Available online at www.sciencedirect.com
ScienceDirect
Applied and Computational Harmonic Analysis 00 (2014) 1–20
Procedia Computer Science

Rigid-Motion Scattering for Texture Classification☆

Laurent Sifre[a], Stéphane Mallat[b]

[a] *CMAP Ecole Polytechnique*
Route de Saclay, 91128 Palaiseau France
laurent.sifre@gmail.com
[b] *Département d'informatique*
École normale supérieure
45 rue d'Ulm F-75230 Paris Cedex 05 France

Abstract

A rigid-motion scattering computes adaptive invariants along translations and rotations, with a deep convolutional network. Convolutions are calculated on the rigid-motion group, with wavelets defined on the translation and rotation variables. It preserves joint rotation and translation information, while providing global invariants at any desired scale. Texture classification is studied, through the characterization of stationary processes from a single realization. State-of-the-art results are obtained on multiple texture databases, with important rotation and scaling variabilities.

Keywords: Deep network, scattering, wavelet, rigid-motion, texture, classification

1. Introduction

Image classification requires to find representations which reduce non-informative intra-class variability, and hence which are partly invariant, while preserving discriminative information across classes. Deep neural networks build hierarchical invariant representations by applying a succession of linear and non-linear operators which are learned from training data. They provide state of the art results for complex image classifications tasks [1, 2, 4, 5, 6]. A major issue is to understand the properties of these networks, what needs to be learned and what is generic and common to most image classification problems. Translations, rotations and scaling are common sources of variability for most images, because of changes of view points and perspective projections of three dimensional surfaces. Building adaptive invariants to such transformation is usually considered as a first necessary steps for classification [3]. We concentrate on this generic part, which is adapted to the physical properties of the imaging environment, as opposed to the specific content of images which needs to be learned.

This paper defines deep convolution scattering networks which can provide invariant to translations and rotations, and hence to rigid motions in $\mathbb{R}^2$. The level of invariance is adapted to the classification task. Scattering transforms have been introduced to build translation invariant representations, which are stable to deformations [8], with applications to image classification [7]. They are implemented as a convolutional network, with successive spatial wavelet convolutions at each layer. Translations is a simple commutative group, parameterized by the location of the input pixels. Rigid-motions is a non-commutative group whose parameters are not explicitly given by the input image, which raises new issues. The first one is to understand how to represent the joint information between translations

☆Work supported by ANR 10-BLAN-0126 and Advanced ERC InvariantClass 320959

and rotations. We shall explain why separating both variables leads to important loss of information and yields representations which are not sufficiently discriminative. This leads to the construction of a scattering transform on the full rigid-motion group, with rigid-motion convolutions on the joint rotation and translation variables. Rotations variables are explicitly introduced in the second network layer, where convolutions are performed on the rigid-motion group along the joint translation and rotation variables. As opposed to translation scattering where linear transforms are performed along spatial variables only, rigid-motion scattering recombines the new variables created at the second network layer, which is usually done in deep neural networks. However, a rigid-motion scattering involves no learning since convolutions are computed with predefined wavelets along spatial and rotation variables. The stability is guaranteed by its contraction properties, which are explained.

We study applications of rigid-motion scattering to texture classification, where translations, rotations and scaling are major sources of variability. Image textures can be modeled as stationary processes, which are typically non Gaussian and non Markovian, with long range dependencies. Texture recognition is a fundamental problem of visual perception, with applications to medical, satellite imaging, material recognition [20, 25, 26], object or scene recognition [33]. Recognition is performed from a single image, and hence can not involve high order moments, because their estimators have a variance which is too large. Finding a low-variance ergodic representation, which can discriminate these non-Gaussian stationary processes, is a fundamental probability and statistical issue.

Translation invariant scattering representation of stationary processes have been studied to discriminate texture which do not involve important rotation or scaling variability [7, 11]. These results are extended to joint translation and rotation invariance. Invariance to scaling variability is incorporated through linear projectors. It provides effective invariants, which yield state of the art classification results on a large range of texture data bases.

Section 2 reviews the construction of translation invariant scattering transforms. Section 2.4 explains why invariants to rigid motion can not be computed by separating the translation and rotation variables, without losing important information. Joint translation and rotation operators defines a rigid motion group, also called special Euclidean group. Rigid-motion scattering transforms are studied in Section 3. Convolutions on the rigid-motion group are introduced in Section 3.1 in order to define wavelet tranforms over this group. Their properties are described in Section 3.2. A rigid-motion scattering iteratively computes the modulus of such wavelet transforms. The wavelet transforms jointly process translations and rotations, but can be computed with separable convolutions along spatial and rotation variables. A fast filter bank implementation is described in Section 4, with a cascade of spatial convolutions and downsampling. Invariant scattering representations are applied to image texture classification in Section 5. State of the art results on four texture datasets containing different types and ranges of variability [34, 35, 36, 37]. All numerical experiments are reproducible with the ScatNet [38] MATLAB toolbox.

2. Invariance to Translations, Rotations and Deformations

Section 2.1 reviews the property of translation invariant representations and their stability relatively to deformations. The use of wavelet transform is justified because of their stability to deformations. Their properties are summarized in Section 2.2. Section 2.3 describes translation scattering transforms, implemented with a deep convolutional network. Separable extensions to translation and rotation invariance is discussed in Section 2.4. It is shown that this simple strategy leads to an important loss of information.

2.1. Translation Invariance and Deformation Stability

Building invariants to translations and small deformations is a prototypical representation issue for classification, which carries major ingredients that makes this problem difficult. Translation invariance is simple to compute. There are many possible strategies that we briefly review. The main difficulty is to build a representation $\Phi(x)$ which is also stable to deformations.

A representation $\Phi(x)$ is said to be translation invariant if $x_v(u) = x(u-v)$ has the same representation

$$\forall v \in \mathbb{R}^2 \ , \ \Phi(x) = \Phi(x_v) \ .$$

Besides translation invariance, it is often necessary to build invariants to any specific class of deformations through linear projectors. Invariant to translation can be computed with a registration $\Phi x(u) = x(u - a(x))$ where $a(x)$ is an anchor point which is translated when x is translated. It means that if $x_v(u) = x(u-v)$ then $a(x_v) = a(x) + v$. For

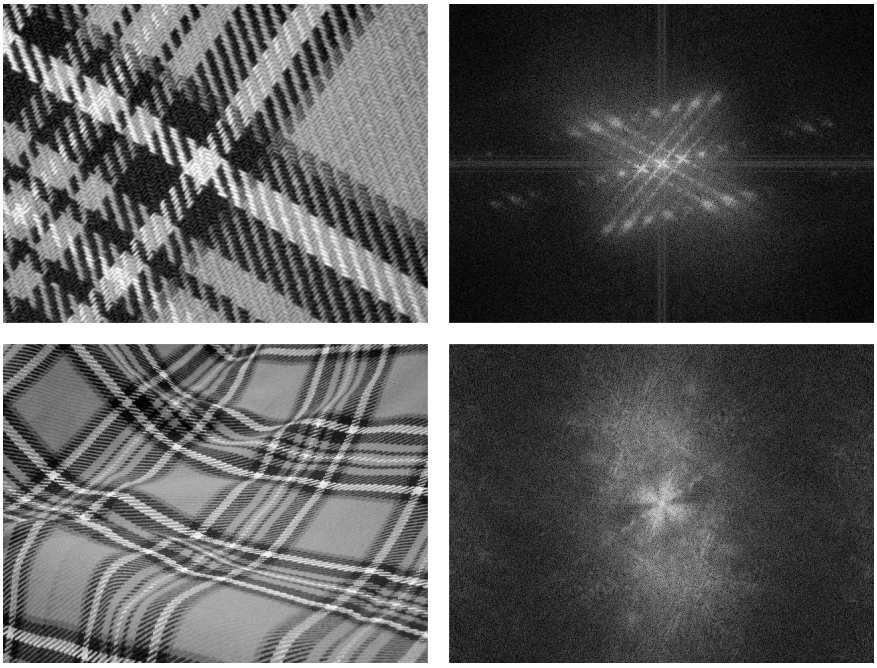


Figure 1: Two images of the same texture (left) from the UIUCTex dataset [35] and the log of their modulus of Fourier transform (right). The periodic patterns of the texture corresponds to fine grained dots on the Fourier plane. When the texture is deformed, the dots spread on the Fourier plane, which illustrates the fact that modulus of Fourier transform is unstable to elastic deformation.

example, $a(x) = \arg\max_u |x \star h(u)|$, for some filter $h(u)$. These invariants are simple and preserve as much information as possible. The Fourier transform modulus $|\hat{x}(\omega)|$ is also invariant to translation.

Invariance to translations is often not enough. Suppose that x is not just translated but also deformed to give $x_\tau(u) = x(u - \tau(u))$ with $|\nabla\tau(u)| < 1$. Deformations belong to the infinite dimensional group of diffeomorphisms. Computing invariants to deformations would mean losing too much information. In a digit classification problem, a deformation invariant representation would confuse a 1 with a 7. We then do not want to be invariant to any deformations, but only to the specific deformations within the digit class, while preserving information to discriminate different classes. Such deformation invariants need to be learned as an optimized linear combinations.

Constructing such linear invariants requires the representation to be stable to deformations. A representation $\Phi(x)$ is stable to deformations if $\|\Phi(x) - \Phi(x_\tau)\|$ is small when the deformation is small. The deformation size is measured by $\|\nabla\tau\|_\infty = \sup_u |\nabla\tau(u)|$. If this quantity vanishes then τ is a "pure" translation without deformation. Stability is formally defined as Lipschitz continuity relatively to this metric. It means that there exists $C > 0$ such that for all $x(u)$ and $\tau(u)$ with $\|\nabla\tau\|_\infty < 1$

$$\|\Phi(x) - \Phi(x_\tau)\| \leq C \|\nabla\tau\|_\infty \|x\| \,. \tag{1}$$

This Lipschitz continuity property implies that deformations are locally linearized by the representation Φ. Indeed, Lipschitz continuous operators are almost everywhere differentiable in the sense of Gateau. It results that $\Phi(x) - \Phi(x_\tau)$ can be approximated by a linear operator of $\nabla\tau$ if $\|\nabla\tau\|_\infty$ is small. A family of small deformations thus generates a linear space $\mathrm{span}_\tau(\Phi(x_\tau))$. In the transformed space, an invariant to these deformations can then be computed with a linear projector on the orthogonal complement $\mathrm{span}_\tau(\Phi(x_\tau))^\perp$.

Registration invariants are not stable to deformations. If $x(u) = 1_{[0,1]^2}(u) + 1_{[\alpha,\alpha+1]^2}(u)$ then for $\tau(u) = \varepsilon u$ one can verify that $\|x - x_\tau\| \geq 1$ if $|\alpha| > \varepsilon^{-1}$. It results that (1) is not valid. One can similarly prove that the Fourier transform modulus $\Phi(x) = |\hat{x}|$ is not stable to deformations because high frequencies move too much with deformations as can be seen on figure 1.

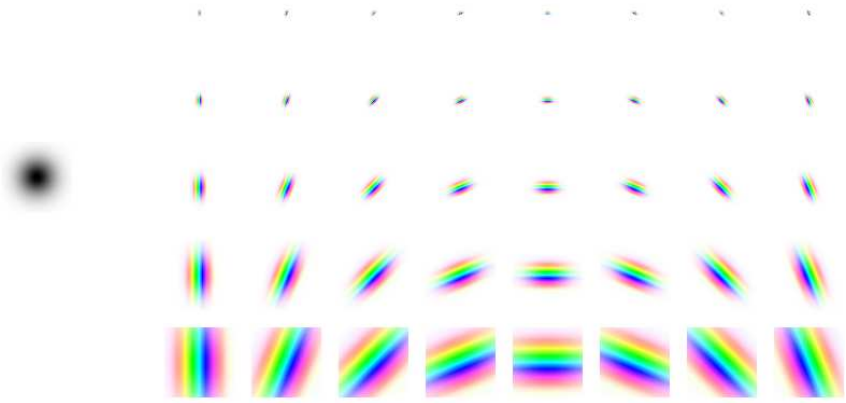


Figure 2: The gaussian window ϕ_J (left) and oriented and dilated Morlet wavelets $\psi_{\theta,j}$ (right). Saturation corresponds to amplitude while color corresponds to complex phase.

Translation invariance often needs to be computed locally. Translation invariant descriptors which are stable to deformations can be obtained by averaging. If translation invariant is only needed within a limited range smaller than 2^J then it is sufficient to average x with a smooth window $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$ of width 2^J:

$$x \star \phi_J(u) = \int x(v)\phi_J(u-v)\,dv. \tag{2}$$

It is proved in [8] that if $\|\nabla\phi\|_1 < +\infty$ and $\||u|\nabla\phi(u)\|_1 < +\infty$ and $\|\nabla\tau\|_\infty \leq 1-\varepsilon$ with $\varepsilon > 0$ then there exists C such that

$$\|x_\tau \star \phi_J - x \star \phi_J\| \leq C\,\|x\|\left(2^{-J}\|\tau\|_\infty + \|\nabla\tau\|_\infty\right). \tag{3}$$

Averaging operators lose all high frequencies, and hence eliminate most signal information. These high frequencies can be recovered with a wavelet transform.

2.2. Wavelet Transform Invariants

Contrarily to sinusoidal waves, wavelets are localized functions which are stable to deformations. They are thus well adapted to construct translation invariants which are stable to deformations. We briefly review wavelet transforms and their applications in computer vision. Wavelet transform has been used to analyze stationary processes and image textures. They provide a set of coefficients closely related to the power spectrum.

A directional wavelet transform extracts the signal high-frequencies within different frequency bands and orientations. Two-dimensional directional wavelets are obtained by scaling and rotating a single band-pass filter ψ. Multiscale directional wavelet filters are defined for any $j \in \mathbb{Z}$ and rotation r_θ of angle $\theta \in [0, 2\pi]$ by

$$\psi_{\theta,j}(u) = 2^{-2j}\psi(2^{-j}r_{-\theta}u)\,. \tag{4}$$

If the Fourier transform $\hat{\psi}(\omega)$ is centered at a frequency η then $\hat{\psi}_{\theta,j}(\omega) = \hat{\psi}(2^j r_{-\theta}\omega)$ has a support centered at $2^{-j}r_\theta\eta$, with a bandwidth proportional to 2^{-j}. We consider a group G of rotations r_θ which is either a finite subgroup of $SO(2)$ or which is equal to $SO(2)$. A finite rotation group is indexed by $\Theta = \{2k\pi/K \,:\, 0 \leq k < K\}$ and if $G = SO(2)$ then $\Theta = [0, 2\pi)$. The wavelet transform at a scale 2^J is defined by

$$Wx = \left\{x \star \phi_J(u)\,,\, x \star \psi_{\theta,j}(u)\right\}_{u\in\mathbb{R}^2,\theta\in\Theta,j<J}. \tag{5}$$

It decomposes x along different orientations θ and scales 2^j in the neighborhood of each location u.

The choice of wavelet ψ depends upon the desired angular resolution. In the following we shall concentrate on Morlet wavelets. A Morlet wavelet is defined by

$$\psi(u_1,u_2) = \exp\left(-\frac{u_1^2+u_2^2/\zeta^2}{2}\right)(\exp(i\xi u_1)-K) \tag{6}$$

The slant ζ of the envelope control the angular sensitivity of ψ. The second factor is an horizontal sine wave of frequency ξ. The constant $K>0$ is adjusted so that $\int\psi = 0$. Morlet wavelets for $\pi \le \theta < 2\pi$ are not computed since they verified $\psi_{\theta+\pi,j} = \psi^*_{\theta,j}$, where z^* denotes the complex conjugate of z. The averaging function is chosen to be a Gaussian window

$$\phi(u) = (2\pi\sigma^2)^{-1}\exp(-u^2/(2\pi\sigma^2)) \tag{7}$$

Figure 2 shows such window and Morlet wavelets.

To simplify notations, we shall write $\sum_{\theta\in\Theta} h(\theta)$ a summation over Θ even when $\Theta = [0,2\pi)$ in which case this discrete sum represents the integral $\int_0^{2\pi} h(\theta)\,d\theta$. We consider wavelets which satisfy the following Littlewood-Paley condition, for $\varepsilon > 0$ and almost all $\omega \in \mathbb{R}^2$

$$1-\varepsilon \le |\hat\phi(\omega)|^2 + \sum_{j<0}\sum_{\theta\in\Theta} |\hat\psi(2^j r_\theta \omega)|^2 \le 1\,. \tag{8}$$

Applying the Plancherel formula proves that if x is real then $Wx = \{x\star\phi_{2^J}\,, x\star\psi_{\theta,j}\}_{\theta,j}$ satisfies

$$(1-\varepsilon)\,\|x\|^2 \le \|Wx\|^2 \le \|x\|^2\,, \tag{9}$$

with

$$\|Wx\|^2 = \|x\star\phi_J\|^2 + \sum_{j<J}\sum_{\theta\in\Theta}\|x\star\psi_{\theta,j}\|^2\,.$$

In the following we suppose that $\varepsilon < 1$ and hence that the wavelet transform is a nonexpansive and invertible operator, with a stable inverse. If $\varepsilon = 0$ then W is unitary.

The Morlet wavelet ψ shown in Figure 2 together with $\phi(u) = \exp(-|u|^2/(2\sigma^2))/(2\pi\sigma^2)$ for $\sigma = 0.7$ satisfy (8) with $\varepsilon = 0.25$. These functions are used in all classification applications.

Unlike Fourier waveforms, Morlet wavelets $\psi_{\theta,j}$ are smooth and localized which makes them stable to deformation. Figure 3 shows that the responses of the same wavelet for two highly deformed images are comparable but displaced. Indeed, wavelet coefficients $x\star\psi_{\theta,j}(u)$ are computed with convolutions. They are therefore translation covariant, which means that if x is translated then $x\star\psi_{\theta,j}(u)$ is translated. Removing the complex phase like in a Fourier transform defines a positive envelope $|x\star\psi_{\theta,j}(u)|$ which is still covariant to translation, not invariant. Averaging this positive envelope defines locally translation invariant coefficients which depends upon (u,θ,j):

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

Such averaged wavelet coefficients are used under various forms in computer vision. Global histograms of quantized filter responses have been used for texture recognition in Leung & Malik [17]. SIFT[19] and DAISY[22] descriptors computes local histogram of orientation. This is similar to S_1x definition, but with different wavelets and non-linearity. Due to their stability properties, SIFT-like descriptors have been used extensively for a wide range of applications where stability to deformation is important, such as key point matching in pair of images from different view points, and generic object recognition.

2.3. Transation Invariant Scattering

The convolution by ϕ_J provides a local translation invariance but also loses spatial variability of the wavelet transform. A scattering successively recovers the information lost by the averaging which computes the invariants. Scattering consists in a cascade of wavelet modulus transforms, which can be interpreted as a deep neural network.

A scattering transform is computed by iterating on wavelet transforms and modulus operators. To simplify notations, we shall write $\lambda = (\theta,j)$ and $\Lambda = \{(\theta,j)\ :\ \theta\in[0,2\pi]\}$. The wavelet transform and modulus operations are combined in a single wavelet modulus operator defined by:

$$|W|x = \left\{x\star\phi_J\,, |x\star\psi_\lambda|\right\}_{\lambda\in\Lambda}\,. \tag{10}$$

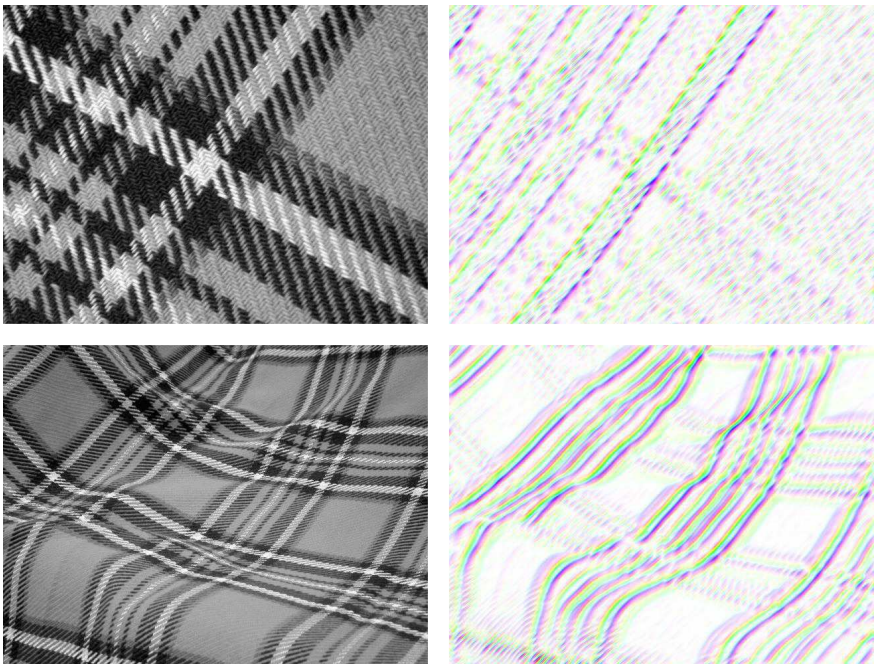


Figure 3: Two images of the same texture (left) and their convolution with the same Morlet wavelet (right). Even though the texture is highly deformed, the wavelet responds to roughly the same oriented pattern in both images, which illustrates its stability to deformation.

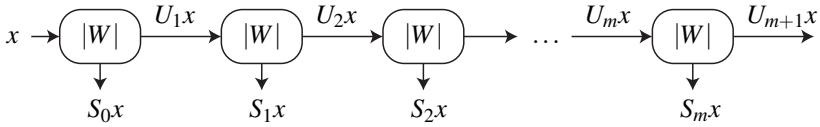


Figure 4: Translation scattering can be seen as a neural network which iterates over wavelet modulus operators $|W|$. Each layer m outputs averaged invariant $S_m x$ and covariant coefficients $U_{m+1}x$.

This operator averages coefficients with ϕ_J to produce invariants to translations and computes higher frequency wavelet transform envelopes which carry the lost information. A scattering transform can be interpreted as a neural network illustrated in Figure 4 which propagates a signal x across multiple layers of the network and which outputs at each layer m scattering invariant coefficients $S_m x$.

The input of the network is the original signal $U_0 x = x$. The scattering transform is then defined by induction. For any $m \geq 0$, applying the wavelet modulus operator $|W|$ on $U_m x$ outputs the scattering coefficients $S_m x$ and computes the next layer of coefficients $U_{m+1}x$:

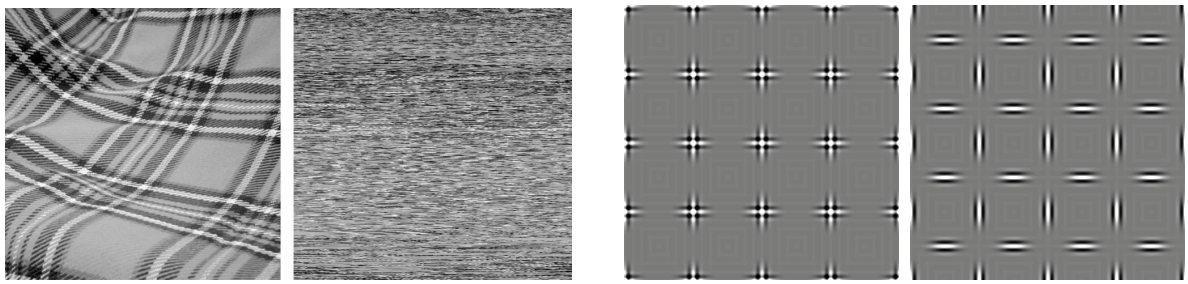
$$|W|\, U_m x = (S_m x\,,\, U_{m+1} x)\;, \tag{11}$$

with

$$\begin{aligned} S_m x(u, \lambda_1, \ldots, \lambda_m) &= U_m x(., \lambda_1, \ldots, \lambda_m) \star \phi_J(u) \\ &= \big|\,|x \star \psi_{\lambda_1}| \star \ldots | \star \psi_{\lambda_m}| \star \phi_J(u) \end{aligned}$$

and

$$\begin{aligned} U_{m+1} x(u, \lambda_1, \ldots, \lambda_m, \lambda_{m+1}) &= |U_m x(., \lambda_1, \ldots, \lambda_m) \star \psi_{\lambda_{m+1}}(u)| \\ &= \big|\,|x \star \psi_{\lambda_1}| \star \ldots | \star \psi_{\lambda_m}| \star \psi_{\lambda_{m+1}}(u)| \end{aligned}$$



(a) (b)

Figure 5: (a): Two images where each row of the second image is translated by a different amount $v(u_1)$. A separable translation invariant that would start by computing a translation invariant for each row would output the same value, which illustrates the fact that such separable invariants are too strong. (b) Two textures whose first internal layer is translated by different values for different orientations. In this example, vertical orientations are not translated while horizontal orientations are translated by $1/2(1,1)$. Translation scattering and other separable invariants cannot distinguish these two textures because it does not connect vertical and horizontal nodes.

This scattering transform is illustrated in Figure 4. The final scattering vector concatenates all scattering coefficients for $0 \leq m \leq M$:

$$Sx = (S_m x)_{0 \leq m \leq M}. \tag{12}$$

A scattering tranform is a non-expansive operator, which is stable to deformations. Let $\|Sx\| = \sum_m \|S_m x\|^2$, one can prove that

$$\|Sx - Sy\| \leq \|x - y\| . \tag{13}$$

Because wavelets are localized and separate scale we can also prove [8] that if x has a compact support then there exists $C > 0$ such that

$$\|Sx_\tau - Sx\| \leq C \|x\| \left(2^{-J} \|\tau\|_\infty + \|\nabla \tau\|_\infty + \|H\tau\|_\infty \right) . \tag{14}$$

Most of the energy of scattering coefficients is concentrated on the first two layers $m = 1, 2$. As a result, applications thus typically concentrate on these two layers. Among second layer scattering coefficients

$$S_2 x(u, \lambda_1, \lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi_J(u)$$

coefficients $\lambda_2 = 2^{j_2} r_{\theta_2}$ with $2^{j_2} \leq 2^{j_1}$ have a small energy. Indeed, $|x \star \psi_{\lambda_1}|$ has an energy concentrated in a lower frequency band. As a result, we only compute scattering coefficients for increasing scales $2^{j_2} > 2^{j_1}$.

2.4. Separable Versus Joint Rigid Motion Invariants

An invariant to a group which is a product of two sub-groups can be implemented as a separable product of two invariant operators on each subgroup. However, this separable invariant is often too strong, and loses important information. This is shown for translations and rotations.

To understand the loss of information produced by separable invariants let us first consider the two-dimensional translation group over $\mathbb{R}^2$. A two-dimensional translation invariant operator applied to $x(u_1, u_2)$ can be computed by applying 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 invariant to any two-dimensional translation. However, if $x_v(u_1, u_2) = x(u_1 - v(u_2), u_2)$ then $\Phi_1 x_v = \Phi_1 x$ for all $v(u_2)$, although x_v is not a translation of x because $v(u_2)$ is not constant. It results that $\Phi x = \Phi x_v$. 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 5a. 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 invariant scattering operators in $\mathbb{R}^2$ are not computed as products of scattering operators along horizontal and vertical variables.

The same phenomena appears for invariants along translations and rotations, although it is more subtle because translations and rotations interfere. Suppose that we apply a translation invariant operator Φ_1, such as a scattering transform, which separate image components along different orientations indexed by an orientation parameter $\theta \in [0, 2\pi)$. Applying a second rotation invariant operator Φ_2 which acts along θ produces a translation and rotation invariant operator.

Locally Binary Pattern [27] follows this approach. It first builds translation invariance with an histogram of oriented pattern. Then, it builds rotation invariance on top, by either pooling all patterns that are rotated versions of one another, or by computing modulus of Fourier transform on the angular difference that relates rotated patterns.

Such separable invariant operators have the advantage of simplicity and have thus been used in several computer vision applications. However, as in the separable translation case, separable products of translation and rotation invariants can confuse very different images. 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 5b we get 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 [9] or any of the separable translation and rotation invariant in used in [25, 27].

Taking into account the joint structure of the rigid-motion group of rotations and translations in $\mathbb{R}^2$ was proposed by several researchers [13, 15, 14], to preserve image structures in applications such as noise removal or image enhancement with directional diffusion operators [16]. Similarly, a joint scattering invariant to translations and rotations is constructed directly on the rigid-motion group in order to take into account the joint information between positions and orientations.

3. Rigid-motion Scattering

Translation invariant scattering operators are extended to define invariant representations over any Lie group, by calculating wavelet transforms on this group. Such wavelet transforms are well defined with weak conditions on the Lie group. We concentrate on invariance to the action of rotations and translations, which belong to the special Euclidean group. Next section briefly reviews the properties of the special Euclidean group. A scattering operator [8] computes an invariant image representation relatively to the action of a group by applying wavelet transforms to functions defined on the group.

3.1. Rigid-Motion Group

The set of rigid-motions is called the special Euclidean group $SE(2)$. We briefly review its properties. A rigid-motion in $\mathbb{R}^2$ is parameterized by a translation $v \in \mathbb{R}^2$ and a rotation $r_\theta \in SO(2)$ of angle $\theta \in [0, 2\pi)$. We write $g = (v, \theta)$. Such a rigid-motion g maps $u \in \mathbb{R}^2$ to

$$gu = v + r_\theta u \,. \tag{15}$$

A rigid-motion g applied to an image $x(u)$ translates and rotates the image accordingly:

$$g.x(u) = x(g^{-1}u) = x(r_{-\theta}(u - v)) \,. \tag{16}$$

The group action (15) must be compatible with the product $g'.(gu) = (g'.g)u$, so that successive applications of two rigid-motions g and g' are equivalent to the application of a single product rigid-motion $g'.g$. This combined to (15) implies that

$$g'.g = (v' + r_{\theta'}v,\ \theta + \theta') \,. \tag{17}$$

This group product is not commutative. The neutral element is $(0, 0)$, and the inverse of g is

$$g^{-1} = (-r_{-\theta}v, -\theta). \tag{18}$$

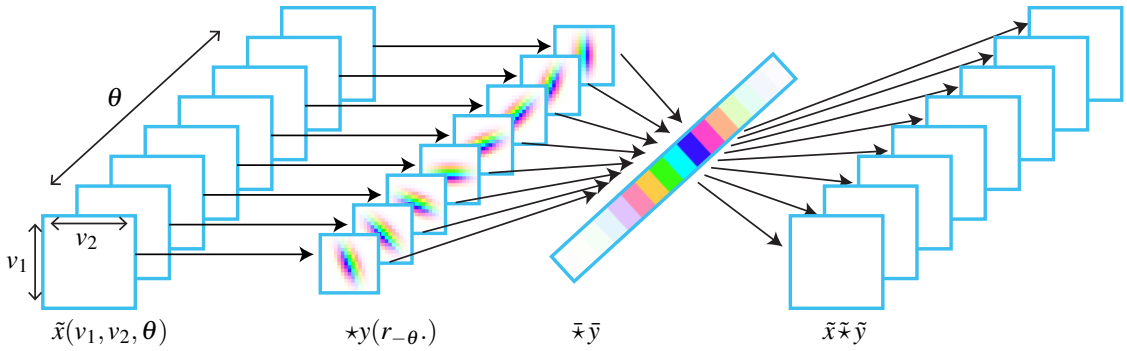


Figure 6: A rigid-motion convolution (20) with a separable filter $\tilde{y}(v,\theta) = y(v)\bar{y}(\theta)$ in $SE(2)$ can be factorized into a two dimensional convolution with rotated filters $y(r_{-\theta}v)$ and a one dimensional convolution with $\bar{y}(\theta)$.

The product (17) of $SE(2)$ is the definition of the semidirect product of the translation group $\mathbb{R}^2$ and the rotation group $SO(2)$:

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

It is a Lie group, and the left invariant Haar measure of $SE(2)$ is $dg = dv\,d\theta$, obtained as a product of the Haar measures on $\mathbb{R}^2$ and $SO(2)$.

The space $\mathbf{L^2}(SE(2))$ of finite energy measurable functions $\tilde{x}(v,\theta)$ is a Hilbert space

$$\mathbf{L^2}(SE(2)) = \left\{ \tilde{x} \,:\, \int_{\mathbb{R}^2} \int_0^{2\pi} |\tilde{x}(v,\theta)|^2 \, d\theta dv < \infty \right\} .$$

The left-invariant convolution of two functions $\tilde{x}(g)$ and $\tilde{y}(g)$ is defined by

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

Since $(v',\theta')^{-1} = (-r_{-\theta'}v', -\theta')$

$$\tilde{x} \tilde{\star} \tilde{y}(v,\theta) = \int_{\mathbb{R}^2} \int_0^{2\pi} \tilde{x}(v',\theta') \, \tilde{y}(r_{-\theta'}(v-v'),\ \theta-\theta') \, dv' d\theta' . \tag{19}$$

For separable filters $\tilde{y}(v,\theta) = y(v)\bar{y}(\theta)$, this convolution can be factorized into a spatial convolution with rotated filters $y(r_{-\theta}v)$ followed by convolution with $\bar{y}(\theta)$:

$$\tilde{x} \tilde{\star} \tilde{y}(v,\theta) = \int_0^{2\pi} \left(\int_{\mathbb{R}^2} \tilde{x}(v',\theta') y(r_{-\theta'}(v-v')) dv' \right) \bar{y}(\theta-\theta') \, d\theta' . \tag{20}$$

This is illustrated in Figure 6.

3.2. Wavelet Transform on the Rigid-Motion Group

A wavelet transform $\widetilde{W}$ in $\mathbf{L^2}(SE(2))$ is defined as convolutions with averaging window and wavelets in $\mathbf{L^2}(SE(2))$. The wavelets are constructed as separable products of wavelets in $\mathbf{L^2}(\mathbb{R}^2)$ and in $\mathbf{L^2}(SO(2))$.

A spatial wavelet transform in $\mathbf{L^2}(\mathbb{R}^2)$ is defined from L mother wavelets $\psi_l(u)$ which are dilated $\psi_{l,j}(u) = 2^{-2j}\psi_l(2^{-j}u)$, and a rotationally symmetric averaging function $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$ at the maximum scale 2^J:

$$Wx = \left\{ x \star \phi_J(u) \,,\, x \star \psi_{l,j}(u) \right\}_{u \in \mathbb{R}^2, 0 \le l < L, j < J} .$$

Since rotations in $\mathbb{R}^2$ parametrized by an angle in $[0,2\pi)$, the space $\mathbf{L^2}(SO(2))$ is equivalent to the space $\mathbf{L^2}([0,2\pi))$. We denote by $\bar{x}(\theta)$ functions which are 2π periodic and belong to $\mathbf{L^2}(SO(2))$. Circular convolutions of such functions are written

$$\bar{x} \bar{\star} \bar{y}(\theta) = \int_0^{2\pi} \bar{x}(\theta') \, \bar{y}(\theta - \theta') \, d\theta' \, .$$

Periodic wavelets are obtained by periodizing a one-dimensional scaling function $\phi_K^1(\theta) = 2^{-K}\phi^1(2^{-K}\theta)$ and one-dimensional wavelets $\psi_k^1(\theta) = 2^{-k}\psi^1(2^{-k}\theta)$

$$\overline{\phi}_K(\theta) = \sum_{m\in\mathbb{Z}} \phi_K^1(\theta - 2\pi m) \tag{21}$$

$$\overline{\psi}_k(\theta) = \sum_{m\in\mathbb{Z}} \psi_k^1(\theta - 2\pi m) \, . \tag{22}$$

The resulting one-dimensional periodic wavelet tranform applied to a function $\bar{x} \in \mathbf{L^2}([0,2\pi))$ is calculated with circular convolutions on $[0,2\pi)$:

$$\overline{W}\bar{x} = \left\{ \bar{x} \bar{\star} \overline{\phi}_K \, , \bar{x} \bar{\star} \overline{\psi}_k \right\}_{k<K} \, . \tag{23}$$

A separable wavelet family in $\mathbf{L^2}(SE(2))$ is constructed as a separable product of wavelets in $\mathbf{L^2}(\mathbb{R}^2)$ and wavelets in $\mathbf{L^2}(SO(2))$

$$\widetilde{\phi}_{J,K}(v,\theta) = \phi_J(v) \, \overline{\phi}_K(\theta) \tag{24}$$

and for all $0 \le l < L$

$$\widetilde{\psi}_{l,j,k}(v,\theta) \triangleq \begin{cases} \psi_{l,j}(v) \, \overline{\psi}_k(\theta) & \text{if } j < J \text{ and } k < K \\ \psi_{l,j}(v) \, \overline{\phi}_K(\theta) & \text{if } j < J \text{ and } k = K \\ \phi_J(v) \, \overline{\psi}_k(\theta) & \text{if } j = J \text{ and } k < K \end{cases} \, . \tag{25}$$

The resulting wavelet transform is defined by

$$\widetilde{W}\tilde{x} = \left\{ \tilde{x} \tilde{\star} \widetilde{\phi}_{J,K}(v,\theta) \, , \tilde{x} \tilde{\star} \widetilde{\psi}_{l,j,k}(v,\theta) \right\}_{l,j,k} \, .$$

Its energy is defined as the sum of the squared $\mathbf{L^2}(SE(2))$ norm of each of its component

$$\|\widetilde{W}\tilde{x}\|^2 \triangleq \|\tilde{x} \tilde{\star} \widetilde{\phi}_{J,K}\|^2 + \sum_{l,j,k} \|\tilde{x} \tilde{\star} \widetilde{\psi}_{l,j,k}\|^2 \, . \tag{26}$$

The following theorem gives conditions on the one and two-dimensional wavelets so that $\widetilde{W}$ is a bounded linear operator which satisfies an energy conservation.

Theorem 1. *If there exists* $\varepsilon_1 > 0$ *and* $\varepsilon_2 > 0$ *such that*

$$\forall \omega \in \mathbb{R}, \;\; 1 - \varepsilon_1 \le |\hat{\phi}^1(\omega)|^2 + \sum_{k<0} |\hat{\psi}^1(2^k\omega)|^2 \le 1 \, , \tag{27}$$

$$\forall \omega \in \mathbb{R}^2, \; 1 - \varepsilon_2 \le |\hat{\phi}(\omega)|^2 + \sum_{\substack{0 \le l < L \\ j<0}} |\hat{\psi}_l(2^j\omega)|^2 \le 1 \, , \tag{28}$$

then

$$(1-\varepsilon_1)\,(1-\varepsilon_2)\,\|\tilde{x}\|^2 \le \|\widetilde{W}\tilde{x}\|^2 \le \|\tilde{x}\|^2 \, . \tag{29}$$

Proof: we denote $Wx(u,J) = x \star \phi_J(u)$, $Wx(u,l,j) = x \star \psi_{l,j}(u)$, $\overline{W}\bar{x}(\theta,K) = \bar{x} \bar{\star} \overline{\phi}_K(\theta)$, $\overline{W}\bar{x}(\theta,k) = \bar{x} \bar{\star} \overline{\psi}_k(\theta)$. For $j < J$ and $k < K$, applying the separable convolution formula (20) to $\widetilde{\psi}_{l,j,k}(v,\theta) = \psi_{l,j}(v) \, \overline{\psi}_k(\theta)$ proves that

$$\begin{aligned} \tilde{x} \tilde{\star} \widetilde{\psi}_{l,j,k}(v,\theta) &= \iint \tilde{x}(v',\theta') \psi_{l,j}(r_{-\theta'}(v-v')) dv' \overline{\psi}_k(\theta - \theta') d\theta' \\ &= \iint \tilde{x}(r_{\theta'}w, \theta') \psi_{l,j}(r_{-\theta'}v - w) dw \overline{\psi}_k(\theta - \theta') d\theta' . \end{aligned} \tag{30}$$

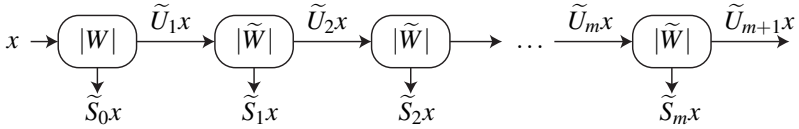


Figure 7: Rigid-motion scattering is similar to translation scattering of Figure 4, but deep wavelet modulus operators $|W|$ are replaced with rigid-motion wavelet modulus operators $|\widetilde{W}|$ where convolutions are applied along the rigid-motion group.

A similar result is obtained for all other $\widetilde{\psi}_{l,j,k}(v,\theta)$. It proves that

$$\widetilde{W}\tilde{x} = \overline{W} R^{-1} W R \tilde{x} \tag{31}$$

where $R\tilde{x}(v,\theta) = \tilde{x}(r_\theta v, \theta)$ and $W\tilde{x}$ computes $Wx_\theta(v,l,j)$ from each $x_\theta(v) = \tilde{x}(v,\theta)$. We saw in (9) that for all x, $(1-\varepsilon_1)\|x\| \leq \|Wx\| \leq \|x\|$. The rotation operator R is unitary $\|Rx\| = \|x\|$. We are now going to prove that for all $\bar{x}$, $(1-\varepsilon_2)\|\bar{x}\| \leq \|\overline{W}\bar{x}\| \leq \|\bar{x}\|$. Since $\tilde{W} = \overline{W}R^{-1}WR$ this last inequality will prove (29).

Let $x_0(\theta) = \overline{x}(\theta)$ for $\theta \in [0, 2\pi)$ and $x_0(\theta) = 0$ otherwise. Observe that

$$\overline{W}\bar{x}(\theta,k) = \int_{-\infty}^{\infty} x_0(\theta')\, \psi_k^1(\theta - \theta')\, d\theta' = x_0 \star \psi_k^1(\theta)$$

and

$$\overline{W}\bar{x}(\theta,K) = \int_{-\infty}^{\infty} x_0(\theta')\phi_K^1(\theta - \theta')d\theta' = x_0 \star \phi_K^1(\theta)\ ,$$

so that

$$\|\overline{W}\bar{x}\|^2 = \|x_0 \star \phi_K^1\|^2 + \sum_{k<K} \|x_0 \star \psi_k^1\|^2$$

where the norms on the right are norms in $\mathbf{L^2}(\mathbb{R})$. By applying the Plancherel formula together with (27) we verify that

$$(1-\varepsilon_2)\|x_0\|^2 \leq \|\overline{W}\bar{x}\|^2 \leq \|x_0\|^2$$

and since $\|x_0\|^2 = \int_0^{2\pi} |x(\theta)|^2 d\theta = \|\bar{x}\|^2$we conclude that $(1-\varepsilon_2)\|\bar{x}\| \leq \|\overline{W}\bar{x}\| \leq \|\bar{x}\|$ over $\mathbf{L^2}([0,2\pi))$. □.

3.3. Rigid-Motion Invariant Scattering Transform

A rigid-motion invariant scattering has the same architecture as the translation invariant scattering of Section 2.3. It is illustrated in Figure 7. It computes a first spatial wavelet modulus operator $|W|$ and then iterates on rigid-motion wavelet modulus operators $|\widetilde{W}|$. To simplify notations, we denote $\lambda = (\mathfrak{l}, j, k)$ and $\tilde{\Lambda} = \{(\mathfrak{l}, j, k)\}$. The rigid-motion wavelet modulus operator can be applied to any function of the rigid-motion group $\tilde{x}(g)$ for $g = (u, \theta)$:

$$|\widetilde{W}|\tilde{x}(g) = \left(\tilde{x} \tilde{\star} \widetilde{\phi}_{J,K}(g)\,,\, |\tilde{x} \tilde{\star} \widetilde{\psi}_\lambda(g)|\right)_{\lambda \in \tilde{\Lambda}}.$$

Its norm is defined by

$$\|\widetilde{W}\tilde{x}\|^2 = \|\tilde{x} \tilde{\star} \widetilde{\phi}_{J,K}\|^2 + \sum_{\lambda \in \tilde{\Lambda}} \|\tilde{x} \tilde{\star} \widetilde{\psi}_\lambda\|^2\ .$$

The rigid-motion scattering begins with applying a spatial wavelet modulus operator (32) to $x(u)$,

$$|W|x = \left\{x \star \phi_{2^J}\,,\, |x \star \psi_{\theta,j}|\right\}_{(\theta,j)\in\Lambda}\ . \tag{32}$$

It computes the first scattering network layer

$$\widetilde{U}_1 x(u,\theta,j) = |x \star \psi_{\theta,j}(u)|\ .$$

$\widetilde{U}_1 x$ is considered as a function of $g = (u, \theta)$, for each j fixed. The scattering transform is then defined by induction, with successive applications of rigid-motion wavelet modulus transforms along the g variable. For any $m \geq 1$, applying the wavelet modulus operator $|\widetilde{W}|$ on $\widetilde{U}_m x$ outputs the scattering coefficients $\widetilde{S}_m x$ and computes the next layer of coefficients $\widetilde{U}_{m+1} x$:

$$|\widetilde{W}|\,\widetilde{U}_m x = (\widetilde{S}_m x\,,\, \widetilde{U}_{m+1} x)\ , \tag{33}$$

with

$$\begin{aligned}\widetilde{S}_m x(g, j_1, \lambda_2, \ldots, \lambda_m) &= \widetilde{U}_m x(., j_1, \lambda_2, \ldots, \lambda_m) \,\tilde{\star}\, \widetilde{\phi}_{J,K}(g) \\ &= |\,|x \star \psi_{.,j_1}| \,\tilde{\star}\, \widetilde{\psi}_{\lambda_2} \ldots \,\tilde{\star}\, \widetilde{\psi}_{\lambda_m}| \,\tilde{\star}\, \widetilde{\phi}_{J,K}(g)\end{aligned}$$

and

$$\begin{aligned}\widetilde{U}_{m+1} x(g, j_1, \lambda_2, \ldots, \lambda_m, \lambda_{m+1}) &= |\widetilde{U}_m x(., j_1, \lambda_2, \ldots, \lambda_m) \,\tilde{\star}\, \widetilde{\psi}_{\lambda_{m+1}}(g)| \\ &= |\,|x \star \psi_{.,j_1}| \star \widetilde{\psi}_{\lambda_2} \ldots | \star \widetilde{\psi}_{\lambda_m}| \star \widetilde{\psi}_{\lambda_{m+1}}(g)|\end{aligned} \tag{34}$$

This rigid-motion scattering transform is illustrated in Figure 7.

The final scattering vector concatenates all scattering coefficients for $0 \leq m \leq M$:

$$\widetilde{S} x = (\widetilde{S}_m x)_{0 \leq m \leq M}. \tag{35}$$

The following theorem proves that a scattering transform is a non-expansive operator.

Theorem 2. *For any $M \in \mathbb{N}$ and any $(x, y) \in \mathbf{L^2}(\mathbb{R}^2)$*

$$\|\widetilde{S} x - \widetilde{S} y\| \leq \|x - y\|\ . \tag{36}$$

Proof: A modulus is non-expansive in the sense that for any $(a, b) \in \mathbb{C}^2$, $||a| - |b|| \leq |a - b|$. Since $\widetilde{W}$ is a linear non-expansive operator, it results that the wavelet modulus operator $|\widetilde{W}|$ is also non-expansive

$$\|\,|\widetilde{W}|x - |\widetilde{W}|y\| \leq \|x - y\|\ .$$

Since $\widetilde{W}$ is non-expansive, it results from (33) that

$$\begin{aligned}&\|\,|\widetilde{W}|\,\widetilde{U}_m x - |\widetilde{W}|\,\widetilde{U}_m y\| \\ &\qquad = \|\widetilde{S}_m x - \widetilde{S}_m y\|^2 + \|\widetilde{U}_{m+1} x - \widetilde{U}_{m+1} y\|^2 \\ &\qquad \leq \|\widetilde{U}_m x - \widetilde{U}_m y\|^2\ .\end{aligned} \tag{37}$$

Summing this equation from $m = 1$ to M gives

$$\begin{aligned}\sum_{m=1}^{M} \|\widetilde{S}_m x - \widetilde{S}_m y\|^2 + \|\widetilde{U}_{M+1} x - \widetilde{U}_{M+1} y\|^2 \\ \leq \|\widetilde{U}_1 x - \widetilde{U}_1 y\|^2\ .\end{aligned} \tag{38}$$

Since $|W|x = (S_0 x, \widetilde{U}_1 x)$ which is also non-expansive, we get

$$\|S_0 x - S_0 y\|^2 + \|\widetilde{U}_1 x - \widetilde{U}_1 y\|^2 \leq \|x - y\|^2\ . \tag{39}$$

Inserting (39) in (38) proves (36). □

4. Fast Rigid-Motion Scattering

For texture classification applications, first and second layers of scattering are sufficient for achieving state-of-the-art results. This section describes a fast implementation of rigid-motion scattering based on a filter bank implementation of the wavelet transform.

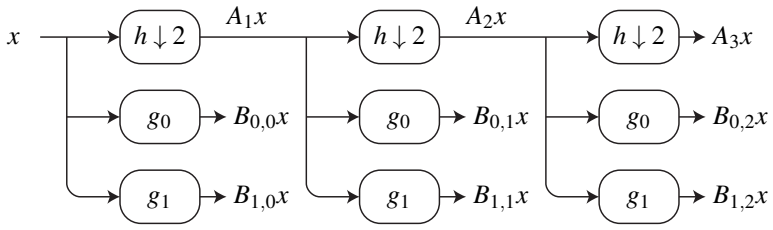


Figure 8: Filter bank implementation of the wavelet transform W with $J = 3$ scales and $C = 2$ orientations. A cascade of low pass filter h and downsampling computes low frequencies $A_j x = x \star \phi_j$ and filters g_θ compute high frequencies $B_{\theta,j} x = x \star \psi_{\theta,j}$. This cascade results in a tree whose internal nodes are intermediate computations and whose leaves are the output of the downsampled wavelet transform.

4.1. Wavelet Filter Bank Implementation

Rigid-motion scattering coefficients are computed by applying a spatial wavelet tranform W and then a rigid-motion wavelet tranform $\widetilde{W}$. This section describes filter bank implementations of the spatial wavelet transform.

A wavelet tranform

$$Wx = \left\{ x \star \phi_J(u) \, , \, x \star \psi_{\theta,j}(u) \right\}_{u \in \mathbb{R}, \theta \in \Theta, j < J} \tag{40}$$

is computed with a filter bank algorithm, also called "algorithm à trous". This assumes that the Fourier transform of the window $\phi(u)$ and each wavelet $\psi_\theta(u) = \psi(r_{-\theta} u)$ can be written as a product of Fourier transforms of discrete dilated filters h and g:

$$\hat{\phi}(\omega) = \prod_{j<0}^{\infty} \hat{h}(2^j \omega) \tag{41}$$

and for all $\theta \in \Theta$

$$\hat{\psi}_\theta(\omega) = \hat{g}_\theta(\omega) \hat{\phi}(\omega) \, . \tag{42}$$

Let us initialize $A_0 x = x \star \phi$ and denote $A_j x(n) = x \star \phi_j(2^j n)$ and $B_{\theta,j} x(n) = x \star \psi_{\theta,j}(2^j n)$ for $n \in \mathbb{Z}^2$ It results from (41) and (42) that

$$\begin{aligned} A_{j+1} x(n) &= \sum_p A_j x(2p) h(n-2p) \\ B_{\theta,j} x(n) &= \sum_p A_j x(p) \, g_\theta(n-p) \, . \end{aligned}$$

Thus, the subsampled wavelet transform operator can be implemented as a cascade of convolution and downsampling. The convolutions are done with filters h and g_θ whose support do not change with the scale of the wavelet transform. This allows to use spatial convolutions in the regime where they are faster than FFT-based convolutions. This is compactly expressed as

$$\begin{aligned} A_{j+1} x &= (A_j x \star h) \downarrow 2 \\ B_{\theta,j} x &= A_j x \star g_\theta \end{aligned}$$

This filter bank cascade is illustrated in Figure 8. Let N be the size of the input image x and P be the size of the filters h and g_θ. A convolution at the finest resolution requires NP operations and N memory. The cascade computes $1 + C$ convolutions at each resolution 2^{-j}. The resulting time complexity is thus $(1 + C) \sum_j 2^{-2j} NP = O(CNP)$ and the required memory is $O(CN)$ 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.

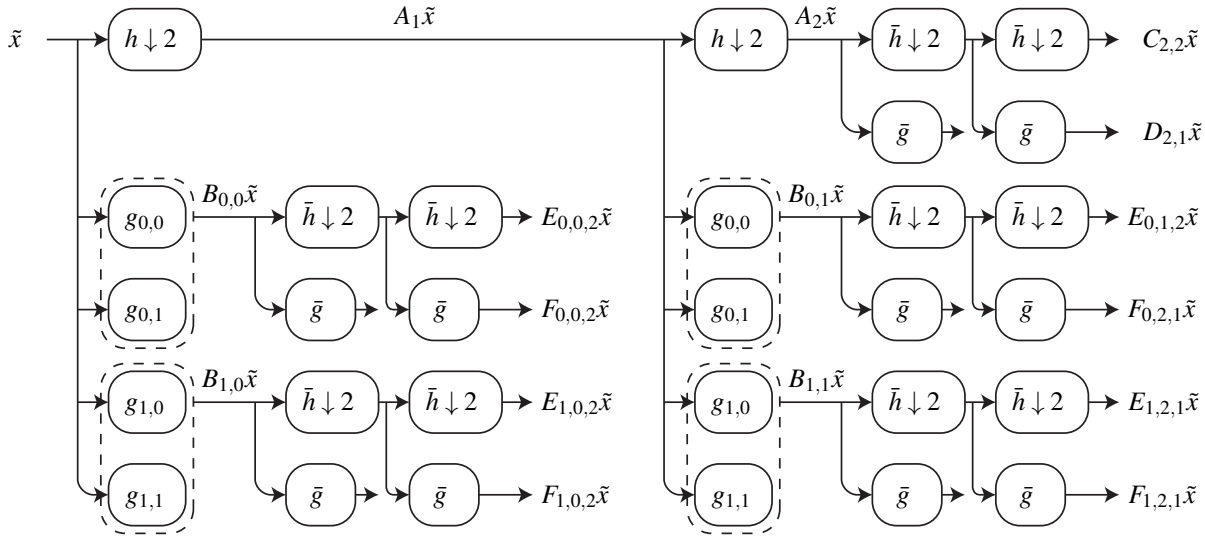


Figure 9: Filter bank implementation of the rigid-motion wavelet transform $\widetilde{W}$ with $J = 2$ spatial scales, $C = 2$ orientations, $L = 2$ spatial wavelets, $K = 2$ orientation scales . A first cascade computes spatial downsampling and filtering with h and $g_{l,\theta}$. The first cascade 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 $\bar{h}$ and $\bar{g}$ 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 $\tilde{x} \tilde{\star} \widetilde{\phi}_{J,K}\tilde{x}$, $\tilde{x} \tilde{\star} \widetilde{\psi}_{J,k}$, $\tilde{x} \tilde{\star} \widetilde{\psi}_{l,j,K}$, $\tilde{x} \tilde{\star} \widetilde{\psi}_{l,j,k}$ appropriately downsampled along the spatial and the orientation variable.

4.2. Rigid Motion Wavelet Filter Bank Implementation

Rigid motion wavelet transform $\widetilde{W}$ takes as input a discretized signal $\tilde{x}(n,\theta)$ indexed by position n and orientation θ and computes a set of convolutions with wavelet $\widetilde{W}\tilde{x} = \{\tilde{x} \tilde{\star} \widetilde{\phi}_{J,K},\ \tilde{x} \tilde{\star} \widetilde{\psi}_{\lambda}\}_{\lambda}$. Similarly to Section 4.1, it is computed with two successive cascades of convolution and downsampling along the spatial and orientation variable. Figure 9 illustrates this algorithm. We start with the spatial cascade. As previously we initialize $A_0\tilde{x} = \tilde{x}$ and compute

$$\begin{aligned} A_{j+1}\tilde{x} &= (A_j\tilde{x} \star h) \downarrow 2 \\ B_{l,j}\tilde{x} &= A_j\tilde{x} \star g_{l,\theta} \end{aligned}$$

The computation of $B_{l,j}\tilde{x}(n,\theta) = (A_j\tilde{x})(.,\theta) \star g_{l,\theta}(n)$ involves rotated filters $g_{l,\theta}(n) = g_l(r_{-\theta}n)$ that naturally appear in the factorization (20). There are LC such filters. In our classification experiments, we have chosen to use oriented filters for g_l, so that $g_{l,\theta} = g_{l+\theta}$ and there are only $L = C$ such filters. The spatial convolution is followed by convolutions along the orientation. Let us denote the subsampled rigid-motion wavelet transform coefficients:

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

These subsampled coefficients are initialized from A and B with $C_{J,0}\tilde{x} = A_J\tilde{x}$ and $E_{l,j,0}\tilde{x} = B_{l,j}\tilde{x}$. We compute them by induction

$$\begin{aligned} C_{J,k+1}\tilde{x} &= (C_{J,k}\tilde{x} \bar{\star} \bar{h})\overline{\downarrow}2 \\ D_{J,k}\tilde{x} &= C_{J,k}\tilde{x} \bar{\star} \bar{g} \\ E_{l,j,k+1}\tilde{x} &= (E_{j,l,k}\tilde{x} \bar{\star} \bar{h})\overline{\downarrow}2 \\ F_{l,j,k}\tilde{x} &= E_{j,l,k}\tilde{x} \bar{\star} \bar{g} \end{aligned}$$

where $\bar{\star}$, $\overline{\downarrow}$, $\bar{h}$, $\bar{g}$ are the discrete convolution, downsampling, low pass and high pass filters along the orientation variable θ.

The first spatial cascade computes CL convolutions at each spatial resolution, which requires $O(CLNP)$ operations and $O(CLN)$ memory. Each leaf is then retransformed by a cascade along the orientation variable θ of cardinality C. Convolutions along the orientations are periodic and since the size of the filter $\bar{h}$, $\bar{g}$ is of the same order as C, we use FFT-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 $\widetilde{W}$ 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 wavelets, N is the size of the input image, P is the size of the spatial filters.

5. Image Texture Classification

Image Texture classification has many applications including satellite, medical and material imaging. It is a relatively well posed problem of computer vision, since the different sources of variability contained in texture images can be accurately modeled. This section presents application of rigid-motion scattering on four texture datasets containing different types and ranges of variability: [34, 35, 36] texture datasets, and the more challenging FMD [28, 37] materials dataset. Results are compared with state-of-the-art algorithms in table 1, 2, 3 and 4. All classification experiments are reproducible with the ScatNet [38] toolbox for MATLAB.

5.1. Dilation, Shear and Deformation Invariance with a PCA Classifier

Rigid-motion scattering builds invariance to the rigid-motion group. Yet, texture images also undergo other geometric transformations such as dilations, shears or elastic deformations. Dilations and shears, combined with rotations and translations, generates the group of affine transforms. One can define wavelets [31] and a scattering transform on the affine group to build affine invariance. However this group is much larger and it would involve heavy and unnecessary computations. A limited range of dilations and shears is available for finite resolution images which allows one to linearizes these variations. Invariance to dilations, shears and deformations are obtained with linear projectors implemented at the classifier level, by taking advantage of the scattering's stability to small deformation. In texture application 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 [7].

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 moments are partially invariant to scaling, we augment the training set by dilating each $X_{c,i}$ by typically 4 scaling factors $\{1,\ \sqrt{2},\ 2,\ 2\sqrt{2}\}$. In the following, we consider $\{X_{c,i}\}_i$ as

Train size	5	20	40
COX [23]	80.2 ± 2.2	92.4 ± 1.1	95.7 ± 0.5
BIF [24]	-	-	98.5
SRP [26]	-	-	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 & scale invariance	$\mathbf{84.3} \pm 3.1$	$\mathbf{98.3} \pm 0.9$	$\mathbf{99.4} \pm 0.4$

Table 1: Classification accuracy with standard deviations on [34] database. Columns correspond to different training sizes 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 [20]	-	92.6	96.0
WMFS [25]	**93.4**	97.0	98.6
BIF [24]	-	-	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 & scale invariance	93.3 ± 1.4	$\mathbf{97.8} \pm 0.6$	$\mathbf{99.4} \pm 0.4$

Table 2: Classification accuracy on [35] database.

the set of training examples augmented by dilation, which are incorporated in the empirical average estimation μ_c of $\mathbb{E}(\log \widetilde{S}X_c)$.

The principal components space $\mathbf{V}_c$ is estimated from the singular value decomposition (SVD) of the matrix of centered training example $\log \widetilde{S}X_{i,c} - \mu_c$. The number of non-zero eigenvectors which can be computed is equal to the total number of training examples. We define $\mathbf{V}_c$ as the space generated by all eigenvectors. In texture discrimination applications, it is not necessary to regularize the estimation by reducing the dimension of this space because there is a small number of training examples.

Given a test image X, we abusively denote by $\log \widetilde{S}X$ the average of the log scattering transform of X and its dilated versions. It is therefore a scaled averaged scattering tranform, which provides a partial scaling invariance. We denote by $P_{\mathbf{V}_c} \log \widetilde{S}X$ the orthogonal projection of $\log \widetilde{S}X$ 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 $\|(Id - P_{\mathbf{V}_c})(\log \widetilde{S}X - \mu_c)\|$ between $\widetilde{S}X$ and the affine space $\mu_c + \mathbf{V}_c$:

$$\hat{c}(X) = \arg\min_c \|(Id - P_{\mathbf{V}_c})(\log \widetilde{S}X - \mu_c)\|^2 \tag{43}$$

The translation and rotation invariance of a rigid-motion scattering $\widetilde{S}X$ results from the spatial and angle averaging implemented by the convolution with $\widetilde{\phi}_{J,K}$. It is nearly translation invariant over spatial domains of size 2^J and rotations of angles at most 2^K. The parameters J and K can be adjusted by cross-validation. One can also avoid performing any such averaging and let the linear supervised classifier optimize directly the averaging. This last approach is possible only if there are enough supervised training examples to learn the appropriate averaging kernel. This is not the case in the texture experiments of Section 5.2 where few training examples are available, but where the classification task is known to be fully translation and rotation invariant. The values of J and K are thus maximum.

5.2. Texture Classification Experiments

This sections details classification results on image texture datasets KTH-TIPS [34], UIUC [20, 35] and UMD [36]. Those datasets contains 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.

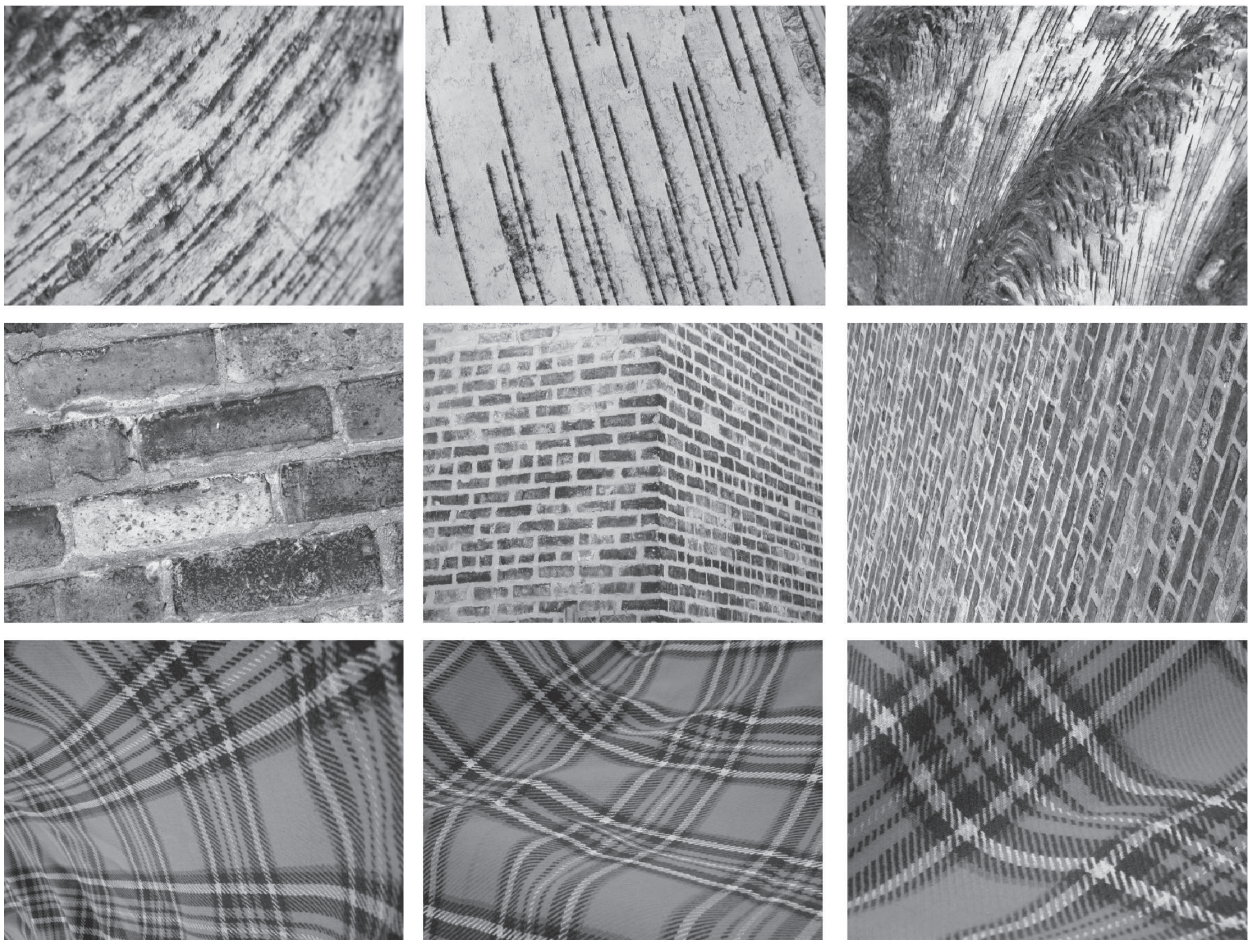


Figure 10: Each row shows images from the same texture class in the UIUC database [20], with important rotation, scaling and deformation variability.

Training size	5	10	20
WMFS [25]	93.4	97.0	98.7
SRP [26]	-	-	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 & scale invariance	$\mathbf{96.6} \pm 1.0$	$\mathbf{98.9} \pm 0.6$	$\mathbf{99.7} \pm 0.3$

Table 3: Classification accuracy on [36] database.

Training size	50
SRP [26]	48.2
Best single feature (SIFT) in [29]	41.2
Rigid-motion scattering + log on grey images	51.22
Rigid-motion scattering + log on YUV images	53.28

Table 4: Classification accuracy on [37] database.

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. This is the case of [20] where rotation invariance is obtained through histograms along concentric circles, as well as Log Gaussian Cox processes (COX) [23] and Basic Image Features (BIF) [24] which use rotation invariant patch descriptors calculated from small filter responses. Sorted Random Projection (SRP) [26] 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) [25] 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 [20, 23, 24, 25, 26] and scattering invariants on KTH-TIPS (table 1), UIUC (table 2) and UMD (table 3) texture databases. For the KTH-TIPS, UIUC and UMD database, Tables 1,2,3 give the mean classification accuracy and standard deviation over 200 random splits between training and testing for different training sizes. Classification accuracy is computed with scattering representations implemented with progressively more invariants, and with the PCA classifier of Section 5.1. 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 training size. The space $\mathbf{V}_c$ is thus generated by the D scattering vectors of the training set. For larger training databases, it must be adjusted with a cross validation as in [7].

Classification accuracy in Tables 1,2,3 are given for different scattering representations. The rows "Translation scattering" correspond to the scattering described in Section 2.3 and initially introduced in [7]. The rows "Rigid-motion scattering" replace the translation invariant scattering by the rigid-motion scattering of Section 3.3. Finally, the rows "+ log & scale invariance" corresponds to the rigid-motion scattering, with a logarithm non-linearity to linearize scaling, and with the partial scale invariance described in Section 5.1, with augmentation at training and averaging at testing along a limited range of dilation.

[34] contains 10 classes of 81 samples with controlled scaling, shear and illumination variations but no rotation. The Rigid-motion scattering does not degrade results but the scale invariant provides significant improvement.

[35] and [36] both contains 25 classes of 40 samples with uncontrolled deformations including shear, perspectivity effects and non-rigid deformations. For both these databases, rigid-motion scattering and the scale invariance provide considerable improvements over translation scattering. The overall approach achieves and often exceeds state-of-the-art results on all these databases.

[37] contains 10 classes of 100 samples. 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 5.1 can not linearize deformation and discriminative classifiers tend to give better results. The scattering results reported in table 4 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 which is to our knowledge the best for a single feature. Better results can be obtained using multiple features and a feature selection framework [29].

6. Conclusion

Rigid motion scattering provides stable translation and rotation invariants through a cascade of wavelet transform along the spatial and orientation variables. We have shown that such joint operators provide tighter invariants than separable operators, which tends to be too strong and thus lose too much information. A wavelet transform on the rigid-motion group has been introduced, with a fast implementation based on two downsampling and filtering cascade. Rigid-motion scattering has been applied to texture classification in presence of large geometric transformations and provide state-of-the-art classification results on most texture datasets.

Recent work [12] has shown that rigid-motion scattering, with extension to dilation, could also be used for more generic vision task such as object recognition, with promising results on the CalTech 101 and 256 datasets. For large scale deep networks, group convolution might also be useful to learn more structured and meaningful multidimensional filters.

References

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

[2] Y. LeCun, K. Kavukvuoglu and C. Farabet, "Convolutional Networks and Applications in Vision", Proc. of ISCAS 2010.

[3] T. Poggio, J. Mutch, F. Anselmi, L. Rosasco, J.Z. Leibo, and A. Tacchetti, "The computational magic of the ventral stream: sketch of a theory (and why some deep architectures work)", MIT-CSAIL-TR-2012-035, December 2012.

[4] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, "Pedestrian Detection with Unsupervised Multi-Stage Feature Learning", Proc. of Computer Vision and Pattern Recognition (CVPR), 2013.

[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Proc. of Neural Information Processing Systems (NIPS), 2012

[6] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M.A. Ranzato, A. Senior, P. Tucker, K. Yang, A. Y. Ng, "Large Scale Distributed Deep Networks", Proc. of Neural Information Processing Systems (NIPS), 2012.

[7] J. Bruna, S. Mallat, "Invariant Scattering Convolution Networks", Trans. on PAMI, vol. 35, no. 8, pp. 1872-1886, 2013.

[8] S. Mallat "Group Invariant Scattering", Communications in Pure and Applied Mathematics, vol. 65, no. 10. pp. 1331-1398, 2012.

[9] L. Sifre, S. Mallat, "Combined scattering for rotation invariant texture analysis", Proc. of European Symposium on Artificial Neural Networks (ESANN), 2012.

[10] L. Sifre, S. Mallat, "Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination", Proc. of Computer Vision and Pattern Recognition (CVPR), 2013.

[11] J. Bruna, S. Mallat, E. Bacry and J-F. Muzy, "Intermittent Process Analysis with Scattering Moments", submitted to Annals of Statistics, Nov 2013.

[12] E. Oyallon, S. Mallat, L. Sifre "Generic Deep Networks with Wavelet Scattering", submitted to International Conference on Learning Representations (ICLR), 2014.

[13] G. Citti, A. Sarti, "A Cortical Based Model of Perceptual Completion in the Roto-Translation Space", Journal of Mathematical Imaging and Vision archive, Vol. 24, no. 3, p. 307=326, 2006.

[14] U. Boscain, J. Duplaix, J.P. Gauthier, F. Rossi, "Anthropomorphic Image Reconstruction via Hypoelliptic Diffusion", SIAM Journal on Control and Optimization, Volume 50, Issue 3, pp. 1071-1733, 2012.

[15] R. Duits, B. Burgeth, "Scale Spaces on Lie Groups", in Scale Space and Variational Methods in Computer Vision, Springer Lecture Notes in Computer Science, Vol. 4485, pp 300-312, 2007.

[16] R. Duits, E. 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, Volume 92, Issue 3, pp 231-264, 2011.

[17] T. Leung, J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons", International Journal of Computer Vision , Volume 43, Issue 1, pp 29-44, 2001.

[18] R. Girshick1, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", arXiv preprint:1311.2524, 2013.

[19] D. Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, 60(4):91–110, 2004.

[20] S. Lazebnik, C. Schmid and J. Ponce, "A sparse texture representation using local affine regions", Trans. on PAMI, vol. 27, no. 8, pp. 1265-1278, 2005.

[21] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", Proc. of Computer Vision and Pattern Recognition (CVPR), 2006.

[22] E. Tola, V. Lepetit, P. Fua "DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo" Trans. on PAMI, Vol. 32, Nr. 5, pp. 815 - 830, 2010.

[23] H.-G. Nguyen, R. Fablet, and J.-M. Boucher, "Visual textures as realizations of multivariate log-Gaussian Cox processes", Proc. of Computer Vision and Pattern Recognition (CVPR), 2011.

[24] M. Crosier and L.D. Griffin, "Texture classification with a dictionary of basic image features", Proc. of Computer Vision and Pattern Recognition (CVPR), 2008.

[25] Y. Xu, X. Yang, H. Ling and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid", Proc. of Computer Vision and Pattern Recognition (CVPR), 2010.

[26] L. Liu, P. Fieguth, G. Kuang, H. Zha, "Sorted Random Projections for Robust Texture Classification", Proc. of ICCV, 2011.

[27] G. Zhao, T. Ahonen, J. Matas, M. Pietikäinen, "Rotation-invariant image and video description with local binary pattern features", Trans. on Image Processing, 21(4):1465-1467, 2012.
[28] L. Sharan, R. Rosenholtz, E. H. Adelson, "Material perception: What can you see in a brief glance?", Journal of Vision, 9(8):784, 2009.
[29] L. Sharan, C. Liu, Ruth Rosenholtz, Edward H. Adelson, "Recognizing Materials Using Perceptually Inspired Features", International Journal of Computer Vision Volume 103, Issue 3, pp 348-371, 2013.
[30] G. Yu and J.M. Morel, "A Fully Affine Invariant Image Comparison Method", Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, 2009.
[31] D. L. Donoho, G. Kutyniok, M. Shahram and X. Zhuang. "A Rational Design of Discrete Shearlet Transform", Proc. of SampTA'11 (Singapore), 2011.
[32] S. Mallat, "A Wavelet Tour of Signal Processing, 3rd ed.", Academic Press, 2008.
[33] L. W. Renninger and J. Malik, "When is scene recognition just texture recognition?", Vision Research, 44, pp. 2301-2311, 2004.
[34] KTH-TIPS: `http://www.nada.kth.se/cvap/databases/kth-tips/`
[35] UIUC : `http://www-cvr.ai.uiuc.edu/ponce_grp/data/`
[36] UMD : `http://www.cfar.umd.edu/~fer/website-texture/texture.htm`
[37] FMD : `http://people.csail.mit.edu/celiu/CVPR2010/FMD/`
[38] ScatNet, a MATLAB toolbox for scattering network : `http://www.di.ens.fr/data/software/scatnet/`