given two discrete finite periodic images x and ψ we have

$$\mathcal{F}(x \star \psi)(\omega) = \mathcal{F}x(\omega)\mathcal{F}\psi(\omega) \tag{2.42}$$

Therefore, to compute the convolutions with all wavelets $\psi_{\theta,j}$ one can compute the Fourier transform of x, multiply it with all the Fourier transforms of $\mathcal{F}\psi_{\theta,j}$, and compute the inverse Fourier transforms of the resulting signals. A Fourier transform is implemented with a fast Fourier transform (FFT) algorithm which has a complexity of $O(N \log N)$ for an image of N pixels. If the wavelets are fixed, as it is the case for our applications, their Fourier transform can be precomputed. Also, $x \star \psi_{j,\theta}$ has a spatial regularity and can therefore be subsampled by 2^j. Instead of computing the inverse Fourier transform and then subsampling, it is equivalent, and more efficient, to periodize the product of

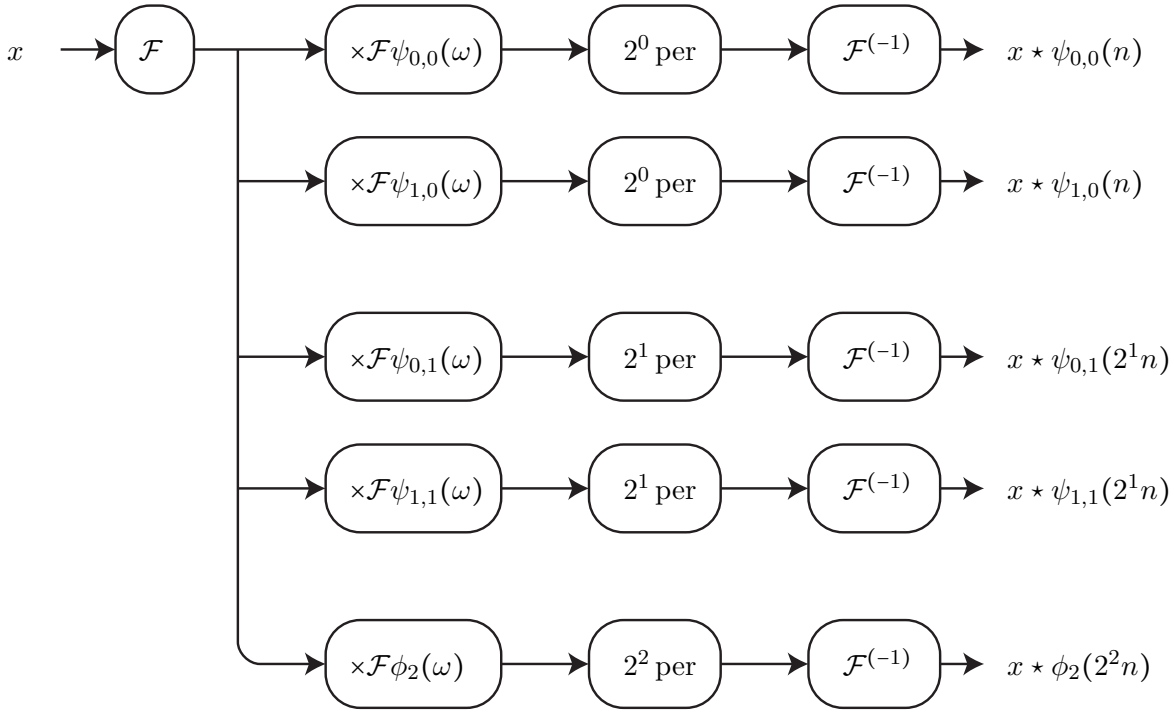


Figure 2.4: Fourier implementation of the wavelet transform $\mathcal{W}$ with $J = 2$ scales and $C = 2$. The Fourier transform of x is calculated and multiplied with the Fourier transform of each function in the wavelet family $\{\phi_J, \psi_{\theta,j}\}_{\theta,j}$. The results are periodized according to the target resolution, and then an inverse Fourier transform is applied.

the Fourier transforms and then to compute the inverse Fourier transform at the lowered resolution. The resulting algorithm is summarized in Figure 2.4 and has a complexity of

$$T_{\text{Fourier}}(N) = \underbrace{O(N \log N)}_{\text{FFT of } x} + \underbrace{O(JCN)}_{\substack{\text{Fourier multiplication} \\ \text{and periodization}}} + \underbrace{O(C \sum_j 2^{-j} N \log 2^{-j} N)}_{\substack{\text{IFFT of peridiodized} \\ \text{versions of } \mathcal{F}x(\omega)\mathcal{F}\psi_\lambda(\omega)}} . \quad (2.43)$$

$$= O(CN \log N) + O(JCN) \quad (2.44)$$

In practice, the term $O(JCN)$ for multiplication and periodization in Fourier domain is negligible and most of the time is spent on the initial FFT at the initial resolution, and on the IFFT at lower resolution. Another issue is that the equation (2.42) is valid for periodic convolution. Natural images are not periodic and therefore computing a periodic wavelet transform generates large amplitude coefficients at the boundaries of the image. Those large coefficients are artifacts that do not relate to the image content and should

preferably be discarded. One solution is to simply discard all wavelet coefficients within an inner margin but this can lead to too much information loss for small images. A better solution is to pad the image symmetrically within a certain outer margin proportional to the maximum scale 2^J of the wavelet family, perform the convolution, and then discard the coefficients inside the outer margin. This strategy seems inefficient because for small scales, the margin is much larger than the size of the region that is actually affected by boundary effects. Adapting the outer margin to the scale 2^j might seem more efficient, but it would also require to compute one additional Fourier transform of x per scale, at the maximum resolution and those extra Fourier transforms can actually be more expensive than what could have been gained by having a tighter margin when computing the inverse Fourier transform at the lower resolution. Therefore, we use the same padding scheme for all scales and targeted resolution. The Fourier implementation of the wavelet transform is summarized in Figure 2.4.

Since convolutions are implemented by multiplication of Fourier transforms at the initial resolution, the Fourier-based wavelet transform also offers the flexibility of choosing a less aggressive downsampling scheme. We typically downsample $x \star \psi_{\theta,j}$ by

$$\text{downsampling} = 2^{\max(0,\ j-\text{oversampling})} \tag{2.45}$$

instead of 2^j. The oversampling is a parameter typically set to 1 in our experiments but that can be set to ∞ in which case all the output signals have the same resolution, equals to the resolution of the input signal x. This increases the running time to $O(JCN \log N)$ which remains acceptable for visualization and debugging purposes.

Filter Bank Implementation

The Fourier implementation of the wavelet transform has some practical advantages, but it does not leverage the fact the the different wavelets of the family $\{\psi_{\theta,j}\}$ are dilated versions of one another. A fast wavelet transform [Mal08], also called "algorithme a trou", takes advantage of this by computing approximations $A_j x$ of the input signal at different resolutions and computing $x \star \psi_{\theta,j}$ by filtering $A_j x$ at the output resolution 2^j.

This assumes that the Fourier transform of the window ϕ_0 at scale 1 and each mother wavelet ψ_θ can be written as a product of Fourier transform of discrete dilated filters h and g_θ

$$\mathcal{F}\phi_0(\omega) = \prod_{j<0} \mathcal{F}h(2^j\omega) \tag{2.46}$$

$$\mathcal{F}\psi_\theta(\omega) = \mathcal{F}g_\theta(\omega)\mathcal{F}\phi(\omega). \tag{2.47}$$

Let us initialize the approximation $A_0 x = x \star \phi_0$ and denote

$$A_j x(n) = x \star \phi_j(2^j n) \tag{2.48}$$

$$B_{\theta,j} x(n) = x \star \psi_{\theta,j}(2^j n) \tag{2.49}$$

for $n \in \mathbb{Z}^2$. In practice we do not have access to either the true continuous image x nor can we compute $\mathcal{F}\phi_0$ as an infinite product. Thus we do not actually compute $A_0 x$, but rather assume that it is the input image itself. It results from the definitions (2.46-2.49) that

$$A_{j+1}x(n) = \sum_p A_j x(2p) h(n-2p) \tag{2.50}$$

$$B_{\theta,j}x(n) = \sum_p A_j x(p) g(n-p). \tag{2.51}$$

Thus the subsampled wavelet transform $\{x \star \phi_J(2^J n),\ x \star \psi_{\theta,j}(2^j n)\}_{n,\theta,j}$ is implemented as a cascade of convolutions and downsampling. Those convolutions are done with filters h and g_θ whose support do not change with the scale 2^j of the wavelet $\psi_{\theta,j}$. Therefore, we implement these convolutions in the spatial domain, in the small filter support regime where they are faster than FFT-based convolutions. Equations (2.50-2.51) are more compactly expressed as

$$A_{j+1}x = (A_j x \star h) \downarrow 2 \tag{2.52}$$

$$B_{\theta,j}x = A_j x \star g_\theta. \tag{2.53}$$

Let N be the size of image and P be the size of filters. A convolution at the finest resolution is implemented as a weighted sum over P elements for each position n

$$x \star h(n) = \sum_p x(n-p) h(n) \tag{2.54}$$

and therefore requires NP operations. The cascade requires $1+C$ such convolution at each resolution 2^j. The "a trou" algorithm thus has a complexity of

$$T_{\text{A-trou}}(N) = (1+C) \sum_j 2^{-2j} NP \tag{2.55}$$

$$= O(CNP) \tag{2.56}$$

and requires a memory of $O(CNP)$ where C is the number of orientations, N is the size of the input image and P is the size of the filters h and g_θ. In practice, we typically use filters of size $P = (2q+1) \times (2q+1)$ where $q = 3$, in which case our implementation of the "a trou" algorithm is faster than our implementation of the FFT-based algorithm.

As for the FFT-based algorithm, it is preferable to pad and unpad the signal symetrically with an outer margin to avoid large coefficients, or having to discard coefficients within an inner margin. Instead of padding the signal with a margin proportionally to 2^J, we rather pad and unpad the signal before and after each convolution in the cascade, with a margin exactly equal to the half width q of the support of filters. This intertwined padding scheme is not strictly equivalent to the global padding scheme of the Fourier implementation of wavelet transform, but it serves the same purpose of avoiding large coefficients at

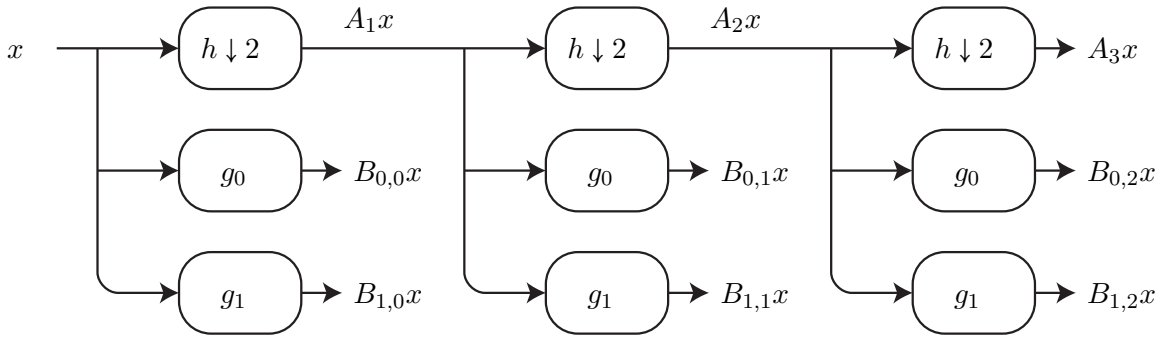


Figure 2.5: Filter bank implementation of the wavelet transform $\mathcal{W}$ with $J = 3$ scales and $C = 2$ orientations. A cascade of low pass filter h and downsampling computes low frequencies $A_j x = x \star \phi_j$ and filters g_θ compute high frequencies $B_{\theta,j} x = x \star \psi_{\theta,j}$. This cascade results in a tree whose internal nodes are intermediate computations and whose leaves are the output of the downsampled wavelet transform.

the boundary and it does so significantly faster. The "a trou" algorithm is summarized in figure 2.5.

The "a trou" algorithm is slightly less flexible than the FFT-based algorithm since it heavily relies on downsampling. To implement oversampling as in (2.45), we would need to compute convolutions with dilated filters h and g_θ, which then would have larger size. This would prohibitively increase the complexity of the algorithm, which would eventually becomes quadratic with respect to N if no downsampling were to be done, while the FFT-based algorithm remains linearithmic.

2.4 Translation Scattering

2.4.1 Wavelet Modulus Operator

The averaging builds stability to deformation and translation invariance up to 2^J. Indeed, it is proved in the early version of [Mal12] that there exist a constant $K > 0$ such that for any deformed image $\mathcal{L}_\tau x(u) = x(u - \tau(u))$,

$$\|\mathcal{L}_\tau x \star \phi_J - x \star \phi_J\| \leq K \|x\| (2^{-J} \|\tau\|_\infty + \|\nabla \tau\|_\infty). \tag{2.57}$$

Yet, averaging only captures a very coarse approximation of x and loses all its high frequency components. A wavelet transform recovers the high frequencies through convolutions with high pass wavelets $x \star \psi_{\theta,j}$ but these high frequencies only have translation invariance up to 2^j, not to the maximum scale 2^J. Indeed, one can similarly prove that

$$\|\mathcal{L}_\tau x \star \psi_{\theta,j} - x \star \psi_{\theta,j} \| \leq K \|x\| (2^{-j} \|\tau\|_\infty + \|\nabla \tau\|_\infty). \tag{2.58}$$

To enforce invariance to 2^J translations while maintaining stability to deformations of wavelet coefficients $x \star \psi_{\theta,j}$, the most obvious thing to do is to average them into $x \star \psi_{\theta,j} \star \phi_J$ but this yields almost zero coefficients since $\psi_{\theta,j}$ and ϕ_J are specifically designed to be almost orthogonal to each other. Therefore, to build non-trivial translation invariant, a possible strategy is to intertwine a non-linearity between the wavelet convolution $\star\psi_{\theta,j}$ and the averaging $\star\phi_J$.

The convolutions of an input signal x with an analytical oriented wavelet such as Gabor or Morlet wavelet, is a complex signal $x \star \psi_{\theta,j}(u)$. The amplitude of this signal is a regular envelope that quantifies how the signal is correlated to the orientation θ and scale 2^j in the neighborhood of u, while its phase varies almost linearly and is sensitive to local displacements. Figure 2.2 shows the result of such convolutions. Applying a complex modulus to such a response discards local displacements which are irrelevant to describe the image content. It is thus a good candidate of non-linearity to apply before averaging, because the averaging will capture more information as its input becomes more regular. Complex modulus also has the advantages of being contractive, preserving the norm and the stability to deformation.

The wavelet modulus operator $|\mathcal{W}x|$ consists in the approximation $x \star \phi_J$ coefficients and the modulus of the wavelet coefficients $|x \star \psi_\lambda|$,

$$|\mathcal{W}x| = \{x \star \phi_J,\ |x \star \psi_\lambda|\}_\lambda. \tag{2.59}$$

It decomposes the signal into a first linear invariant S_0x, called scattering of order 0, defined by

$$S_0x(u) = x \star \phi_J(u) \tag{2.60}$$

which is invariant to translation up to 2^J and stable to deformation, and a first non-linear covariant part

$$U_1x(u,\theta,j) = |x \star \psi_{\theta,j}(u)| \tag{2.61}$$

which is invariant to small translation up to 2^j and stable to deformation.

2.4.2 Cascading Wavelet Transform

To build homogeneous wavelet up to 2^J, U_1x is also averaged which yields a second invariant S_1x, called scattering of order 1

$$S_1x(u,\theta,j) = |x \star \psi_{\theta,j}| \star \phi_J(u). \tag{2.62}$$

As it was the case for S_0x, the averaging in S_1x also loses information, namely the high frequencies of U_1x. These are recovered through wavelet modulus coefficients

$$U_2x(u,\theta_1,j_1,\theta_2,j_2) = ||x \star \psi_{\theta_1,j_1}| \star \psi_{\theta_2,j_2}(u)|. \tag{2.63}$$

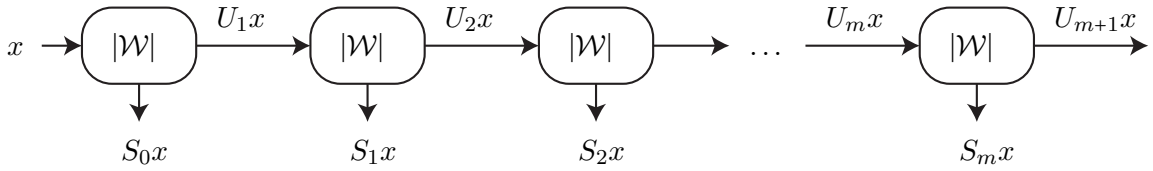


Figure 2.6: Translation scattering can be seen as a neural network which iterates over wavelet modulus operators $|W|$. Each layer m outputs averaged invariant $S_m x$ and covariant coefficients $U_{m+1}x$.

Because $U_1 x(., \theta_1, j_1)$ is invariant to translations up to 2^{j_1}, its fine scale coefficients $U_1 x(., \theta_1, j_1) \star \psi_{\theta_2, j_2}$ are almost zero for fine scale $j_2 \leq j_1$. Therefore, we usually only compute those coefficient for increasing scale $j_2 > j_1$.

$U_2 x$ is invariant to translations up to 2^{j_2}, but not to 2^J. Therefore $U_2 x$ is averaged into the second order scattering coefficients

$$S_2 x(u, \theta_1, j_1, \theta_2, j_2) = ||x \star \psi_{\theta_1, j_1}| \star \psi_{\theta_2, j_2}| \star \phi_J(u). \tag{2.64}$$

This non-linear decomposition is generalized to any order m by

$$\begin{aligned} S_m x(u, \theta_1, j_1, \dots \theta_m, j_m) &= U_m(., \theta_1, j_1, \dots \theta_m, j_m) \star \phi_J(u) && (2.65)\\ &= |\dots|x \star \psi_{\theta_1, j_1}| \dots \star \psi_{\theta_m, j_m}| \star \phi_J(u) && (2.66)\\ U_{m+1} x(u, \theta_1, j_1, \dots, \theta_{m+1}, j_{m+1}) &= |U_m x(., \theta_1, j_1, \dots \theta_m, j_m)| \star \psi_{\theta_{m+1}, j_{m+1}}(u)| && (2.67)\\ &= |\dots|x \star \psi_{\theta_1, j_1}| \dots \star \psi_{\theta_{m+1}, j_{m+1}}(u)|. && (2.68) \end{aligned}$$

For compactness we denote the scattering path $p_m = (\theta_1, j_1, \dots, \theta_m, j_m)$ and thus have

$$\begin{aligned} S_m x(u, p_m) &= U_m x(., p_m) \star \phi_J(u) && (2.69)\\ U_{m+1} x(u, p_{m+1}) &= |U_m(., p_m) \star \psi_{\theta_{m+1}, j_{m+1}}(u)|. && (2.70) \end{aligned}$$

One can therefore compactly write that

$$\{S_m x, U_{m+1} x\} = |\mathcal{W}| U_m x. \tag{2.71}$$

The scattering thus successively applies the wavelet modulus operator $|\mathcal{W}|$ to decompose the residual layer $U_m x$ into a translation invariant layer $S_m x$ and a non-linear covariant deeper layer $U_{m+1} x$ which will be either retransformed or simply discarded when it becomes negligeable. The scattering structure is illustrated in Figure 2.6.

The scattering invariant vector consists of scattering vector of all orders up to a maximum order M

$$Sx = \{S_0 x, \dots, S_M x\}. \tag{2.72}$$

Figure 2.7 shows an input texture image x and its scattering coefficients of order $m = 0, 1, 2$.

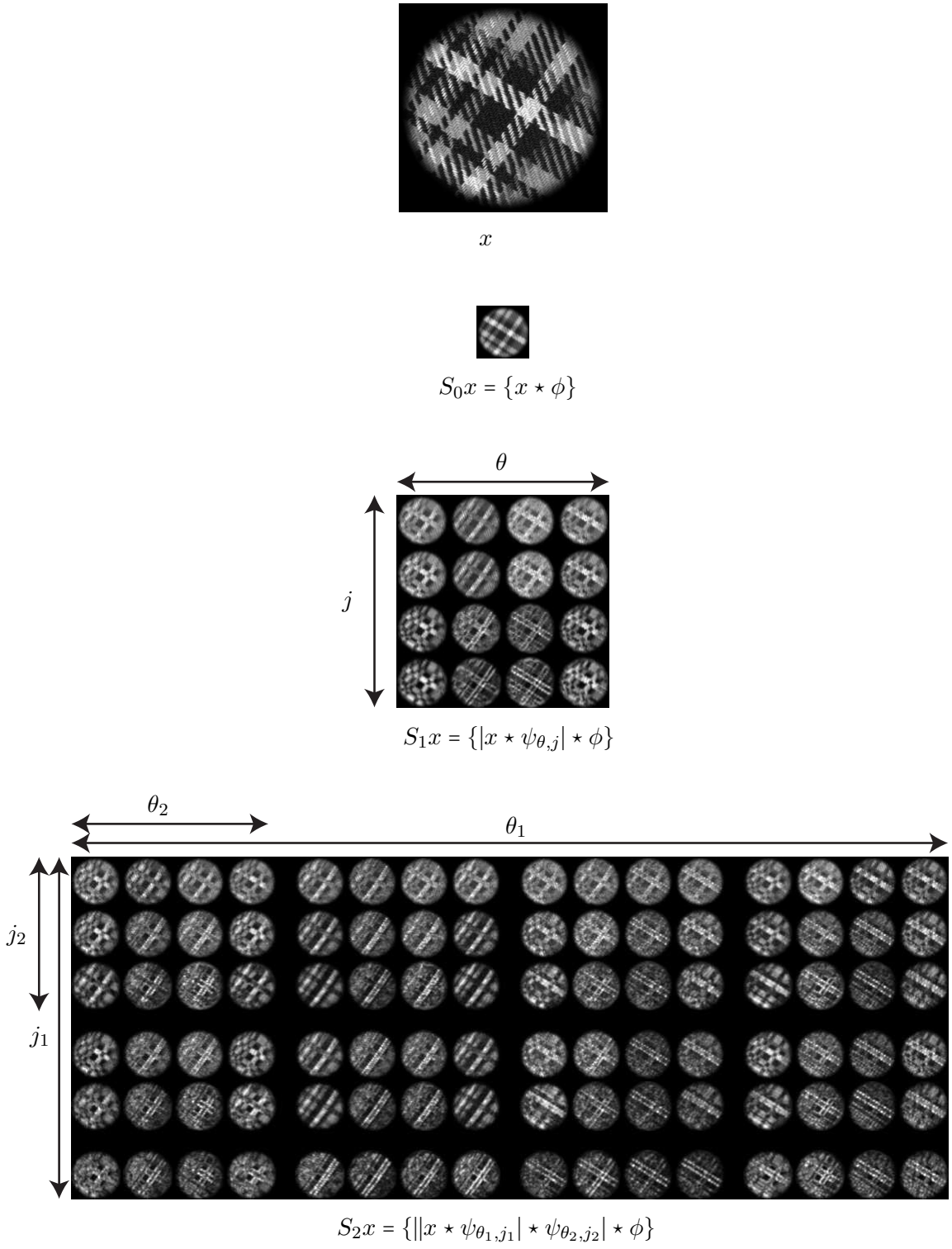


Figure 2.7: An input texture image x and its scattering coefficients of order $m = 0, 1, 2$, computed with $J = 4$ scales and $C = 4$ orientations. The paths of second order scattering S_2 are laid out with rows corresponding to lexicographic order on (j_1, j_2) and columns corresponding to lexicographic order on (θ_1, θ_2). For second order coefficients S_2x, only paths with increasing scale $j_2 > j_1$ are computed which is why there are three rows on the first block of rows, two for the second block and one for the last.

2.4.3 Scattering Properties

The $\mathbf{L}^2$ norm of the scattering covariant coefficients $U_m x$ is defined as

$$\|U_m x\|^2 = \sum_{p_m} \|U_m x(., p_m)\|^2 \tag{2.73}$$

$$= \sum_{p_m} \int_{\mathbb{R}^2} |U_m x(u, p_m)|^2 du \tag{2.74}$$

and similarly

$$\|S_m x\|^2 = \sum_{p_m} \|S_m x(., p_m)\|^2 \tag{2.75}$$

$$= \sum_{p_m} \int_{\mathbb{R}^2} |S_m x(u, p_m)|^2 du. \tag{2.76}$$

Finally, the norm of the scattering vector is

$$\|Sx\|^2 = \sum_{0 \le m \le M} \|S_m x\|^2. \tag{2.77}$$

If the wavelet family $\{\phi_J, \psi_\lambda\}_\lambda$ is an α frame (2.24) then the wavelet transform operator almost preserves the norm (2.25). Since the modulus preserves the norm, the wavelet modulus operator also almost preservers the norm,

$$(1 - \alpha)\|x\| \le \||\mathcal{W}|x\| \le \|x\|. \tag{2.78}$$

For $\alpha = 0$ the wavelet modulus operator is unitary $\||\mathcal{W}|x\| = \|x\|$. Applying this equality to signals $U_m x(., p_m)$ for every p_m yields

$$\|S_m x\|^2 + \|U_{m+1} x\|^2 = \|U_m x\|^2. \tag{2.79}$$

Cascading (2.79) shows that

$$\|Sx\|^2 + \|U_{M+1} x\|^2 = \|x\|^2. \tag{2.80}$$

It is proved in [Mal12] that under reasonable assumptions on the wavelets $\{\phi_j, \psi_{\theta,j}\}$,

$$\|U_m x\|^2 \xrightarrow[m \to \infty]{} 0 \tag{2.81}$$

and more specifically that $U_m x$ has an exponential decay, so that scattering almost preserves the norm when M is sufficiently large.

$$\sum_{0 \le m \le M} \|S_m x\|^2 \approx \|x\|^2 \tag{2.82}$$

In practice, with the Morlet wavelets the decay is very fast so that only scattering of order $0, 1$ and 2 are computed and higher order paths are discarded.

If $\alpha > 0$ then a weaker version of (2.80) holds

$$(1-\alpha)^{M+1}\|x\|^2 \leq \|Sx\|^2 + \|U_{M+1}x\|^2 \leq \|x\|^2. \tag{2.83}$$

The lower bound is actually pessimistic. A tighter inequality is

$$\|x\|^2 \leq \frac{\|S_0 x\|^2}{1-\alpha} + \frac{\|S_1 x\|^2}{(1-\alpha)^2} + \ldots + \frac{\|S_M x\|^2}{(1-\alpha)^{M+1}} + \frac{\|U_{M+1}x\|^2}{(1-\alpha)^{M+1}} \tag{2.84}$$

but does not relate the norm of the scattering vector to the norm of x.

Since the wavelet transform is contractive (2.26) and that the complex modulus is contractive $\forall a, b \in \mathbb{C} ||a| - |b|| \leq |a - b|$, the wavelet modulus operator is also contractive

$$\forall x, y \in \mathbf{L}^2(\mathbb{R}^2), \quad \||\mathcal{W}|x - |\mathcal{W}|y\| \leq \|x - y\|. \tag{2.85}$$

Cascading this inequality yields

$$\|Sx - Sy\|^2 + \|U_{M+1}x - U_{M+1}y\|^2 \leq \|x - y\|^2 \tag{2.86}$$

and in particular, this proves that the scattering vector is contractive

$$\|Sx - Sy\|^2 \leq \|x - y\|^2. \tag{2.87}$$

Contractivness is an important property since it proves that the representation is robust to additive noise h. Indeed, the scattering of the noised version $x + h$ of x is not too far from the scattering of the original signal x

$$\|S(x + h) - Sx\| \leq \|h\|. \tag{2.88}$$

Finally, the scattering vector is obtained with a cascade of operators that are stable to deformations, and ends with an averaging that builds a translation invariance up to 2^J. Therefore, it is proved in [Mal12] that the scattering vector also has translation invariance up to 2^J and stability to deformation in the sense that there exists a constant $C > 0$ such that for any $x \in \mathbf{L}^2(\mathbb{R}^2)$ and any twice differentiable deformation τ for which $\|\nabla\tau\|_\infty \leq 1/2$,

$$\|S\mathcal{L}_\tau x - Sx\| \leq CMK(\tau)\|x\| \tag{2.89}$$

with

$$K(\tau) = 2^{-J}\|\tau\|_\infty + \|\nabla\tau\|_\infty \max\left(\log \frac{\|\Delta\tau\|_\infty}{\|\nabla\tau\|_\infty}, 1\right) + \|H\tau\|_\infty \tag{2.90}$$

The deformation operator $\mathcal{L}_\tau$ can also be applied to multivariate signals $\mathcal{L}_\tau y(u, p) = y(u - \tau(u),\ p)$. The proof essentially decomposes the difference $S\mathcal{L}_\tau x - Sx$ into two terms

$$S\mathcal{L}_\tau x - Sx = (\mathcal{L}_\tau Sx - Sx) \ + \ (S\mathcal{L}_\tau x - \mathcal{L}_\tau Sx) \tag{2.91}$$

The first term is bounded by the term $2^{-J}\|\tau\|_\infty$ because Sx ends with an averaging which builds a spatial regularity of 2^J. A change of variable shows that an upper bound on the second term can be obtained as an upper bound on a deformation of ψ. Since ψ is localized, this deformation can be bounded which results in the remaining term of $K(\tau)$.

Scattering vector only achieves local translation invariance, up to the size 2^J of the low-pass window. As shown in [Mal12], one can replace the windowing with a global averaging normalized by an appropriate constants μ_p that depends upon the path p.

$$\bar{S}x(p) = \mu_p^{-1} \int Ux(u,p)du \quad \text{with} \quad \mu_p = \int U\delta(u,p)du. \tag{2.92}$$

where δ is a centered dirac. As J grows, the set of computed paths also increases, so that defining a norm of $\bar{S}x$ over all paths is technical [Mal12]. The resulting fully delocalized scattering transform is strictly invariant to any translation $\bar{S}\mathcal{L}_v x = \bar{S}x$. In practice, images contains a finite number of samples and measurable scales. When 2^J becomes of the same order as the size of the image, it is reasonable to replace it with an averaging over the whole image, and the resulting scattering operators becomes fully translation invariant.

2.4.4 Scattering Implementation

The scattering vectors are computed by cascading several wavelet transform operators. The wavelet transform implementation can be chosen to be either the Fourier or the filter bank of Section 2.3.3. A first wavelet modulus transform computes $S_0 x$, subsampled at resolution 2^J, and $U_1 x(., \theta_1, j_1)$, subsampled at resolution 2^{j_1}. A second wavelet modulus operator is applied to every $U_1 x(., \theta_1, j_1)$ at input resolution j_1. This results in C wavelets modulus at each resolution 2^{j_1}, all of which costs $O(CNP)$ if the filter bank algorithm is used to implement the wavelet transform. Cascading this process results in a time complexity of $O(C^M NP)$ and a memory cost of $O(C^M N)$ where C is the number of orientations, M is the scattering order, N is the size of the input and P is the size of the filters. For classification purposes, we are often mostly interested in the invariant scattering coefficients Sx and the intermediate computations Ux can be discarded. Because all its components are downsampled by a factor 2^J, storing Sx requires $O(2^{-2J}(JC)^M N)$ coefficients, which in practice, is much smaller than the size N of the original image. For the fully translation invariant scattering $2^{2J} = N$ so that the representation has $O((C/2\log N)^M)$ coefficients.

2.5 Deep Convolutional Neural Networks

Deep neural networks are flexible architecture to learn signal representations. They have recently obtained state-of-the-art results in various applications including generic object classification [DCM+12, KSH12, ZF13]. A deep network usually consists in a cascade of linear operators, intertwined with a non-linearity f which computes a sequence of layers

$\Phi_m x$,

$$\Phi_m x = f(\mathcal{W}_m \dots f(\mathcal{W}_2 f(\mathcal{W}_1 x))). \tag{2.93}$$

These layers are computed one after another with a forward propagation

$$\Phi_m x = f(\mathcal{W}_m \Phi_{m-1} x). \tag{2.94}$$

The last layer $\Phi_M x$ should compute a desired function $y(x)$. For a classification task, the target function $y(x)$ is the label of the input image. The discrepancy between Φ_M and the target function is quantified by an error $E(\Phi x)$. The derivative of this errors with respect to the values of the output of each layer is computed by back propagation [LLB+98]. The top error $\frac{\partial E}{\partial \Phi_M}$ is initialized by simply computing the derivative of E in the point $\Phi_M x$, then the other derivatives are obtained recursively by applying the chain rule

$$\frac{\partial E}{\partial \Phi_{m-1} x} = \frac{\partial E}{\partial \Phi_m x} \frac{\partial f(\mathcal{W}_m \Phi_{m-1} x)}{\partial \Phi_{m-1} x}. \tag{2.95}$$

Once those errors derivatives are computed, the derivatives of the error with respect to the weights are deducted by applying the chain rule

$$\frac{\partial E}{\partial \mathcal{W}_m} = \frac{\partial E}{\partial \Phi_m} \frac{\partial f(\mathcal{W}_m \Phi_{m-1} x)}{\partial \mathcal{W}_m}. \tag{2.96}$$

The learning is done typically with a gradient descent algorithm which updates the weights with

$$\mathcal{W}_m \leftarrow (1-\eta)\mathcal{W}_m + \eta \frac{\partial E}{\partial \mathcal{W}_m} \tag{2.97}$$

where η is called the learning rate.

When the input signal has a large dimension (e.g. typically 100.000 for images), the number of weights becomes impractical. Convolutional neural networks [LKF10, KSH12, DCM+12, ZF13] limit the number of weights by localizing the operators $\mathcal{W}_m$ and sharing their weights across different positions. Their layers $\Phi_m x(u, p_m)$ are indexed by a spatial variable and a depth variable p_m. The computation of the weights becomes

$$\Phi_m x(u, p_m) = f\left(\sum_{\substack{v \in \Omega \\ p_{m-1}}} \Phi_{m-1}(u, p_{m-1}) w_{v, p_m, p_{m-1}} \right). \tag{2.98}$$

The localization corresponds to the fact that the sum is limited to a compact support Ω, while the convolution corresponds to the fact that $w_{v,p_m,p_{m-1}}$ does not depend upon the position u but only upon the offset v and the input and output path p_{m-1} and p_m.

In most deep network architecture for images, the spatial resolution of signals (i.e. the number of samples for u) decreases from one layer to the next, while the depth (i.e. the number of samples for p_m) grows, which yields a progressively more rich and more

invariant representation. This is similar to translation scattering. A major difference between the translation scattering and convolutional neural network as defined in (2.98) is that in (2.98), every output depth p_m is connected to every input depth p_{m-1}. On the contrary, a scattering path $p_m = (\theta_1, j_1, \dots \theta_{m-1}, j_{m-1}, \theta_m, j_m)$ is connected to only one previous path, its ancestor $p_{m-1} = (\theta_1, j_1, \dots \theta_{m-1}, j_{m-1})$. This implies that the translation invariance is built independently for different path, which can lead to information loss, as we shall explain in Section 4.2.

Chapter 3

Separable Scattering

3.1 Introduction

The translation scattering introduced in Chapter 2 builds a translation invariant representation by iterating wavelet modulus operators. Scattering inherits from the wavelet modulus operator the property of preserving the $\mathbf{L}^2$ norm of the signal, and the stability to deformations. Translation scattering provides a stable, informative, translation invariant representation suitable for some image recognition tasks such as digits recognition or texture classification [BM13] in situations where images undergo little geometric transformations.

Yet, due to the nature of the imaging process, which consists in a projection of three dimensional objects onto a two dimensional image plane, natural images undergo a wide range of geometric transformations. If the object is a smooth surface, as it is often the case for texture images, the three dimensional projection can be locally approximated by an affine transform. Therefore the construction of affine invariants has been a fundamental issue for texture recognition.

An affine transform consists in a translation and a linear transformation. Section 3.2 reviews the properties of affine transforms. A reasonable strategy to build affine invariant representations is to cascade a first translation invariant representation and second representation that builds invariance on linear transform, or the other way around. Such a strategies have been extensively developed for computer vision and in particular for texture recognition [LM01, OMP+02, LW03, LSP05]. Section 3.3 exhibits the similarities of such representations.

As it is the case for translations, a representation that is invariant to affine transformations while retaining sufficient information will require less examples to learn a model of classes or discriminative frontiers between distinct classes. Stability to deformations and retaining information are as important while building affine invariants as they were while building translation invariants.

Section 3.4 introduces a separable scattering representation that builds invariance to translations and rotations by applying successively two scatterings, the first one on the position variable, and the second one on the orientation variable. Both scattering are stable to deformations, preserve the norm of the signal and build invariance respectively to translations and rotations. Therefore, the resulting separable scattering is also stable, preserves the norm and is invariant to both translations and rotations.

3.2 Transformation Groups

The imaging process maps three dimensional surfaces onto the image plane. Projections of smooth surfaces can be locally approximated by affine transforms. This section reviews the properties of the affine group and some of its subgroups.

3.2.1 The Affine Group

An affine transform $g = (v, A)$ consists in an inversible linear mapping $A \in GL(\mathbb{R}^2)$ followed by a translation $v \in \mathbb{R}^2$. Affine transforms act upon image position $u \in \mathbb{R}^2$ through the group action

$$gu = v + Au. \tag{3.1}$$

The successive application of two affine transforms $g = (v, A)$ and $g' = (v', A')$ corresponds to the application of a single affine transform, denoted $g'g$

$$(g'g)u = g'(gu). \tag{3.2}$$

Equation (3.2) is called the compatibility of the group action with the group product. For the compatibility equation to hold, the group product must be defined as

$$g'g = (v' + A'v,\ A'A). \tag{3.3}$$

Unlike the translation addition, the affine group product (3.3) is not commutative for two reasons. First, the term $v' + A'v$ is not symmetric, secondly the matrix multiplication $A'A$ in $GL(\mathbb{R}^2)$ is not commutative. Because of the non-symmetric term $v' + A'v$, the affine group is not the product group $\mathbb{R}^2 \times GL(\mathbb{R}^2)$ of the translation group $\mathbb{R}^2$ and the linear group $GL(\mathbb{R}^2)$, but rather their semi-direct product, denoted $\mathbb{R}^2 \rtimes GL(\mathbb{R}^2)$.

A smooth three dimensional surface can be locally parameterized by a two dimensional position $u \in \mathbb{R}^2$. When an image x is lying on such a surface, which is projected onto the image plane, the parameterization u is mapped to a point $d(u)$ on the image plane. If the local flat image lying on the physical surface is $x(u)$, the observed projected image lying on the image plan is $x(d^{(-1)}(u))$. In the pinhole camera model, when the surface is smooth and the normal to the surface is not orthogonal to the ray from the surface point to the cameral pinhole, the projection d is a smooth deformation. A first order

Taylor approximation shows that any smooth deformation can be locally approximated by a two-dimensional affine transform

$$d(u_0 + u) = d(u_0) + \nabla d(u_0)u + O(|u|^2 \|Hd\|_\infty). \tag{3.4}$$

This shows that around u_0, the deformation d behaves almost like an affine transform $g = (d(u_0),\ \nabla d(u_0))$. Therefore, local affine invariants can be used to build more sophisticated invariant such as deformation invariants.

An affine transform acts on images by reverse action on the position

$$\mathcal{L}_g x(u) = x(g^{-1}u). \tag{3.5}$$

As translation operators $\mathcal{L}_v$, an affine transform $\mathcal{L}_g$ is an operator from $\mathbf{L}^2(\mathbb{R}^2)$ to $\mathbf{L}^2(\mathbb{R}^2)$. For translation, the action of $\mathcal{L}_v$ corresponds to the intuitive definition of x translated by a constant vector v, which is $\mathcal{L}_v x(u) = x_v(u) = x(u - v)$. The rotation by angle θ is the action $\mathcal{L}_\theta x(u) = x(r_{-\theta}u)$. For a general affine transform $g = (v, A)$, one can verify that the group inverse is

$$g^{-1} = (-A^{-1}v,\ A^{-1}) \tag{3.6}$$

so that

$$g^{-1}u = A^{-1}(u - v). \tag{3.7}$$

3.2.2 Subgroups of the Affine Group

It can be shown [MY09] that any element A of the linear group $GL(\mathbb{R}^2)$ can be decomposed into a rotation r_θ, a dilation s and a shear $T_{t,\beta}$

$$gu = v + r_\theta s T_{t,\beta}. \tag{3.8}$$

The shear $T_{t,\beta}$ consists in a dilation of t in the direction β, which can be expressed as

$$T_{t,\beta} = r_{-\beta} \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} r_{-\beta}. \tag{3.9}$$

By restricting A to subgroups of the linear group, one can define interesting subgroups of the affine group, for which the construction of invariant can be easier. For example, the rigid motion group, also called special euclidean group is the semi direct product of translations and rotations

$$SE(2) = \mathbb{R}^2 \rtimes SO(2) \tag{3.10}$$

It is of particular interest because the rotation groups has some convenient properties that the linear group does not have. In particular, the group of rotation of $SO(2)$ is commutative, periodic and is isomorphic to $\mathbb{R}/\mathbb{Z}$ endowed with the addition, on which there are already plenty of existing methods to build invariance. Also, in practice, we will

often consider discrete subgroups of the rotations with $2C$ elements by restricting A to rotations of angles that are multiples of π/C.

Another important example is the semi-separable product group of translation and dilations. The dilation group is isomorphic to $\mathbb{R}^{+*}$ endowed with the multiplication, or to $\mathbb{R}$ endowed with the addition if we consider the logarithm of the dilation factor. Unlike the rotation group, it is not periodic, and when working with images with a finite number of samples, only a limited range of dilations is available. Therefore, building invariant to dilation in more difficult than rotation because of these boundary effects. As rotations, dilations can be discretized for example by restricting dilations of factors $2^{j/Q}$ where Q is the number of scales per octave.

Both two dimensional rotations and dilations are isomorphic to a one dimensional additive group, on which there a plenty of tools to build invariance representations. The resulting separable representations are reviewed in section 3.3. In particular, one can build stable invariants through integration, such as separable scattering introduced in Section 3.4.

Combination of translations, rotations and dilations also form a group called the similitude group. On the other hand, shears $T_{t,\beta}$ do not constitute a group. Indeed, one can verify that the product of two shears $T_{t,0}T_{t,\pi/2} = t$ is a scaling which does not belong to the set of shears, and hence the set of shears endowed with the matrix multiplication does not constitute a group. Thus, if the construction of shear invariants is possible [LSP05, MY09], it can lead to instabilities or impracticalities.

3.3 Separable Representations

This section shows how to build affine invariant image representations by successively applying invariants to different subgroups of the affine group. Those strategies are called separable, in the sense that the different invariants involved act on different variables.

One family of separable strategy to build affine invariance starts with a first translation invariant operator

$$\Phi_1 : \mathbf{L}^2(\mathbb{R}^2) \to \mathbf{L}^2(\Lambda) \tag{3.11}$$

which is invariant to translation but covariant to the linear group $GL(\mathbb{R}^2)$. Covariance to A means that there exist an action on the path variable such that $\Phi\mathcal{L}_A x(p) = \Phi x(A^{-1}p)$ or more compactly

$$\forall A \in GL(\mathbb{R}^2), \quad \Phi\mathcal{L}_A = \mathcal{L}_A\Phi. \tag{3.12}$$

The two symbols $\mathcal{L}_A$ in (3.12) refer to two different mathematical objects. The first one refers to the affine transform of an image into another image, while the second one is an action that maps a function of $\mathbf{L}^2(\Lambda)$ to another function of $\mathbf{L}^2(\Lambda)$. For this to be of practical interest, if we already have computed the representation of the original Φx, the action on the representation $\mathcal{L}_A\Phi x$ should be faster to compute than the representation

of the transformed image $\Phi\mathcal{L}_A x$. Ideally, $\mathcal{L}_A\Phi x$ should be obtained by reindexing the coefficients of Φx.

The covariance property can be restricted to a subgroup $G \subset GL(\mathbb{R}^2)$ of the linear group, such as rotations or dilations. If (3.12) holds, $\Phi\mathcal{L}_A x$ can be obtained from Φx by reindexing the path variable p through the group action Ap. The group action defines an equivalence relationship

$$p \sim p' \iff \exists A \in G / p = Ap'. \tag{3.13}$$

The class of equivalence for this relationship are the orbits Gp defined as

$$Gp = \{Ap\}_{A \in G} \tag{3.14}$$

The set of all orbits defines a partition over the set of all paths Λ. The set of orbits is

$$\bar{\Lambda} = \Lambda / G. \tag{3.15}$$

For each orbit, a particular element $\bar{p}$ that belongs to the orbit is chosen arbitrarly and will uniquely identify the orbit. Any given path p can then be factorized into its unique orbit index $\bar{p}$, and a position A within this orbit

$$p = A\bar{p} \tag{3.16}$$

This corresponds to a factorization of the transformed space Λ into

$$\Lambda = G \times \bar{\Lambda} \tag{3.17}$$

Let us suppose that we know how to build an operator $\Phi_2 : \mathbf{L}^2(G) \to \mathbf{L}^2(\Lambda_2)$ on functions of G (not of Λ) such that

$$\forall A \in G, \quad \Phi_2\mathcal{L}_A = \Phi_2 \tag{3.18}$$

where $\mathcal{L}_A y(B) = y(A^{-1}B)$. Equation (3.18) means that the operator Φ_2 is fully invariant to the action of A. Given such Φ_1 and Φ_2, we can now see $\Phi_1 x$ as a set of functions on G indexed by $\bar{p}$, and apply Φ_2 to each of these functions. This yields a separable invariant

$$\Phi x(\bar{p}, p_2) = \Phi_2(A \mapsto \Phi_1 x(A\bar{p}))(p_2) \tag{3.19}$$

For compactness of notation, given any operator $\Phi_2 : \mathbf{L}^2(G) \to \mathbf{L}^2(\Lambda_2)$ we also denote $\Phi_2^{(\Lambda)}$ the operator that applies Φ_2 along A for fixed values of $\bar{p}$, defined by

$$\Phi_2^{(\Lambda)} : \quad \mathbf{L}^2(G \times \bar{\Lambda}) \to \mathbf{L}^2(\Lambda_2 \times \bar{\Lambda}) \tag{3.20}$$

$$\big((A, \bar{p}) \mapsto y(A\bar{p})\big) \mapsto \big((\bar{p}, p_2) \mapsto \Phi_2(A \mapsto y(A\bar{p}))(p_2)\big) \tag{3.21}$$

With this notation, we simply have

$$\Phi = \Phi_2^{(\Lambda)}\Phi_1 \tag{3.22}$$

Such a Φ is called a separable invariant because a first operator Φ_1 builds invariance to translation $\mathcal{L}_v, v \in \mathbb{R}^2$ and then a second invariant Φ_2 build invariance to linear transforms $\mathcal{L}_A, A \in G$. The following theorem states the invariance property of Φ.

Theorem 1. *The operator* $\Phi = \Phi_2^{(\Lambda)}\Phi_1$ *is invariant to the action* $\mathcal{L}_g$ *for any* g *in the semi-direct product group* $\mathbb{R}^2 \rtimes G$.

Proof. For $g = (v, A) \in \mathbb{R}^2 \rtimes G$, the action of the affine transform $\mathcal{L}_g$ can be factorized into

$$\mathcal{L}_g x(u) = x(A^{-1}(u - v)) = \mathcal{L}_v \mathcal{L}_A x(u) \tag{3.23}$$

then

$$\begin{aligned}
\Phi\mathcal{L}_g &= \Phi_2^{(\Lambda)}\Phi_1\mathcal{L}_v\mathcal{L}_A && (3.24)\\
&= \Phi_2^{(\Lambda)}\Phi_1\mathcal{L}_A \quad \text{since } \Phi_1 \text{ is translation invariant} && (3.25)\\
&= \Phi_2^{(\Lambda)}\mathcal{L}_A\Phi_1 \quad \text{since } \Phi_1 \text{ is G covariant} && (3.26)\\
&= \Phi_2^{(\Lambda)}\Phi_1 \quad \text{since } \Phi_2^{(\Lambda)} \text{ is G invariant.} && (3.27)\\
&= \Phi && (3.28)
\end{aligned}$$

which proves that Φ is invariant to the action of $\mathcal{L}_g$ for any $g \in \mathbb{R}^2 \rtimes G$. □

Theorem 1 states that the cascade of a translation invariant and a linear transform invariant yields an affine invariant which is called separable. We will further refer to these type of invariant as separable type I. There is another possible construction of separable invariant, which is to start the cascade with a local invariant to linear transform and then to build a translation invariant on top. A similar theorem can be derived for these other invariants, which will be referred to as separable type II.

Separable invariants have the advantage of simplicity. Each operator builds invariance to a single variable, which is conceptually straightforward. Separable invariants have been widely used in texture recognition. A typical example of type I separable invariant is the Local Binary Pattern Histogram Fourier (LBP-HF) [ZAMP12]. First, local binary pattern (LBP) are computed. They consist in local descriptors of the image , obtained by thresholding the image pixel values, or differences of values, on several circular grids around the patch center. This results in a binary vector $p(u)$ that is different for each position u of the patch center. The number of different possible binary vectors is 2^d where d is the dimension of the vector $p(u)$. Typically $2^d = 256$ is relatively small, so that one can compute an histogram over all the patches of the image

$$\Phi_1^{\text{LBP-H}}(p) = \sum_u \mathbb{1}_{p(u)=p}. \tag{3.29}$$

Due to the global spatial averaging, the histogram Φ_1^{LBP} is fully translation invariant. Since the patterns p are computed from samplings on circular grids, they are covariant to rotations and can therefore be expressed as a rotation r_θ of a normalized pattern $\bar{p}$

$$\Phi_1^{\text{LBP-H}}(p) = \Phi_1^{\text{LBP-H}}(\mathcal{L}_\theta \bar{p}). \tag{3.30}$$

To achieve rotation invariant, a first possible strategy presented in [ZAMP12] is to average together all the pattern that are equals up to a rotation. They denote the resulting descriptor $\mathrm{LBP}^{\mathrm{riu2}}$

$$\Phi^{\mathrm{LBP}^{\mathrm{riu2}}}(\bar{p}) = \sum_{\theta} \Phi_1^{\text{LBP-H}}(\mathcal{L}_\theta \bar{p}). \tag{3.31}$$

The authors of [ZAMP12] observes that the averaging along θ loses too much information along the orbit compared to what is necessary to build invariance. Therefore, they also build another descriptor that they call LBP-HF (Histogram Fourier), by replacing the averaging of (3.31) with a Fourier modulus operator along each rotation orbit

$$\Phi^{\text{LBP-HF}}(\omega, \bar{p}) = |\mathcal{F}|\left(\theta \mapsto \Phi_1^{\text{LBP-H}}(\mathcal{L}_\theta \bar{p})\right)(\omega). \tag{3.32}$$

The Fourier modulus only discards a global phase, which relates to the global orientation of the image, but preserves the relative angular distribution of patterns. It is reported in [ZAMP12] that LBP-HF provides a significant improvement of classification accuracy over $\mathrm{LBP}^{\mathrm{riu2}}$.

$\mathrm{LBP}^{\mathrm{riu2}}$ and LBP-HF are separable type I in the sense that they start with a translation invariant operator, followed by an invariant operator over a subgroup of linear transforms, in this case, the rotation group. There are also many instances of separable type II invariants in the texture classification literature. A typical example is the Rotation Invariant Feature Transform (RIFT) or Spin Image introduced in [LSP05]. This approach starts by computing patch descriptors that are invariant to any linear transform centered on the patch. Local invariance to scaling and shear is obtained by detection of an Harris or Laplacian region and normalization with respect to these regions. On the normalized patch, an histogram of gradient orientation, where the orientation is measured with respect to the eccentricity vector, and is therefore invariant to patch-centered rotations. The resulting $\Phi_1^{(RIFT)}$ description is translation covariant, but invariant to local linear transform. Is is then vector quantized and spatially averaged into a global histogram to build translation invariance.

3.4 Separable Rigid-Motion Scattering

We argue that affine invariant representations have the same desirable properties than translation invariant representations. They should also be stable to deformation and retain as much information as possible. Therefore, when building separable invariance, all the operators involved should have those properties, and therefore, a good candidate for Φ_2 is a scattering invariant. Separable rigid-motion scattering $\mathring{S}$ is a separable invariant where both operators are scattering operators that act respectively on the translation and rotation variable.

3.4.1 Covariance Property of the Translation Scattering

The translation scattering reviewed in Chapter 2 consists in a cascade of convolutions with oriented and dilated wavelets. Let us first focus on rotations and oriented wavelets $\psi_\theta(u) = \psi(r_{-\theta}u)$. For any rotation r_θ,

$$\begin{aligned}(\mathcal{L}_\theta x) \star \psi_{\theta_1}(u) &= \int_{\mathbb{R}^2} x(r_{-\theta}v)\psi(r_{-\theta_1}(u-v))dv && (3.33)\\ &= \int x(w)\psi(r_{-\theta_1}(u-r_\theta w))dw && (3.34)\\ &= \int x(w)\psi(r_{-(\theta_1-\theta)}(r_{-\theta}u-w))dw && (3.35)\\ &= \mathcal{L}_\theta(x \star \psi_{\theta_1-\theta})(u). && (3.36)\end{aligned}$$

Since the modulus is covariant to rotation we also have

$$|(\mathcal{L}_\theta x) \star \psi_{\theta_1}| = \mathcal{L}_\theta |x \star \psi_{\theta_1-\theta}|. \tag{3.37}$$

Cascading this equality yields

$$U_m(\mathcal{L}_\theta x)(u, \theta_1, j_1 \ldots, \theta_m, j_m) = U_m x(r_{-\theta}u, \theta_1 - \theta, j_1, \ldots, \theta_m - \theta, j_m). \tag{3.38}$$

For a fully delocalized scattering $J = \infty$, the scattering is simply

$$S_m x(p) = \int_{\mathbb{R}^2} U_m x(u, p)du \tag{3.39}$$

which is fully translation invariant and therefore does not depend upon u. From (3.38) we thus derive

$$S_m(\mathcal{L}_\theta x)(\theta_1, j_1 \ldots, \theta_m, j_m) = S_m x(\theta_1 - \theta, j_1, \ldots, \theta_m - \theta, j_m). \tag{3.40}$$

Therefore, if we define the action of the rotation group $SO(2)$ over $\mathbf{L}^2(\mathfrak{P}_m)$ as

$$\forall y \in \mathbf{L}^2(\mathfrak{P}_m), \quad \mathcal{L}_\theta y(\theta_1, j_1, \ldots, \theta_m) = y(\theta_1 - \theta, j_1, \ldots, \theta_m - \theta, j_m), \tag{3.41}$$

then the scattering has the covariance property

$$S_m \mathcal{L}_\theta = \mathcal{L}_\theta S_m. \tag{3.42}$$

3.4.2 Wavelets on the Rotation Parameter

The group action (3.41) is nothing but a translation of θ in the direction $(1, 0, \ldots, 1, 0)^T$. Therefore, building an invariant to this action can be done with any translation invariant operator. As it is the case for spatial translation, a good invariant operator should be stable and retain as much information as possible. We therefore define a scattering operator on $SO(2)$ and apply it along the rotation orbit of the translation scattering.

To define a scattering operator on $\mathbf{L}^2(SO(2))$ we need to define a wavelet transform, and hence convolutions and wavelets on this space. $\mathbf{L}^2(SO(2))$ is equivalent to the space of $\mathbf{L}^2([0,2\pi))$. For $\mathring{x} \in \mathbf{L}^2([0,2\pi))$, we abusively denote $\mathring{x}(\theta) = \mathring{x}(\theta \mod 2\pi)$ which is valid for any $\theta \in \mathbb{R}$ not only for $\theta \in [0,2\pi)$. With this notation, the convolution of two functions $\mathring{x}, \mathring{y} \in \mathbf{L}^2(SO(2))$ is simply

$$\mathring{x} \circledast \mathring{y}(\theta) = \int_{\theta' \in [0,2\pi)} \mathring{x}(\theta') \mathring{y}(\theta - \theta') d\theta'. \tag{3.43}$$

The Fourier transform on $\mathbf{L}^2([0,2\pi))$, also called the Fourier series in that case, is defined as

$$\begin{aligned} \mathcal{F} : \mathbf{L}^2([0,2\pi)) &\to \mathbf{L}^2(\mathbb{Z}) && (3.44)\\ \mathring{x} &\mapsto \left(m \mapsto \int_0^{2\pi} \mathring{x}(u) e^{-im\theta} d\theta \right). && (3.45) \end{aligned}$$

If $x \in \mathbf{L}^2(\mathbb{R})$ we define its periodization

$$\mathring{x}(\theta) = \sum_{n \in \mathbb{Z}} x(\theta - 2\pi n), \tag{3.46}$$

which belongs to $\mathbf{L}^2([0,2\pi))$. Then,

$$\begin{aligned} \mathcal{F}\mathring{x}(m) &= \sum_{n \in \mathbb{Z}} \int_0^{2\pi} x(\theta - 2\pi n) e^{-im\theta} d\theta && (3.47)\\ &= \sum_{n \in \mathbb{Z}} \int_{2n\pi}^{2(n+1)\pi} x(\theta) e^{-im\theta} d\theta && (3.48)\\ &= \mathcal{F}x(m). && (3.49) \end{aligned}$$

The Fourier transform of the periodized signal $\mathring{x}$ is the same as the Fourier transform of the original signal x, but restricted to integer frequencies $m \in \mathbb{Z}$. With these definitions, the Fourier transform preserves the norm

$$\forall \mathring{x} \in \mathbf{L}^2([0,2\pi)), \quad \int_0^{2\pi} |\mathring{x}(du)|^2 du = (2\pi)^{-1} \sum_{m \in \mathbb{Z}} |\mathcal{F}\mathring{x}(m)|^2 \tag{3.50}$$

and the convolution theorem also holds for any $\mathring{x}, \mathring{y} \in \mathbf{L}^2([0,2\pi))$

$$\forall m \in \mathbb{Z}, \quad \mathcal{F}(\mathring{x} \circledast \mathring{y})(m) = \mathcal{F}\mathring{x}(m) \mathcal{F}\mathring{y}(m). \tag{3.51}$$

A periodic wavelet family $\{\mathring{\phi}_K, \mathring{\psi}_k\}$ is obtained by periodizing a one-dimensional scaling function $\phi_K(\theta) = 2^{-K}\phi(2^{-K}\theta)$ and a one-dimensional wavelet $\psi_k(\theta) = 2^{-k}\psi(2^{-k}\theta)$ with

$$\begin{aligned} \mathring{\phi}_K(\theta) &= \sum_{n \in \mathbb{Z}} \phi_K(\theta - 2\pi n) && (3.52)\\ \mathring{\psi}_k(\theta) &= \sum_{n \in \mathbb{Z}} \psi_k(\theta - 2\pi n). && (3.53) \end{aligned}$$

Having defined wavelets, we can define the wavelet transform on $\mathbf{L}^2([0,2\pi))$ as

$$\mathcal{W}\mathring{x} = \{\mathring{x} \circledast \mathring{\phi}_K, \mathring{x} \circledast \mathring{\psi}_k\}_k. \tag{3.54}$$

As the wavelet transform on $\mathbf{L}^2(\mathbb{R}^2)$, the wavelet transform on $\mathbf{L}^2([0,2\pi))$ also preserves the norm, as stated by the following property.

Property 1. *If $\{\phi_K, \psi_k\}_{-\infty<k<K}$ is an α frame of $\mathbf{L}^2(\mathbb{R})$, then $\{\mathring{\phi}_K, \mathring{\psi}_k\}_{-\infty<k<K}$ defined by (3.52-3.53) is also an α frame of $\mathbf{L}^2([0,2\pi))$.*

Proof. Indeed, let us suppose that $\{\phi_K, \psi_k\}_{-\infty<k<K}$ is an α frame. Since the Fourier transform of a periodized signal is the discretized Fourier transform of the original signal (3.47-3.49), α is also a bound on the discretized Littlewood-Paley sum of $\{\mathring{\phi}_K, \mathring{\psi}_k\}_{-\infty<k<K}$

$$\forall m \in \mathbb{Z}, \quad 1-\alpha \leq |\mathcal{F}\mathring{\phi}_K(m)|^2 + \sum_{k<K} |\mathcal{F}\mathring{\psi}_k(m)|^2 \leq 1. \tag{3.55}$$

In the following equations, we apply the Plancherel theorem, the convolution theorem (3.57), the bound on the Littlewood Paley sum on $\{\mathring{\phi}_K, \mathring{\psi}_k\}_k$ (3.58) and the Plancherel theorem back (3.59)

$$\begin{aligned}
\|\mathcal{W}\mathring{x}\|^2 &= \|\mathring{x} \circledast \mathring{\phi}_K\|^2 + \sum_{k<K} \|\mathring{x} \circledast \mathring{\psi}_k\|^2 && (3.56)\\
&= (2\pi)^{-1} \left(\|\mathcal{F}(\mathring{x} \circledast \mathring{\phi}_K)\|^2 + \sum_{k<K} \|\mathcal{F}(\mathring{x} \circledast \mathring{\psi}_k)\|^2 \right) && (3.57)\\
&= (2\pi)^{-1} \sum_{m\in\mathbb{Z}} |\mathcal{F}\mathring{x}(m)|^2 \left(|\mathcal{F}\mathring{\phi}_K(m)|^2 + \sum_{k<K} |\mathcal{F}\mathring{\psi}_k(m)|^2 \right) && (3.58)\\
&\leq (2\pi)^{-1} \sum_{m\in\mathbb{Z}} |\mathcal{F}\mathring{x}(m)|^2 && (3.59)\\
&= \|\mathring{x}\|^2. && (3.60)
\end{aligned}$$

This proves that $\|\mathcal{W}\mathring{x}\|^2 \leq \|\mathring{x}\|^2$. One can prove similarly that $(1-\alpha)\|\mathring{x}\|^2 \leq \|\mathcal{W}\mathring{x}\|^2$ and thus, $\{\mathring{\phi}_K, \mathring{\psi}_k\}_k$ is an α frame. □

As the wavelet transform over $\mathbb{R}^2$, the periodic wavelet transform separate the signal $\mathring{x}$ into an invariant $\mathring{x} \circledast \mathring{\phi}_K$ and a covariant part $\{\mathring{x} \circledast \mathring{\psi}_k\}_k$. The invariant part $\mathring{x} \circledast \mathring{\phi}_K$ is invariant up to translation of θ of size 2^K. The following property states that full invariance can be achieved with a finite maximum scale K in the periodic case.

Property 2. *If $\mathcal{F}\phi$ has a compact support, there exists a finite $K \in \mathbb{Z}$ such that $\mathring{\phi}_K$ is constant, in which case $\mathring{x} \circledast \mathring{\phi}_K$ is fully invariant by translation of the variable θ.*

Proof. Indeed, in this case, there exists a finite K such that if the support of $\mathcal{F}\phi$ is included in $]2^K, 2^K[$. Therefore, all the terms of the Fourier series of $\mathcal{F}\mathring{\phi}_K$ are null, except $\mathcal{F}\mathring{\phi}_K(0)$. Thus $\mathring{\phi}_K$ is constant and so is $\mathring{x} \circledast \mathring{\phi}_K$. □

3.4.3 Periodic Scattering on the Rotation Parameter

Having defined the periodic wavelet transform on $\mathbf{L}^2([0,2\pi))$, the periodic scattering of a function $\mathring{x} \in \mathbf{L}^2([0,2\pi))$ is obtained by iterating over a wavelet modulus operator, as the two dimensional scattering over $\mathbb{R}^2$

$$\mathring{U}_{\mathring{m}}\mathring{x}(\theta, k_1, \dots, k_{\mathring{m}}) = |\dots|\mathring{x} \circledast \mathring{\psi}_{k_1}| \dots \star \mathring{\psi}_{k_{\mathring{m}}}|(\theta) \tag{3.61}$$

$$\mathring{S}_{\mathring{m}}\mathring{x}(\theta, k_1, \dots, k_{\mathring{m}}) = |\dots|\mathring{x} \circledast \mathring{\psi}_{k_1}| \dots \star \mathring{\psi}_{k_{\mathring{m}}}| \circledast \mathring{\phi}_K(\theta). \tag{3.62}$$

Which can be summarized as

$$\{\mathring{S}_{\mathring{m}}\mathring{x},\ \mathring{U}_{\mathring{m}+1}\mathring{x}\} = |\mathcal{W}|\mathring{U}_{\mathring{m}}x \tag{3.63}$$

The resulting periodic scattering has the same properties as the translation scattering of Chapter 2, that is invariance to translations of the variable θ, stability to deformations and it retains most of the information contained in $\mathring{x}$. Similarly to the translation scattering, we define the scattering path along the rotation parameter as the sequence of indices of one dimensional wavelets modulus that $\mathring{U}_{\mathring{m}}\mathring{x}(\theta, \mathring{p})$ or $\mathring{S}_{\mathring{m}}\mathring{x}(\theta, \mathring{p})$ has been through

$$\mathring{p}_{\mathring{m}} = (k_1, \dots, k_{\mathring{m}}) \tag{3.64}$$

The length $\mathring{m}$ of such a path is the rotation scattering order. As in translation scattering, we compute all scattering order up to a maximum order $\mathring{M}$. Choosing $\mathring{M}$ to 0 is equivalent to averaging along the rotation parameter which provides a limited set of invariants. Increasing $\mathring{M}$ provides richer invariant coefficients and potentially better classification results at a higher computational cost.

The periodic wavelet transform and scattering implementation can be done using either the Fourier or the filter bank implementation described in Section 2.3.3. Theoretically, after a wavelet transform, one should be able to downsample the signal along the rotation parameter proportionaly to the scale of the wavelet $\mathring{\psi}_k$. Yet, in practice, we operate with a small number of angles $C = 8$ and any intermediate downsampling introduce non-negligible aliasing that degrades the rotation invariance. We therefore only perform downsampling after the low-pass along rotation $\mathring{\phi}_K$. Also, since the number of angles is small, and the convolutions are periodic, there are no padding issues and the filters support are about the same size as the signal, therefore Fourier based convolutions are faster in this regime. A periodic wavelet transform without downsampling actually consists in K Fourier based convolutions thus has a cost of $O(KC \log C)$ where K is the number of scales and C is the number of angles. The periodic scattering consists in a cascade of $\mathring{M}$ such wavelet modulus operators and therefore has a cost of $O\left((KC \log C)^{\mathring{M}}\right)$. The intermediate memory cost is $O(KC)$. After the low-pass $\mathring{\phi}_K$, we perform a final downsampling of 2^K so that the storage requirement is $O\left(K2^{-K}C\right)$.

3.4.4 Separable Scattering

Now let us go back to the fully translation invariant scattering. It is a function

$$S_m x(\theta_1, j_1, \dots, \theta_m, j_m) = \int_{\mathbb{R}^2} |\dots|x \star \psi_{\theta_1,j_1}| \dots \star \psi_{\theta_m,j_m}|(u)du \tag{3.65}$$

that does not depend upon the spatial position u and which is therefore invariant to any translation

$$\forall v \in \mathbb{R}^2, \quad S_m x = S_m \mathcal{L}_v x. \tag{3.66}$$

It has covariance property with respect to rotation

$$S_m \mathcal{L}_\theta x = \mathcal{L}_\theta S_m x. \tag{3.67}$$

We factorize $p_m = (\theta_1, j_1, \dots, \theta_m, j_m)$ into

$$\begin{aligned} p_m &= \theta_1(0, j_1, \theta_2 - \theta_1, j_2, \dots, \theta_m - \theta_1, j_m) && (3.68)\\ &= \theta_1 \bar{p}_m && (3.69) \end{aligned}$$

Now we consider $S_m x$ as a set of orbit functions $\theta \mapsto S_m x(\theta \bar{p}_m)$, indexed by $\bar{p}_m$. Each of these functions belong to $\mathbf{L}^2(SO(2))$, we can therefore apply to them the periodic scattering operator $\mathring{U}_{\mathring{m}}$ or $\mathring{S}_{\mathring{m}}$. This yields a separable scattering operators $\mathring{U}_{m,\mathring{m}}$ and $\mathring{S}_{m,\mathring{m}}$

$$\begin{aligned} \mathring{U}_{m,\mathring{m}} x(\theta_1, \bar{p}_m, \mathring{p}_{\mathring{m}}) &= \mathring{U}_{\mathring{m}}\big(\theta \mapsto S_m x(\theta \bar{p})\big)(\theta_1, \mathring{p}) && (3.70)\\ \mathring{S}_{m,\mathring{m}} x(\theta_1, \bar{p}_m, \mathring{p}_{\mathring{m}}) &= \mathring{S}_{\mathring{m}}\big(\theta \mapsto S_m x(\theta \bar{p})\big)(\theta_1, \mathring{p}) && (3.71) \end{aligned}$$

Separable scattering is indexed by an orbit position θ_1, an orbit $\bar{p}$ that are reminiscent from the initial translation scattering, and a periodic scattering path $\mathring{p}$ that has been created by the second scattering along orientation.

The separable scattering vector $\mathring{S}x$ is the concatenation of all scattering coefficients at all orders up to maximal order M and $\mathring{M}$, at all orbit positions, orbit indices and periodic scattering paths.

$$\mathring{S}x = \{\mathring{S}_{m,\mathring{m}} x(\theta_1, \bar{p}_m, \mathring{p}_{\mathring{m}})\}_{0 \le m \le M,\ 0 \le \mathring{m} \le \mathring{M},\ \theta_1, \bar{p}_m, \mathring{p}_{\mathring{m}}} \tag{3.72}$$

Since the initial translation scattering is fully translation invariant, the separable scattering is also fully translation invariant. If the scale 2^K of the rotation scattering is sufficiently large, the separable scattering is also fully invariant to rotations. In these case, it does not depends on the orbit position θ_1 but only on the orbit index $\bar{p}_m$ and on the rotation path $\mathring{p}_{\mathring{m}}$.

The computation of the separable scattering is illustrated in figure 3.1. First, a translation scattering is computed. The rotation orbit are extracted from the translation scattering. Each of these orbit are one dimensional signals that we apply a periodic scattering to.

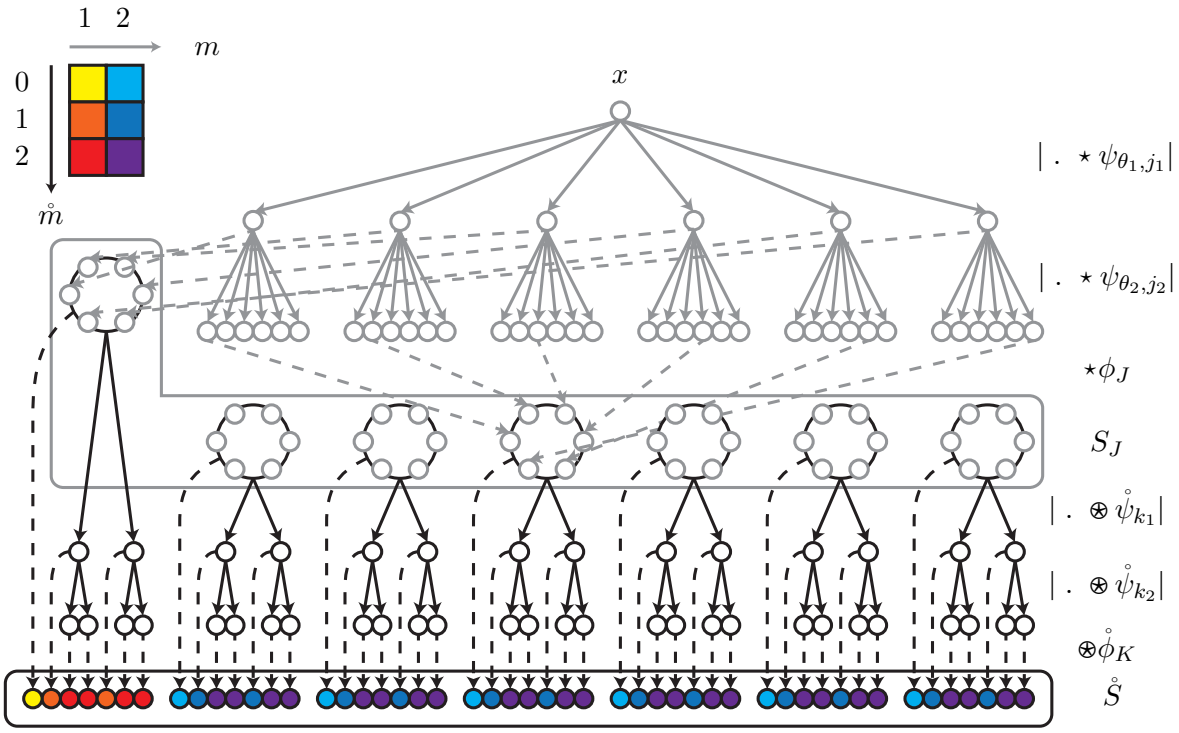


Figure 3.1: Separable scattering architecture. First spatial scattering layers in grey, second scattering layers in black. Spatial wavelet-modulus operators (grey arrows) are averaged (doted grey arrows), as in [BM13]. Outputs of the first scattering are reorganized in different orbits (large black circles) of the action of the rotation on the representation. A second cascade of wavelet-modulus operators along the orbits (black arrows) splits the angular information in several paths that are averaged (doted black arrows) along the rotation to achieve rotation invariance. Output nodes are colored with respect to the order $m, \mathring{m}$ of their corresponding paths.

Finally, all the output of all periodic scattering are concatenated. The implementation of the translation an rotation scattering can be chosen depending upon the application. Since the translation scattering is not periodic, a filter bank implementation is often faster. Since the rotation scattering is periodic, with wavelet sizes of the same order than the size of signal, a FFT-based implementation is faster. For these choices the total time complexity of the separable scattering is $O\left(C^M NP + C^{M-1}(\log N/2)^M (KC\log C)^{\mathring{M}}\right)$ where C is the number of angles, N the size of the image, P the size of the spatial filters, M is the maximum spatial scattering order and $\mathring{M}$ is the maximum rotation scattering order. Indeed, the cost of the translation scattering is $O(C^M NP)$. Then, we apply a one dimensional Fourier based scattering with no intermediate downsampling, on signals of size C which has a cost

$O(KC \log C)^{\mathring{M}}$). These signal corresponds to rotational orbits. Since the translation scattering representation has a size $O(C^M (\log N/2)^M)$, there are $O(C^{M-1}(\log N/2)^M)$ such orbits.

Chapter 4

Joint Scattering

4.1 Introduction

The separable scattering introduced in Chapter 3 builds stable invariant to translations and rotations by cascading a first scattering that builds full translation invariance, and a second scattering that builds rotational invariance. This approach has the advantage of simplicity but it also discards important information about the joint distribution of position and orientation of intermediate layers. Section 4.2 argues that all separable invariants, which are common in the texture classification literature, also discard similar information. Retaining this information requires to adapt signal processing tools to the specificities of transformation groups, as proposed by several authors [CS06, BDGR12, DB07, DF11]. Another possibility is to learn much more general representations such as deep neural networks [LLB$^+$98, HS06, LKF10, DCM$^+$12, KSH12], that have enough expressive power to capture those specificities. The translation group has a simple structure that allows to build convolutional operators which dramatically simplify the task of designing or learning a translation invariant representation. We argue that the same properties hold for slightly more complex subgroups of the affine group, in particular for the rigid-motion group consisting of translations and rotations. Section 4.3 constructs a multiresolution analysis on the rigid-motion group, with convolutions, wavelets, wavelet transform operators and fast implementations. Section 4.4 builds a joint rigid-motion scattering, that iterates over the rigid-motion wavelet transform. Joint scattering provides a representation that is invariant to translations and rotations, but which provides much tighter invariance compared to separable scattering or other separable representations.

4.2 Joint versus Separable Invariants

Chapter 3 has reviewed some separable invariants in the texture classification and introduced a stable translation and rotation invariant separable scattering. The separable

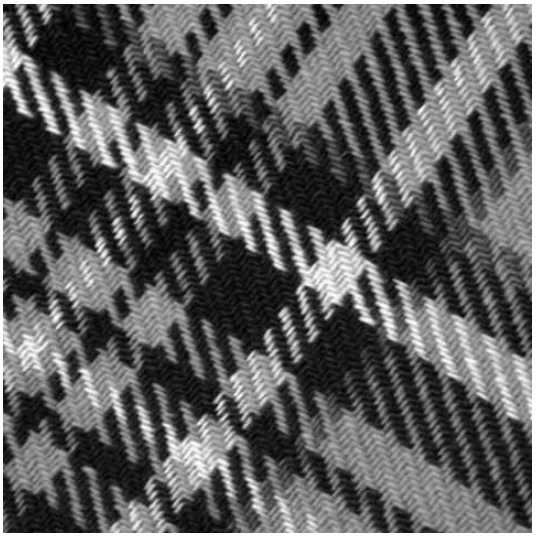
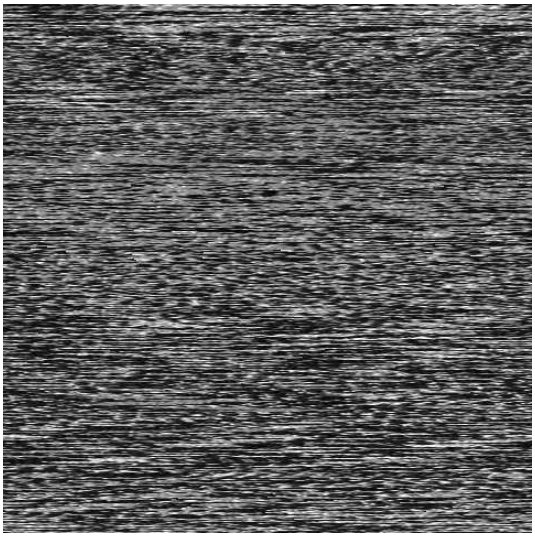



Figure 4.1: A texture image x (left) from the UIUCTex dataset [LSP05]. Translating all the rows independently yields an image $\mathcal{L}_\tau x$ (right) that is perceptually very different from the original. Yet, separable translation invariants are also invariant to such transformations, which shows that they potentially discard important information for classification.

scattering is a type 1 separable invariant, in the sense that it builds a first translation invariant, rotation covariant representation, and applies a second scattering transform on the rotations orbits of the first scattering transform. We argue that such invariant, by separating the translation and orientation variables, loses important information that may be helpful to discriminate different classes.

4.2.1 Separable Translation Invariance

To illustrate the loss of information of separable invariants, let us first consider the two-dimensional translation group $\mathbb{R}^2$. The two-dimensional translation group can be considered as the product of two one dimensional translation groups $\mathbb{R}$. A separable invariant to these groups can be build by applying, at each vertical position u_2, a first translation invariant operator $\Phi_1 : \mathbf{L}^2(\mathbb{R}) \to \mathbf{L}^2(\Lambda)$ along the horizontal position u_1. Φ_1 could be any one dimensional translation invariant operator, such as Fourier modulus or scattering. This defines an extended operator $\Phi_1^{(\mathbb{R}^2)} : \mathbf{L}^2(\mathbb{R}^2) \to \mathbf{L}^2(\mathbb{R} \times \Lambda)$ as

$$\Phi_1^{(\mathbb{R}^2)} x(u_2, p) = \Phi_1\big(u_1 \mapsto x(u_1, u_2)\big)(p). \tag{4.1}$$

The operator $\Phi_1^{(\mathbb{R}^2)}$ is invariant to translations of the variable u_1, and covariant to translations of the variable u_2,

$$\Phi_1^{(\mathbb{R}^2)} \mathcal{L}_{(v_1,v_2)} x(u_2, p) = \Phi_1^{(\mathbb{R}^2)} x(u_2 - v_2, p). \tag{4.2}$$

A second one dimensional translation invariant operator $\Phi_2 : \mathbf{L}^2(\mathbb{R}) \to \mathbf{L}^2(\mathbb{R})$ can now be applied to $\Phi_1^{(\mathbb{R}^2)} x$ along the vertical variable u_2. This yields a separable operator $\Phi : \mathbf{L}^2(\mathbb{R}^2) \to \mathbf{L}^2(\Lambda^2)$ defined by

$$\Phi x(p_1, p_2) = \Phi_2 \left(u_2 \mapsto \Phi_1^{(\mathbb{R}^2)} x(u_2, p_1) \right)(p_2). \tag{4.3}$$

Applying the translation invariance of Φ_2 to (4.2) proves that the resulting separable operator Φ is invariant to any two dimensional translations $\mathcal{L}_v$. Yet, let $\tau(u) = (v_1(u_2), v_2)$ be independent horizontal translations $v_1(u_2)$ of each rows, and a global translation v_2 of all columns. Then Φ is also invariant to $\mathcal{L}_\tau$. Indeed

$$\begin{aligned}
\Phi_1^{(\mathbb{R}^2)} \mathcal{L}_\tau x(u_2, p) &= \Phi_1\big(u_1 \mapsto x(u_1 - v_1(u_2), u_2 - v_2)\big)(p) && (4.4)\\
&= \Phi_1\big(u_1 \mapsto x(u_1, u_2 - v_2)\big)(p) && (4.5)\\
&= \Phi_1^{(\mathbb{R}^2)} x(u_2 - v_2, p) && (4.6)
\end{aligned}$$

and since Φ_2 is also translation invariant, Φ is invariant to $\mathcal{L}_\tau$. Such a τ belongs to the group $\mathbb{R}^{\mathbb{R}} \times \mathbb{R} = \{(u_1, u_2) \mapsto (v_1(u_2), v_2)\}$ which is a much larger group than the translation group $\mathbb{R}^2$. This toy example shows that translation separable invariants are too loose. Indeed, they are invariant to a much wider set of transformations than what was originally intended by design, i.e. global two dimensional translations. Figure 4.1 shows an image obtained by applying such τ to a texture image. The original and the transformed images are perceptually very different and an operator that maps those two images to the same representation is not likely to perform well at perception related tasks such as image classification. Figure 4.2 shows that a separable Fourier modulus transform is invariant to such $\mathcal{L}_\tau, \tau \in \mathbb{R}^{\mathbb{R}} \times \mathbb{R}$, whereas the usual two dimensional Fourier modulus transform is not, because it takes into account the joint structure of the translation group. In particular, a two dimensional Fourier modulus captures oriented periodic patterns that are aligned with neither the horizontal nor the vertical axis. This is why translation invariant scattering operators in $\mathbb{R}^2$ described in Chapter 2 are not computed as a separable product of one dimensional scattering along horizontal and vertical variables. Instead, two dimensional scattering iterates over a two dimensional wavelet transform that captures covariation of horizontal and vertical variable throughs oriented wavelets.

4.2.2 Separable Rigid-Motion Invariance

Two dimensional translation separable invariants built in section 4.2.1 are in fact invariant to a much larger group $\mathbb{R}^{\mathbb{R}} \times \mathbb{R}$ than intended because they treat both variables of the

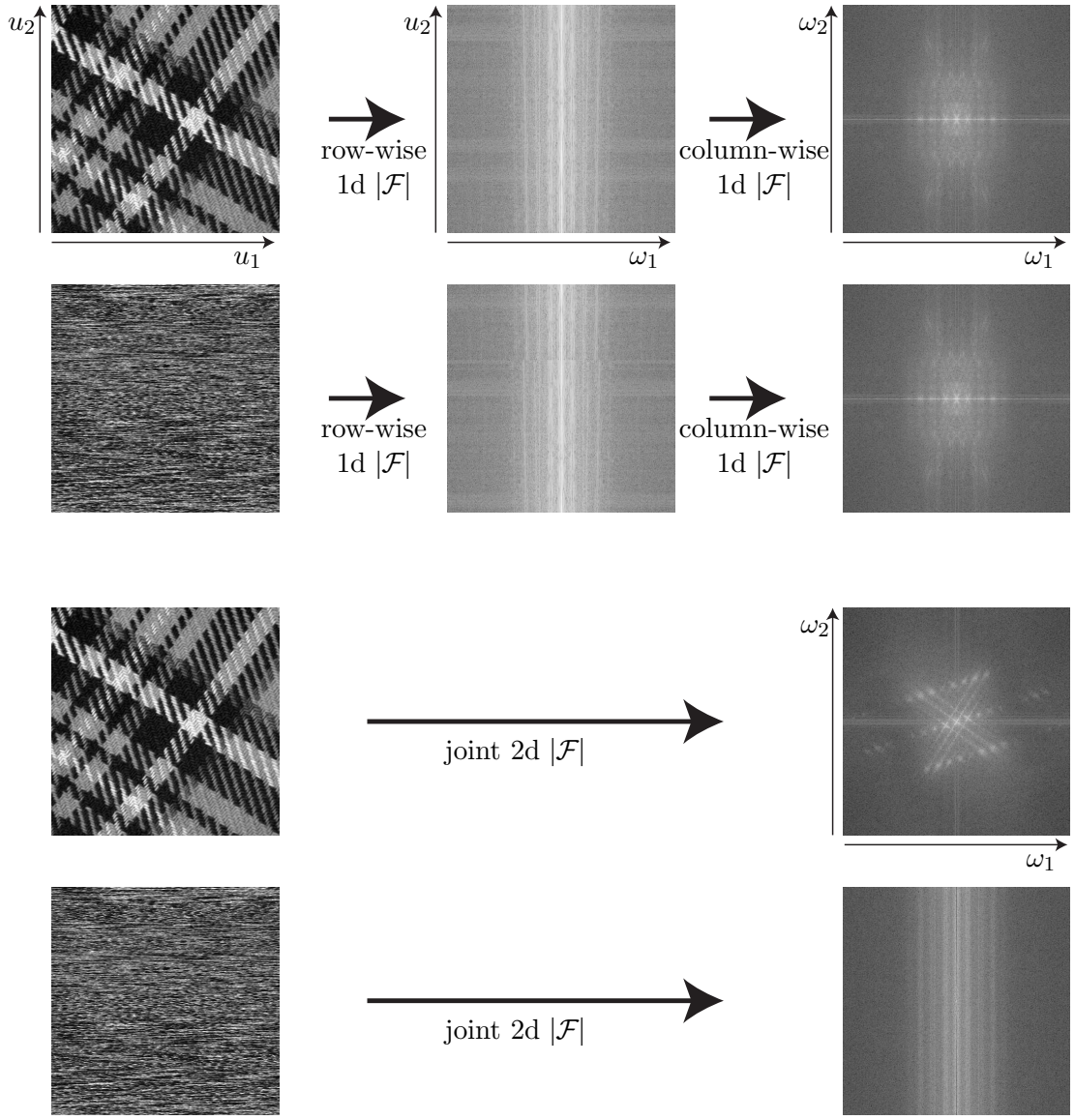


Figure 4.2: A separable two-dimensional translation invariant (top) can be obtained by cascading a one-dimensional translation invariant along rows and along columns. In this illustration, we have chosen one-dimensional Fourier modulus. Such a separable invariant is invariant to independent translation of each rows and global translations of columns, that is to the group $\mathbb{R}^{\mathbb{R}} \times \mathbb{R}$. On the contrary, a two-dimensional Fourier modulus takes into account the group structure of the two-dimensional translation groups and is not invariant to $\mathbb{R}^{\mathbb{R}} \times \mathbb{R}$ but only to $\mathbb{R}^2$. This illustrates the fact that separable invariant may be too loose invariant in the sense that they are invariant to a much larger group than the one intended.

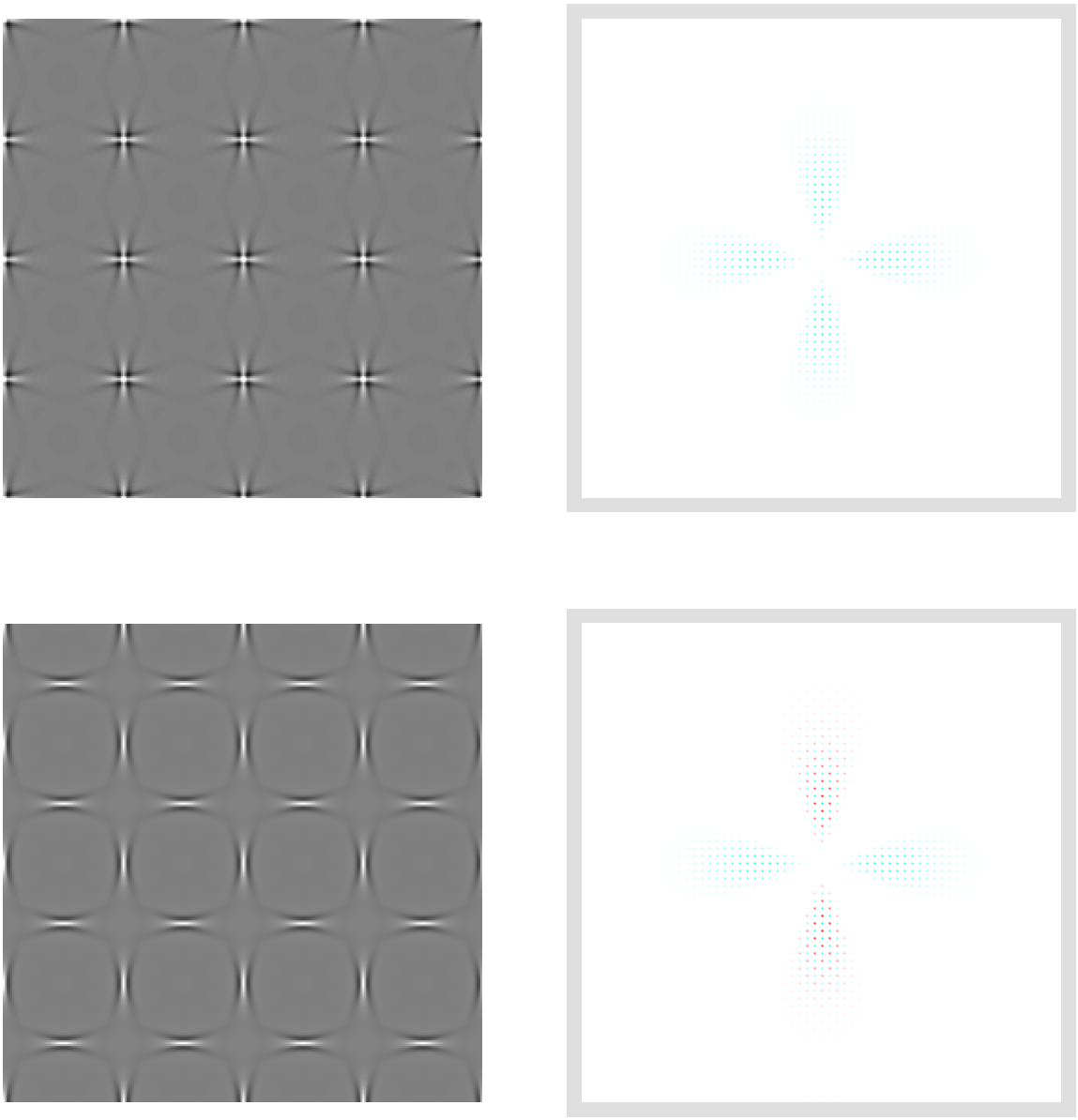


Figure 4.3: Two synthesized texture images of grid of horizontal and vertical Cauchy wavelets spaced out by 1 (left) , and their Fourier transform (right). In the top image, both grids are aligned, while in the bottom image, the grid of vertical wavelets is offsetted by $(1/2, 1/2)^T$. The Fourier transform are identical, up to non-global phase. Therefore, a Fourier modulus will not be able to distinguish these two texture, even though they are not related to a global translation.

translation group as if they were independant. The same phenomenon appears for separable invariants along translations and rotations, although it is more subtle because rotations affect translations. Indeed, let us consider both synthesized texture images x and y of figure 4.3. They both consist in a grid of horizontal ψ_0 and vertical $\psi_{\pi/2}$ wavelets and are defined as

$$x = \text{real}\left(\sum_{v\in\mathbb{Z}^2}\left(\mathcal{L}_v\psi_0 + \mathcal{L}_v\psi_{\pi/2}\right)\right) \tag{4.7}$$

$$y = \text{real}\left(\sum_{v\in\mathbb{Z}^2}\mathcal{L}_v\psi_0 + \sum_{v\in\mathbb{Z}^2+(1/2,1/2)^T}\mathcal{L}_v\psi_{\pi/2}\right). \tag{4.8}$$

The only difference is that in image y, the grid of vertical wavelets $\psi_{\pi/2}$ has an offset of $(1/2,1/2)^T$ with respect to the horizontal wavelets ψ_0. Let us suppose that the wavelet ψ is sufficiently localized in angle so that, among the wavelet family, any wavelet $\psi_{\theta,j}$ has to be orthogonal to either both ψ_0 and ψ_π or both $\psi_{\pi/2}$ and $\psi_{3\pi/2}$

$$\forall\theta, j \in [0,2\pi)\times\mathbb{Z} \quad \begin{pmatrix}\psi_0 \star \psi_{\theta,j} = 0\\ \text{and}\\ \psi_\pi \star \psi_{\theta,j} = 0\end{pmatrix} \text{ or } \begin{pmatrix}\psi_{\pi/2} \star \psi_{\theta,j} = 0\\ \text{and}\\ \psi_{3\pi/2} \star \psi_{\theta,j} = 0\end{pmatrix}. \tag{4.9}$$

This can be achieved for example with Cauchy directional wavelets [AMV99]. Let $\alpha < \beta$ be two directions and denote $\omega_\alpha = (\cos\alpha, \sin\alpha)^T$. A Cauchy directional wavelet is defined by its Fourier transform as

$$\mathcal{F}\psi(\omega) = \mathbb{1}_{\alpha\le\arg\omega\le\beta}(\omega.\omega_{\alpha+\pi/2})^l(\omega.\omega_{\beta+\pi/2})^l e^{-\omega.\omega_{(\alpha+\beta)/2}} \tag{4.10}$$

The first factor $\mathbb{1}_{\alpha\le\arg\omega\le\beta}$ enforces that the support of the Fourier transform of the wavelet is in the cone of direction $\alpha \le \arg\omega \le \beta$. The next two factors $(\omega.\omega_{\alpha+\pi/2})^l(\omega.\omega_{\beta+\pi/2})^l$ of (4.10) guarantee that the Fourier transform is smooth at the boundaries of that cone, while the last factor $e^{-\omega.\omega_{(\alpha+\beta)/2}}$ localizes the wavelet in scale. Cauchy wavelets are exactly analytical as soon as $\beta - \alpha < \pi$. As Morlet or Gabor wavelets, Cauchy wavelets have an hermitian symmetry and therefore we can limit the analysis of real signal to wavelets whose angle is between $[0,\pi)$. To construct a frame with C orientations between $[0,\pi)$, we use a wavelet ψ with $\alpha = -\pi/C$ and $\beta = \pi/C$. Therefore, the support of two wavelets of successive orientations will overlap, which will provides a reasonable Littlewood-Paley sum (2.23). Nevertheless, one wavelet of such a frame is strictly orthogonal to all other wavelets excepts those whore are in its immediate angular neighborhodd. In particular,

Property 3. *Cauchy wavelets defined as (4.10) with $\alpha = -\pi/C$ and $\beta = \pi/C$ exactly verifies (4.9) as soon as $C \ge 6$.*

Figure 4.4: An oriented Cauchy wavelet (left), and its Fourier transform (right), whose support is included in the cone of direction $\theta \in [-\pi/8, \pi/8]$.

Proof. Indeed, for $C \geq 6$, $\psi_{\pi/2}$ is at least the fourth angular wavelet, ψ_0 being the first. Thus, at least two orientations separate ψ_0 and ψ_π from $\psi_{\pi/2}$ and $\psi_{3\pi/2}$. Therefore, since Cauchy wavelets defined with $\alpha = -\pi/C$ and $\beta = \pi/C$ are orthogonal to all other Cauchy wavelets but their immediate angular neighbors, they must be orthogonal to either both ψ_0 and ψ_π or both $\psi_{\pi/2}$ and $\psi_{3\pi/2}$. □

Figure 4.5 shows all the wavelets of such a frame with $C = 8$, along with their Fourier transform and their Littlewood Paley sum (2.23).

Let us suppose that (4.9) is verified for some family of wavelet and that the two grids (4.7-4.8) are defined with theses wavelets, as in Figure 4.3. Now let us apply a wavelet transform operator defined with the same wavelets. The following property relates the wavelet transforms of such images x and y.

Property 4. *If wavelets $\{\psi_{\theta,j}\}_{j,\theta}$ have Hermitian symmetry $\psi(-u) = \psi^*(u)$ and verifies (4.9). Let x and y be defined as (4.7-4.8), then each signal of the wavelet transforms of x and y are either equal or translated:*

$$\forall \theta, j \in [0, 2\pi) \times \mathbb{Z}, \quad x \star \psi_{\theta,j} = y \star \psi_{\theta,j} \quad or \quad x \star \psi_{\theta,j} = \mathcal{L}_{-(1/2,1/2)^T} y \star \psi_{\theta,j}. \tag{4.11}$$

This property is illustrated with Cauchy wavelets in Figure 4.3.

Proof. Indeed, since the Cauchy wavelets have Hermitian symmetry we have

$$\operatorname{real} \psi_0 = 1/2(\psi_0 + \psi_\pi) \tag{4.12}$$

$$\operatorname{real} \psi_{\pi/2} = 1/2(\psi_{\pi/2} + \psi_{3\pi/2}). \tag{4.13}$$

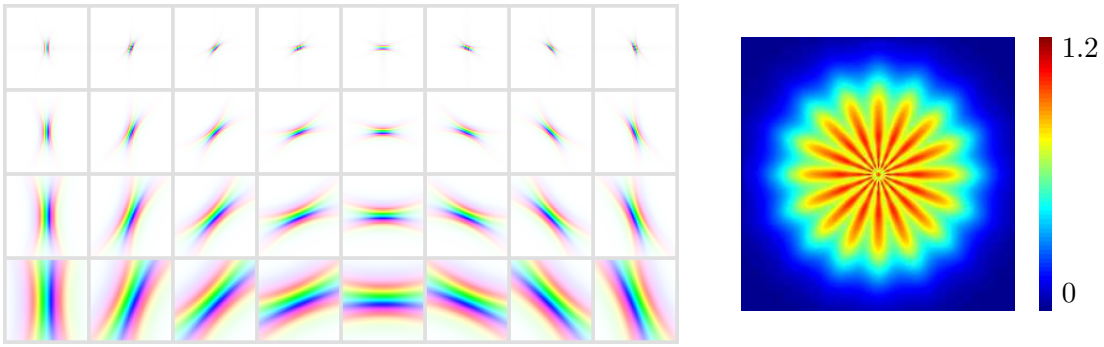


Figure 4.5: A family of oriented Cauchy wavelets, and their associated Littlewood Paley function.

Therefore, if, for example, $\psi_0 \star \psi_{\theta,j} = \psi_\pi \star \psi_{\theta,j} = 0$, then

$$x \star \psi_{\theta,j} = 1/2 \sum_{v \in \mathbb{Z}^2} (\mathcal{L}_v \psi_{\pi/2} + \mathcal{L}_v \psi_{3\pi/2}) \star \psi_{\theta_j} \tag{4.14}$$

$$= \mathcal{L}_{-(1/2,1/2)^T} 1/2 \sum_{v \in \mathbb{Z}^2 + (1/2,1/2)^T} (\mathcal{L}_v \psi_{\pi/2} + \mathcal{L}_v \psi_{3\pi/2}) \star \psi_{\theta_j} \tag{4.15}$$

$$= \mathcal{L}_{-(1/2,1/2)^T} y \star \psi_{\theta,j}. \tag{4.16}$$

and if $\psi_{\pi/2} \star \psi_{\theta,j} = \psi_{3\pi/2} \star \psi_{\theta,j} = 0$ then $x \star \psi_{\theta,j} = y \star \psi_{\theta,j}$. □

Property 4 shows that it is possible to construct two signals that are not related by a translation but whose wavelet transform are related to different translations for different wavelet indices. Iterating over two dimensional wavelet modulus operators will propagate property 4 to $U_m x$ and $U_m y$ for all $m \geq 1$. Their translation scattering vector Sx and Sy, obtained by averaging Ux and Uy along spatial position, will thus be equal. Therefore, their rigid-motion separable scattering transform, which is obtained by applying a scattering transform on the rotation orbit of the translation scattering will also be equal. This toy example shows that separable invariants on the rigid-motion group may be too lose, because they do not capture the local covariation of different orientations.

4.3 Multiresolution Analysis on the Rigid Motion Group

Section 4.2 has shown that separable invariants build invariance to a large group of transformation by applying successive invariants to different subgroups that generate the large group. This is a reasonnable and simple approach that breaks the problem of building invariance to a large group into smaller problems of building invariance to small groups. Yet, by considering the subgroups as independent, separable invariants discard the information

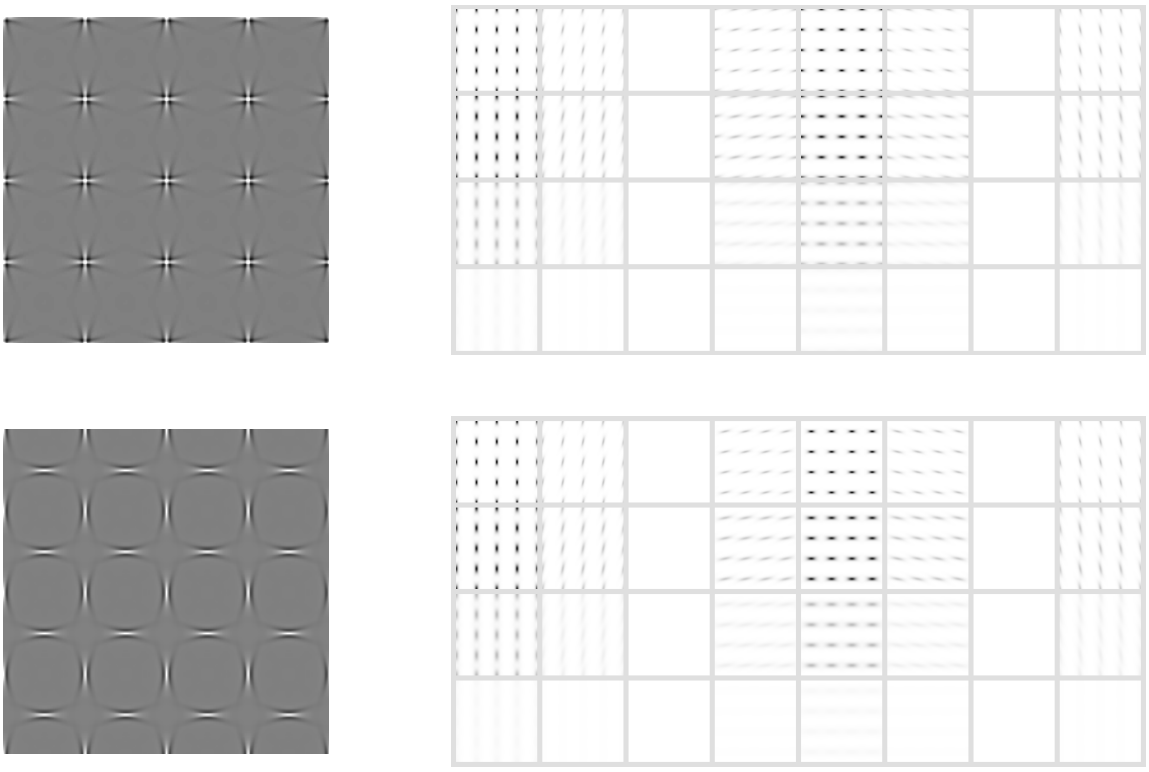


Figure 4.6: Two images of grids of horizontal and vertical Cauchy wavelets (left), and their wavelet modulus coefficients (right). In the top image, both grids are aligned, while in the bottom image, the grid corresponding to the vertical wavelet is offsetted by $(1/2, 1/2)^T$. Cauchy wavelets separate the horizontal and vertical grids into independent channels. The wavelet coefficients $|x \star \psi_{\theta,j}|$ of both images are strictly equal, but they are translated by $(0,0)^T$ for $\theta \in [0, \pi/4) \cup [3\pi/4, \pi)$, and by $(1/2, 1/2)^T$ for $\theta \in [\pi/4, 3\pi/4)$. The translation scattering builds translation invariance by processing all these channels independently and therefore will be equal for both images. The separable rigid-motion scattering, which is build by appending a scattering along the orientation on top of the translation scattering, will thus also be equal for both images.

contained in the covariations of different variables of the larger group. Therefore, separable operators are invariant to a wider class of transformation than what was originally intended and may not be able to distinguish two classes of signals that are perceptually distinct.

Instead of breaking the large group invariance problem into smaller group invariance subproblems, a joint invariant takes into account the group law of the larger group and adapts signal processing tools to the group law. Most two dimensional image processing tools, such as Fourier or wavelet transforms, are generalization of one dimensional signal processing tools and take into account the two dimensional translation group structure.

In the same spirit, several authors [CS06, DB07, DF11, BDGR12] have proposed to generalize some signal processing tools to the rigid-motion group consisting of two dimensional translations and rotations. Indeed a large body of works including [RP99] suggest that in the human visual brain, the first area called V1 is responsible for extracting orientation information at each position in the visual field. Therefore V1 would lift the two dimensional retina image into a three dimensional signal indexed by position and orientation, i.e. an element of the rigid-motion group. [CS06] transforms two dimensional curves into three dimensional curves augmented by the orientation of their tangent vector and proposes to process the lifted curves according to the group law of rigid-motions to perform completion of occluded curves. [DB07, DF11] generalizes the approach to images by considering a single-scale wavelet transform of an image as a three dimensional signal indexed by position and orientation, i.e. and element of the rigid-motion group. Given an image x, [DB07, DF11] computes its single-scale wavelet transform $\mathcal{W}x$, performs a diffusion according to the rigid-motion group on the wavelet transform $\mathcal{D}\mathcal{W}x$, and inverses it which results in a new image $\mathcal{W}^{-1}\mathcal{D}\mathcal{W}x$ where the contours have been enhanced. This has medical applications such as vessel or fiber tissue imaging [DF11].

Similarly, but in the different context of building invariant representations, this section introduces a joint scattering operators. Joint scattering applies a first wavelet modulus operators that plays a role similar to the lifting in [CS06, DB07, DF11, BDGR12, RP99] of transforming a two dimensional image into higher dimensional signals of a larger group. Then, a wavelet transform is applied to the higher dimensional signals according to the group law of the larger group. The rest of this section introduces the necessary tools to build that wavelet transform. Section 4.3.1 introduces rigid-motion convolutions and fast implementations for separable wavelets. Section 4.3.2 builds a separable rigid-motion wavelet basis, the corresponding wavelet transform operator and section 4.3.3 discusses fast implementations.

4.3.1 Rigid-Motion Convolutions

Group Convolutions

For any group G endowed with a measure $\tilde{\mu}$, one can define the set of squared integrable functions relatively to the measure $\tilde{\mu}$ as

$$\mathbf{L}^2(G) = \{\tilde{x} : G \to \mathbb{C} \;\text{ s.t}\; \int_{g\in G} |\tilde{x}(g)|^2 d\tilde{\mu}(g)\}. \tag{4.17}$$

The left translation of a function $x \in \mathbf{L}^2(G)$ by $h \in G$ is

$$\mathcal{L}_h \tilde{x}(g) = \tilde{x}(h^{-1}g). \tag{4.18}$$

$\mathcal{L}_h$ is an operator from $\mathbf{L}^2(G) \to \mathbf{L}^2(G)$. Given two functions $\tilde{x}, \tilde{y} \in \mathbf{L}^2(G)$ their convolution is

$$\tilde{x} \tilde{\star} \tilde{y}(g) = \int_{h\in G} \tilde{x}(h)\tilde{y}(h^{-1}g) d\tilde{\mu}(h). \tag{4.19}$$

Let us make the assumption that $\tilde{\mu}$ is a left invariant haar measure that is $\tilde{\mu}(h^{-1}S) = \tilde{\mu}(S)$ for all $h \in G, S \subset G$. Then, the definition (4.19) of convolutions is invariant to left translation of any $h \in G$

$$(\mathcal{L}_h \tilde{x}) \star \tilde{y} = \mathcal{L}_h(\tilde{x} \star \tilde{y}). \tag{4.20}$$

Indeed,

$$\begin{aligned}
(\mathcal{L}_h \tilde{x}) \tilde{\star} \tilde{y}(g) &= \int_{h'\in G} \mathcal{L}_h \tilde{x}(h')\tilde{y}(h'^{-1}g) d\tilde{\mu}(h') && (4.21)\\
&= \int_{h'\in G} \tilde{x}(h^{-1}h')\tilde{y}(h'^{-1}g) d\tilde{\mu}(h') && (4.22)\\
&= \int_{h''\in G} \tilde{x}(h'')\tilde{y}(h''^{-1}h^{-1}g) d\tilde{\mu}(hh'') && (4.23)\\
&= \int_{h''\in G} \tilde{x}(h'')\tilde{y}(h''^{-1}h^{-1}g) d\tilde{\mu}(h) && (4.24)\\
&= \tilde{x} \tilde{\star} \tilde{y}(h^{-1}g) && (4.25)\\
&= \mathcal{L}_h(\tilde{x} \tilde{\star} \tilde{y})(g). && (4.26)
\end{aligned}$$

In particular, let us consider an affine subgroup of the form $\mathbb{R}^2 \rtimes G$ where G is a subgroup of $GL(\mathbb{R}^2)$ as introduced in Section 3.2.2. Inserting the affine law defined in equation (3.3) as $g'g = (v' + A'v,\ A'A)$ into the group convolution (4.19) yields, for any $\tilde{x}, \tilde{y} \in \mathbf{L}^2(\mathbb{R}^2 \rtimes G)$

$$\tilde{x} \tilde{\star} \tilde{y}(v, A) = \int_{v', A' \in \mathbb{R}^2 \rtimes G} \tilde{x}(v', A')\ \tilde{y}(A'^{-1}(v - v'), A'^{-1}A)\ d\tilde{\mu}(v', A'). \tag{4.27}$$

Fast Separable Group Convolutions

If there are N samples for $\mathbb{R}^2$ and C samples for G, computing a naive discretized version of (4.27) would involve, for each discretized position $(v, A) \in \mathbb{R}^2 \rtimes G$, a sum over $(v', A') \in \mathbb{R}^2 \rtimes G$. The total cost of a group convolutions would therefore be $O(N^2C^2)$ operations which may be prohibitive. This paragraph shows how to lower this cost for separable wavelets.

Let us assume that the measure $\tilde{\mu}$ and the wavelet $\tilde{y}$ are separable,

$$\tilde{\mu}(v, A) = \mu(v)\mathring{\mu}(A) \tag{4.28}$$

and

$$\tilde{y}(v, A) = y(v)\mathring{y}(A). \tag{4.29}$$

Then, the group convolution can be factorized into

$$\tilde{x}\tilde{\star}\tilde{y}(v, A) = \int_{A' \in G} \left(\int_{v' \in \mathbb{R}^2} \tilde{x}(v', A') y(A'^{-1}(v - v')) d\mu(v') \right) \mathring{y}(A'^{-1}A) d\mathring{\mu}(A'). \tag{4.30}$$

Assuming that the measure μ on $v \in \mathbb{R}^2$ is the usual Lebesgue measure $d\mu(v) = dv$, the inner term $\left(\int_{v' \in \mathbb{R}^2} \tilde{x}(v', A') y(A'^{-1}(v - v')) dv' \right)$ is a spatial convolution of $v \mapsto \tilde{x}(v, A')$ with the two dimensional warped wavelet $v \mapsto y(A'^{-1}v)$. This spatial convolution can be implemented either in Fourier domain with a cost $O(N \log N)$ or in spatial domain with a cost $O(NP)$ where P is an upper bound on the support of the warped wavelets $v \mapsto y(A'^{-1}v)$ for all $A' \in G$. The spatial convolution has to be computed for all $A' \in G$ a thus requires either $O(N \log NC)$ for a Fourier implementation or $O(NPC)$ for a spatial implementation. Then, a convolution along A must be applied at all spatial location $v \in \mathbb{R}^2$. A naive implementation for this final convolution thus requires $O(NC^2)$ operations. The total cost of computing a discretized version of (4.30) is thus $O(N \log NC + NC^2)$ for a Fourier implementation of the spatial convolution or $O(NPC + NC^2)$ for a spatial implementation of the spatial convolution, instead of $O(N^2C^2)$ for the naive implementation.

An even more advantageous situation is when the linear subgroup $G \subset GL(\mathbb{R}^2)$ has a particular structure for which there exists a fast convolution algorithm. That is the case for the rotation group $G = SO(2)$ which consists in rotation matrices r_θ for $\theta \in [0, 2\pi)$. The corresponding affine subgroup $\mathbb{R}^2 \rtimes G$ is the rigid motion group, which is denoted $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ and consists in all compositions of translations an rotations. The separable group convolution (4.30) becomes

$$\tilde{x}\tilde{\star}\tilde{y}(v, \theta) = \int_{\theta \in [0, 2\pi)} \left(\int_{v' \in \mathbb{R}^2} \tilde{x}(v', \theta') y(r_{-\theta'}(v - v')) dv' \right) \mathring{y}(\theta - \theta') d\theta'. \tag{4.31}$$

The inner term is a two dimensional convolution of $v \mapsto \tilde{x}(v, \theta')$ with the rotated wavelet $v \mapsto y(r_{-\theta'}v)$. The outer term is a periodic one dimensional convolution on $SO(2)$ of the inner term with the one dimensional 2π periodic wavelet $\mathring{y}$, as reviewed in Section

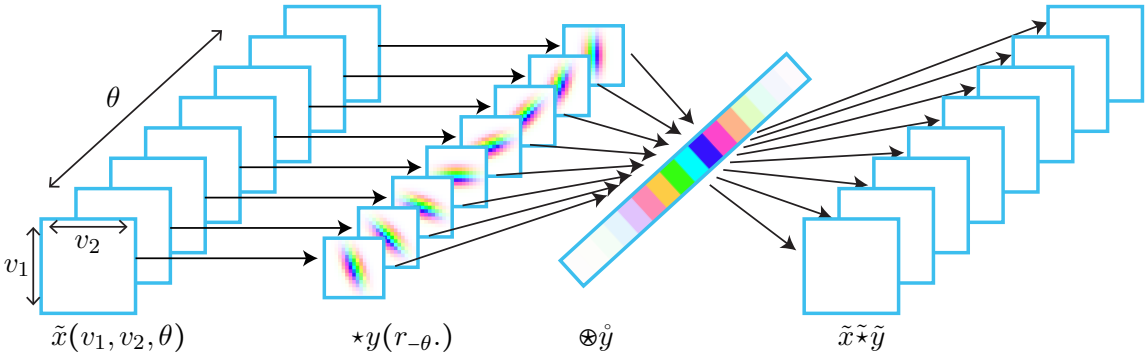


Figure 4.7: A rigid-motion convolution (4.31) with a separable wavelet $\tilde{y}(v, \theta) = y(v)\mathring{y}(\theta)$ in $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ can be factorized into two dimensional convolutions with rotated wavelets $y(r_{-\theta}v)$ followed by one dimensional convolutions with $y(\theta)$. The wavelets displayed are two dimensional complex elongated Morlet wavelet (2.41) for y and one dimensional complex Morlet wavelet for $\mathring{y}$.

3.4.2. Figure 4.7 illustrates this separable implementation of such rigid-motion convolutions. For periodic one dimensional convolutions, a Fourier domain implementation requires $O(C \log C)$ operations and a spatial domain implementation requires $O(CP')$ where P' would be the support of the wavelet $\mathring{y}$. Computing the outer term of (4.31) involves N such convolutions. Thus, the cost of all the convolutions along G becomes $O(NC \log C)$ or $O(NCP')$ for Fourier or spatial implementation of convolution along θ.

More generally, the separable convolution (4.30) is a succession of convolutions along $\mathbb{R}^2$ followed by convolutions along G. Therefore, one can independently choose an implementation of the convolutions along $\mathbb{R}^2$ and another implementation of the convolutions along G. Table 4.1 summarizes the total cost of computing an affine convolution, depending on the choice of both implementations.

The group of dilation $G = \mathbb{R}^{+*}$ also has a simple one dimensional structure. Yet, the dilation group is not periodic. Thus, one must take extra care at the boundaries when computing convolutions along dilations. Real world images are finite and therefore have a limited range of dilations available. This makes convolutions along dilation difficult to compute because boundary effects tends to affect a large part if not all of the support of the signal. Therefore, in most applications we do not compute convolutions along the dilation group.

4.3.2 Rigid-Motion Wavelet Transform

Section 4.3.1 has defined the group convolution on affine subgroups and has introduced fast implementations for separable wavelets. This section introduces a family of separable

G-conv VS $\mathbb{R}^2$-conv	Naive	Fourier	Spatial
Naive	N^2C^2	$N \log NC + NC^2$	$NPC + NC^2$
Fourier	$N^2C + NC \log C$	$N \log NC + NC \log C$	$NPC + NC \log C$
Spatial	$N^2C + NCP'$	$N \log NC + NCP'$	$NPC + NCP'$

Table 4.1: Cost of separable implementation of convolution (4.30) on an affine subgroup $\mathbb{R}^2 \rtimes G$ with $G \subset GL(\mathbb{R}^2)$. A separable implementation is implemented by convolutions along $\mathbb{R}^2$, followed by convolutions along G. The columns of this table corresponds to the possible choice of implementations of the convolutions along $\mathbb{R}^2$, and the rows corresponds to the convolutions along G. N is the number of samples for $\mathbb{R}^2$, C is the number of samples for G, P is an upper bound of the support's size for warped spatial wavelets $v \mapsto y(A^{-1}v)$, P' is the size of the support's size for wavelet $\mathring{y}$. For the second convolution, all implementations may not be available depending on the group structure of the linear subgroup G. If G is the rotation group $G = SO(2)$, all implementations are available.

wavelets on affine subgroups, with fast implementations of the associated wavelet transform.

A wavelet family on a product space can be obtained as a separable product of wavelet families on smaller space. This approach can be used for example to compute a two dimensional $\{\phi_J, \psi_{l,j}\}_{1 \leq l \leq 3,\ j<J}$ wavelet family from a one dimensional wavelet family $\{\phi_J, \psi_j\}_{j<J}$ with

$$\begin{aligned}
\phi_J(u_1, u_2) &= \phi_J(u_1)\phi_J(u_2) && (4.32)\\
\psi_{1,j}(u_1, u_2) &= \psi_j(u_1)\psi_j(u_2) && (4.33)\\
\psi_{2,j}(u_1, u_2) &= \psi_j(u_1)\phi_j(u_2) && (4.34)\\
\psi_{3,j}(u_1, u_2) &= \phi_j(u_1)\psi_j(u_2). && (4.35)
\end{aligned}$$

This construction imposes that both factors of the product wavelet have roughly the same scale 2^j. This makes sense for images since both variables u_1 and u_2 have a similar role. For other two dimensional signal where u_1 and u_2 would be physically different quantities, one could imagine to chose the scales along u_1 and u_2 independently. Another separable family would thus be

$$\begin{aligned}
\phi_J(u_1, u_2) &= \phi_J(u_1)\phi_J(u_2) && (4.36)\\
\psi_{j_1,j_2}(u_1, u_2) &= \psi_{j_1}(u_1)\psi_{j_2}(u_2) && (4.37)\\
\psi_{J,j_2}(u_1, u_2) &= \phi_J(u_1)\psi_{j_2}(u_2) && (4.38)\\
\psi_{j_1,J}(u_1, u_2) &= \psi_{j_1}(u_1)\phi_J(u_2). && (4.39)
\end{aligned}$$

This family makes more sense if u_1 and u_2 are physically different quantities since it does not assume anything on the joint distribution of scales of both variables. Figure 4.8 shows the support of both families of wavelets (4.32-4.35) and (4.36-4.39).

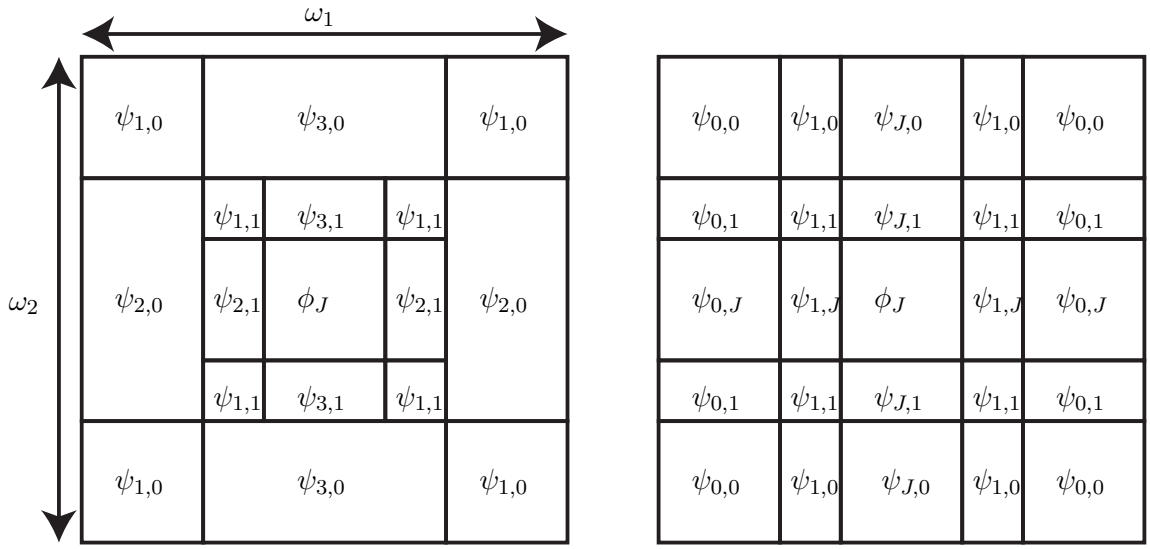


Figure 4.8: Left: the Fourier support of two dimensional separable wavelets with same scale on both variables defined (4.32-4.35). Right: the Fourier support of wavelets with independent scales for both variables as defined in (4.36-4.39).

Similarly, we define a separable wavelet family $\{\tilde{\phi}, \tilde{\psi}_\lambda\}_\lambda$ on affine subgroups of the form $\mathbb{R}^2 \rtimes G$ where $G \subset GL(\mathbb{R}^2)$ by multiplying a first wavelet family $\{\phi_J, \psi_{l,j}\}_{0 \le l < L,\ j < J}$ and a second wavelet family $\{\mathring{\phi}, \mathring{\psi}_k\}_{k<K}$ on G. Since scales on $\mathbb{R}^2$ and on G have a priori no reason to relate, we chose to define our wavelets with independent scales

$$\tilde{\phi}_{J,K}(u, A) = \phi_J(u_1)\mathring{\phi}_K(A) \tag{4.40}$$

$$\tilde{\psi}_{l,j,k}(u, A) = \psi_{l,j}(u)\mathring{\psi}_k(A) \quad \text{for } j < J,\ k < K \tag{4.41}$$

$$\tilde{\psi}_{0,J,k}(u, A) = \phi_J(u)\mathring{\psi}_k(A) \quad \text{for } k < K \tag{4.42}$$

$$\tilde{\psi}_{l,j,K}(u, A) = \psi_{l,j}(u)\mathring{\phi}_K(A) \quad \text{for } j < J. \tag{4.43}$$

In our application we focus on rigid motion group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$. For this group, we can define $\{\phi_K, \psi_k\}_k$ as any one dimensional periodic wavelet family. Section 3.4.2 builds such a wavelet family and reviews a discrete Fourier transform on $SO(2)$. The Littlewood Paley function of the wavelet family $\{\phi_K, \psi_k\}_k$ has been defined as $|\mathcal{F}\mathring{\phi}_K(m)|^2 + \sum_{k<K} |\mathcal{F}\mathring{\psi}_k(m)|^2$. The following theorem shows that if both wavelet families used to build the separable rigid-motion wavelet family are frame, then the separable family is also a frame.

Theorem 2. *If there exists ϵ and $\mathring{\epsilon}$ such that*

$$\forall \omega \in \mathbb{R}^2,\ 1 - \epsilon \le |\mathcal{F}\phi_J(\omega)|^2 + \sum_{0 \le l < L,\ j < J} |\mathcal{F}\psi_{l,j}(\omega)|^2 \le 1 \tag{4.44}$$

and

$$\forall m \in \mathbb{Z},\ 1 - \mathring{\epsilon} \le |\mathcal{F}\mathring{\phi}_K(m)|^2 + \sum_{k<K} |\mathcal{F}\psi_k(m)|^2 \le 1 \tag{4.45}$$

then the wavelet family $\{\tilde{\phi}_{J,K}, \tilde{\psi}_{l,j,k}\}_{l,j,k}$ defined in (4.40-4.43) is a $\tilde{\epsilon}$ frame of $SE(2) = \mathbb{R}^2 \rtimes SO(2)$, i.e. for all $\tilde{x} \in \mathbf{L}^2(\mathbb{R}^2 \rtimes G)$,

$$(1 - \tilde{\epsilon})\|\tilde{x}\|^2 \le \|\tilde{x} \tilde{\star} \tilde{\phi}_{J,K}\|^2 + \sum_{l,j,k} \|\tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}\|^2 \le \|\tilde{x}\|^2 \tag{4.46}$$

where

$$1 - \tilde{\epsilon} = (1 - \epsilon)(1 - \mathring{\epsilon}) \tag{4.47}$$

Proof. The proof factorizes the wavelet operator on $SE(2)$ into a product of operators for which we have an upper and lower bound. Let us define the set of indices $\Lambda = \{J, (l, j)\}_{l,\ j<J}$, $\mathring{\Lambda} = \{K, k\}_{k<K}$ and $\tilde{\Lambda} = \{(J, K), (l, j, k)\}_{l,j,k}$ that respectively correspond to the families of functions $\{\phi_J, \psi_{l,j}\}_{l,j}$, $\{\mathring{\phi}_K, \mathring{\psi}_k\}_k$ and $\{\tilde{\phi}_{J,K}, \tilde{\psi}_{l,j,k}\}_{l,j,k}$. For compactness, we respectively denote these families $\{\psi_\lambda\}_{\lambda \in \Lambda}$, $\{\mathring{\psi}_{\mathring{\lambda}}\}_{\mathring{\lambda} \in \mathring{\Lambda}}$ and $\{\tilde{\psi}_{\tilde{\lambda}}\}_{\tilde{\lambda} \in \tilde{\Lambda}}$ with the abuse of notation $\psi_\lambda = \phi_J$ if $\lambda = J$. By definition of separable wavelets (4.40-4.43) we have

$$\tilde{\Lambda} = \Lambda \times \mathring{\Lambda}. \tag{4.48}$$

Let use define the following operators:

$$\begin{aligned} \mathcal{R}_1 : \mathbf{L}^2(SE(2)) &\rightarrow \mathbf{L}^2(SE(2)) && (4.49)\\ \tilde{x} &\mapsto ((v,\theta) \mapsto \tilde{x}(r_\theta v, \theta)) && (4.50)\end{aligned}$$

$$\begin{aligned} \mathcal{W} : \mathbf{L}^2(SE(2)) &\rightarrow \mathbf{L}^2(SE(2)\times\Lambda) && (4.51)\\ \tilde{x} &\mapsto ((v,\theta,\lambda) \mapsto \tilde{x}(.,\theta) \star \psi_\lambda(v)\ ) && (4.52)\end{aligned}$$

$$\begin{aligned} \mathcal{R}_2 : \mathbf{L}^2(SE(2)\times\Lambda) &\rightarrow \mathbf{L}^2(SE(2)\times\Lambda) && (4.53)\\ \tilde{x} &\mapsto ((v,\theta,\lambda) \mapsto \tilde{x}(r_\theta v, \theta, \lambda)) && (4.54)\end{aligned}$$

$$\begin{aligned} \mathring{\mathcal{W}} : \mathbf{L}^2(SE(2)\times\Lambda) &\rightarrow \mathbf{L}^2(SE(2)\times\Lambda\times\mathring{\Lambda}) && (4.55)\\ \tilde{x} &\mapsto \left((v,\theta,\lambda,\mathring{\lambda}) \mapsto \tilde{x}(v,.,\lambda) \circledast \psi_{\mathring{\lambda}}(\theta)\ \right) && (4.56)\end{aligned}$$

and

$$\begin{aligned} \widetilde{\mathcal{W}} : \mathbf{L}^2(SE(2)) &\rightarrow \mathbf{L}^2(SE(2)\times\Lambda\times\mathring{\Lambda}) && (4.57)\\ \tilde{x} &\mapsto \left((v,\theta,\lambda,\mathring{\lambda}) \mapsto \tilde{x} \tilde{\star} \tilde{\psi}_{\lambda,\mathring{\lambda}}(v,\theta)\ \right). && (4.58)\end{aligned}$$

We now verify that

$$\widetilde{\mathcal{W}} = \mathring{\mathcal{W}} \mathcal{R}_2^{-1} W \mathcal{R}_1. \tag{4.59}$$

Indeed, the separable rigid-motion convolution (4.31) can be rewritten

$$\begin{aligned} \tilde{x} \tilde{\star} \tilde{\psi}_{\lambda,\mathring{\lambda}}(v,\theta) &= \int_{\theta'\in[0,2\pi)} \left(\int_{v'\in\mathbb{R}^2} \tilde{x}(v',\theta') \psi_\lambda(r_{-\theta'}(v-v')) dv' \right) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta') d\theta' && (4.60)\\ &= \int_{\theta'\in[0,2\pi)} \left(\int_{w\in\mathbb{R}^2} x(r_{\theta'} w, \theta') \psi_\lambda(r_{-\theta'} v - w) dw \right) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta') d\theta' && (4.61)\\ &= \int_{\theta'\in[0,2\pi)} \left(\int_{w\in\mathbb{R}^2} (\mathcal{R}_1 x)(w,\theta') \psi_\lambda(r_{-\theta'} v - w) dw \right) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta') d\theta' && (4.62)\\ &= \int_{\theta'\in[0,2\pi)} \left(\mathcal{W}\mathcal{R}_1 x\right)(r_{-\theta'} v, \theta', \lambda) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta') d\theta' && (4.63)\\ &= \int_{\theta'\in[0,2\pi)} \left(\mathcal{R}_2^{-1}\mathcal{W}\mathcal{R}_1 x\right)(v,\theta',\lambda) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta') d\theta' && (4.64)\\ &= \mathring{\mathcal{W}} \mathcal{R}_2^{-1} W \mathcal{R}_1 x(v,\theta,\lambda,\mathring{\lambda}). && (4.65)\end{aligned}$$

Since rotation preserves the norm,

$$\forall \tilde{x} \in \mathbf{L}^2(SE(2)), \|\mathcal{R}_1 \tilde{x}\|^2 = \|\tilde{x}\|^2\ , \tag{4.66}$$

$$\forall \tilde{x} \in \mathbf{L}^2(SE(2)\times\Lambda), \|\mathcal{R}_2^{-1} \tilde{x}\|^2 = \|\tilde{x}\|^2. \tag{4.67}$$

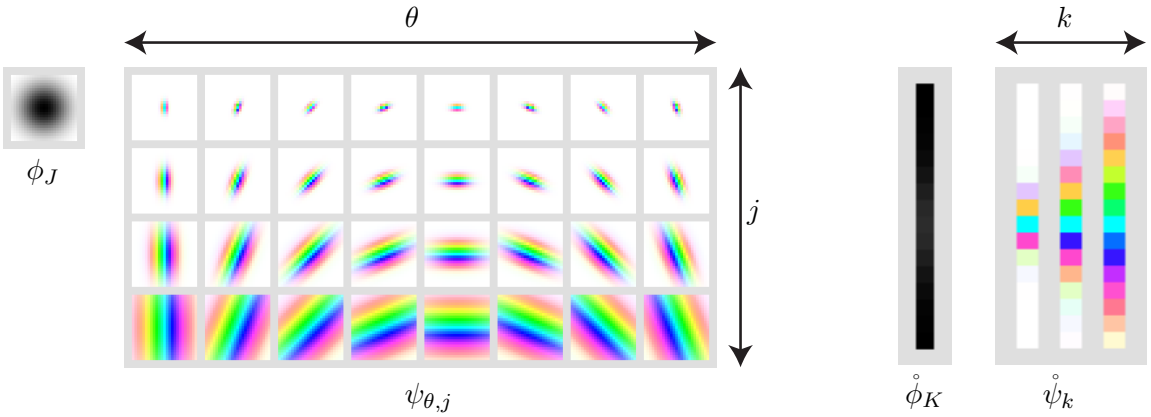


Figure 4.9: The two dimensional wavelet family $\{\phi_J, \psi_{\theta,j}\}_{\theta,j}$ and one dimensional wavelet family $\{\mathring{\phi}_K, \mathring{\psi}_k\}_k$ used to build the rigid-motion separable wavelets $\{\tilde{\phi}, \tilde{\psi}_{\theta,j,k}\}_{\theta,j,k}$ (4.40-4.43) in all classification experiments.

and (4.44-4.45) implies that

$$\forall \tilde{x} \in \mathbf{L}^2(SE(2)),\ (1-\epsilon)\|\tilde{x}\|^2 \leq \|\mathcal{W}\tilde{x}\|^2 \leq \|\tilde{x}\|^2. \tag{4.68}$$

$$\forall \tilde{x} \in \mathbf{L}^2(SE(2)\times\Lambda),\ (1-\mathring{\epsilon})\|\tilde{x}\|^2 \leq \|\mathring{\mathcal{W}}\tilde{x}\|^2 \leq \|\tilde{x}\|^2, \tag{4.69}$$

Cascading (4.66-4.69) into (4.59) yields

$$\forall \tilde{x} \in \mathbf{L}^2(SE(2)),\ (1-\tilde{\epsilon})\|\tilde{x}\|^2 \leq \|\widetilde{\mathcal{W}}\tilde{x}\|^2 \leq \|\tilde{x}\|^2 \tag{4.70}$$

which is equivalent to (6.13). □

In all classification applications we use two dimensional elongated Morlet wavelets (2.41) for $\{\phi_J, \psi_{l,j}\}_{l,j}$ and one dimensional Morlet wavelet for $\{\mathring{\phi}_K, \mathring{\psi}_k\}_{k<K}$. Since two dimensional Morlet wavelets are oriented, they can be indexed $\psi_{\theta_2,j}$ with their orientation θ_2 instead of the generic index l. The rigid-motion convolutions (4.60) becomes

$$\tilde{x}\tilde{\star}\tilde{\psi}_{\theta_2,j,l}(v,\theta) = \int_{\theta'\in[0,2\pi)} \left(\int_{v'\in\mathbb{R}^2} \tilde{x}(v',\theta')\psi_{\theta_2+\theta',j}(v-v')dv' \right) \mathring{\psi}_{\mathring{\lambda}}(\theta-\theta')d\theta'. \tag{4.71}$$

Thus, there are no additional wavelets to compute, since the warped wavelets $v \mapsto y(A'^{-1}v)$ are actually oriented wavelets $\psi_{\theta_2+\theta',j}$ that already belong to the family $\{\phi_J, \psi_{\theta,j}\}_{\theta,j}$. A convolution with such a wavelet is illustrated in Figure 4.7. Figure 4.9 illustrates the two wavelet families used to build the rigid-motion separable Morlet wavelet family.

4.3.3 Fast Rigid-Motion Wavelet Transform

Section 4.3.1 has introduced rigid-motion convolutions and fast implementations for separable wavelets. Section 4.3.2 has described the construction of separable rigid-motion wavelets $\{\tilde{\phi}_{J,K}, \tilde{\psi}_{l,j,k}\}_{l,j,k}$ and the associated wavelet transform operator $\widetilde{\mathcal{W}}$. This section introduces fast algorithms that implement the rigid-motion wavelet transform $\widetilde{\mathcal{W}}$ leveraging the fast implementation of separable rigid-motion convolution and the multi scale structure of wavelets.

The discrete rigid-motion wavelet transform takes as input a discrete signal $\tilde{x}(n,\theta)$ for $n \in \mathbb{Z}^2, \theta \in 2\pi\mathbb{Z}/(2C\mathbb{Z})$ where C is the number of orientations in $[0,\pi)$. All the following definitions and algorithms are also valid for the practical case of finite images where $n \in \mathbb{Z}^2/(N_1\mathbb{Z}\times N_2\mathbb{Z})$ where (N_1, N_2) are the width and height of the image and $N = N_1N_2$ is the number of pixels of the image. The discrete convolution of $\tilde{x}$ with a separable rigid-motion wavelet $\tilde{\psi}_{l,j,k}(n,\theta) = \psi_{l,j}(n)\mathring{\psi}_k(\theta)$ is

$$\tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}(n,\theta) = \sum_{\theta \in 2\pi\mathbb{Z}/(2C\mathbb{Z})} \left(\sum_{n' \in \mathbb{Z}^2} \tilde{x}(n',\theta') \psi_{l,j}(r_{-\theta'}(n-n')) \right) \mathring{\psi}_k(\theta - \theta'). \tag{4.72}$$

Since $\tilde{\psi}_{l,j,k}$ has a scale 2^j with respect to n and 2^k with respect to θ, we actually want to compute a downsampled version of the convolutions $\tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}$. More precisely, we want to compute the following quantities

$$\begin{aligned}
C_{J,K}\tilde{x}(n,\theta) &= \tilde{x} \tilde{\star} \tilde{\phi}_{J,K}(2^J n, 2^K \theta) && (4.73)\\
D_{J,k}\tilde{x}(n,\theta) &= \tilde{x} \tilde{\star} \tilde{\psi}_{0,J,k}(2^J n, 2^k \theta) && (4.74)\\
E_{l,j,K}\tilde{x}(n,\theta) &= \tilde{x} \tilde{\star} \tilde{\psi}_{l,j,K}(2^j n, 2^K \theta) && (4.75)\\
F_{l,j,k}\tilde{x}(n,\theta) &= \tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}(2^j n, 2^k \theta). && (4.76)
\end{aligned}$$

Therefore, to compute the downsampled convolutions of $\tilde{x} \in \mathbf{L}^2(SE(2))$ with all the functions in the family $\{\tilde{\phi}_{J,K}, \tilde{\psi}_{l,j,k}\}_{l,j,k}$, we start be computing all possible downsampled convolutions of $\tilde{x}$ with ϕ_J and $\psi_{l,j}(r_\theta .)$ along the variable n.

$$\begin{aligned}
A_J\tilde{x}(n,\theta) &= \tilde{x}(.,\theta) \star \phi_J(2^J n) && (4.77)\\
B_{l,j}\tilde{x}(n,\theta) &= \tilde{x}(.,\theta) \star \psi_{l,j}(r_{-\theta}.)(2^J n). && (4.78)
\end{aligned}$$

Then, we compute downsampled convolutions of previous quantities with $\mathring{\phi}_K$ and $\mathring{\psi}_k$ along the variable θ.

$$\begin{aligned}
C_{J,K}\tilde{x}(n,\theta) &= A_J\tilde{x}(n,.) \circledast \mathring{\phi}_K(2^K\theta) && (4.79)\\
D_{J,k}\tilde{x}(n,\theta) &= A_J\tilde{x}(n,.) \circledast \mathring{\psi}_k(2^k\theta) && (4.80)\\
E_{l,j,K}\tilde{x}(n,\theta) &= B_{l,j}\tilde{x}(n,.) \circledast \mathring{\phi}_K(2^K\theta) && (4.81)\\
F_{l,j,k}\tilde{x}(n,\theta) &= B_{l,j}\tilde{x}(n,.) \circledast \mathring{\psi}_k(2^k\theta). && (4.82)
\end{aligned}$$

Indeed, from the definition of discrete rigid-motion convolution (4.72), cascading (4.77-4.78) with (4.79-4.82) is equivalent to definitions (4.73-4.76).

This algorithm cascades two wavelet transforms along two different variables n and θ. The convolution along θ happens after the downsampling of the variables n has been done, which dramatically reduces the number of such convolutions. Similarly to the separable rigid-motion convolution algorithm presented in Section 4.3.1, the two steps (4.77-4.78) and (4.79-4.82) can be implemented either with naive convolutions, with a Fourier wavelet transform or with a cascade filter bank wavelet transform. The details of Fourier or filter bank implementations of the wavelet transform have been presented in Section 2.3.3.

Each choice of implementation for the two steps yields a different implementation of the rigid-motion wavelet transform. As in 2.3.3, the Fourier implementation is often slower, but offers the possibility of oversampling the signal for a reasonable extra computational cost, which can be helpful for debugging or for applications that requires high resolutions. Here, we describe in details only the fastest implementation, where both wavelet transforms are implemented with a cascade filter bank algorithm. This cascade filter bank rigid-motion wavelet transform is illustrated in figure 4.10. To actually compute $A_J\tilde{x}$ and $B_{l,j}\tilde{x}$ we use a filter bank similar to the one used in Section 2.3.3. We assume that the Fourier transform of the window ϕ and each mother wavelet ψ_l can be written as a product

$$\mathcal{F}\phi(\omega) = \prod_{j<0} \mathcal{F}h(2^j\omega) \tag{4.83}$$

$$\mathcal{F}\psi_l(\omega) = \mathcal{F}g_l(\omega)\mathcal{F}\phi(\omega). \tag{4.84}$$

and that there exists similar filters $\mathring{h}, \mathring{g}$ respectively corresponding to $\mathring{\phi}, \mathring{\psi}$. We define the rotated filters $g_{l,\theta}(v) = g_l(r_{-\theta}v)$. Then, we initialize $A_0\tilde{x} = \tilde{x}$ and compute a first cascade of filtering and downsampling along n

$$A_{j+1}\tilde{x}(n,\theta) = A_j(.,\theta)\tilde{x} \star h(2n) \tag{4.85}$$

$$B_{l,j}\tilde{x}(n,\theta) = A_j(.,\theta)\tilde{x} \star g_{l,\theta}(n). \tag{4.86}$$

From this we initialize $C_{J,0}\tilde{x} = A_J\tilde{x}$ and $E_{l,j,0}\tilde{x} = B_{l,j}\tilde{x}$ and compute a second cascade of filtering and downsampling along θ

$$C_{J,k+1}\tilde{x}(n,\theta) = C_{J,k}\tilde{x}(n,.) \circledast \mathring{h}(2\theta) \tag{4.87}$$

$$D_{J,k}\tilde{x}(n,\theta) = C_{J,k}\tilde{x}(n,.) \circledast \mathring{g}(\theta) \tag{4.88}$$

$$E_{l,j,k+1}\tilde{x}(n,\theta) = E_{j,l,k}\tilde{x}(n,.) \circledast \mathring{h}(2\theta) \tag{4.89}$$

$$F_{l,j,k}\tilde{x}(n,\theta) = E_{j,l,k}\tilde{x}(n,.) \circledast \mathring{g}(\theta). \tag{4.90}$$

The first spatial cascade computes CL convolutions at each spatial resolution, which requires $O(CLNP)$ operations and $O(CLN)$ memory. The first cascade is a tree whose leaves are $C_{J,0}\tilde{x}$ and $E_{l,j,0}\tilde{x}$. Each leaf is then retransformed by a second cascade where convolutions are applied along the orientation variable θ. Convolutions along the orientations

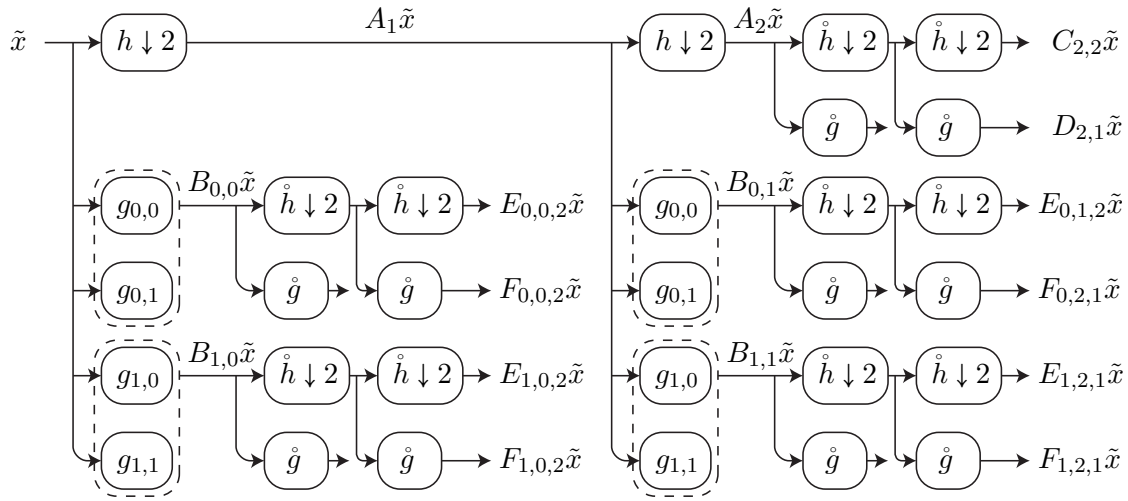


Figure 4.10: Filter bank implementation of the rigid-motion wavelet transform $\widetilde{\mathcal{W}}$ with $J = 2$ spatial scales, $2C = 2$ orientations, $L = 2$ spatial wavelets, $K = 2$ orientation scales . A first cascade computes spatial downsampling and filtering with h and $g_{l,\theta}$. The first cascade is a tree whose leaves are $A_J\tilde{x}$ and $B_{l,j}\tilde{x}$. Each leaf is retransformed with a second cascade of downsampling and filtering with $\mathring{h}$ and $\mathring{g}$ along the orientation variable. The leaves of the second cascade are $C_{J,K}\tilde{x}$, $D_{J,k}\tilde{x}$ (whose ancestor is $A_J\tilde{x}$) and $E_{l,j,K}\tilde{x}$, $F_{l,j,k}\tilde{x}$ (whose ancestors are the $B_{l,j}\tilde{x}$). These leaves constitute the output of the downsampled rigid-motion wavelet transform. They correspond to signals $\tilde{x} \tilde{\star} \tilde{\phi}_{J,K}\tilde{x}$, $\tilde{x} \tilde{\star} \tilde{\psi}_{J,k}$, $\tilde{x} \tilde{\star} \tilde{\psi}_{l,j,K}$, $\tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}$ appropriately downsampled along the spatial and the orientation variable.

are periodic and since the size of the filter $\mathring{h}$, $\mathring{g}$ is of the same order as $2C$, we use Fourier based convolutions. One such convolution requires $O(C \log C)$ operations. One cascade of filtering and downsampling along orientations requires $\sum_k C2^{-k} \log(C2^{-k}) = O(C \log C)$ time and $O(C)$ memory. There are $O(LN)$ such cascades so that the total cost for processing along orientation is $O(CLN \log C)$ operations and $O(CLN)$ memory. Thus, the total cost for the full rigid-motion wavelet transform $\widetilde{\mathcal{W}}$ is $O(CLN(P + \log C))$ operations and $O(CLN)$ memory where C is the number of orientations of the input signal, L is the number of spatial filters $\{h_l\}_l$, N is the size of the input image, P is the size of the spatial filters.

4.4 Joint Rigid-Motion Scattering

Previous section has defined rigid-motion wavelet transform, which is an operator from $\mathcal{L}^2(\mathbb{R}^2 \rtimes SO(2)) \to \mathcal{L}^2(\mathbb{R}^2 \rtimes SO(2) \times \tilde{\Lambda})$ which maps a three dimensional signal $\tilde{x}(u, \theta)$ to $\{\tilde{x} \tilde{\star} \tilde{\phi}_{J,L}, \ \tilde{x} \tilde{\star} \tilde{\psi}_{l,j,k}\}_{l,j,k}$. Images are not signal from $\mathcal{L}^2(\mathbb{R}^2 \rtimes SO(2))$ but are two dimensional

functions belonging to $\mathcal{L}^2(\mathbb{R}^2)$, thus we cannot apply to them a rigid-motion wavelet transform. This section shows how a first spatial wavelet transform maps an input x to $\mathcal{W}x$ which is covariant to rigid-motions. The orbits of $\mathcal{W}x$ are functions of $\mathcal{L}^2(\mathbb{R}^2 \rtimes SO(2))$, on which we can apply $\widetilde{\mathcal{W}}$. Iterating this yields the rigid-motion scattering, a representation of an image that is invariant to rigid-motions, but preserves joint information along the rigid-motion group.

4.4.1 Covariance of the Spatial Wavelet Transform

The oriented spatial wavelet transform has been reviewed in Section 2.3. It maps a signal x to

$$\mathcal{W}x(u, J) = x \star \phi_J(u) \tag{4.91}$$
$$\mathcal{W}x(u, \theta, j) = x \star \psi_{\theta,j}(u). \tag{4.92}$$

The wavelet transform is covariant to rigid-motion. Let us define the rigid-motion operator on $\mathbf{L}^2(\mathbb{R}^2)$ as

$$\mathcal{L}_{(v,\theta)}x(u) = x((v, \theta)^{-1}u) \tag{4.93}$$
$$= x(r_{-\theta}(u - v)) \tag{4.94}$$

and on $\mathbf{L}^2(\mathbb{R}^2 \times \Lambda)$ as

$$\mathcal{L}_{(v',\theta')}x(u, J) = x((v', \theta')^{-1}u, J) \tag{4.95}$$
$$\mathcal{L}_{(v',\theta')}x(u, \theta, j) = x((v', \theta')^{-1}(u, \theta), j) \tag{4.96}$$

then the following properties claim that the wavelet transform is covariant with respect to those actions.

Property 5. *The wavelet transform defined with a radial low pass ϕ_J and oriented wavelets $\psi_{\theta,j}$ is covariant to the action of the rigid-motion group. For any rigid motion $(v, \theta) \in \mathbb{R}^2 \rtimes SO(2)$ and any image $x \in \mathbf{L}^2(\mathbb{R}^2)$,*

$$\mathcal{L}_{v,\theta}\mathcal{W}x = \mathcal{W}\mathcal{L}_{v,\theta}x \tag{4.97}$$

where the action $\mathcal{L}_{v,\theta}$ on the left and right term of (4.97) are respectively defined as (4.93) and (4.95-4.96).

Proof. Indeed, a change of variable $w = r_{-\theta'}(v - v')$ shows that

$$\begin{aligned}
(\mathcal{W}\mathcal{L}_{v',\theta'}x)(u,\theta,j) &= (\mathcal{L}_{v',\theta'}x) \star \psi_{\theta,j}(u) && (4.98)\\
&= \int_{v\in\mathbb{R}^2} (\mathcal{L}_{v',\theta'}x)(v)\psi_{\theta,j}(u-v)dv && (4.99)\\
&= \int_{v\in\mathbb{R}^2} x(r_{-\theta'}(v-v'))\psi_{\theta,j}(u-v)dv && (4.100)\\
&= \int_{w\in\mathbb{R}^2} x(w)\psi_{\theta,j}(u-(r_{\theta'}w+v'))dw && (4.101)\\
&= \int_{w\in\mathbb{R}^2} x(w)\psi_{\theta,j}(r_{\theta'}(r_{-\theta'}(u-v')-w))dw && (4.102)\\
&= x \star \psi_{\theta-\theta',j}(r_{-\theta'}(u-v')) && (4.103)\\
&= (\mathcal{L}_{v',\theta'}\mathcal{W}x)(u,\theta,j). && (4.104)
\end{aligned}$$

Similarly, since the low pass is a radial window,

$$\phi_J(r_{\theta'}(r_{-\theta'}(u-v')-w)) = \phi_J(r_{-\theta'}(u-v')-w) \tag{4.105}$$

and thus

$$(\mathcal{W}\mathcal{L}_{v',\theta'})x(u,J) = (\mathcal{L}_{v',\theta'}\mathcal{W}x)(u,J). \tag{4.106}$$

□

An immediate corollary is that the complex modulus of wavelet transform is also covariant to rigid-motions

$$|\mathcal{W}\mathcal{L}_{v',\theta'}x| = \mathcal{L}_{v',\theta'}|\mathcal{W}x|. \tag{4.107}$$

The next section iterates over the rigid-motion wavelet modulus operator to define a rigid-motion scattering operator.

4.4.2 Rigid-Motion Orbit and Rigid-Motion Scattering

Given the covariance property (4.107) of the wavelet modulus transform with respect to the rigid-motions, we can apply a rigid-motion wavelet transform to their orbits. Section 3.3, factorized the spatial scattering space $\mathfrak{P}_m = \{(\theta_1, j_1, \ldots, \theta_m, j_m)\}$ into rotation orbits

$$\mathfrak{P}_m = SO(2) \times \bar{\mathfrak{P}}_m \tag{4.108}$$

and applied one dimensional scattering along $SO(2)$ for each of these orbits. Similarly, this section factorizes the spatial wavelet space $\mathbb{R}^2 \times \Lambda$ into rigid-motion orbits

$$\mathbb{R}^2 \times \Lambda = (\mathbb{R}^2 \rtimes SO(2)) \times \bar{\Lambda} \tag{4.109}$$

where $\bar{\Lambda} = \{J, j\}_{j<J}$. The covariance property states that when x is translated and rotated, a coefficient $|x \star \psi_{\theta,j}|(u)$ that belongs to the orbit j is going the move within this orbit from point (u, θ) to some other point $(r_{\theta'}(u - v'), \theta - \theta')$ within the same orbit j.

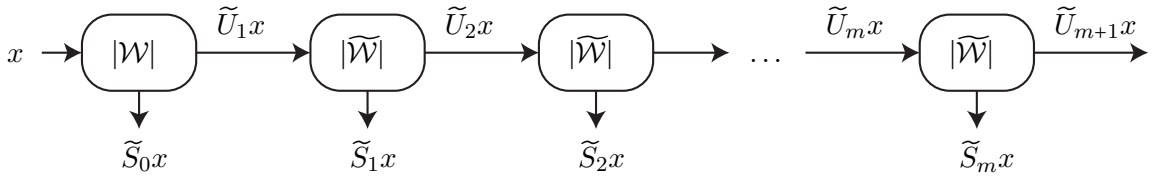


Figure 4.11: Rigid-motion scattering is similar to translation scattering of Figure 2.6, but deep wavelet modulus operators $|\mathcal{W}|$ are replaced with rigid-motion wavelet modulus operators $|\widetilde{\mathcal{W}}|$ where convolutions are applied along the rigid-motion group.

The rigid-motion joint scattering applies a rigid-motion wavelet modulus transform $\widetilde{\mathcal{W}}$ along $\mathbb{R}^2 \rtimes SO(2)$ for each orbit j. The first layer is thus a regular wavelet transform

$$\widetilde{S}_0 x(u, J) = x \star \phi_J(u) \tag{4.110}$$
$$\widetilde{U}_1 x(u, \theta, j) = |x \star \psi_{\theta,j}(u)|. \tag{4.111}$$

$\widetilde{S}_0 x$ is already invariant to rotations and translations up to 2^J. $\widetilde{U}_1 x$ is not invariant but covariant to rigid-motions. It is considered as a set of orbits, an orbit being a function of the rigid-motion variable $g \in \mathbb{R}^2 \rtimes SO(2)$. The rigid-motion joint scattering iterates over a rigid-motion wavelet modulus transform along the rigid-motion variable, which decomposes the previous layer into an invariant and a covariant part,

$$|\widetilde{\mathcal{W}}|\widetilde{U}_m x = (\widetilde{S}_m x,\ \widetilde{U}_{m+1} x) \tag{4.112}$$

with

$$\widetilde{S}_m x(g, j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_m) = \widetilde{U}_m x(., j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_m) \tilde{\star} \tilde{\phi}_{J,K}(g) \tag{4.113}$$
$$= |\ldots \| x \star \psi_{.,j_1}| \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_2}| \ldots \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_m}| \tilde{\star} \tilde{\phi}_{J,K}(g) \tag{4.114}$$

and

$$\widetilde{U}_{m+1} x(g, j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{m+1}) = |\widetilde{U}_m x(., j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_m) \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_{m+1}}(g)| \tag{4.115}$$
$$= |\ldots \| x \star \psi_{.,j_1}| \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_2}| \ldots \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_m}| \tilde{\star} \tilde{\psi}_{\tilde{\lambda}_{m+1}}(g)|. \tag{4.116}$$

We more compactly denote

$$\widetilde{U}_m(g, p_m) = \widetilde{U}_m x(g, j_1, \tilde{\lambda}_2, \ldots \tilde{\lambda}_m) \tag{4.117}$$
$$\widetilde{S}_m(g, p_m) = \widetilde{S}_m x(g, j_1, \tilde{\lambda}_2, \ldots \tilde{\lambda}_m) \tag{4.118}$$

and thus interpret joint rigid-motion scattering as a set of functions of $\mathbf{L}^2(\mathbb{R}^2 \rtimes SO(2))$ indexed by the scattering path $p_m = (j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_m)$.

This rigid-motion scattering is illustrated in Figure 4.11. The final scattering vector concatenates all scattering coefficients for all order $0 \leq m \leq M$.

$$\widetilde{S}x = \{\widetilde{S}_m x\}_{0 \leq m \leq M}. \tag{4.119}$$

Figure 4.12 shows a texture image x with its joint rigid-motion scattering coefficients. Compared to the separable scattering of Chapter 3, the joint rigid-motion scattering captures the joint variability of internal layers along position u and orientation θ, through convolutions with joint wavelet. In particular, joint scattering is able to discriminate both texture of Figure 4.3. The next section demonstrates the covariance and invariance properties of rigid-motion scattering.

4.4.3 Covariance and Invariance Properties of Rigid-Motion Scattering

Property 6. *The internal and output layers of the rigid-motion scattering are covariant to any rigid-motion.* $\forall x \in \mathbf{L}^2(\mathbb{R}^2),\ \forall g \in \mathbb{R}^2 \rtimes SO(2)$,

$$\widetilde{U}_m \mathcal{L}_g x = \mathcal{L}_g \widetilde{U}_m x \tag{4.120}$$

$$\widetilde{S}_m \mathcal{L}_g x = \mathcal{L}_g \widetilde{S}_m x \tag{4.121}$$

where for the left action $\mathcal{L}_g$ on right terms of (4.120-4.121) is defined as $\forall y \in \mathbf{L}^2(\mathbb{R}^2 \rtimes SO(2) \times \bar{\mathfrak{P}}_m), \forall g, g' \in (\mathbb{R}^2 \rtimes SO(2)), \forall p_m \in \bar{\mathfrak{P}}_m$

$$\mathcal{L}_{g'} y(g, p_m) = y(g'^{-1} g, p_m). \tag{4.122}$$

Proof. Indeed, the rigid-motion scattering iterates over the rigid-motion wavelet modulus operator $|\widetilde{\mathcal{W}}|$ which is covariant to rigid-motion, and is therefore also covariant to rigid-motions. □

The following theorem quantifies the rigid-motion invariance of the rigid-motion scattering.

Theorem 3. *There exists a constant A such that for any J, K, the rigid-motion joint scattering at spatial scale 2^J and at rotational scale 2^K verifies*

$\forall x \in \mathbf{L}^2(\mathbb{R}^2),\ \forall g = (v, \theta) \in \mathbb{R}^2 \rtimes SO(2), \forall m \in \mathbb{N}$,

$$\|\widetilde{S}_m x - \widetilde{S}_m \mathcal{L}_g x\| \leq A \left(2^{-J} \sup_{u / \widetilde{U}x(u) \neq 0} |(\mathbb{1} - g)u| + 2^{-K} |\theta \mod 2\pi| \right) \|\widetilde{U}_m x\| \tag{4.123}$$

The term $\sup_{u/\widetilde{U}x(u)\neq 0} |(\mathbb{1} - g)u|$ is the largest inverse displacement induced by g on the spatial support of $\widetilde{U}x$. Assuming that all wavelets have a finite support, the spatial support of $\widetilde{U}x$ is typically the same as the image x but dilated proportionally to 2^J. If the largest

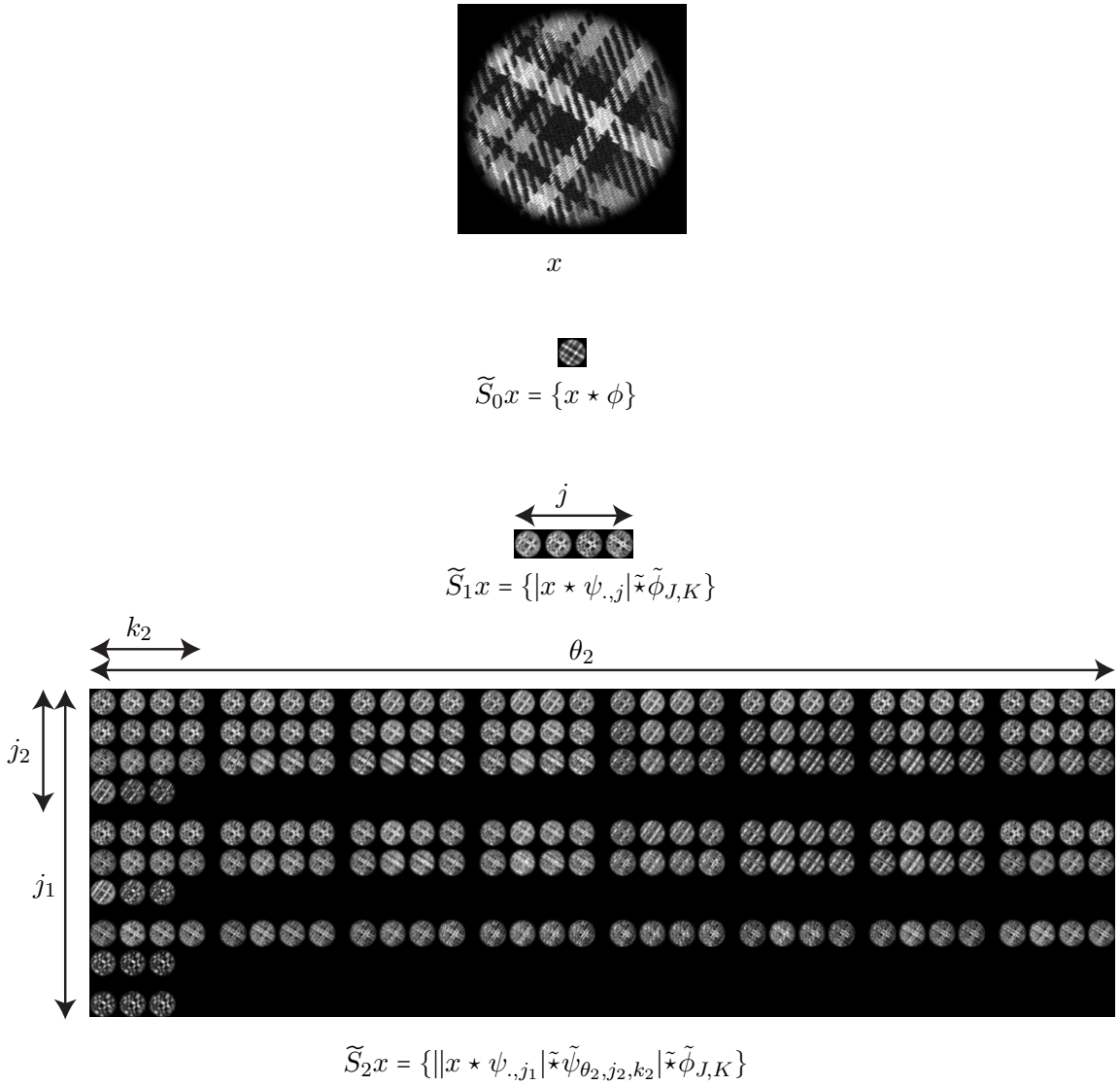


Figure 4.12: An input texture image x and its rigid-motion joint scattering coefficients of order $m = 0, 1, 2$, computed with $J = 4$ scales, $C = 8$ orientations and a orientation scale $K = 3$. In this setting, $2^K = C$ so that that full rotation invariance is achieved and we can keep only one sample along the orientation θ. The paths of second order scattering S_2 are laid out with rows corresponding to lexicographic order on (j_1, j_2) and columns corresponding to lexicographic order on (θ_2, k_2). For rows where $j_2 \neq J$ there , there are 8×4 valid columns corresponding to wavelets $\{\psi_{\theta_2,j_2,k_2}\}_{\theta_2,k_2}$ for $\theta_2 \in \pi/8[0, 7]$ and $k_2 \in [0, 3]$. For rows where $j_2 = J$, there are only 3 valid columns corresponding to wavelets of the form $\{\psi_{0,J,k_2}\}_{k_2}$ for $k_2 \in \{0, 1, 2\}$.

inverse displacement is small compared to 2^J, and if the rotation angle θ is small compared to 2^J and 2^K, then

$$\widetilde{S}_m x \approx \widetilde{S}_m \mathcal{L}_g x. \tag{4.124}$$

Proof. The proof is adapted from [Mal12]. One of the claim of Property 6 is that the internal layers are covariant to rigid-motions

$$\widetilde{U}_m \mathcal{L}_g x = \mathcal{L}_g \widetilde{U}_m x. \tag{4.125}$$

We then prove the following lemma.

Lemma 1. *For any function $\tilde{x} \in \mathbf{L}^2(\mathbb{R}^2 \rtimes SO(2))$ and any $g = (v, \theta) \in \mathbb{R}^2 \rtimes SO(2)$,*

$$\|\mathcal{L}_g \tilde{x} \tilde{\star} \tilde{\phi}_{J,K} - \tilde{x} \tilde{\star} \tilde{\phi}_{J,K}\| \leq A \left(2^{-J} \sup_{u / \tilde{x}(u,.) \neq 0} |(\mathbb{1} - g)u| + 2^{-K} |\theta \mod 2\pi| \right) \|x\|. \tag{4.126}$$

To prove Lemma 1, let us define $g = (v, \theta)$, $g' = (v', \theta')$ and decompose

$$\begin{aligned} \mathcal{L}_{g'} \tilde{x}(g) - \tilde{x}(g) &= \tilde{x}(r_{-\theta'}(v - v'), \theta - \theta') - \tilde{x}(v, \theta) && (4.127) \\ &= \underbrace{\tilde{x}(r_{-\theta'}(v - v'), \theta - \theta') - \tilde{x}(r_{-\theta'}(v - v'), \theta)}_{= \tilde{y}(g)} && (4.128) \\ &\quad + \underbrace{\tilde{x}(r_{-\theta'}(v - v'), \theta) - \tilde{x}(v, \theta)}_{= \tilde{z}(g)}. && (4.129) \end{aligned}$$

Lemma 2.11 of version 3 of [Mal12] claims that there exists A such that for all image $x \in \mathbf{L}^2(\mathbb{R}^2)$ and all displacement field $\tau \in \mathbb{C}^2(\mathbb{R}^2)$,

$$\|\mathcal{L}_\tau x \star \phi_J - x \star \phi_J\| \leq A 2^{-J} \|\tau\|_\infty \|x\| \tag{4.130}$$

where

$$\mathcal{L}_\tau x(u) = x(u - \tau(u)) \tag{4.131}$$

and

$$\|\tau\|_\infty = \sup_{u \in \mathbb{R}^2} |\tau(u)| \tag{4.132}$$

This Lemma can be adapted so that the sup in (4.132) is taken over the support of $\mathcal{L}_\tau x$

$$\|\mathcal{L}_\tau x \star \phi_J - x \star \phi_J\| \leq A 2^{-J} \sup_{u / x(u - \tau(u)) \neq 0} |\tau(u)| \|x\|. \tag{4.133}$$

Indeed, the value of τ outside the support of $\mathcal{L}_\tau x$ does not affect $\mathcal{L}_\tau x$. Let us apply (4.133) to $1 - \tau(v) = g^{-1} v$ so that $\mathcal{L}_\tau = \mathcal{L}_g$.

$$\|\mathcal{L}_g x \star \phi_J - x \star \phi_J\| \leq A 2^{-J} \sup_{v / x(g^{-1} v) \neq 0} |(\mathbb{1} - g^{-1})v| \|x\|. \tag{4.134}$$

A change of variable $u = g^{-1}v$ yields $(\mathbb{1} - g^{-1})v = (\mathbb{1} - g^{-1})gu = (g - \mathbb{1})u$ and thus

$$\|\mathcal{L}_g x \star \phi_J - x \star \phi_J\| \leq A 2^{-J} \sup_{u/x(u)\neq 0} |(\mathbb{1} - g)u| \|x\|. \tag{4.135}$$

Now we can find upper bound for both $\|\tilde{y}(g)\tilde{\star}\tilde{\phi}_{J,K}\|$ and $\|\tilde{z}(g)\tilde{\star}\tilde{\phi}_{J,K}\|$. Indeed since $\phi_J(r_{-\theta}v) = \phi_J(v)$ we have for any $\tilde{x}$

$$\begin{aligned} \tilde{x}\tilde{\star}\tilde{\phi}_{J,K}(g) &= \int_{\theta'} \left(\int_{v'} \tilde{x}(v', \theta')\phi_J(v - v')dv \right) \mathring{\phi}_K(\theta - \theta')d\theta' && (4.136) \\ &= (\tilde{x} \star \phi_J) \circledast \mathring{\phi}_K(g) && (4.137) \end{aligned}$$

where the first convolution $\star$ is applied along u and the second convolution $\circledast$ is applied along θ. Therefore, since the convolution with ϕ_J is contractive $\|\tilde{x} \star \phi_J\| \leq \|\tilde{x}\|$, applying Lemma 2.11 of [Mal12] along θ yields

$$\|\tilde{y} \star \tilde{\phi}_{J,K}\| \leq A 2^{-K} |\theta \mod 2\pi| \|\tilde{x}\|. \tag{4.138}$$

Similarly, since the convolution with $\mathring{\phi}_K$ is contractive, applying (4.135) along u yields

$$\|\tilde{z} \star \tilde{\phi}_{J,K}\| \leq A 2^{-J} \sup_{u/\tilde{x}(u,.)\neq 0} |(\mathbb{1} - g)u| \|\tilde{x}\|. \tag{4.139}$$

which finish to prove Lemma 1. Since $\widetilde{S}_m x(g, p_m) = \widetilde{U}_m x(., p_m)\tilde{\star}\tilde{\phi}_{J,K}(g)$, applying Lemma 1 to all scattering path p_m proves theorem 3. □

When the orientation scale equals the number of orientations $2^K = C$, then the orientation window $\mathring{\phi}_K$ becomes constant and $\widetilde{S}_m x(u, \theta, p_m)$ thus does not depend upon θ. Because rotation also induces spatial displacement, this does not mean that the rigid-motion scattering of two rotated images will be equal. Indeed, the first term $2^{-J} \sup_{u/\tilde{x}(u,.)\neq 0} |(\mathbb{1} - g)u|$ of theorem 3 does not vanishes when K increases. Nevertheless, the following property states that the scattering of the original and a rotated and translated image are equal, up to the global rigid-motion.

Property 7. *If $2^K = C$, then for*

$$\widetilde{S}_m \mathcal{L}_g x(u, p_m) = \widetilde{S}_m(g^{-1}u, p_m) \tag{4.140}$$

This property is not true for the translation scattering S_m, where rotation would induce an additional displacement along orientations. It is illustrated in figure 4.13. This is practical in the context of texture analysis where a global averaging will discard the global rigid-motion.

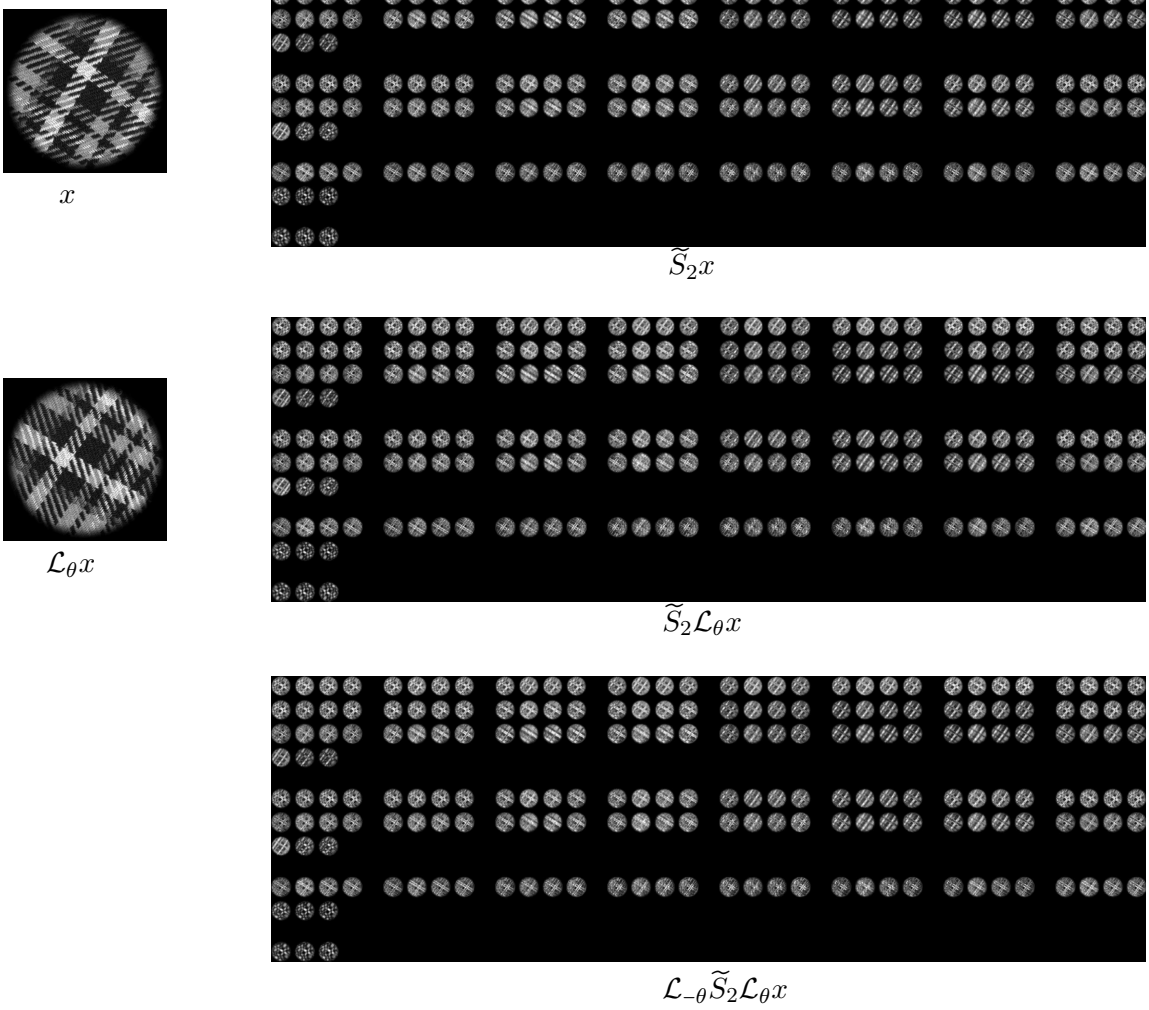


Figure 4.13: A test of rotation invariance of rigid-motion scattering when then orientation window $\mathring{\phi}_K$ is fully delocalized $2^K = C$. At the top, a texture image x and its second order rigid-motion scattering coefficients $\widetilde{S}_2 x$. We also compute the rigid-motion scattering of a rotated version of the image $r_\theta x$, and we rotate back the scattering coefficients into $\mathcal{L}_{-\theta} \widetilde{S}_2 \mathcal{L}_\theta x$. The back rotated coefficients $\mathcal{L}_{-\theta} \widetilde{S}_2 \mathcal{L}_\theta x$ are almost equal to the original coefficients $\widetilde{S}_2 x$.

Chapter 5

Texture Classification

5.1 Introduction

Image texture classification has many applications including satellite, medical and material imaging. It is also a relatively well-posed problem, compared to other computer vision tasks such as generic image recognition. A texture image can be modeled as a realization of a stationary ergodic process X. A texture dataset contains a set of images $\{X_{c,i}\}_{c,i}$, i being the index of the image within a class c. Ideally, for a given class c, all images in $\{X_{c,i}\}_i$ would be independent and identically distributed. The texture classification problem is, given a training set $\{X_{c,i}\}$, predict the class of all images in a testing set.

The usual way to tackle this problem is to train a classifier $\hat{c}$ on the training set, and use this classifier to predict the class of the images in the testing set. Because images live in a large dimensional space (typically 10^6), classifiers cannot be used directly on the raw pixel values. The image dimensionality must be lowered by computing a feature representation ΦX. The representation should discard enough information so that the classifier can generalize its training data, but also retain enough discriminative information to be able to separate different classes.

In the context of texture classification, there are an important part of the content of the image that is known to be non-discriminative and should therefore be discarded. Julesz [Jul81] has shown that our ability to distinguish different textures mainly depends upon the first order statistics of local descriptors that he has called textons, and that we are not sensitive to higher order statistics of these textons without consciously examining local differences. Yet, textons themselves depends upon local higher order statistics of the pixels values, and cannot be limited to local first or second order statistics. The notion of texton is the foundation of many texture representations including bag-of-words [LM01, VZ05, LSP05, CG10] that are global average of non-linear local quantity.

The images X are modeled as stationary ergodic process. Therefore, one should focus on quantities that do not depend on a particular realization but only on the law of the

underlying process. Stationarity means that the law of the process does not depend upon the position within the image. Ergodicity means that spatial averaging of local quantities is equivalent to ensemble averaging for sufficiently large images. Therefore, most representations used in the texture classification litterature discard locality and consist in global statistics such as histograms or bag of words. Section 5.2.1 reviews the expected scattering, a deterministic quantity that only depends on the law of the underlying process. Expected scattering cannot be computed given a finite number of finite images, but it can be estimated from one or a few finite images with a windowed scattering.

Another important source of non-discriminative variability is geometric transformations. Indeed, due to the nature of the imaging process, texture images may undergo rotations, scaling, and elastic deformation. Therefore, some texture datasets would be more adequately described as a set of transformed stationary processes $\mathcal{L}_{c,i}X_{c,i}$ where $\mathcal{L}_{c,i}$ is unknown transformation. $\mathcal{L}_{c,i}$ induces some additional variance which may not be relevant and should thus be discarded to avoid wasting training examples on learning this additional source of variability. Chapter 3 and 4 have respectively introduced a separable and a joint scattering representation, specifically designed to build invariance to translation and rotations while being stable to dilations, scaling and elastic deformation. Section 5.2.2 and 5.2.3 extend those representations to stationary processes. Section 5.4 shows how a generative PCA classifier can leverage the stability to deformation of scattering representations and how to enhance the invariance to dilation of scattering representations with data augmentation and a logarithmic non-linearity. Finally, section 5.3 and 5.5 will present classification results where scattering representations are compared with other state-of-the-art representations.

5.2 Scattering of Stationary Processes

This section briefly reviews the expected and windowed scattering that were initially introduced in [Mal12]. Section 5.2.1 reviews the translation expected scattering $\bar{S}X$. Expected scattering $\bar{S}X$ is a deterministic quantity that characterizes the law of the stationary process X. Since it involves an expected value, it cannot be computed in practice. Yet, it can be estimated by a windowed scattering SX computed from a single finite realization. Expected and windowed scattering are reviewed and generalized to separable and joint rigid-motion scattering in Sections 5.2.2 and 5.2.3.

5.2.1 Translation Expected and Windowed Scattering

We denote by $X(u)$ a stationary process with finite second-order moments $\mathbb{E}(|X(u)|^2)$. A multidimensional process $Y(u,p)$ for a spatial variable $u \in \mathbb{R}^2$ and path variable $p \in \mathfrak{P}$ is said to be stationary when it is stationary with respect to u. This means that the law of $Y(.,.)$ is the same as the law of the translated process $Y(.-v,.)$ for any $v \in \mathbb{R}^2$. Of course, this does not imply that the law of Y is the same as the law of any $Y(.,\mathcal{L}.)$ where

$\mathcal{L}$ is a mapping from $\mathfrak{P}$ to $\mathfrak{P}$. Since X is stationary, the moments do not depends upon the position u and are simply denoted $\mathbb{E}|X|^2$. Since X is stationary and since convolutions and complex modulus preserve stationarity, the multidimensional process $UX(u,p)$ is also stationary, its law does not depend upon the position u. The expected scattering of X is defined as, $\forall m \in \mathbb{N}, \forall p_m = (\lambda_1, \dots, \lambda_m) \in \mathfrak{P}_m$

$$\begin{aligned} \bar{S}X(p_m) &= \mathbb{E}(UX(u,p_m)) && (5.1)\\ &= \mathbb{E}\left(||X \star \psi_{\lambda_1}| \dots \star \psi_{\lambda_m}|\right). && (5.2) \end{aligned}$$

The expected scattering $\bar{S}X(p_m)$ is a deterministic quantity that only depends upon the law of the random process X, not upon a particular realization nor the spatial position u. Because it involves an expected value $\mathbb{E}$, it cannot be computed from a finite number of finite realizations. Yet, it can be approximated by the windowed scattering

$$\begin{aligned} SX(u,p_m) &= UX(.,p_m) \star \phi_J(u) && (5.3)\\ &= |\dots|X \star \psi_{\lambda_1}| \dots \star \psi_{\lambda_m}| \star \phi_J(u) && (5.4) \end{aligned}$$

which is exactly the translation scattering described in Chapter 2 but applied to a random process X. Since X is random, the windowed scattering SX is also a random quantity that depends upon the particular realization, and upon the position u, but which can be actually computed from that single, finite realization. Since $\int \phi_J = 1$, the windowed scattering is an unbiased estimator of the expected scattering

$$\forall u \in \mathbb{R}^2,\ \mathbb{E}SX(u,p_m) = \bar{S}X(p_m). \tag{5.5}$$

The ergodicity of the process means that the averaging on the position converges toward the expected value. Therefore, Conjecture 4.4 of [Mal12] states that

$$SX(u,p_m) \xrightarrow[J\to\infty]{} \bar{S}X(p_m) \tag{5.6}$$

in the sense of mean square $X_n \xrightarrow[n\to\infty]{} x \Leftrightarrow \mathbb{E}|X_n - x|^2 \xrightarrow[n\to\infty]{} 0$. In practice, we can only compute the windowed scattering $SX(u,p_m)$ for a window size 2^J up to the size of the image.

5.2.2 Separable Expected and Windowed Scattering

By definition, the law of a stationary process, and therefore its expected scattering, is invariant to translations. Yet, the law of a stationary process is a priori not invariant to other transformations $\mathcal{L}$ such as rotations, scaling and shears. Neither is their expected or windowed translation scattering. Therefore, the scattering of transformed images contain additional variance which is induced by transformations and does not help to discriminate

different classes of process. Similarly to Section 5.2.1, this section defines expected separable scattering which is invariant to rotations, and applies the separable rigid-motion scattering of Chapters 3 and 4 to estimate theses expected quantities.

For each rotation orbit $\bar{p}$ defined in (3.68-3.69), the function $\theta \mapsto \bar{S}X(\theta\bar{p})$ is a deterministic function on $SO(2)$, of which one can compute a scattering. The expected separable rigid-motion scattering is built on top of the expected scattering $\bar{S}_m$, with a second periodic scattering $\mathring{S}_{\mathring{m}}$ along the rotation orbit

$$\begin{aligned} \bar{\bar{S}}_{m,\mathring{m}}X(\theta_1,\bar{p},\mathring{p}) &= \mathring{S}_{\mathring{m}}\left(\theta \mapsto \bar{S}_m X(\theta\bar{p})\right)(\mathring{p}) && (5.7)\\ &= \mathring{S}_{\mathring{m}}\left(\theta \mapsto \mathbb{E}U_m X(u,\theta\bar{p})\right)(\mathring{p}). && (5.8)\end{aligned}$$

As the expected scattering $\bar{S}X$, the expected separable rigid-motion $\bar{\bar{S}}X$ only depends upon the law of the process X, not on a particular realization. In addition, it is also invariant to rotation of this law, up to the window width 2^K of the second scattering $\mathring{S}_{\mathring{m}}$. If $2^K = C$ the expected scattering is strictly invariant to any rotation. The expected separable rigid-motion scattering $\bar{\bar{S}}X$ is estimated with the windowed separable rigid-motion scattering, which is exactly the separable scattering described in chapter 3, but applied to a random process

$$\mathring{S}_{m,\mathring{m}}X(\theta_1,\bar{p},\mathring{p}) = \mathring{S}_{\mathring{m}}\left(\theta \mapsto S_m X(\theta\bar{p})\right)(\mathring{p}). \quad (5.9)$$

Separable scattering has been introduced in our paper [SM12]. Section 5.3 describes the texture classification experiments conducted in [SM12] on relatively easy texture datasets OUTex 10 [OMP+02].

5.2.3 Joint Expected and Windowed Scattering

As explained in Section 4.2, the separable expected scattering considers the different paths $U_m X(.,p)$ independently and thus capture quantities that depend only upon their marginal distributions. It does not capture correlations between different paths of the same order. A joint rigid-motion expected scattering applies the expected values after group convolutions and thus has the opportunity to capture quantity that does depend upon the joint law of different paths. It is defined for $\tilde{p} = (j_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_m)$ as the expected value of the convolution of the internal layers of joint scattering $\widetilde{U}_m X$ with a separable window $\delta \times \phi_K(g) = \delta(v)\phi_K(\theta)$ where δ is a dirac.

$$\begin{aligned} \overline{\widetilde{S}}_m X(\theta,\tilde{p}) &= \mathbb{E}\left(\widetilde{U}_m X(.,\tilde{p}) \tilde{\star} (\delta \times \phi_K)(g)\right) && (5.10)\\ &= \mathbb{E}\left(\int_{g' \in \mathbb{R}^2 \rtimes SO(2)} \widetilde{U}_m X(g',\tilde{p})\delta(r_{-\theta'}(v-v'))\phi_K(\theta-\theta')dg'\right) && (5.11)\\ &= \mathbb{E}\left(\int_{\theta' \in SO(2)} \widetilde{U}_m X(v,\theta-\theta',\tilde{p})\phi_K(\theta-\theta')d\theta'\right) && (5.12)\\ &= \mathbb{E}\left(\widetilde{U}_m X(v,.,\tilde{p}) \circledast \phi_K(\theta)\right). && (5.13)\end{aligned}$$

This is a correct definition which does not depend upon the position v but only on $\theta, \tilde{p}$. Indeed, let us suppose that for a fixed $m \in \mathbb{N}$, $\widetilde{U}_m X(v, \theta, \tilde{p})$ is stationary, then

$$\begin{aligned} \widetilde{U}_{m+1} X(v, \theta, \tilde{p}, \tilde{\lambda}_{m+1}) &= \left| \widetilde{U}_m X(., \tilde{p}) \circledast \tilde{\psi}_{\tilde{\lambda}_{m+1}}(g) \right| \\ &= \left| \int_{\theta' \in SO(2)} \left(\widetilde{U}_m X(., \theta', \tilde{p}) \star \psi_{\theta_{m+1}+\theta', j_{m+1}}(u) \right) \mathring{\psi}_k(\theta - \theta') d\theta \right| \end{aligned} \tag{5.14}$$

which is a complex modulus of a sum of convolutions of stationary processes, and is therefore a stationary process. Thus, since $\widetilde{U}_1 X$ is a stationary process, we prove by induction that $\widetilde{U}_m X$ is a stationary process for every $m \in \mathbb{N}$. Therefore, the definition (5.10-5.13) of the expected joint scattering $\overline{\widetilde{S}}$ is correct and does not depend upon the position v. Contrary to the expected separable scattering $\bar{\mathring{S}}$, the expected joint scattering $\overline{\widetilde{S}}$ does not process internal layers independently and therefore captures information about the joint laws of internal processes $\widetilde{U}_m(., \tilde{p})$. As the expected scattering, it is invariant to rotation of the process X up to rotation angles of the order of the orientation window width 2^K. If $2^K = C$, the expected scattering is strictly invariant to any rotation. The expected joint scattering cannot be computed from a single realization but is estimated by a windowed joint scattering, which is exactly the joint scattering presented in Chapter 4, but applied to a stationary process X

$$\widetilde{S}_m X(g, \theta, \tilde{p}) = \widetilde{U}_m X(., \tilde{p}) \tilde{\star} \phi_{J,K}(g). \tag{5.15}$$

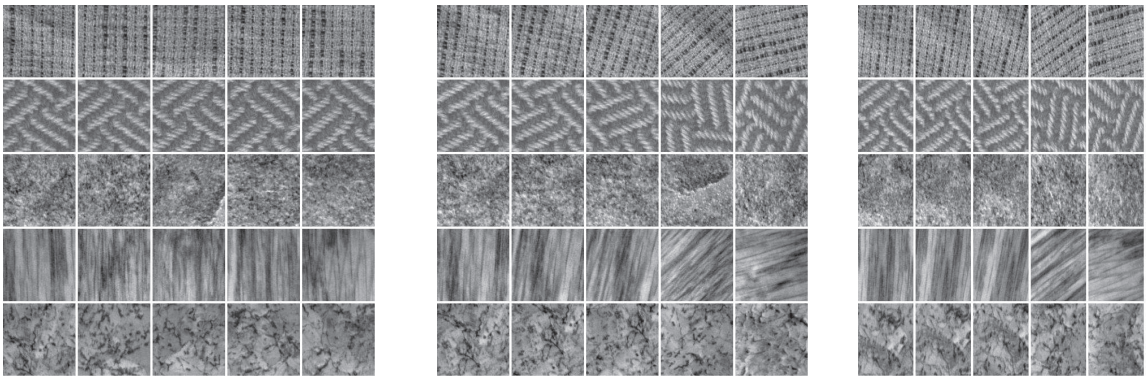
The definition of the windowed joint scattering is the same as the expected joint scattering (5.10-5.13) but the combination of expected value and group convolution with $\delta \times \phi_K$ is replaced with a group convolution with $\phi_{J,K}$. Since it is sensitive to the joint law of internal layers, joint scattering is a tighter invariant and can thus be used on harder problems than separable scattering. Section 5.5 presents texture classification experiments with challenging datasets conducted in our papers [SM13, SM14] where joint scattering obtains comparable or better results than existing state-of-the art methods.

5.3 Classification with Separable Scattering

A first texture classification experiments has been performed with separable scattering in our paper [SM12] on the OUTex10 database [OMP+02] (*rot* experiment). The OUTex10 database contains 24 different texture classes. Each class has 20 training samples with a single orientation which is normalized to 0°. There are $24 \times 20 \times 8$ testing samples corresponding to 20 samples in each class that are rotated by $5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°$. Since only one orientation of each texture class is present in the testing sets, the classifier cannot learn the rotation invariance from the data. The dataset essentially contains the same images in the training and in the testing sets, but rotated and cropped differently. Thus, it is a fairly easy dataset, where the class assignment is more a matching task than a

true classification problem. Simple approaches such as the separable rigid-motion scattering described in chapter 3 or other separable descriptors based on Locally Binary Pattern (LBP) [OMP+02] provide very high classification rates. Yet, the latter are not stable to deformations, and a slight shear can dramatically lower their performances. A separable rigid-motion scattering is stable to deformation and preserves good results in presence of such slight deformations.

A second experiment (*rot-shear*) simulates an horizontal shear. A shear $T_{t,\beta}$ corresponds to a dilation by factor t in the direction β, as defined in 3.9. In experiment (*rot-shear*), an horizontal shear $T_{1.3,0}$ is implemented with a gaussian blur with $\sigma = \sqrt{1.3^2 - 1}$ and a subsampling at intervals 1.3 in the horizontal direction only, for all images in the testing set. Images in the training set are cropped to keep the same image size in both sets. This experiments was designed to compare the robustness to deformations of separable scattering and other texture descriptors. Figure 5.1 shows some training and testing samples from both experiments.



Training set Testing for *rot* Testing for *rot-shear*

Figure 5.1: A few samples of the databases used for experiments *rot* and *rot-shear*. The *rot* experiment was conducted with the original OUTex 10 training and testing sets, in which testing images are rotated but training images are not, and thus the classifier cannot learn the rotation invariance from the data. In the *rot-shear* experiments, all the images of the testing set have undergone an additional horizontal shear of amplitude 1.3 to test the robustness of descriptors to deformations.

The images of the OUTex texture dataset are 128×128 grey level images. The separable scattering is computed with a a first spatial scattering SX, with $C = 8$ orientations and a maximum scale $2^J = 2^5$. This first scattering is then averaged spatially along the whole image. This would corresponds to a scattering with a scale equal to the image size $2^J = 2^8$ where the paths with the largest scale have been discarded. The reason why we discard paths with large scales is that they contain more variance as an estimator of the expected

	rot	*rot-shear*
$\text{LBP}^{riu2}/\text{VAR}_{(8,1)+(16,2)+(24,3)}$ (r)	97.7	NC
$\text{LBP-HF}_{(8,1)+(16,2)+(24,3)}$ (c)	96.59	67.50
RI-LPQ (c)	98.26	78.02
$\{\mathring{S}_{m,\mathring{m}}X\}$, $m, \mathring{m} \leq 1,2$	96.72	81.61
$\{\mathring{S}_{m,\mathring{m}}X\}$, $m, \mathring{m} \leq 2,0$	97.73	89.38
$\{\mathring{S}_{m,\mathring{m}}X\}$, $m, \mathring{m} \leq 2,1$	98.62	92.89
$\{\mathring{S}_{m,\mathring{m}}X\}$, $m, \mathring{m} \leq 2,2$	**98.75**	**93.07**

Table 5.1: Classification results on OUTex 10. (r) is reported from papers, (c) is obtained with authors' software [OMP+02]. State-of-the-art approaches are compared with separable rigid motion scattering. Best results are obtained by concatenating all the paths of order less than $2,2$. As can be seen in the *rot-shear* experiment, scattering performances degrade gracefully when images undergo deformations, which is not the case for other methods.

scattering $\bar{S}X$, since fewer independent samples have contributed to their averaged values. The spatial scattering contains $CJ + C^2J(J-1)/2 = 680$ paths of order 1 and 2, which corresponds to $J + CJ(J-1)/2 = 85$ rotational orbits of $C = 8$ orientations each. A second scattering operator transforms each orbit into $1 + K + K^2 = 7$ paths corresponding to order $\mathring{m} = 0, 1, 2$, where $K = 2$ is the maximum scale along the rotation parameter. Unlike the spatial scattering, the second scattering does not discard non-increasing scales along the rotation parameters. The reason for that is that the modulus smoothing effect is limited with so few samples as 8, therefore non-increasing rotational paths are non-negligible. In those experiments, the second scattering performs a complete averaging along the rotation parameter, leaving one sample per rotational path. Finally, the largest separable rigid-motion scattering used in those experiments was a $85 \times 7 = 595$ dimensional vector for $M, \mathring{M} = 2, 2$.

A nearest neighbor classifier is applied to the combined scattering $\{\mathring{S}_{m,\mathring{m}}X\}_{m,\mathring{m} \leq M,\mathring{M}}$ representation with several choices of maximum path length $M, \mathring{M} \leq 2$, and to other state-of-the-art descriptors for rotation invariant texture analysis. We have used $\mathbf{L}_2$ distance except for LBP-HF where the authors recommend to use $\mathbf{L}_1$[ZAMP12] . Local Binary Pattern (LBP) [OPM02] computes histograms of local binary patterns. Bins that correspond to rotated versions of the same pattern are merged, which leads to a loss of discriminability. Local Binary Pattern Histogram Fourier (LBP-HF)[ZAMP12] computes a Fourier transform modulus on the rotation parameter of LBP[OPM02]. It thus maintains variability information along angles while achieving rotation invariance. Rotation Invariant Local Phase Quantization (RI-LPQ) [ORH08] computes windowed Fourier coefficients over a discrete set of frequencies distributed along circles. The phase is quantized to obtain a binary word on which a histogram is computed. As opposed to LBP and LBP-HF, RI-LPQ is robust to image blurring [ORH08].

Classification results obtained with LBP [OMP+02], LBP-HF [ZAMP12], RI-LPQ [ORH08] and separable rigid-motion scattering $\mathring{S}X$ are presented in Table 5.1. The combined scattering achieves the best results with or without shear distortions. The classification accuracy is improved when adding second order paths both in space and in rotation, with $M = \mathring{M} = 2$. Increasing further the path length has a marginal impact because the remaining scattering energy is negligible. For the *rot-shear* experiment which includes tilt deformations, the scattering brings an important improvement because it is stable to deformation, which is not the case for other state-of-the-art rotation invariant representations [ORH08, ZAMP12].

5.4 Scale and Deformation Invariance with Augmented log-PCA Classifier

Rigid-motion scattering performs reasonably well with a nearest-neighbor classifier on relatively easy datasets such as OUTex 10 [OMP+02], where only pure rotations occur in images, or small shears. Yet, texture images can also undergo other geometric transformations such as dilations, shears or elastic deformations. Texture datasets KTH-TIPS [HCFE04], UIUCTex [LSP05] and UMDTex [XJF09] contain instances of such transformations as can be seen on Figures 5.4, 5.5 and 5.6. Explicitly building invariant to those transformations can lower the number of training examples that is required to build an accurate model, thus dramatically improving classification results on small training set. This section describes an ad hoc classifying scheme for enforcing invariance to scaling and deformations. The experiments described in Section 5.5 achieve state-of-the-art results on most texture datasets by applying this scheme to joint rigid-motion scattering.

One can define wavelets [DKSZ11] and a scattering transform on the affine group to build affine invariance. However the full affine group is much larger than the rigid-motion group and a scattering on the affine group would involve heavy and unnecessary computations. A limited range of dilations and shears is available for finite resolution images, which allows one to linearize these variations. Invariance to dilations, shears and deformations is obtained with linear projectors implemented at the classifier level, by taking advantage of the scattering's stability to small deformations. In texture classification, there is typically a small number of training examples per class, in which case PCA generative classifiers can perform better than linear SVM discriminative classifiers [BM13].

Let X_c be a stationary process representing a texture class c. Its rigid-motion scattering transform $\widetilde{S}X_c$ typically has a power law behavior as a function of its scale parameters. It is partially linearized by a logarithm which thus improves linear classifiers. The random process $\log \widetilde{S}X_c$ has an energy which is essentially concentrated in a low-dimensional affine space

$$\mathbf{A}_c = \mathbb{E}(\log \widetilde{S}X_c) + \mathbf{V}_c$$

where $\mathbf{V}_c$ is the principal component linear space, generated by the eigenvalues of the

covariance of $\log \widetilde{S} X_c$ having non-negligible eigenvalues.

The expected value $\mathbb{E}(\log \widetilde{S} X_c)$ is estimated by the empirical average μ_c of the $\log \widetilde{S} X_{c,i}$ for all training examples $X_{c,i}$ of the class c. To guarantee that the scattering is partially invariant to scaling, we augment the training set with dilated versions of each $X_{c,i}$ by different scaling factors 2^j for $0 \leq j < H$. The value H quantifies the range of scale invariance. It is limited by the size of the image. Increasing H theoretically increases invariance to scaling, but also lowers the number of independent samples when computing spatial averaging which increases the variance, thus adversely affecting classification results. We typically chose $H = 2$ and sample j at half integer which results in 4 scaling factors $\{1, \sqrt{2}, 2, 2\sqrt{2}\}$. We compute the log scattering of the dilated examples $\mathcal{L}_j X_{c,i}(u) = X_{c,i}(2^j u)$ and their averaged values

$$\mu_c = \left(\sum_{\substack{0 \leq j < H \\ i}} 1 \right)^{-1} \sum_{\substack{0 \leq j < H \\ i}} \log \widetilde{S} \mathcal{L}_j X_{c,i}. \tag{5.16}$$

Internal layers of scattering have covariance properties with respect to scale which makes it possible to accelerate the computation of $\widetilde{S} \mathcal{L}_j X$ by reusing some paths of internal layers $\widetilde{U} X$.

The principal components space $\mathbf{V}_c$ is estimated from the singular value decomposition (SVD) of the matrix of centered training example $\log \widetilde{S} \mathcal{L}_j X_{i,c} - \mu_c$ with all possible examples i and dilation 2^j for a given class c. The number of non-zero eigenvectors which can be computed is equal to the number of training examples per class, typically 20, times the number of dilations per examples, typically 4, minus one, that is consumed by the estimation of the mean μ_c. The space generated by all eigenvectors is denoted $\mathbf{V}_c$, is thus typically of dimension 79. Therefore, it is not necessary to regularize it, since this dimension is relatively small compared to the dimensionality of scattering, which is of the order of 1000.

Given a test image X, we denote by $A \log \widetilde{S} X$ the average of the log scattering transform of X and its dilated versions $\mathcal{L}_j X$

$$A \log \widetilde{S} X = \left(\sum_{0 \leq j < H} 1 \right)^{-1} \sum_{0 \leq j < H} \log \widetilde{S} \mathcal{L}_j X. \tag{5.17}$$

The representation $A \log \widetilde{S} X$ used at test time is therefore a scale-averaged log scattering transform, which provides an additional partial scaling invariance. We denote by $P_{\mathbf{V}_c} A \log \widetilde{S} X$ the orthogonal projection of $A \log \widetilde{S} X$ in the scattering space $\mathbf{V}_c$ of a given class c. The PCA classification computes the class $\hat{c}(X)$ which minimizes the distance $\| (Id - P_{\mathbf{V}_c})(A \log \widetilde{S} X - \mu_c) \|$ between $A \log \widetilde{S} X$ and the affine space $\mu_c + \mathbf{V}_c$

$$\hat{c}(X) = \arg\min_c \| (Id - P_{\mathbf{V}_c})(A \log \widetilde{S} X - \mu_c) \|^2. \tag{5.18}$$

This minimum projection error classifier is illustrated in Figure 5.2. The complete classification algorithm is illustrated in Figure 5.3.

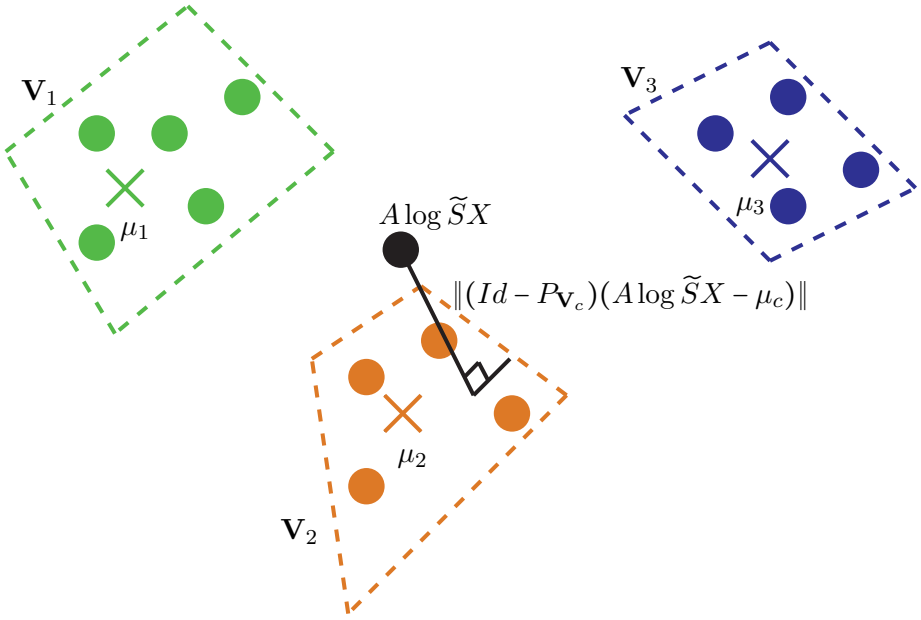


Figure 5.2: A PCA-classifier which classifies a test image X according to the minimum distance of the log scattering $A \log \widetilde{S}X$ to the class affine subspace $\mu_c + \mathbf{V}_c$. The affine subspaces are obtained with the log scattering of training examples of a class, at all scale $\{1, \sqrt{2}, 2, 2\sqrt{2}\}$. For 20 training examples per class, each space V_c is thus of dimension $4 \times 20 - 1 = 79$.

5.5 Classification with Joint Scattering and PCA

This section details classification results on image texture datasets KTH-TIPS [HCFE04], UIUCTex [LSP05], UMDTex [XJF09] and FMD [SRA09]. Those datasets contain images with different range of variability for each different geometric transformation type. We give results for progressively more invariant versions of the scattering and compare with state-of-the-art approaches for all datasets. Since these datasets are much more challenging than the OUTex 10 [OMP+02] , we use the joint rigid-motion scattering $\widetilde{S}X$ described in Chapter 4 instead of the separable rigid-motion scattering $\mathring{S}X$ of Chapter 3. The invariance to scaling, shears and deformations is enforced by the augmented log PCA classifier described in Section 5.4.

Most state of the art algorithms use separable invariants to define a translation and rotation invariant algorithms, and thus lose joint information on positions and orientations.

Train size	5	20	40
COX [HGFB11]	80.2 ± 2.2	92.4 ± 1.1	95.7 ± 0.5
BIF [CG10]	-	-	98.5
SRP [LFKZ11]	-	-	99.3
Translation scattering	69.1 ± 3.5	94.8 ± 1.3	98.0 ± 0.8
Rigid-motion scattering	69.5 ± 3.6	94.9 ± 1.4	98.3 ± 0.9
+ log	76.2 ± 3.3	96.0 ± 1.1	98.8 ± 0.7
+ multiscale avg test	77.8 ± 3.6	97.4 ± 1.0	99.2 ± 0.6
+ multiscale svd train	$\mathbf{84.3} \pm 3.1$	$\mathbf{98.3} \pm 0.9$	$\mathbf{99.4} \pm 0.4$

Table 5.2: Classification accuracy with standard deviation on KTH-TIPS [HCFE04] database. Columns correspond to different numbers of training examples per class. The first few rows give the best published results. The last rows give results obtained with progressively refined scattering invariants. Best results are bolded.

Training size	5	10	20
Lazebnik [LSP05]	-	92.6	96.0
WMFS [XYLJ10]	**93.4**	97.0	98.6
BIF [CG10]	-	-	98.8 ± 0.5
Translation scattering	50.0 ± 2.1	65.2 ± 1.9	79.8 ± 1.8
Rigid-motion scattering	77.1 ± 2.7	90.2 ± 1.4	96.7 ± 0.8
+ log	84.3 ± 2.1	94.5 ± 1.1	98.2 ± 0.6
+ multiscale avg test	86.6 ± 2.0	95.4 ± 1.0	98.6 ± 0.6
+ multiscale svd train	93.3 ± 1.4	$\mathbf{97.8} \pm 0.6$	$\mathbf{99.4} \pm 0.4$

Table 5.3: Classification accuracy on UIUCTex [LSP05] database.

Training size	5	10	20
WMFS [XYLJ10]	93.4	97.0	98.7
SRP [LFKZ11]	-	-	99.3
Translation scattering	80.2 ± 1.9	91.8 ± 1.4	97.4 ± 0.9
Rigid-motion scattering	87.5 ± 2.2	96.5 ± 1.1	99.2 ± 0.5
+ log	91.9 ± 1.7	97.6 ± 0.8	99.3 ± 0.4
+ multiscale avg test	91.6 ± 1.6	97.7 ± 0.9	99.6 ± 0.4
+ multiscale svd train	$\mathbf{96.6} \pm 1.0$	$\mathbf{98.9} \pm 0.6$	$\mathbf{99.7} \pm 0.3$

Table 5.4: Classification accuracy on UMDTex [XJF09] database.

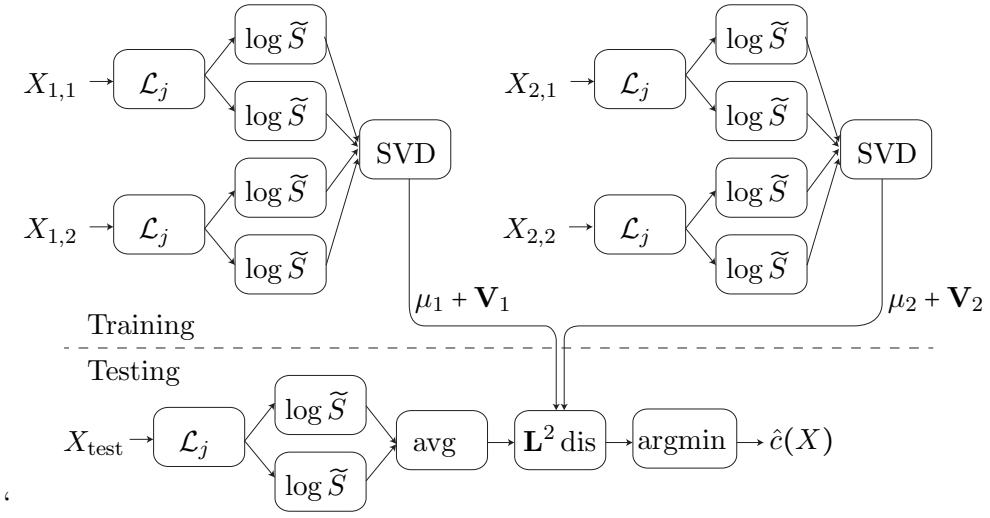


Figure 5.3: The complete diagram of the classification experiments conducted with KTH-TIPS, UIUCTex and UMDTex datasets. Training images $X_{c,i}$ where c is the class and i is the index within the class are dilated by $\mathcal{L}_j$. The figure only displays two dilations per image but we have used four dilations $\{1, \sqrt{2}, 2, 2\sqrt{2}\}$ in the experiments. We compute the log of the joint rigid motion scattering $\widetilde{S}$ of the dilated images. All the vector of a class c are aggregated into an affine space $\mu_c + \mathbf{V}_c$ computed with a singular value decomposition (SVD). The dimension the space is the total number of dilated training images minus one. Given a test image, we compute the log joint scattering of its dilated versions, and we average them which provides an additional invariance to scaling. The testing image is classified according to the minimum distance of its scale-averaged log scattering to the affine space $\mu_c + \mathbf{V}_c$.

This is the case of [LSP05] where rotation invariance is obtained through histograms along concentric circles, as well as Log Gaussian Cox processes (COX) [HGFB11] and Basic Image Features (BIF) [CG10] which use rotation invariant patch descriptors calculated from small filter responses. Sorted Random Projection (SRP) [LFKZ11] replaces histogram with a similar sorting algorithm and adds fine scale joint information between orientations and spatial positions by calculating radial and angular differences before sorting. Wavelet Multifractal Spectrum (WMFS) [XYLJ10] computes wavelet descriptors which are averaged in space and rotations, and are similar to first order scattering coefficients $S_1 x$.

We compare the best published results [LSP05, HGFB11, CG10, XYLJ10, LFKZ11] and scattering invariants on KTH-TIPS (table 5.2), UIUCTex (table 5.3) and UMDTex (table 5.4) texture databases. For the KTH-TIPS, UIUCTex and UMDTex database, Tables 1, 2 and 3 give the mean classification accuracy and standard deviation over 200 random splits between training and testing for different training sizes. Classification accuracy is

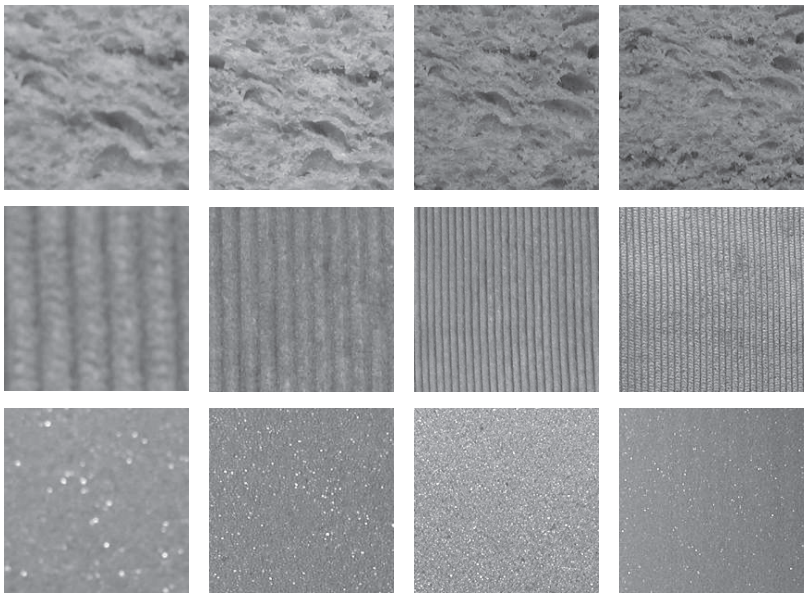


Figure 5.4: Each row shows images from the same texture class in the KTH-TIPS database [HCFE04]. In this database, each class contains 9 different scales, 3 different view points, 3 different illumination for a total of 81 images per class.

computed with scattering representations implemented with progressively more invariants, and with the PCA classifier of Section 5.4. As the training sets are small for each class c, the dimension D of the high variability space $\mathbf{V}_c$ is set to the number of dilated training images. The space $\mathbf{V}_c$ is thus generated by the D scattering vectors of the dilated training set. For larger training databases, it should be adjusted with a cross validation as in [BM13].

Classification accuracy in Tables 5.2, 5.3 and 5.4 are given for progressively more invariant scattering representations and classification scheme. The rows "Translation scattering" correspond to the scattering described in Chapter 2 and initially introduced in [BM13]. The rows "Rigid-motion scattering" replace the translation invariant scattering by the joint rigid-motion scattering of Chapter 4, with no dilation of the training or the testing set. The rows "+ log " corresponds to the joint rigid-motion scattering, with a logarithm non-linearity to linearize scaling. The rows "+ multiscale avg test " is obtained by averaging the scattering of dilated versions of a test image. Finally, the rows "+ multiscale svd train " is obtained by augmenting the training space $\mu_c + \mathbf{V}_c$ with the joint scattering of all the dilated versions of images in the training set of a class c, as described in Section 5.4, and illustrated in figure 5.3.

KTH-TIPS [HCFE04] dataset contains 10 classes of 81 samples per texture class with 9 different dilations, 3 different shears and 3 different illuminations but no rotation. Figure

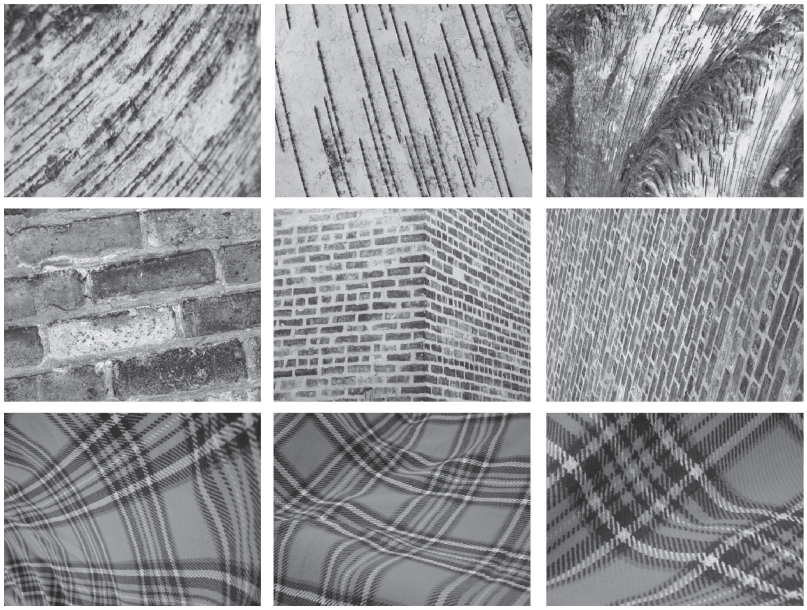


Figure 5.5: Each row shows images from the same texture class in the UIUCTex database [LSP05] which contains 25 classes of 40 images at resolution 640×480 with important rotation, scaling and deformation variability.

Training size	50
SRP [LFKZ11]	48.2
Best single feature (SIFT) in [SLRA13]	41.2
Feature selection framework [SLRA13]	60.6
Rigid-motion scattering + log on grey images	51.22
Rigid-motion scattering + log on YUV images	53.28

Table 5.5: Classification accuracy on FMD [SRA09] database.

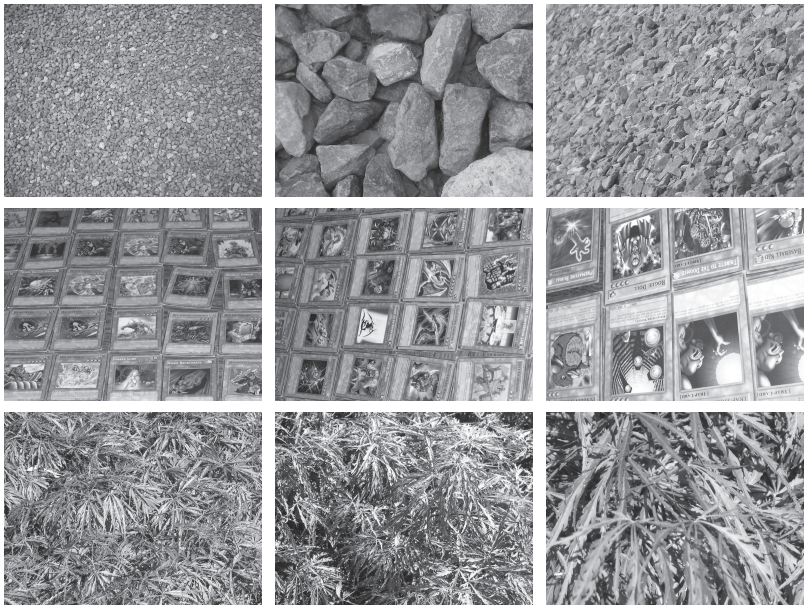


Figure 5.6: Each row shows images from the same texture class in the UMDTex database [XJF09]. UMDTex is similar to UIUCTex but has higher resolution 1280×960 images which allows to compute more scattering scale.

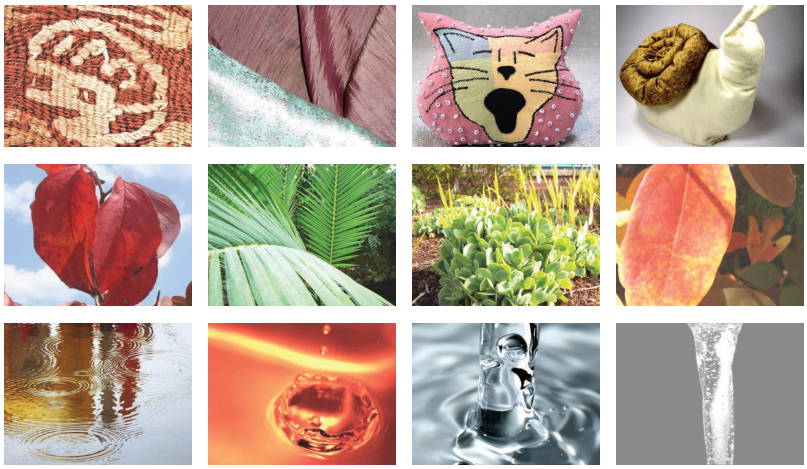


Figure 5.7: Each row shows images from the same texture class in the FMD database [SLRA13]. In addition to objects with non-frontal non-flat surfaces, this dataset contains much more diverse classes of texture.

5.4 shows a few sample images of this dataset. Table 5.2 shows that the rigid-motion scattering does not degrade results but the logarithm, the training and testing scale invariants all provide significant improvements.

UIUCTex [LSP05] and UMDTex [XJF09] datasets both contain 25 classes of 40 samples with uncontrolled deformations including shears, perspectivity effects and non-rigid deformations. These datasets are illustrated in Figures 5.5 and 5.6. For both these databases, rigid-motion scattering and the scale invariance provide considerable improvements over translation scattering. As demonstrated in Tables 5.3 and 5.4, the overall approach achieves and often exceeds state-of-the-art results on these two databases.

FMD [SRA09] dataset contains 10 classes of 100 samples, some of which are shown in Figure 5.7. Each class contains images of the same material manually extracted from Flickr. Unlike the three previous databases, images within a class are not taken from a single physical sample object but comes with variety of material sub-types which can be very different. Therefore, the PCA classifier of Section 5.4 can not linearize deformations and discriminative classifiers tend to give better results. The scattering results reported in Table 5.5 are obtained with a one versus all linear SVM. Rigid-motion log scattering applied to each channel of YUV image and concatenated achieves 52.2 % accuracy. To our knowledge, this is the best result for a single feature. Better results can be obtained using multiple features and a feature selection framework [SLRA13].

Chapter 6

Generic Object Classification

6.1 Introduction

Chapters 3 and 4 have introduced two representations that have the same invariance to rigid-motion but with very different discriminative power. As argued in Section 4.2, separable representations process the different paths of internal layers completely independently, and by doing so, they lose all the information contained in the local correlation of the different paths of these layers. On the contrary, joint representations consider an internal layer as a multidimensional function. In the proposed joint rigid-motion scattering, an internal layer was described as a set of functions, or orbits, of the rigid-motion group, but internal layers can also be interpreted as a function of some much larger group, which eventually is not hardcoded but learned.

Deep networks [HS06, LLB+98] are representation with little a priori information on the data. They can be trained to discriminate different classes of signal, and by doing so, they possibly learn some class invariant. Deep networks consist in a cascade of linear operators, whose weights are learned, and non-linear operators. For large images, deep networks, as is, would involve too many weights to be learned. Convolutional neural networks (ConvNets) [LBD+89, LKF10, KSH12, DCM+12, ZF13] take advantage of the image structure to limit the number of weights while maintaining the ability to learn generic invariance.

During an internship at Google, I have had the opportunity to experiment large scale generic object recognition with distbelief, Google's distributed implementation of deep networks [DCM+12]. Those networks involve relatively expensive convolutions, some of which whose trained weights have appeared to be highly redondant. From this finding, we have adapted the efficient rigid-motion convolution introduced in section 4.3.1 into a generic separable convolutional layer. This separable convolutional layer factorizes multidimensional convolutions into purely spatial convolutions followed by pointwise matrix multiplications along the path variable. Early experiments have shown that such separable convolutional layers require less data to train, less resources per data and result in identical to slightly

better final accuracy on ImageNet ILSVRC2012 classification dataset. Section 6.2 details these separable convolutions and briefly describes the classification experiments.

In collaboration with Edouard Oyallon, we have also experimented with pure scattering, that is without any learning of the weights of internal layers, on generic object classification datasets such as Caltech 101 and 256. Oyallon [OMS14] has shown that a slightly modified version of the joint rigid-motion scattering introduced in chapter 4 can achieve similar performances as the first layers of fully trained deep networks. These experiments are described in section 6.3. If the performances of scattering are still behind a fully trained deep network with 7 layers, these experiments open the possibility to simplify deep neural network learning by initializing the first few layers with wavelets.

6.2 Separable Convolutions for Generic Deep Networks

This sections briefly reviews the convolutional deep network for generic image classification [KSH12, DCM$^+$12, ZF13]. These networks have obtained state-of-the-art results on most large scale generic image classification datasets, such as ImageNet ILSVRC2012. Section 6.2.1 reviews the dense convolutions that implement the linear operators of the first layers of [KSH12, DCM$^+$12, ZF13]. Section 6.2.2 observes that some of the weights in dense convolutional layers are highly redundant and proposes a separable convolutional layer inspired by the rigid-motion convolution of Chapter 4. Some preliminary experiments on ImageNet have shown that such separable convolutions can achieve similar or slightly better accuracy at a lower computational cost. This is related to recents works [DSD$^+$13, LCY13] where other forms of factorizations of the first convolutional layers are proposed.

6.2.1 Dense Multidimensional Convolutions

ConvNets, as described in [KSH12, DCM$^+$12, ZF13] consist in a cascade of linear operators $\mathcal{W}_1, \dots, \mathcal{W}_m$ intertwined with non-linearties f. Given an input image x, they compute successive layers $\Phi_m x$ defined as

$$\Phi_m x = f(\mathcal{W}_m f(\mathcal{W}_{m-1} \dots f(\mathcal{W}_1 x))) \quad (6.1)$$

$$= f(\mathcal{W}_m \Phi_{m-1} x). \quad (6.2)$$

A Layer $\Phi_m x$ is indexed by a spatial position u and a depth d that plays a role similar to scattering paths p_m. Let us call N_m, D_m the size and the depth of a layer $\Phi_m x$, which are respectively the number of samples for u, d in this layer . A fully connected linear operator, that is without any convolutional structure, that connect $\Phi_m x$ to $\Phi_{m+1} x$ would require $N_m D_m N_{m+1} D_m$ weights which is impractical for first layers where both N_m and D_m are large. Convolutional networks reduce the number of weights by leveraging the structure of images. They enforce locality by connecting an output position only to the input position which lies within a window of size Q_m centered around the output position. Locality

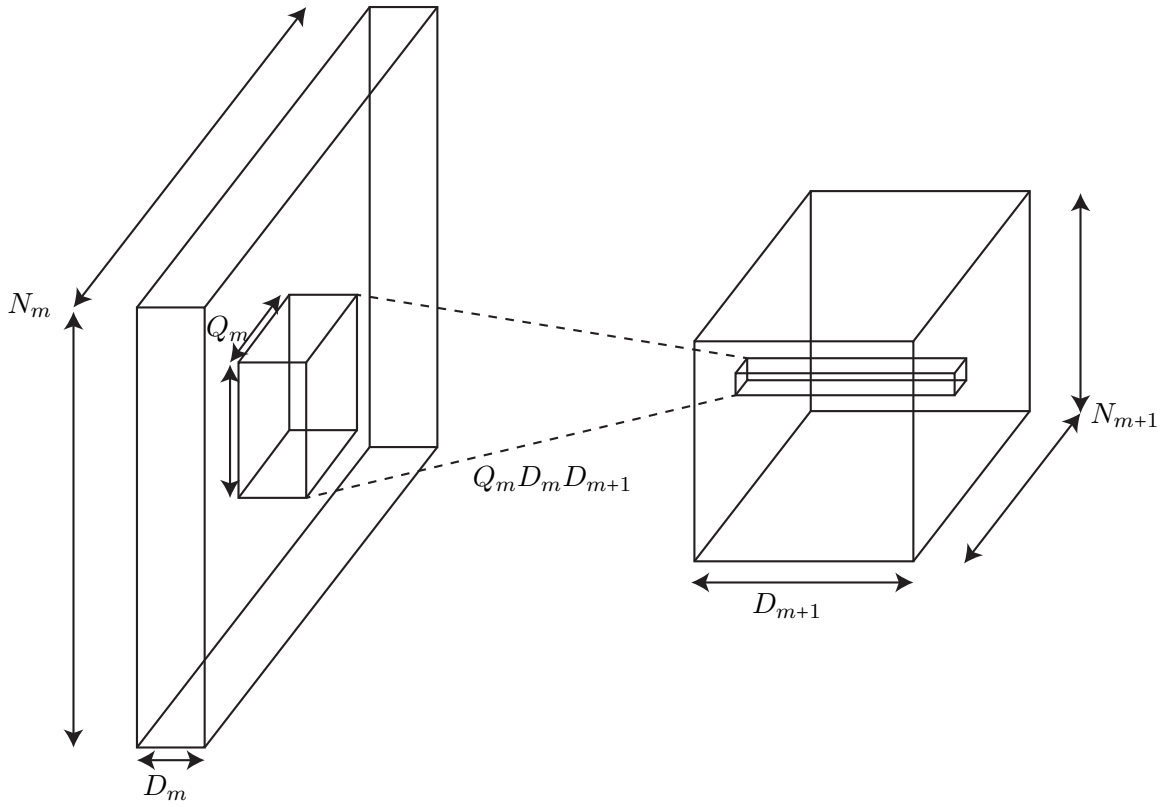


Figure 6.1: Dense multidimensional convolutions in deep networks for image classification [KSH12, DCM+12, ZF13]. A block of $Q_m \times D_m$ values in the input layer is considered as a vector, which is multiplied by a a matrix with $Q_m \times D_m \times D_{m+1}$ weights to generate a vector of dimension D_{m+1} in the output layer. This linear operator is local in the sense that the input block is much smaller than the input image, and convolutional in the sense that the same matrix is applied to every block, independently of their location.

reduces the number of weights to $Q_m D_m N_{m+1} D_{m+1}$, that is by a factor of N_m/Q_m. Q_m is typically equal to 11×11 for the first convolutional layers and 3×3 for the last convolutional layers, whereas D_m typically increases from 3 to 2048. The number of weights is further reduced by convolution, that is by forcing the weights corresponding to windows centered around different output position to be equal. Convolution further lowers the number of weights to $Q_m D_m D_{m+1}$ which then does not depend upon the size of the image. Locality and convolutions are essential to the practicability of back-propagation algorithm, which would otherwise require an enormous amount of processing time and memory to learn the weights. The dense convolution is computed as

$$\mathcal{W}_m \Phi_m x(u, d_{m+1}) = \sum_{\substack{0 \le d_m < D_m \\ v \in Q_m}} w_{v,d_m,d_{m+1}} \Phi_m x(u+v, d_m). \tag{6.3}$$

This dense convolution is illustrated in figure 6.1.

6.2.2 Separable Convolutions

The dense convolutions (6.3) can be interpreted as a sum of two dimensional convolutions with filters $w_{.,d_m,d_{m+1}}$

$$\mathcal{W}_m \Phi_m x(u, d_{m+1}) = \sum_{0 \le d_m < D_m} \Phi_m x(., d_m) \star w_{.,d_m,d_{m+1}}(u). \tag{6.4}$$

Figure 6.2 shows the filters that are learned in the first two layers of a network similar to [KSH12]. One can denote that for a given output depth, these filters are highly redundant along the input depth. This suggest to factorize the matrix of weights $w_{v,d_m,d_{m+1}}$ into purely spatial convolutions with weights w followed by a pointwise matrix multiplication with weights $\mathring{w}$ along the depth variable. A separable multidimensional convolution, as opposed to the dense multidimensional convolution (6.3), is defined as

$$\widetilde{\mathcal{W}}_m \Phi_m x(u, d_{m+1}) = \sum_{\substack{0 \le k_m \le K_m \\ 0 \le d_m}} \mathring{w}_{k_m,d_m,d_{m+1}} \left(w_{.,k_m,d_m} \star \Phi_m x(., d_m)(u) \right) \tag{6.5}$$

The separable multidimensional convolution (6.5) is very similar to the separable group convolution (4.30) introduced in Chapter 4. Each two dimensional signal of $\Phi_m x(., d_m)$ is filtered with K_M two dimensional filters $w_{.,k_m,d_m}$ which play the role of warped filters $y(A'^{-1}.)$ in the separable group convolution (4.30). The output of those first convolutions is an intermediate layer of depth $D_m \times K_m$, therefore we call K_m the depth-multiplier. In the affine wavelet transform (4.40-4.43) of Chapter 4, the depth-multiplier was typically LJ, which was the cardinality of the first wavelet family $\{\phi_J, \psi_{l,j}\}_{0 \le l < L,\ j < J}$. The intermediate layer is then retransformed by multiplying its pointwise vector of dimension $1 \times K_m \times D_m$ with a matrix $\mathring{w}$ of size $D_{m+1} \times (K_m \times D_m)$, which results in a output layer of depth D_{m+1}. This matrix multiplication with $\mathring{w}$ is similar to the filtering with $\mathring{y}$ along A in the

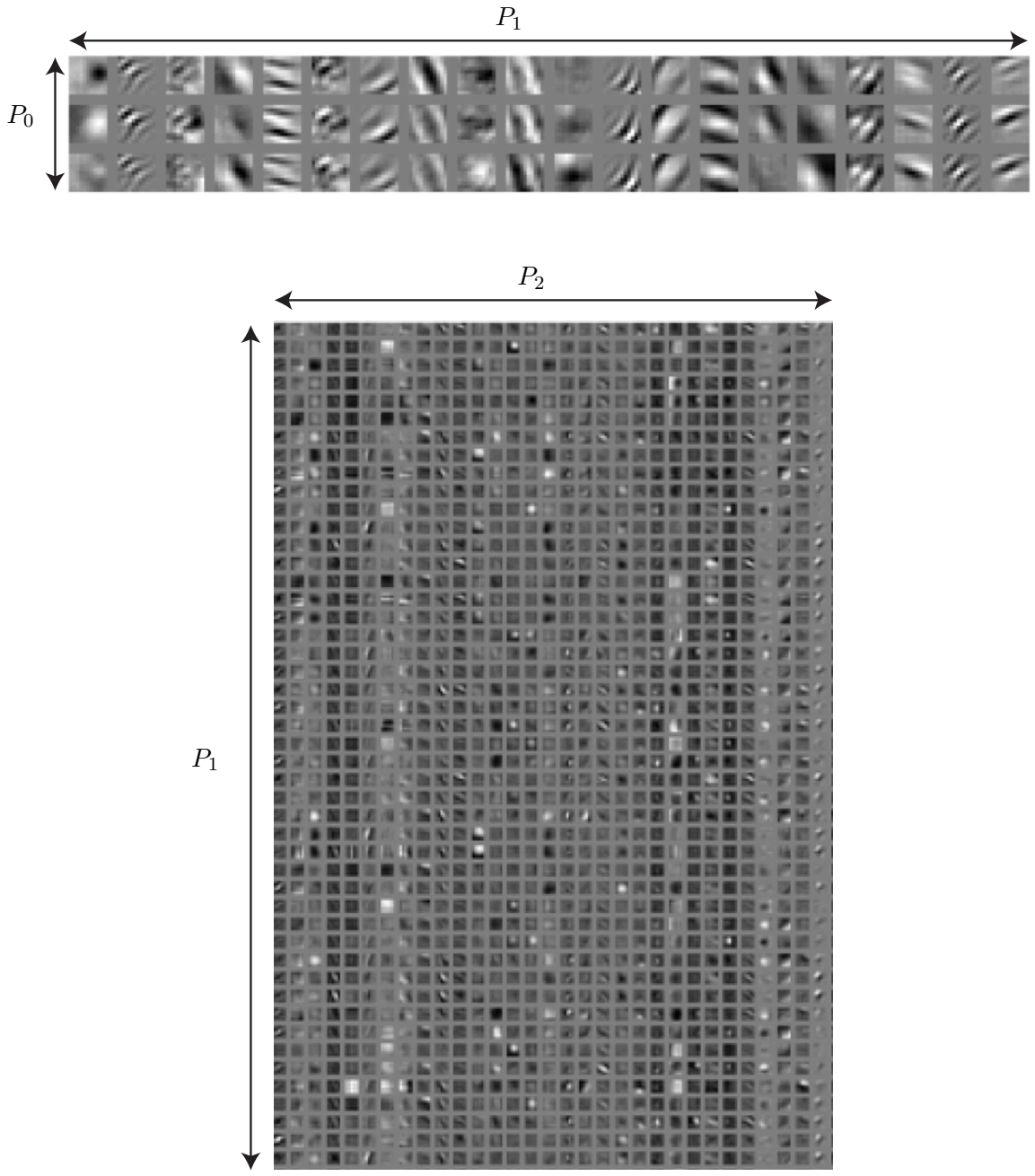


Figure 6.2: A visualization of the learned weights of the first two layers of deep networks similar to [KSH12, DCM+12, ZF13]. Rows corresponds to input depth and columns corresponds to output depth. Computing one output depth of a dense convolution, 6.3 is equivalent to compute a two dimensional convolution of each input depth $\Phi_m(., d_m)$ with the two dimensional filter $w_{.,d_m,d_{m+1}}$ located at rows d_m and column d_{m+1} in these arrays of filters.

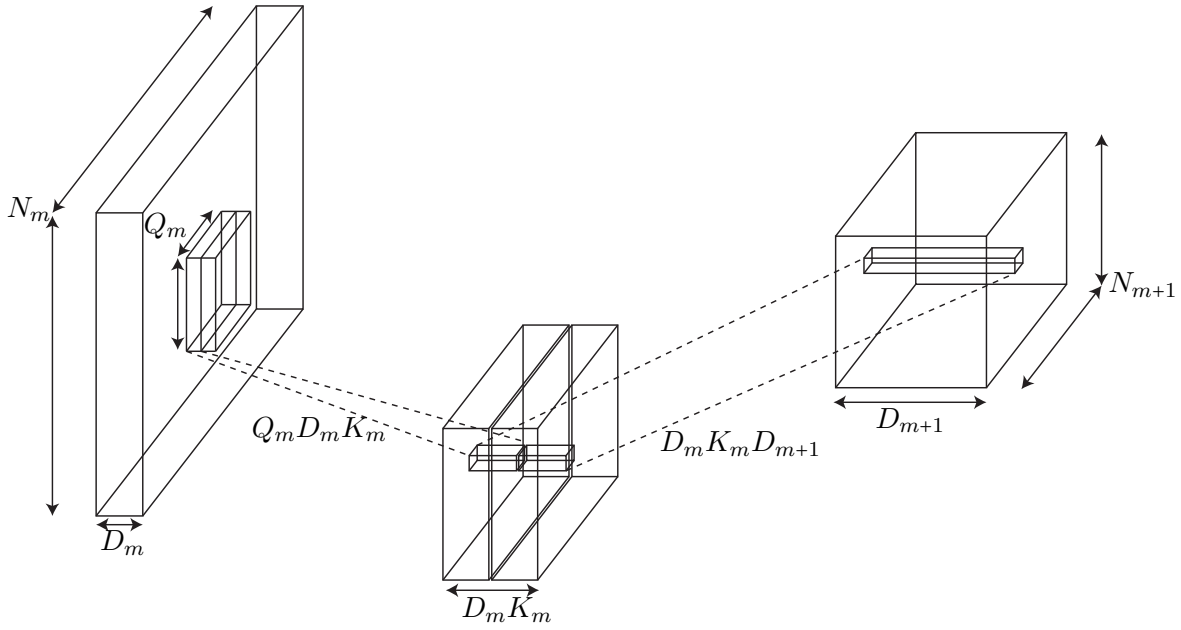


Figure 6.3: The proposed separable multidimensional convolution. First, each input depth is independantly filtered with K_m filters of compact support Q_m, which results in $Q_m D_m K_m$ weights. The result is an intermediate layer of depth $D_m K_m$. At each position u, the vector consisting of all the depths of the intermediate layer is retransformed with a matrix multiplication, which results in $D_m K_m D_{m+1}$ weights and an output layer of depth D_{m+1}.

separable affine convolution (4.30), except that in the separable convolutional layer, the underlying group is unknown and therefore may not have a convolutional structure. Figure 6.3 illustrates the architecture of a separable convolutional layer.

A separable convolutional layer can have the same input and output depth and can therefore act as a replacement of a dense convolutional layer, with less redundancy. The number of free parameters can be adjusted by the depth multiplier K_m, which can be chosen arbitrarily without changing the input and output depth. A separable convolutional layer has $Q_m D_m K_m$weights in its first spatial component and $K_m D_m D_{m+1}$ weights in its second depth component and has therefore a total of $K_m D_m (Q_m + D_{m+1})$ weights. This has to be compared to a dense convolutional layer with $Q_m D_m D_{m+1}$ free parameters. In particular, one can note that the dependency in Q_m, D_{m+1} goes from $Q_m D_{m+1}$ for a dense layer to $Q_m + D_{m+1}$ for a separable layer.

m	D_m	D_{m+1}	Q_m	K_m	#dense param	#separable param	ratio
1	3	96	7×7	4	14112	1740	87%
				8		3480	75%
2	96	256	5×5	4	614k	107k	82%
				8		215k	64%

Table 6.1: Comparison of the number of parameters for the first layers in the network of [ZF13] with dense versus separable convolutions. m is the index of the layer, D_m is the input depth, D_{m+1} is the output depth, Q_m is the spatial window sizes, K_m is the depth multiplier for separable layers. A separable architecture has fewer parameters and therefore requires less computational resources per data. Experiments on ImageNet have shown that a network with separable layers also requires less data to achieve similar or slightly better performance than a network with dense layers with the exact same architecture.

6.2.3 ImageNet Classification with Separable Convolutional Layers

Table 6.1 gives the number of parameters for the network used in [ZF13] which has obtained state-of-the-art results on Imagenet, where the first two layers are implemented with dense or separable convolutional layers. During the second part of my internship at Google, I have replaced convolutional dense layers with separable layers in the distbelief [DCM+12] implementation of the deep network described in [ZF13]. We have trained it on ImageNet ILSVRC2012, which contains 1000 classes, 1.2M training images and 150k testing images of generic objects with clutter. On this dataset, we have recorded that the separable versions required 20% fewer steps to converge to an identical or slightly better final accuracy, compared to the dense version. This suggests that separable convolutions are an appropriate way to factorize dense convolutions without losing expressive power, and that the architecture benefits from this factorization since it then needs less data to achieve the same accuracy. The training and inference time per step was also smaller.

6.3 Object Classification with Joint Scattering

Convolutional networks (ConvNets) such as the one described in [KSH12, DCM+12, ZF13] have obtained state-of-the-art results on all large scale image classification datasets such as ImageNet. It appears that the first layers of a deep network trained on ImageNet also perform very well on smaller datasets such as CalTech101 or CalTech256 [GDDM13, ZF13, DJV+13]. This suggests that these first layers capture some intrinsic properties of real world images.

Other experiments, mainly conducted by Oyallon [OMS14], have shown that a slightly modified version of the joint rigid-motion scattering introduced in Chapter 4 can also obtain results that are competitive with the first layers of a ConvNet trained on ImageNet. This suggest that the first layers of ConvNets capture some basic geometric properties of the

image, and do not necessarily have to be trained. Section 6.3.1 introduces a modified, non-invariant joint scattering and Section 6.3.2 describes the numerical results obtained in [OMS14].

6.3.1 Non-Invariant Joint Scattering

This section describes the unaveraged joint scattering representations that will be applied in Section 6.3.2 to CalTech101 and 256 classification, with comparison with ConvNets.

For a generic object recognition task such as CalTech101 or CalTech256, global rotation or scale invariance are not necessarily desired properties of an image representation. Indeed, these datasets contain objects in a human environment context, in which some objects tend to appear in specific orientations. Discarding those hints by building a rotation invariant is likely to have an adversary effect on performances. Therefore, the joint rigid-motion scattering, as described in chapter 4 may not be a good representation for object recognition because it builds invariance to global rotation. Translation scattering do not build rotational invariance and thus may be a better candidate. Yet, as argued in Section 4.2, translation scattering does not connect the different paths of a layer and therefore fails to capture the joint variability of different paths within a layer. Thus translation scattering may not be discriminant enough. This is corroborated by the fact that most state-of-the-art ConvNets [KSH12, DCM+12, ZF13] contain dense convolutional layers such as described in Section 6.2.1, that do connect the different paths within a layer.

Similarly, the rigid-motion wavelet transform of Section 4.3.2 also connects different orientations within the orbit of a layer through convolutions with rigid-motion wavelets $\tilde{\psi}_{\theta_2,j_2,k_2}(u,\theta) = \psi_{\theta_2,j_2}(u)\mathring{\psi}_{k_2}(\theta)$. The implementation of rigid-convolutions with separable wavelets (4.71) is subtle and necessitates to rotate the spatial part $\psi_{\theta_2+\theta,j_2}$ of the wavelet along the orbit. This was designed to maintain covariance with rigid-motion within an orbit, a property which is no longer needed when we do not seek to build a rotational invariant. Therefore, we propose a simplified version of the joint scattering where we replace the group convolution $\circledast$ (4.27) on an affine subgroup $\mathbb{R}^2 \rtimes G$ with a simple, separable convolution $\star$ on the product space $\mathbb{R}^2 \times G$. For two functions $\overset{\times}{x}, \overset{\times}{y} \in \mathbf{L}^2(\mathbb{R}^2 \times G)$, their separable convolution $\overset{\times}{\star}$ is defined as

$$\overset{\times}{y} \overset{\times}{\star} \overset{\times}{x}(v,A) = \int_{\substack{v' \in \mathbb{R}^2 \\ A' \in G}} \overset{\times}{x}(v',A')\overset{\times}{y}(v-v', A'^{-1}A) d\overset{\times}{\mu}(v',A'). \tag{6.6}$$

as opposed to (4.27) where the linear group variable A' also acts on the spatial variable, i.e. where $\overset{\times}{y}(v-v', A'^{-1}A)$ is replaced with $\overset{\times}{y}(A'^{-1}(v-v'), A'^{-1}A)$. Assuming the measure is separable $d\overset{\times}{\mu}(v,A) = d\mu(v)d\mathring{\mu}(A)$, the separable convolution with a separable filter $\overset{\times}{y}(v,A) = y(v)\mathring{y}(A)$ can be implemented by either a convolution in G followed by a convo-

lution in $\mathbb{R}^2$ or the other way around

$$\overset{\times}{y} \overset{\times}{\star} \overset{\times}{x}(v,A) = \int_{v' \in \mathbb{R}^2} \left(\int_{A' \in G} \overset{\times}{x}(v',A') \mathring{y}(A'^{-1}A) d\mathring{\mu}(A') \right) y(v-v') d\mu(v') \quad (6.7)$$

$$= \int_{A' \in G} \left(\int_{v' \in \mathbb{R}^2} \overset{\times}{x}(v',A') y(v-v') d\mu(v') \right) \mathring{y}(A'^{-1}A) d\mathring{\mu}(A'). \quad (6.8)$$

A non-group invariant joint wavelet family $\{\overset{\times}{\phi}_J, \overset{\times}{\psi}_{\theta,j,k}\}_{\theta,j,k}$ is defined as

$$\overset{\times}{\phi}_J(v,A) = \phi_J(v)\delta(A) \quad (6.9)$$

$$\overset{\times}{\psi}_{l,j,k}(v,A) = \psi_{\theta,j}(v)\mathring{\psi}_k(A) \quad (6.10)$$

$$\overset{\times}{\psi}_{l,j,K}(v,A) = \psi_{\theta,j}(v)\mathring{\phi}_K(A). \quad (6.11)$$

Since a convolution with a dirac δ does not modify a signal, the convolution with the window $\overset{\times}{\phi}_J$ is equivalent to a simple spatial convolution of each component of $\overset{\times}{x}$ with ϕ_J, without any smoothing occurring along the linear group variable A

$$\overset{\times}{x} \overset{\times}{\star} \overset{\times}{\phi}_J(v,A) = \left(\overset{\times}{x}(.,A) \star \phi_J \right)(v). \quad (6.12)$$

As it is the case for the joint wavelet family with Theorem 2, if the two wavelet families, of which the non-invariant separable family is the product, are frame, then the following theorem states that the non-invariant separable family is also a frame.

Theorem 4. *If $\{\phi_J, \psi_{\theta,j}\}_{\theta,j}$ and $\{\mathring{\phi}_K, \mathring{\psi}_k\}_k$ are respectively ϵ and $\mathring{\epsilon}$ frame as defined in (4.44-4.45) then the wavelet family $\{\overset{\times}{\phi}_J, \overset{\times}{\psi}_{l,j,k}\}_{l,j,k}$ defined in (6.9-6.11) is a $\overset{\times}{\epsilon}$ frame of $SE(2) = \mathbb{R}^2 \rtimes SO(2)$, i.e. for all $\overset{\times}{x} \in \mathbf{L}^2(\mathbb{R}^2 \rtimes G)$,*

$$(1-\overset{\times}{\epsilon})\|\overset{\times}{x}\|^2 \leq \|\overset{\times}{x} \overset{\times}{\star} \overset{\times}{\phi}_{J,K}\|^2 + \sum_{l,j,k} \|\overset{\times}{x} \overset{\times}{\star} \overset{\times}{\psi}_{l,j,k}\|^2 \leq \|\overset{\times}{x}\|^2 \quad (6.13)$$

where

$$1-\overset{\times}{\epsilon} = (1-\epsilon)(1-\mathring{\epsilon}) \quad (6.14)$$

Proof. As opposed to the separable joint wavelet family (4.40-4.43), the non-invariant joint wavelet family contains only 3 different types of wavelets. The wavelet $\phi_J(u)\mathring{\psi}_k(A)$ is not present because it would be redundant with $\overset{\times}{\phi}_J(u,A) = \phi_J(u)\delta(A)$ which does not smooth along A and therefore captures all frequencies along A. Similarly to the proof of Theorem 2, we factorize $\overset{\times}{\mathcal{W}}$ in

$$\overset{\times}{\mathcal{W}} = \mathring{\mathcal{W}}\mathcal{W} \quad (6.15)$$

where

$$\begin{aligned} \mathcal{W} : \mathbf{L}^2(\mathbb{R}^2 \times G) &\rightarrow \mathbf{L}^2(\mathbb{R}^2 \times G \times \Lambda) && (6.16) \\ \overset{\times}{x} &\mapsto \Big((v, A, \lambda) \mapsto \overset{\times}{x}(., A) \star \psi_\lambda(v) \Big) && (6.17) \end{aligned}$$

and

$$\begin{aligned} \mathring{\mathcal{W}} : \mathbf{L}^2(\mathbb{R}^2 \times G \times \Lambda) &\rightarrow \mathbf{L}^2(\mathbb{R}^2 \times G + \mathbb{R}^2 \times G \times \{\Lambda - J\} \times \mathring{\Lambda}) && (6.18) \\ \overset{\times}{y} &\mapsto \big\{ \overset{\times}{y}(., ., J), && (6.19) \\ & \qquad (v, A, \lambda, \mathring{\lambda}) \mapsto \overset{\times}{y}(v, ., \lambda) \circledast \psi_{\mathring{\lambda}}(A) \big\}. && (6.20) \end{aligned}$$

The second operator applies the identity to the slice $\lambda = J$ and a wavelet transform along A to all other slices $\lambda \neq J$ of $\overset{\times}{y} \in \mathbf{L}^2(\mathbb{R}^2 \times G \times \Lambda)$. Since the identity is a 0 frame operator and the wavelet transform along A is a $\mathring{\epsilon}$ operator, one can verify that $\mathring{\mathcal{W}}$ is also an $\mathring{\epsilon}$ frame operator. Since $\mathcal{W}$ is an ϵ frame operator, applying the frame bound inequality to both $\mathcal{W}$ and $\mathring{\mathcal{W}}$ in (6.15) finishes the proof. □

The non-invariant joint scattering is defined as the joint scattering, but by replacing the joint wavelet transform with non-invariant wavelet transform. We only compute wavelets along the spatial u and the orientation variable θ, that is that the linear subgroup that we are considering is the group of rotations $G = SO(2)$. We start by computing a first spatial wavelet modulus transform

$$\begin{aligned} S_0 x(v) &= x \star \phi_J(v) && (6.21) \\ U_1 x(v, \theta, j) &= |x \star \psi_{\theta, j}|(v). && (6.22) \end{aligned}$$

Then the first internal layer $U_1 x$ is retransformed with a non-invariant joint wavelet modulus operator along each orbit

$$\begin{aligned} S_1 x(v, \theta, j) &= U_1 x(., ., j) \circledast \overset{\times}{\phi}_J(v, \theta) && (6.23) \\ &= U_1 x(., \theta, j) \star \phi_J(v) && (6.24) \\ \overset{\times}{U}_2 x(v, \theta_1, j_1, \theta_2, j_2, k_2) &= |U_1 x(., ., j_1) \star \overset{\times}{\psi}_{\theta_2, j_2, k_2}|(v, \theta_1). && (6.25) \end{aligned}$$

As in all scattering networks, $U_2 x$ is iteratively retransformed into $S_2 x, U_3 x$. In practice, we do not compute $U_3 x$ because it has negligible energy. Yet, we do compute $S_2 x$ with a spatial windowing

$$\begin{aligned} \overset{\times}{S}_2 x(v, \theta_1, j_1, \theta_2, j_2, k_2) &= \overset{\times}{U}_2 x(., ., j_1, \theta_2, j_2, k_2) \overset{\times}{\star} \overset{\times}{\phi}_J(v, \theta_1) && (6.26) \\ &= \overset{\times}{U}_2 x(., \theta_1, j, \theta_2, j_2, k_2) \star \phi_J(v) && (6.27) \end{aligned}$$

which simply corresponds to an average along v. In the classification experiments of Section 6.3.2 we concatenate the non-invariant joint scattering of order $0, 1, 2$ into a single vector

$$\overset{\times}{S}x = \{S_0 x, S_1 x, \overset{\times}{S}_2 x\}. \tag{6.28}$$

As the joint scattering $\widetilde{S}x$ introduced in Chapter 4, and unlike the translation and separable scattering Sx and $\mathring{S}x$ of Chapter 2 and 3, the non-invariant joint scattering $\overset{\times}{S}x$ connects the different paths of internal layers through convolutions with multidimensional wavelets. Yet it is only invariant to translations up to 2^J but has no invariance to the linear transform group $GL(\mathbb{R}^2)$. The invariance to transformations is delegated to the classifier which will hopefully learn the appropriate level of invariance for each transformation, given enough data.

6.3.2 Caltech101, 256 Classification with Non-Invariant Joint Scattering

The classification performance of the non-invariant joint scattering $\overset{\times}{S}x$ is evaluated on the CalTech101 and 256 databases. CalTech101 has 102 classes of different objects, each containing 31 or more images per class with 10k images. CalTech256 is similar but larger with 256 classes and 30k images. All images are first resized to 128×128 by a linear rescaling. The scattering of each YUV channel of each image are computed separately and are concatenated. The first wavelet transform $\mathcal{W}_1$ is computed with Morlet wavelets of Section 2.3.2, over 5 scales $0 \leq 2^{j1} < 2^5$ with $C = 8$ angles θ_1. The second wavelet transform $\overset{\times}{\mathcal{W}}_2$ is computed with the same two dimensional and one dimensional wavelet family as the one described in Section 4.3.2. The final scattering coefficients $\overset{\times}{S}x$ are computed with a spatial pooling at a scale $2^J = 32$, as opposed to the maxima selection used in most convolution networks. Scattering coefficients $\overset{\times}{S}x$ are renormalized by a standardization which subtracts their mean and sets their variance to 1. The mean and variance are computed on the training databases. Standardized scattering coefficients are then provided to a linear SVM classifier.

Almost state of the art classification results are obtained on Caltech-101 and Caltech-256, with a ConvNet [ZF13] pretrained on ImageNet. Table 6.2 shows that with 7 layers a ConvNet can achieve 85.5% accuracy on Caltech101 and 72.6% accuracy on Caltech256, using respectively 30 and 60 training images. The classification is performed with a linear SVM. With only two layers, the ConvNet performances drop to 66.2% on Caltech101 and 39.6% on Caltech256. A non-invariant joint scattering achieves similar performances without any learning except at the renormalization and classifier stages. The similarity between those results suggests that the first layers of the deep networks used in [KSH12, DCM+12, ZF13] are responsible for basic geometric properties images and do not necessarily have to be trained.

Representation	Layers	CalTech101	CalTech256
Joint Scattering	1	51.2 ± 0.8	19.3 ± 0.2
ConvNet pretrained on ImageNet [ZF13]	1	44.8 ± 0.7	24.6 ± 0.4
Joint Scattering	2	68.8 ± 0.5	34.6 ± 0.2
ConvNet pretrained on ImageNet [ZF13]	2	66.2 ± 0.5	39.6 ± 0.3
ConvNet pretrained on ImageNet [ZF13]	3	72.3 ± 0.4	46.0 ± 0.3
ConvNet pretrained on ImageNet [ZF13]	7	85.5 ± 0.4	72.6 ± 0.2

Table 6.2: Classification accuracy on CalTech101 with 30 training images per class and on CalTech256 with 60 images per class, for different number of layers and different architecture. For one or two layers, the joint scattering performs similarly to the ConvNet of [ZF13] pre trained on ImageNet.

Summary of Notations

Groups

- $\mathbb{R}^2$ the set of all real position in the two dimensional plane.
- $u, v, w \in \mathbb{R}^2$ two dimensional positions.
- $GL(\mathbb{R}^2)$ the linear group, consisting of all 2×2 real invertible matrices.
- $A \in GL(\mathbb{R}^2)$ invertible matrix.
- $\mathbb{R}^2 \rtimes GL(\mathbb{R}^2)$ the affine group, endowed with its semi-separable product law.
- $g = (v, A) \in \mathbb{R}^2 \rtimes GL(\mathbb{R}^2)$ element of the affine group.
- $G \subset GL(\mathbb{R}^2)$ a subgroup of the linear group.
- $\mathbb{R}^2 \rtimes G \subset \mathbb{R}^2 \rtimes GL(\mathbb{R}^2)$ a subgroup of the affine group.
- $SO(2)$ the group of two dimensional rotations.
- $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ the rigid-motion group.

Fourier and Wavelets

- $x, y, z \in \mathbf{L}^2(\mathbb{R}^2)$ squared integrable two dimensional functions.
- $\mathcal{F}$ the Fourier transform.
- $\omega \in \mathbb{R}^2$ a frequency.
- ϕ scaling function, low-pass window.
- ψ mother, high-pass wavelet.
- $\theta \in [0, 2\pi)$ orientation.

- j index of the scale 2^j.
- $\lambda = (\theta, j)$ a compact index of a wavelet.
- J index of a maximum scale 2^J.
- ϕ_J low-pass at scale 2^J.
- $\psi_{\theta,j}$ oriented high pass at orientation θ and scale j.
- $\star$ the convolution of two functions in $\mathbf{L}^2(\mathbb{R}^2)$.
- $\mathcal{W}$ the wavelet transform.

Scattering

- m scattering order, index of a scattering layer, number of high pass modulus operations necessary to compute a given coefficient.
- M maximum scattering order.
- $U_m x$ the internal scattering layer of order m.
- $S_m x$ the output scattering layer of order m.
- Sx the scattering vector, consisting of all $S_m x$.
- $p_m = (\theta_1, j_1, \ldots, \theta_m, j_m)$ a scattering path of order m, an sequence of indices of wavelet.
- $\mathfrak{P}_m$ the set of all possible scattering paths.

Separable, Joint, Non-invariant Joint Scattering

- k index of a one dimensional wavelet.
- K index of maximum scale 2^K.
- $\mathring{.}$ any object related to separable scattering of Chapter 3.
- $\tilde{.}$ any object related to joint scattering of Chapter 4.
- $\overset{\times}{.}$ any object related to non-invariant joint scattering of Section 6.3.

Bibliography

[AMV99] J.-P. Antoine, R. Murenzi, and P. Vandergheynst. Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Applied and Computational Harmonic Analysis*, 6(3):314–345, 1999.

[BDGR12] Ugo Boscain, Jean Duplaix, Jean-Paul Gauthier, and Francesco Rossi. Anthropomorphic image reconstruction via hypoelliptic diffusion. *SIAM J. Control and Optimization*, 50(3):1309–1336, 2012.

[BM13] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[BN84] Richard Barakat and Garry Newsam. Necessary conditions for a unique solution to two dimensional phase recovery. *Journal of Mathematical Physics*, 25(11):3190–3193, 1984.

[BRB+11] Y-Lan Boureau, Nicolas Le Roux, Francis Bach, Jean Ponce, and Yann LeCun. Ask the locals: Multi-way local pooling for image recognition. *International Conference on Computer Vision*, pages 2651–2658, 2011.

[CG10] M. Crosier and L.D. Griffin. Using basic image features for texture classification. *International Journal of Computer Vision*, 88(3):447–460, 2010.

[CS06] G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *Journal of Mathematical Imaging and Vision*, 24(3):307–326, 2006.

[DB07] Remco Duits and Bernhard Burgeth. Scale space and variational methods in computer vision. *Scale Space and Variational Methods in Computer Vision, Springer Lecture Notes in Computer Science*, 4485:300–312, 2007.

[DCM+12] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. Advances in neural information processing systems. *Advances in Neural Information Processing Systems*, 2012.

[DF11] Remco Duits and Erik Franken. Left-invariant diffusions on the space of positions and orientations and their application to crossing-preserving smoothing of hardi images. *International Journal of Computer Vision*, 92(3):231–264, 2011.

[DJV+13] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv.org*, (1310.1531), 2013.

[DKSZ11] L. Donoho, G. Kutyniok, M. Shahram, and X. Zhuang. A rational design of discrete shearlet transform. *Proc. of SampTA'11 (Singapore)*, 2011.

[DSD+13] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. Predicting parameters in deep learning. *arXiv.org*, (1306.0543), 2013.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:886–893, 2005.

[FS89] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, 1989.

[GDDM13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv.org*, (1311.2524v3), 2013.

[HCFE04] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the significance of real-world conditions for material classification. *Proceedings of European Conference on Computer Vision*, 3024:253–266, 2004.

[HGFB11] Nguyen Huu-Giao, R. Fablet, and J.-M. Boucher. Visual textures s realizations of multivariate log-gaussian cox processes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2945–2952, 2011.

[HS06] G.E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Jul81] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, (290):91 – 97, 1981.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, 1989.

[LCY13] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv.org*, (1312.4400), 2013.

[LFKZ11] Li Liu, P. Fieguth, Gangyao Kuang, and Hongbin Zha. Sorted random projections for robust texture classification. *Proceedings of International Conference on Computer Vision*, pages 391–398, 2011.

[LKF10] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

[LLB+98] Yann LeCun, Leon, Bottou, Genevieve B. Orr, and Klaus Robert Müller. Efficient backprop. *Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, Springer Berlin Heidelberg*, 1524:9–50, 1998.

[LM01] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LSP05] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:2169–2178, 2006.

[LW03] Xiuwen Liu and DeLiang Wang. Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, 12(6):661–670, 2003.

[Mal08] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3 edition, 2008.

[Mal12] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[MY09] Jean-Michel Morel, , and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469, 2009.

[OMP+02] Timo Ojala, Topi Mäenpää, Matti Pietikäinen, Jaakko Viertola, Juha Kyllönen, and Sami Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. *Proceedings of International Conference on Pattern Recognition*, 1:701–706, 2002.

[OMS14] Edouard Oyallon, Stéphane Mallat, and Laurent Sifre. Generic deep networks with wavelet scattering. *submitted to International Conference on Learning Representations*, 2014.

[OPM02] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[ORH08] Ville Ojansivu, Esa Rahtu, and Janne Heikkilä. Rotation invariant local phase quantization for blur insensitive texture analysis. *Proceedings of International Conference on Pattern Recognition*, pages 1–4, 2008.

[RP99] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(1019–1025), 1999.

[SLRA13] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H. Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision*, 103(3):348–371, 2013.

[SM12] Laurent Sifre and Stéphane Mallat. Combined scattering for rotation invariant texture. *Proceedings of European Symposium on Artificial Neural Networks*, 2012.

[SM13] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1233–1240, 2013.

[SM14] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for texture classification. *submitted to International Journal of Computer Vision*, 2014.

[SRA09] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8), 2009.

[TLF10] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[VZ05] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1–2):61–81, 2005.

[XJF09] Yong Xu, Hui Ji, and Cornelia Fermuller. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, 83(1):85–100, 2009.

[XYLJ10] Yong Xu, Xiong Yang, Haibin Ling, and Hui Ji. A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 161–168, 2010.

[ZAMP12] Guoying Zhao, Timo Ahonen, Jiří Matas, and Matti Pietikäinen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, 21(4):1465–1477, 2012.

[ZF13] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *arXiv.org*, (1311.2901), 2013.

[ZYZH10] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S Huang. Image classification using super-vector coding of local image descriptors. *European Conference on Computer Vision*, 6315:141–154, 2010.