Group-invariant colour morphology based on frames

Jasper J. van de Gronde and Jos B.T.M. Roerdink, *Senior Member, IEEE*

***Abstract*—Mathematical morphology is a very popular framework for processing binary or greyscale images. One of the key problems in applying this framework to colour images is the notorious "false colour problem". We discuss the nature of this problem and its origins. In doing so it becomes apparent that the lack of invariance of operators to certain transformations (forming a group) plays an important role. The main culprits are the basic join and meet operations and the associated lattice structure that form the theoretical basis for mathematical morphology. We show how a lattice that is not group invariant can be related to another lattice that is. When all transformations in a group are linear, these lattices can be related to one another via the theory of frames. This provides all the machinery to let us transform any (greyscale or colour) morphological filter into a group-invariant filter on greyscale or colour images. We then demonstrate the potential for both subjective and objective improvement in selected tasks.**

***Index Terms*—Mathematical morphology, colour morphology, group invariance, frames, image processing, computer vision**

EDICS Category: TEC-PRC, ELI-COL

I. Introduction

Mathematical morphology is a mathematical framework for processing images. Serra [1] and Heijmans [2] (among others) have produced some of the most comprehensive works on this topic. Although originally used (mostly) for binary and monochrome images, the morphological framework allows for shape-centric processing of any data type that admits a (useful) "lattice structure" (partial order plus some additional structure). Typical applications include denoising and shape analysis.

Shapes are generally taken to be the connected components of the level sets of an image in mathematical morphology. Morphological filters thus rely on the ability to order pixel values. For binary and monochrome images this presents no particular problems. For colour images, on the other hand, it is not so clear what order is best. The only options [3, thm. XV.1] for a so-called "vector lattice" are a lexicographical order and a product order (or a mix of those two).

A lexicographical order considers one value less than or equal to another if its first component is less than the first component of the other, or if their first components are equal and the second component of the one is less than the second component of the other, etc.; a lexicographical order is a total order (any two values are comparable). In contrast, a product order (also known as "marginal ordering") considers one value less than or equal to another if all its components are less than or equal to the corresponding components of the other, and is thus a partial order.

Even though product orders often lead to better results than other orders [4, 5], much of the existing work on colour morphology focuses on defining suitable lexicographical orders (or other total orders) [6–17]. This is to a large degree because of the so-called "false colour problem" that product orders suffer from. Section II examines why this "problem" is considered a problem and how this relates to a lack of invariance to groups of transformations. As a result, we propose to fix, rather than avoid, product orders.

Sections III and IV show how to "lift" a colour space (or any other lattice) to a higher-dimensional one that is invariant to a chosen group of transformations, and how for linear transformations on a vector space this gives rise to a representation based on frames. As an example, we demonstrate how enforcing invariance to changes in hue can make the false colour "problem" much less objectionable. Additionally, in Section V we show that in both noise reduction and texture classification tasks our approach gives a (further) improvement over product orders. Our approach thus mitigates the false colour problem *and* leads to better results than a traditional product order-based approach.

Finally, it should be noted that although this work restricts itself to colour images, the reader can easily substitute "vector-valued image" for colour image, if he or she so pleases. Some of the examples will be somewhat specific to colour images, but the techniques naturally extend to other application areas.

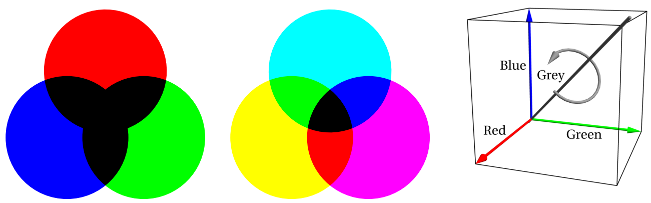
II. The false colour problem

Let us look at the main problem that plagues product orders: the false colour problem. This refers to the fact that the supremum and infimum (least upper bound and greatest lower bound, respectively) of a set of colours need not correspond to any of the colours in the original set, as the colour channels get treated independently. For example, the greatest lower bound of red and blue is black (in the RGB colour space).

Papers like those by Talbot et al. [7] and Serra [17] describe the appearance of "false" colours as a critical problem to be overcome. Many other papers simply take it for granted that false colours are to be avoided. This then forms an argument for developing a range of methods that do not suffer from this problem (but typically, unfortunately, do suffer from other problems that a product order does not suffer from). A related, but slightly different position is taken in a paper by Goutsias et al. [18]. Their main objection to processing channels independently is that this fails to take into account any correlation between channels. This leads them to suggest several remedies (some still based on product orders).

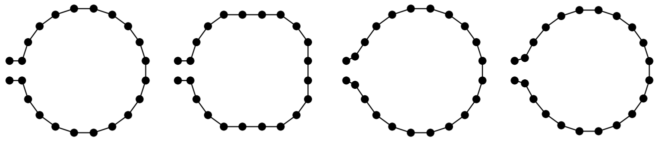
The authors are with the Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands. E-mail: {j.j.van.de.gronde,j.b.t.m.roerdink}@rug.nl

This research is funded by the Dutch National Science Foundation (NWO), project no. 612.001.001.



(a) Primary colours (b) Secondary colours (c) Hue rotation

Figure 1. Infima of all combinations of primary colours (a) and secondary colours (b). Note that the greatest lower bound of any two primary colours is black. On the other hand, if we take cyan and magenta (for example), then the greatest lower bound is blue. The change from red, green and blue to magenta, cyan and yellow can be considered a "hue rotation" by 180°. A hue rotation is a rotation of a colour about the grey axis (c). So a hue rotation of 120° transforms red into green, green into blue and blue into red.



(a) Original (b) Median filter per channel (c) Vector median filter (d) Our result

Figure 2. Astola et al. [21] gave this example to demonstrate one of the problems with product orders. Each figure represents a single sequence of 2D values, plotted as a string of points in the 2D value plane. Filtering the top-left figure using a median filter [22] that operates per channel results in the top-right figure. This result was deemed unsatisfactory, presumably because of the clear flattening of the result along two specific axes. This prompted the development of a proper "vector median filter". In contrast, our approach takes the usual median filter, as it applies to scalar values, and generalizes it to vector-valued data in a way that does not result in any flattening.

So why are false colours considered a problem? In some cases using a product order makes perfect sense [18], and it has been shown to be superior to many alternatives in various contexts [4, 5, 19]. Also, if we take "false colours" to simply be values that were not in the image before, then a greyscale erosion or dilation with a non-flat structuring element will typically introduce "false colours" as well. Linear filters are even almost guaranteed to introduce "false colours", although they have proven very useful. In fact, given that there is usually at least some noise in images, does it really make sense to insist that a filter is not allowed to output colours different from those in the original image? In summary, "false colours" do not appear to be a problem *per se*, and are not limited to multivariate morphology.

Still, false colours are clearly perceived as a problem in mathematical morphology. So what *is* the problem with taking black to be the infimum of red and blue? Black is clearly less red than both, as well as less green and blue than both, and arguably the "largest" such colour in the RGB colour space. We would like to posit that this result is unintuitive because black is also a lot less bright and a lot less purple (for example) than either red or blue. The red and blue used in Fig. 1 are both quite bright, does it then make sense for their greatest lower bound to be the darkest possible colour? Similarly, both red and blue have a reasonably large "purple" component, does it then make sense for their infimum to have no purple at all?

Humans do not perceive colours as being composed of differing amounts of red, green and blue. Roughly speaking, we consider a colour to have a certain hue, saturation and brightness. Typically, the hue of a colour is expressed as an angle, often with red corresponding to zero degrees, and the grey axis being the axis around which the angle is taken (so for pure grey the hue is undefined). When changing the hue of a colour, one usually speaks of applying a "hue rotation". This is illustrated in Fig. 1c.

If a colour is "redder" than another, we consider it larger, if a colour is "greener" we consider it larger, so why not consider a colour that is more yellow or more purple[1] to be larger? This suggests that if we were to rotate the hue of a set of colours, compute their greatest lower bound, and then rotate back, that this should have the same result as not rotating at all. Formally, we can say that we would like the greatest lower bound to be "invariant to hue rotation". In this work we will explore where this desire for invariance takes us.

Interestingly, the importance of invariant properties has been touched on before, but (until now) does not seem to have led to a systematic construction of invariant filters. For example, many references to problems with product orders seem to go back to a paper by Astola et al. [21] (from 1990) on vector median filters. One of the given examples of problems with product orders (see Fig. 2) shows graphically why a rotation-invariant filter might be desirable.

Similarly, a paper from 1993 by Serra [23] stresses the existence of multiple equivalent coordinate systems when using a product order with colour images. He first notes that the product order is the only sensible option when it does not make sense to mix the components of the vectors. He then goes on to note that in many vector spaces (like most colour spaces) there are many, many coordinate systems that could be used, all in principle equivalent. As Serra puts it: "..., there exists an infinite number of other equivalent systems of coordinates for the same vector space; they derive from the first one by rotations, similarities, passages to spherical, cylindric or polar coordinates, etc." It is interesting to note the emphasis put on the operations required to get to such an equivalent system of coordinates, as this is precisely the approach taken here.

At this point, the reader may wonder why this issue does not come up with greyscale images. The answer is that it does, but that it is typically less of an issue than with colour images. For one thing, grey does not have a hue, so (a lack of) hue invariance is not an issue. And since most morphological operators are already invariant to positive scalings and other order-preserving transformations, the only (linear) transformations that are interesting in the greyscale case are those that invert the order. Soille [24] discusses several methods for dealing with such (and similar) transformations; the technique used by Peters [25] can be viewed as a specific instance of the method developed here (when applied to the greyscale case). But note that a lack of invariance is not *necessarily* a problem. For example, astronomical images often have a very natural interpretation in terms of bright regions corresponding to objects (stars) and dark regions corresponding to the background.

[1] It is interesting to note that there is at least some evidence [20] that our colour perception is in fact sensitive to "in-between" hues.

So false colours are not necessarily a problem, and it is likely that the real problem with product orders is a lack of "invariance" to certain transformations. This is backed up by tracing some of the arguments against using product orders through the literature. So rather than abandoning product orders, we will now take a look at "fixing" them.

III. Group-invariant lattices

An operator is called invariant to some transformation if applying the transformation first and the operator second has the same effect as applying the operator first and the transformation second. In other words, the operator and the transformation commute. In this section we formalize the transformations to be invariant to using groups, and present a construction for invariant representations based on lattices. Such representations can then be used to build invariant operators. We will show how to lift a lattice $\mathcal{L}$ that is not invariant to certain group actions to a lattice $\mathcal{M}$ that *is* invariant to group actions, using a simple example that leads to self-dual operators on greyscale images.

A. Definitions

1) Lattices: A greyscale image maps positions to grey values. Take the image domain to be some set E, and assume the grey values lie in $\mathbb{R}$. Greyscale images can then be considered as functions from E to $\mathbb{R}$. The set of all such functions is denoted by $\mathrm{Fun}(E, \mathbb{R})$. Clearly we can define a partial order on this set, such that for any $f, g \in \mathrm{Fun}(E, \mathbb{R})$

$$f \leq g \iff \forall x \in E\,(f(x) \leq g(x))\,.$$

Based on this (partial) order we can define the infimum '$\wedge$' and supremum '$\vee$' by simply taking the pointwise infimum/supremum. The operations '$\wedge$' and '$\vee$' are also called meet and join or greatest lower bound and least upper bound, respectively. A partially ordered set that admits such operations is called a *lattice*.

Note that the above order on $\mathrm{Fun}(E, \mathbb{R})$ essentially applies the usual order on $\mathbb{R}$ component-wise and combines the results. The resulting order is called a "product" order. More generally, if the range of a function is given by a lattice $\mathcal{R}$, then the above construction can be used to turn $\mathrm{Fun}(E, \mathcal{R})$ into a lattice as well. We will make use of this a lot, as just talking about the range is often simpler. Also, we often use $\mathcal{L}$ to denote a general lattice, regardless of what it contains.

Instead of the notation $\mathrm{Fun}(A, B)$ we will sometimes use B^A for the set of functions from A to B. In this case, the arguments of a function are given as indices. For example, applying $f \in \mathrm{Fun}(A, B)$ to $a \in A$ is denoted by $f(a)$, while applying $f \in B^A$ to a is denoted by f_a. This is purely a notational distinction, meant to mimic the way elements of $\mathbb{R}^n$ (with n a positive integer) are usually indexed to give their components.

2) Morphological operators: Several kinds of morphological operators are commonly defined, based on how they behave from the lattice point of view:

- *Erosions* are operators that commute with taking the meet (so if ε is an erosion, then $\varepsilon(a \wedge b) = \varepsilon(a) \wedge \varepsilon(b)$).
- *Dilations* are operators that commute with taking the join.
- *Openings* are operators that are order-preserving (also called increasing), anti-extensive (the output is always less than or equal to the input) and idempotent (repeated application has no effect).
- *Closings* are operators that are order-preserving, extensive (the output is greater than or equal to the input) and idempotent.

In the case of erosions and dilations, the output lattice need not be the same as the input lattice. Furthermore, erosions and dilations are dual, in the sense that for every erosion $\varepsilon : \mathcal{L} \to \mathcal{L}'$ there is a corresponding dilation $\delta : \mathcal{L}' \to \mathcal{L}$ (and vice versa), such that for all $a \in \mathcal{L}'$ and $b \in \mathcal{L}$

$$\delta(a) \leq b \iff a \leq \varepsilon(b).$$

Such a pair of an erosion and a dilation is called an *adjunction*.

A *structural* erosion is defined as the meet over some neighbourhood for each point in an image. This is similar to a convolution, except that we take the meet instead of the sum. The corresponding dilation is then formed by taking the join over the reflected neighbourhood. A structural erosion followed by the corresponding dilation forms a structural opening. Similarly, a structural dilation followed by the corresponding erosion forms a structural closing.

3) Transformation groups: We characterize the transformations to be invariant to as forming a group. Here, a transformation is taken to be an invertible mapping from a lattice $\mathcal{L}$ to itself. A transformation group $\mathbb{T}$ on a lattice $\mathcal{L}$ is then a set of transformations, with composition '$\circ$' as associative binary operation. To be a group it must be closed for composition as well as taking the inverse, and contain an identity element. Usually the operator is not written explicitly, so $\tau_1 \tau_2 = \tau_1 \circ \tau_2$ (with $\tau_1, \tau_2 \in \mathbb{T}$). The identity element is simply given by the identity mapping id: $1_{\mathbb{T}}(a) = \mathrm{id}(a) = a$ for all $a \in \mathcal{L}$. As the elements of a transformation group are mappings from $\mathcal{L}$ to $\mathcal{L}$, we say that $\mathbb{T}$ acts on $\mathcal{L}$.

Often it is convenient to specify a transformation as an anonymous function. For this, the notation $a \mapsto b$ is used, where b is expressed in terms of a. For example: $a \mapsto -a$ is an (anonymous) function that negates its input. The domain and range should be clear from the context.

From now on, $\mathbb{T}$ and $\mathbb{S}$ are used to denote transformation groups acting on the lattices $\mathcal{L}$ and $\mathcal{M}$, respectively (unless specifically stated otherwise).

A group homomorphism ρ is a function from a group $\mathbb{T}$ to another group such that $\rho(\tau_1 \tau_2) = \rho(\tau_1)\,\rho(\tau_2)$ for all $\tau_1, \tau_2 \in \mathbb{T}$. We call a group homomorphism between $\mathbb{T}$ and $\mathbb{S}$ a representation of $\mathbb{T}$ (on $\mathcal{M}$). Similarly, $\rho(\tau) \in \mathbb{S}$ can be said to represent $\tau \in \mathbb{T}$.

Example 1. *(Negation).* Take $\mathbb{T}$ to be the set $\{\mathrm{id}, a \mapsto -a\} \subset \mathrm{Fun}(\mathbb{R}, \mathbb{R})$. Both elements qualify as transformations. Also, the set is closed for composition, contains all inverses of its members, and contains the identity transformation. Thus $\mathbb{T}$ is a transformation group acting on the reals.

Example 2. *(Negation and addition).* Take $\mathbb{T}$ to be the set $\{a \mapsto a + c \mid c \in \mathbb{R}\} \cup \{a \mapsto -a + c \mid c \in \mathbb{R}\}$ (with $a \in \mathbb{R}$). This set is closed for composition, as, with $a, c, d \in \mathbb{R}$,

$(a \mapsto a+c) \circ (a \mapsto a+d)$ is equivalent to $a \mapsto a+(c+d)$, and $(a \mapsto -a+c) \circ (a \mapsto a+d)$ is equivalent to $a \mapsto -a+(c-d)$ (other combinations follow similarly). Also, each transformation is clearly invertible (with the inverse in the set), and the set contains an identity element ($a \mapsto a+0$). This leads us to conclude that $\mathbb{T}$ is a transformation group.

Example 3. *(Hue rotations).* Consider the set of all rotations around the grey axis, as illustrated in Fig. 1c. We will call the elements of this set *hue rotations*. Clearly, a hue rotation followed by another is again a hue rotation. Also, for every hue rotation there is an inverse hue rotation, and the hue rotation by 0° obviously functions as identity element. We can thus conclude that the set of all hue rotations is a (transformation) group, acting on the RGB colour space.

4) Invariance: An operator $\phi : \mathcal{L} \to \mathcal{L}$ is called invariant to a transformation $\tau : \mathcal{L} \to \mathcal{L}$ if $\phi(\tau(f)) = \tau(\phi(f))$. We call a lattice $\mathcal{L}$ invariant to τ if $\tau(a) \wedge \tau(b) = \tau(a \wedge b)$ for all $a, b \in \mathcal{L}$, and similarly for the join. If $\mathcal{L}$ is invariant to τ, then τ is a lattice-homomorphism of $\mathcal{L}$. And since τ is assumed to be a bijection, it is even an *automorphism*. These terms are discussed in more detail by Birkhoff [3, §II.5].

An operator on a lattice $\mathcal{L}$, or the lattice itself, is said to be invariant to a transformation group $\mathbb{T}$ if it is invariant to all group actions in $\mathbb{T}$. Equivalently, we can speak about a $\mathbb{T}$-invariant operator or lattice. Alternatively, as Heijmans [2, ch. 5] does, we can say that $\mathbb{T}$ is an automorphism group. The group of *all* automorphisms of $\mathcal{L}$ is denoted by $\mathrm{Aut}(\mathcal{L})$[2].

Given lattices $\mathcal{L}$ and $\mathcal{M}$, an injective mapping $\Lambda : \mathcal{L} \to \mathcal{M}$ and a transformation group $\mathbb{T}$ on $\mathcal{L}$, we say that $\mathcal{M}$ is invariant to $\mathbb{T}$ on $\mathcal{L}$ (under Λ), if there is a representation $\rho : \mathbb{T} \to \mathrm{Aut}(\mathcal{M})$ of $\mathbb{T}$, such that $\Lambda \circ \tau = \rho(\tau) \circ \Lambda$ for all $\tau \in \mathbb{T}$. Similarly, we say that an operator $\psi : \mathcal{M} \to \mathcal{M}$ is invariant to $\mathbb{T}$ on $\mathcal{L}$ if it commutes with all $\rho(\tau)$.

Example 4. *(Addition invariance).* The reals $\mathbb{R}$ form a lattice under their usual order. Take $\mathbb{T}$ to be the set $\{a \mapsto a+c \mid c \in \mathbb{R}\}$ (with $a \in \mathbb{R}$) of all additions by some real. We can then see that for all $\tau \in \mathbb{T}$ we have $\tau(a) \wedge \tau(b) = (a+c) \wedge (b+c) = (a \wedge b) + c = \tau(a \wedge b)$, and similarly for the join operation. Therefore, $\mathbb{R}$ is invariant to "addition".

Example 5. *(Negation invariance).* An operator $\phi : \mathrm{Fun}(E, \mathbb{R}) \to \mathrm{Fun}(E, \mathbb{R})$ on greyscale images is called self-dual if $\phi(-f) = -\phi(f)$. So a self-dual operator can also be said to be an operator invariant to negation, or more formally, to the transformation group introduced in Example 1. Before looking at negation-invariant operators, we will focus on getting a negation-invariant lattice. To see if $\mathrm{Fun}(E, \mathbb{R})$ is negation-invariant, it is sufficient (because of the product order structure of the lattice) to check whether $\mathbb{R}$ is negation-invariant. Unfortunately, it is not. This can be seen quite easily: $-1 \wedge -2 = -2 \neq -(1 \wedge 2) = -1$.

We can, in fact, see that no lattice on $\mathbb{R}$ can be invariant to negation. For this purpose, assume that we *do* have a lattice on $\mathbb{R}$ that is invariant to negation. Then, for all $a \in \mathbb{R}$, by the commutativity of the meet and the assumed negation invariance of the lattice, we have: $a \wedge -a = -a \wedge a = -(a \wedge -a)$. Thus, $a \wedge -a$ must be zero, as that is the only real equal to its own negation. Similarly, we can show, that $a \vee -a$ must also be zero. Let '$\preceq$' denote the partial order used in the hypothetical lattice. The previous considerations then imply both $0 \preceq a$ and $a \preceq 0$, and thus that $a = 0$. Obviously not all $a \in \mathbb{R}$ are equal to zero. We can thus conclude that no lattice on $\mathbb{R}$ can be invariant to negation. Below we will show how we can construct a new lattice to get around this problem.

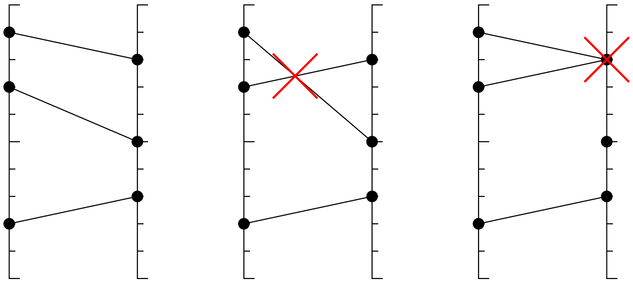


Figure 3. Imagine that a totally ordered set is represented by a line (here with tick marks). A bijection on such a set can be represented by drawing the line twice and drawing line segments (links) between positions on the two copies of the line, such that no two links share an endpoint and all positions on both lines are covered by endpoints. An automorphism is then characterized by not having any crossing links. So the subset of links shown in the left-most illustration is consistent with an automorphism, while the other two illustrations clearly cannot represent an automorphism (the last one is not even a bijection).

B. The nature of automorphisms

Existing work on group-invariant morphology, done by Heijmans [2] and Roerdink [26], deals with groups of automorphisms. Or, put differently, they consider lattices that are already invariant to a certain group. This allows for very neat constructions, but what kind of transformations are realizable as automorphisms?

An automorphism on the lattice of reals $\mathbb{R}$ (and other totally ordered lattices) can be pictured as in Fig. 3: a map from $\mathbb{R}$ to itself such that we never invert the order of any two elements. In most other cases we are interested in (like images), the lattice can be considered to be of the form $\mathrm{Fun}(E, \mathbb{R})$. For example, a greyscale image can be regarded as a function that associates a real with every point in some set (usually a regular grid). Interestingly, it can be shown that automorphisms on such a lattice must be built up by permutation and the per-position application of automorphisms of $\mathbb{R}$.

So, in essence, an automorphism on a greyscale image is anything that permutes the pixels and fiddles with each pixel value in an order-preserving manner. Thus, only hue rotations by multiples of 120 degrees are automorphisms on the RGB colour space, as those are the only hue rotations mapping each basis vector into another basis vector (see Fig. 1c); other hue rotations would always produce mixes of the red, green and blue components. In general, for a lattice $\mathrm{Fun}(E, \mathcal{R})$, any permutation operator based on a permutation of the domain

[2] That $\mathrm{Aut}(\mathcal{L})$ indeed forms a transformation group can be seen be realizing that the composition of two automorphisms must again be an automorphism, that the identity mapping is an automorphism, and that the inverse of an automorphism is an automorphism.

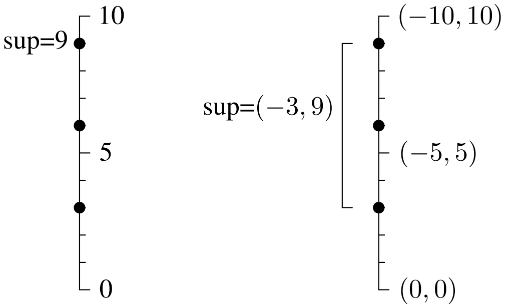


Figure 4. Illustration of a very simple lattice meant to create self-dual operators. On the left we see that in the original lattice, the supremum of the numbers 3, 6 and 9 is 9. On the right we see that if we take the lattice described in Example 6, that then the supremum of the lifted versions of the same set of numbers is $(-3, 9)$.

E is clearly an automorphism. This suggests looking for a way to create a lattice in which all group actions in $\mathbb{T}$ are representable by permutations.

C. Construction

Assume a lattice $\mathcal{L}$ and a transformation group $\mathbb{T}$ on $\mathcal{L}$ are given, such that $\mathcal{L}$ is not invariant to $\mathbb{T}$. What lattice $\mathcal{M}$ based on $\mathcal{L}$ can we then construct, such that the transformations in $\mathbb{T}$ become automorphisms on $\mathcal{M}$? We already concluded that permutation operators lead to automorphisms. So we would like to create a lattice $\mathcal{M}$ and mapping $\Lambda : \mathcal{L} \to \mathcal{M}$ such that $\Lambda(\tau(a))$ is a permuted version of $\Lambda(a)$ for all $\tau \in \mathbb{T}$ and $a \in \mathcal{L}$. The basic idea of our construction for such a lattice and mapping is shown in Example 6.

Example 6. *(Negation invariant lattice).* To construct a new lattice, based on the reals, that *is* invariant to negation, we take the new lattice to consist of all pairs of reals (ordered by a product order), and map any real to a pair of reals: the number itself and its negation. So we map a to $(-a, a)$, and say that $(a_1, a_2) \le (b_1, b_2) \iff a_1 \le b_1$ and $a_2 \le b_2$, this is illustrated in Fig. 4. Since a maps to $(-a, a)$, we have that $-a \mapsto (a, -a)$. The group $\{\mathrm{id}, a \mapsto -a\}$ (on the reals) introduced in Example 1 can thus be represented by the group $\{\mathrm{id}, (a_1, a_2) \mapsto (a_2, a_1)\}$ on pairs of reals. It should be obvious that the given product order is invariant to swapping of the components, and thus to negation on the reals.

It is important to note that the new lattice is *not* invariant to the group $\{\mathrm{id}, (a_1, a_2) \mapsto (-a_1, -a_2)\}$ (which is another obvious way to represent negation on the reals). Groups on the original lattice can often be represented by multiple groups on the new lattice. What matters is that there is at least one such group to which the new lattice is invariant.

Interestingly, the above construction also preserves invariance to addition on the reals. The mapping $a \mapsto a + c$ can be represented by $(a_1, a_2) \mapsto (a_1 - c, a_2 + c)$. Note that c is subtracted from a_1 to ensure that first adding c and then lifting results in the same pair of numbers as first lifting and then performing the corresponding transformation. As we clearly have that $(a_1, a_2) \le (b_1, b_2) \iff (a_1 - c, a_2 + c) \le (b_1 - c, b_2 + c)$, the product order is still invariant to addition on the reals. This will be made more precise by Lemma 2.

To formalize the above construction, take $\mathcal{M}$ to be $\mathcal{L}^{\mathbb{T}}$ (using a product order over $\mathbb{T}$), and Λ to be a mapping that lifts elements of $\mathcal{L}$ to elements of $\mathcal{M}$, given by $\Lambda(a)_\tau = \tau(a)$. The idea is that this mapping allows us to represent $\mathbb{T}$ by permutations on $\mathcal{M}$, to which $\mathcal{M}$ is definitely invariant.

Lemma 1. *$P_\tau : \mathcal{M} \to \mathcal{M}$, defined by $P_\tau(u)_\sigma = u_{\sigma\tau}$ ($\tau, \sigma \in \mathbb{T}$), is a permutation operator. That is, for every $\tau \in \mathbb{T}$ there is a bijection $\pi : \mathbb{T} \to \mathbb{T}$ (depending on τ) such that $P_\tau(u)_\sigma = u_{\pi(\sigma)}$.*

Proof. Given a $\tau \in \mathbb{T}$, simply take $\pi(\sigma) = \sigma\tau$. Due to the properties of a group this is clearly a bijection. This proves the lemma. □

Theorem 1. *Given a lattice $\mathcal{L}$, and a transformation group $\mathbb{T}$ on $\mathcal{L}$, the lattice $\mathcal{M} = \mathcal{L}^{\mathbb{T}}$ is invariant to $\mathbb{T}$ on $\mathcal{L}$ (under Λ). Elements of $\mathcal{L}$ are lifted to $\mathcal{M}$ using Λ, defined by $\Lambda(a)_\tau = \tau(a)$, while $\mathbb{T}$ is represented on $\mathcal{M}$ by $\rho : \mathbb{T} \to \mathrm{Aut}(\mathcal{M})$, defined by $\rho(\tau) = P_\tau$.*

Proof. To show the statement to be true, we give a representation ρ of $\mathbb{T}$, and show that we have $\Lambda(\tau(a)) = \rho(\tau)(\Lambda(a))$.

Take $\rho(\tau) = P_\tau$, with P_τ as defined above. This is a (group) homomorphism from $\mathbb{T}$ to $\mathrm{Aut}(\mathcal{M})$, and thus a representation of $\mathbb{T}$, if (and only if) $P_\tau \in \mathrm{Aut}(\mathcal{M})$ and $P_\tau \circ P_\sigma = P_{\tau\sigma}$ for all $\tau, \sigma \in \mathbb{T}$. Recalling the definition from Lemma 1 in the third step, we see that (for all $u, v \in \mathcal{M}$, $\tau \in \mathbb{T}$)

$$\begin{aligned} u \le v &\iff \forall \sigma \in \mathbb{T}\,(u_\sigma \le v_\sigma) \\ &\iff \forall \sigma \in \mathbb{T}\,(u_{\sigma\tau} \le v_{\sigma\tau}) \\ &\iff \forall \sigma \in \mathbb{T}\,(P_\tau(u)_\sigma \le P_\tau(v)_\sigma) \\ &\iff P_\tau(u) \le P_\tau(v). \end{aligned}$$

This establishes that $P_\tau \in \mathrm{Aut}(\mathcal{M})$ for all $\tau \in \mathbb{T}$. We also have

$$\begin{aligned} (P_{\tau_1} \circ P_{\tau_2})(u)_\sigma = P_{\tau_1}(P_{\tau_2}(u))_\sigma &= P_{\tau_2}(u)_{\sigma\tau_1} \\ &= u_{\sigma\tau_1\tau_2} = P_{\tau_1\tau_2}(u)_\sigma. \end{aligned}$$

This shows that the function ρ that maps a $\tau \in \mathbb{T}$ to P_τ, is in fact a group homomorphism from $\mathbb{T}$ to $\mathrm{Aut}(\mathcal{M})$.

To make the final piece of the puzzle fit, consider that for all $a \in \mathcal{L}$ and $\tau, \sigma \in \mathbb{T}$,

$$\begin{aligned} \Lambda(\tau(a))_\sigma = \sigma(\tau(a)) = (\sigma\,\tau)(a) &= \Lambda(a)_{\sigma\tau} \\ &= P_\tau(\Lambda(a))_\sigma = \rho(\tau)(\Lambda(a))_\sigma. \end{aligned}$$

Thus $\Lambda \circ \tau = \rho(\tau) \circ \Lambda$. We have now shown that both requirements for $\mathcal{M}$ to be invariant to $\mathbb{T}$ on $\mathcal{L}$ (under Λ) are met, using the representation $\rho : \mathbb{T} \to \mathrm{Aut}(\mathcal{M})$, defined by $\rho(\tau) = P_\tau$. □

The above constructions lead to a lattice $\mathcal{M} = \mathcal{L}^{\mathbb{T}}$ that is invariant to some $\mathbb{S}$ on $\mathcal{M}$ that represents $\mathbb{T}$ on $\mathcal{L}$. The group $\mathbb{S}$ equals $\{P_\tau \mid \tau \in \mathbb{T}\}$, as each $\tau \in \mathbb{T}$ was mapped to P_τ using a homomorphism. Using the methods introduced by Heijmans [2] and Roerdink [26], it is now possible to construct morphological operators on $\mathcal{M}$ that are invariant to $\mathbb{S}$, and thus to $\mathbb{T}$ on $\mathcal{L}$. Alternatively, we can easily lift an operator $\phi_0 : \mathcal{L} \to \mathcal{L}$ to an $\mathbb{S}$-invariant operator $\psi : \mathcal{M} \to \mathcal{M}$ by taking

$$\begin{tikzcd} \mathcal{L} \arrow[r, "\Lambda"] \arrow[d, dotted, "\phi?"'] & \mathcal{M} \arrow[d, "\psi"] \\ \mathcal{L} & \mathcal{M} \arrow[l, dotted, "?"'] \end{tikzcd}$$

Figure 5. We have shown how to construct a mapping Λ that can lift $\mathcal{L}$ to a $\mathbb{T}$-invariant lattice $\mathcal{M}$. On this lattice we can easily define a $\mathbb{T}$-invariant operator ψ. But to use this to construct a $\mathbb{T}$-invariant operator ϕ on $\mathcal{L}$, we still need some way of getting back to $\mathcal{L}$.

$\psi(u)_\tau = \phi_0(u_\tau)$. It is easily verified that this operator is $\mathbb{S}$-invariant: $\psi(P_\tau(u))_\sigma = \phi_0(P_\tau(u)_\sigma) = \phi_0(u_{\sigma\tau}) = \psi(u)_{\sigma\tau} = P_\tau(\psi(u))_\sigma$. The general situation is illustrated in Fig. 5.

Now, how to get back to $\mathcal{L}$? Although in some cases it might be sufficient to simply use the new lattice, in most cases (like colour images), we will ultimately want an answer in the form of some element of the original lattice $\mathcal{L}$. Section IV is devoted to exploring the case that $\mathcal{L}$ is a vector space, and $\mathbb{T}$ consists entirely of linear transformations. It will then turn out to be easy to project back to $\mathcal{L}$ in a sensible way.

D. Inheriting invariance

In Example 6 we saw that the newly constructed lattice was not just invariant to the transformation group we used in the construction, it also "inherited" invariance to addition from the original lattice. In this section we will analyse this phenomenon in more detail.

Suppose we have an automorphism α on the original lattice $\mathcal{L}$ (or really any arbitrary transformation on $\mathcal{L}$), how could we lift it to $\mathcal{M}$? It would have to map to some $\alpha_\mathcal{M}$ such that $\Lambda(\alpha(a)) = \alpha_\mathcal{M}(\Lambda(a))$ for all $a \in \mathcal{L}$. For all $\sigma \in \mathbb{T}$

$$\begin{aligned}\Lambda(\alpha(a))_\sigma &= (\sigma\,\alpha)(a) = (\sigma\,\alpha\,\sigma^{-1}\,\sigma)(a)\\ &= (\sigma\,\alpha\,\sigma^{-1})(\sigma(a)) = (\sigma\,\alpha\,\sigma^{-1})(\Lambda(a)_\sigma).\end{aligned}$$

This suggests the definition (with $u \in \mathcal{M}$):

$$\alpha_\mathcal{M}(u)_\sigma = (\sigma\,\alpha\,\sigma^{-1})(u_\sigma).$$

The mapping $\alpha \mapsto \sigma\,\alpha\,\sigma^{-1}$ is a *conjugation*, and $\sigma\,\alpha\,\sigma^{-1}$ a conjugate, with σ and α both transformations on the same lattice. A transformation group $\mathbb{T}_1$ is *closed for conjugation* by $\tau \in \mathbb{T}_2$ if and only if $\mathbb{T}_1 = \tau\,\mathbb{T}_1\,\tau^{-1} = \{\tau\,\sigma\,\tau^{-1} \mid \sigma \in \mathbb{T}_1\}$. Similarly, $\mathbb{T}_1$ is closed for conjugation by $\mathbb{T}_2$ if and only if $\forall \tau \in \mathbb{T}_2 (\mathbb{T}_1 = \tau\,\mathbb{T}_1\,\tau^{-1})$. If $\mathbb{T}_1$ is closed for conjugation by $\mathbb{T}_2$, then $\mathbb{T}_1\,\mathbb{T}_2 = \{\tau_1\,\tau_2 \mid \tau_1 \in \mathbb{T}_1 \text{ and } \tau_2 \in \mathbb{T}_2\}$ equals $\mathbb{T}_2\,\mathbb{T}_1$ and forms a group.

Now define $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ as the largest group of automorphisms on $\mathcal{L}$ that is closed for conjugation by $\mathbb{T}$ (so $\mathrm{Aut}_\mathbb{T}(\mathcal{L}) = \tau\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})\,\tau^{-1}$ for all $\tau \in \mathbb{T}$). This is closely related to the concept of a core or normal interior, as defined by Robinson [27, p. 16] and Suzuki [28, p. 65] for example. If $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ is a subgroup of $\mathbb{T}$, then it is called a normal subgroup. Very slightly generalizing the notation used by Suzuki, we consider $\mathrm{Core}_\mathbb{T}(\mathbb{G})$ to be the largest subgroup of $\mathbb{G}$ that is closed for conjugation by $\mathbb{T}$. We then take

$$\mathrm{Aut}_\mathbb{T}(\mathcal{L}) = \mathrm{Core}_\mathbb{T}(\mathrm{Aut}(\mathcal{L})) = \bigcap\nolimits_{\tau\in\mathbb{T}} \tau\,\mathrm{Aut}(\mathcal{L})\,\tau^{-1}.$$

As each of the conjugates $\tau\,\mathrm{Aut}(\mathcal{L})\,\tau^{-1}$ is a group, the intersection is a group as well. With some more effort, it can be seen that $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ as defined above, is indeed the largest subgroup of $\mathrm{Aut}(\mathcal{L})$ that is closed for conjugation by $\mathbb{T}$.

Getting back to inheriting automorphisms, we can now see that for $\alpha_\mathcal{M}$ to be an automorphism on $\mathcal{M}$, all the conjugates of α must be automorphisms on $\mathcal{L}$. This leads us to Lemma 2.

Lemma 2. *The lattice $\mathcal{M} = \mathcal{L}^\mathbb{T}$ is invariant to the group $\{\alpha_\mathcal{M} \mid \alpha \in \mathrm{Aut}_\mathbb{T}(\mathcal{L})\}$, and thus to $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ on $\mathcal{L}$.*

Proof. First recall that $\mathcal{M}$ is invariant to $\alpha_\mathcal{M}$ if $\alpha_\mathcal{M}(u) \wedge \alpha_\mathcal{M}(v) = \alpha_\mathcal{M}(u \wedge v)$ (and similarly for the join). From the definitions, we have (with $\tau \in \mathbb{T}$ and $\alpha \in \mathrm{Aut}_\mathbb{T}(\mathcal{L})$)

$$\begin{aligned}(\alpha_\mathcal{M}(u) \wedge \alpha_\mathcal{M}(v))_\tau &= \alpha_\mathcal{M}(u)_\tau \wedge \alpha_\mathcal{M}(v)_\tau\\ &= (\tau\,\alpha\,\tau^{-1})(u_\tau) \wedge (\tau\,\alpha\,\tau^{-1})(v_\tau).\end{aligned}$$

Now, since $\tau\,\alpha\,\tau^{-1}$ is (by definition) also in $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ for all $\tau \in \mathbb{T}$:

$$\begin{aligned}(\alpha_\mathcal{M}(u) \wedge \alpha_\mathcal{M}(v))_\tau &= (\tau\,\alpha\,\tau^{-1})(u_\tau \wedge v_\tau)\\ &= (\tau\,\alpha\,\tau^{-1})((u \wedge v)_\tau) = \alpha_\mathcal{M}(u \wedge v)_\tau.\end{aligned}$$

Thus $\alpha_\mathcal{M}(u) \wedge \alpha_\mathcal{M}(v) = \alpha_\mathcal{M}(u \wedge v)$ for all $\alpha \in \mathrm{Aut}_\mathbb{T}(\mathcal{L})$, and $\mathcal{M}$ is invariant to the group of all $\alpha_\mathcal{M}$. Since the mapping $\alpha \mapsto \alpha_\mathcal{M}$ can easily be seen to be a group homomorphism, and by construction $\Lambda \circ \alpha = \alpha_\mathcal{M} \circ \Lambda$, this means that $\mathcal{M}$ is invariant to $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ on $\mathcal{L}$, concluding the proof. □

Making use of the fact that $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ is a group and equal to $\mathrm{Aut}_\mathbb{T}(\mathcal{L})\,\mathbb{T}$, we can combine the previous results.

Corollary 1. *If $\mathbb{T} \cap \mathrm{Aut}_\mathbb{T}(\mathcal{L}) = \{\mathrm{id}\}$, then there is an extension of the representation ρ from Theorem 1 to the transformation group $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ on $\mathcal{L}$. Thus, the lattice $\mathcal{M}$ is invariant to $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$.*

Proof. We give a construction for $\rho : \mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L}) \to \mathrm{Aut}(\mathcal{M})$, proving the statement. Like both previous constructions ρ maps the identity transformation on $\mathcal{L}$ to the identity transformation on $\mathcal{M}$. For all (other) group actions in $\mathbb{T}$ we use the previous definition. For all group actions α in $\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ we use $\rho(\alpha) = \alpha_\mathcal{M}$. Now, without loss of generality (as $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L}) = \mathrm{Aut}_\mathbb{T}(\mathcal{L})\,\mathbb{T}$), all other group actions can be considered as having the form $\tau\,\alpha$ (with $\tau \in \mathbb{T}$ and $\alpha \in \mathrm{Aut}_\mathbb{T}(\mathcal{L})$). These "mixed" group actions are mapped to $\rho(\tau)\,\rho(\alpha)$.

To verify that the extended ρ is still a group homomorphism we can check that $\rho(\alpha)\,\rho(\tau) = \rho(\tau)\,\rho(\beta)$ for $\alpha \in \mathrm{Aut}_\mathbb{T}(\mathcal{L})$, $\tau \in \mathbb{T}$ and $\beta = \tau^{-1}\,\alpha\,\tau$. This is easily verified (with $\sigma \in \mathbb{T}$):

$$\begin{aligned}(\rho(\alpha)\,\rho(\tau)\,u)_\sigma &= \alpha_\mathcal{M}(P_\tau(u))_\sigma = (\sigma\,\alpha\,\sigma^{-1})(P_\tau(u)_\sigma)\\ &= (\sigma\,\alpha\,\sigma^{-1})(u_{\sigma\tau}) = (\sigma\,\tau\,\beta\,\tau^{-1}\,\sigma^{-1})(u_{\sigma\tau}) = \beta_\mathcal{M}(u)_{\sigma\tau}\\ &= P_\tau(\beta_\mathcal{M}(u))_\sigma = (\rho(\tau)\,\rho(\beta)\,u)_\sigma.\end{aligned}$$

We have constructed a mapping ρ from $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ to $\mathrm{Aut}(\mathcal{M})$, and verified that it is a representation. Also, it should be clear that we still have $\Lambda(\tau(a)) = \rho(\tau)(\Lambda(a))$ for all $\tau \in \mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$. We can now conclude that $\mathbb{M}$ is invariant to $\mathbb{T}\,\mathrm{Aut}_\mathbb{T}(\mathcal{L})$ on $\mathcal{L}$. □

IV. Invariant frames

Above, a lattice $\mathcal{M}$ invariant to some transformation group $\mathbb{T}$ on a lattice $\mathcal{L}$ was easily constructed. We did not, however, have any (satisfactory) way of going back to $\mathcal{L}$. Interestingly, if the original lattice is a vector space (with some additional properties), and the group transformations are linear operators acting on that space, then we can use frames [29, 30] to get back to $\mathcal{L}$ in a useful way.

A. Preliminaries

1) Vector lattices and Hilbert spaces: If a lattice $\mathcal{L}$ is also a vector space, and invariant to addition and multiplication by positive scalars, then it is called a vector lattice (or Riesz space). It is known that every vector lattice must be built up from $\mathbb{R}$ by "direct and lexicographic union", as Birkhoff [3, thm. XV.1] puts it. Translated to our setting this means that we can only use a product order, a lexicographical order, or some combination of the two, if we wish to have a vector lattice. Here we stick to using a product order.

To define a product order on a vector lattice, we use a basis. In this context, a basis of a d-dimensional vector space $\mathcal{L}$ is a set of d vectors $\{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$, that together span $\mathcal{L}$. For every vector $\mathbf{a} \in \mathcal{L}$ there is then a unique way to write it as a weighted sum of the basis vectors: $\mathbf{a} = a_1\,\mathbf{e}_1 + \cdots + a_d\,\mathbf{e}_d$. With $\mathcal{K}$ the index set $\{1, 2, \cdots, d\}$, we then have $\mathbf{a} \leq \mathbf{b} \iff \forall k \in \mathcal{K}\,(a_k \leq b_k)$. We say that the vector lattice is ordered on the basis $\{\mathbf{e}_k\}_{k\in\mathcal{K}}$. Note that we will assume all vector spaces to be over the reals, so the a_k will be real.

If a vector space is a Hilbert space, then it also has a positive definite inner product, denoted by '$\cdot$'[3]. The weights $\{a_k\}_{k\in\mathcal{K}}$ are then given by $a_k = \mathbf{e}^k \cdot \mathbf{a}$. Here $\{\mathbf{e}^k\}_{k\in\mathcal{K}}$ is a set of vectors such that $\mathbf{e}_k \cdot \mathbf{e}^m = \delta_{k,m}$ ($k, m \in \mathcal{K}$), with $\delta_{k,m}$ the Kronecker delta (1 if $k = m$, 0 otherwise). There is always exactly one such set, called the "dual basis". If a vector lattice is (also) a Hilbert space, we will call it a Hilbert lattice.

Example 7. *(The reals as a Hilbert space).* As discussed earlier, the usual lattice on the reals is invariant to addition. It is also invariant to multiplication by a positive real; thus, the reals form a (very simple) vector lattice, with the basis $\{1\}$ for example. Furthermore, using ordinary multiplication as "inner product", $\mathbb{R}$ is indeed a Hilbert space. The dual basis is then simply equal to the basis $\{1\}$.

Alternatively, $\{2\}$, or any other singleton set with a real, could also be used as a basis. In general, if we take $\{\mathbf{e}\}$ (with $\mathbf{e}$ non-zero and real) to be our basis over the reals, then $\{1/\mathbf{e}\}$ is the dual basis. For any $\mathbf{a} \in \mathbb{R}$, we then have $\mathbf{a} = a_1\,\mathbf{e}$, with $a_1 = \mathbf{a} \cdot (1/\mathbf{e}) = \mathbf{a}/\mathbf{e}$.

2) Frames: Instead of having a set with precisely enough vectors to span a Hilbert space $\mathcal{L}$, *more* than enough vectors may also be used. A set of vectors $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ spanning $\mathcal{L}$ (not necessarily finite or even countable), is called a frame if there are *finite and positive* constants A and B (called the lower and upper frame bound, respectively), such that for any $\mathbf{a} \in \mathcal{L}$

$$A\,\|\mathbf{a}\|^2 \leq \|F\mathbf{a}\|^2 \leq B\,\|\mathbf{a}\|^2. \qquad (1)$$

Here the linear operator $F : \mathcal{L} \to \mathbb{R}^{\mathcal{I}}$ is called the analysis operator, and is defined by $(F\mathbf{a})_i = \mathbf{f}_i \cdot \mathbf{a}$ for all $i \in \mathcal{I}$. The squared norm of $\mathbf{a} \in \mathcal{L}$ is given by $\|\mathbf{a}\|^2 = \mathbf{a} \cdot \mathbf{a}$. Similarly, we take $\|F\mathbf{a}\|^2 = F\mathbf{a} \cdot F\mathbf{a}$. For simplicity, just assume that an inner product on $\mathbb{R}^{\mathcal{I}}$ exists, we will explicitly give the inner product being used in each example. The general case, using measure theory, is covered by Christensen [30, §5.8].

The adjoint of the analysis operator of a frame is called the synthesis operator. It is characterized, as usual, by $(F\mathbf{a}) \cdot \mathbf{u} = \mathbf{a} \cdot (F^*\mathbf{u})$ (for all $\mathbf{a} \in \mathcal{L}$ and $\mathbf{u} \in \mathbb{R}^{\mathcal{I}}$). A dual frame of $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ is a frame $\{\hat{\mathbf{f}}_i\}_{i\in\mathcal{I}}$ whose synthesis operator $\hat{F}^*$ is a left inverse for the analysis operator F (so $\hat{F}^* F = \mathrm{id}$). When the frame bounds A and B are equal, the frame is called *tight*, and $\{\frac{1}{A}\mathbf{f}_i\}_{i\in\mathcal{I}}$ is a dual frame. In particular, when $A = B = 1$, a frame is its own dual. The pair of a frame and a dual frame together will be called a frame-dual pair.

It should be stressed that in general there is not just one dual frame. In most cases, it is therefore important to specify which dual frame is being used. One particularly important choice is the *canonical* dual frame $\{(F^*F)^{-1}\mathbf{f}_i\}_{i\in\mathcal{I}}$. This choice has the interesting property of minimizing $\mathbf{u} - F\hat{F}^*\mathbf{u}$ in a least-squares sense. This follows from the fact that for this frame $\hat{F}^* = (F^*F)^{-1}F^*$, which is the Moore-Penrose pseudoinverse (or, more generally, the maximal generalized inverse) of F. More details are given by Christensen [30, §1.4] and Ben-Israel and Greville [31, ch. 1 and 8]. Note that a tight frame with $A = B = 1$ is not just its own dual, it is its own canonical dual.

A frame-dual pair $(\{\mathbf{f}_i\}, \{\hat{\mathbf{f}}_i\})$ for $\mathcal{L}$ is considered invariant to a transformation group $\mathbb{T}$ on $\mathcal{L}$, if there is a representation $\rho : \mathbb{T} \to \mathbb{S}$ of $\mathbb{T}$ on $\mathbb{R}^{\mathcal{I}}$, such that $\mathbb{R}^{\mathcal{I}}$ is invariant to $\mathbb{S}$, and for all $\tau \in \mathbb{T}$

$$F \circ \tau = \rho(\tau) \circ F \quad\text{and}\quad \hat{F}^* \circ \rho(\tau) = \tau \circ \hat{F}^*. \qquad (2)$$

Also, if a frame satisfies the first part, then it is considered $\mathbb{T}$-invariant. The relevance of this definition of a $\mathbb{T}$-invariant frame-dual pair is that if we have an operator ψ on $\mathbb{R}^{\mathcal{I}}$ that is invariant to automorphisms of $\mathbb{R}^{\mathcal{I}}$, then the operator ϕ on $\mathcal{L}$, given by $\phi = \hat{F}^* \circ \psi \circ F$, is invariant to $\mathbb{T}$.

Example 8. *(Negation invariant frame-dual pair).* In Example 6 we mapped $\mathbf{a}$ onto $(-\mathbf{a}, \mathbf{a})$ (with $\mathbf{a} \in \mathbb{R}$). Here we write $\mathbf{a}$ in boldface to stress that we are talking about a vector in the Hilbert space $\mathbb{R}$. If we define the frame $\{-1, 1\}$, then the corresponding analysis operator $F : \mathbb{R} \to \mathbb{R}^2$ implements exactly the mapping $\mathbf{a} \mapsto (-\mathbf{a}, \mathbf{a})$. The analysis operator corresponding to the canonical dual frame is then given by

$$\hat{F} = \begin{bmatrix}-1\\1\end{bmatrix}\left(\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)^{-1} = \begin{bmatrix}-1\\1\end{bmatrix}2^{-1} = \begin{bmatrix}-1/2\\1/2\end{bmatrix}.$$

We can now verify that this frame-dual pair is invariant to negation on the reals. In Example 6 we already saw that the first half of Eq. (2) can be satisfied by choosing to represent negation by swapping. The second half is satisfied by the dual frame found above:

$$\begin{aligned}\hat{F}^*\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} &= \begin{bmatrix}-1/2 & 1/2\end{bmatrix}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\\ &= \begin{bmatrix}1/2 & -1/2\end{bmatrix} = -\begin{bmatrix}-1/2 & 1/2\end{bmatrix}.\end{aligned}$$

[3] Note that a vector space must also be "complete" to be called a Hilbert space. In our examples the vector spaces are all complete.

Note that any set of the form $\{-p, q\}$ with $p+q=1$ can be seen to form a frame dual to $\{-1, 1\}$, but (in this case) only the canonical dual frame results in a negation-invariant frame-dual pair:

$$\begin{bmatrix}-p & q\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}q & -p\end{bmatrix} = -\begin{bmatrix}-p & q\end{bmatrix} \iff p = q,$$

and since $p+q=1$ we have $p=q=1/2$.

B. Constructing invariant frame-dual pairs

In Section III-C we showed how to construct a lattice $\mathcal{M}$ that was invariant to a given transformation group $\mathbb{T}$ on the lattice $\mathcal{L}$. This construction boiled down to taking $\mathcal{M} = \mathcal{L}^{\mathbb{T}}$ and mapping elements from $\mathcal{L}$ to $\mathcal{M}$ using Λ, with $\Lambda(\mathbf{a})_\tau = \tau\,\mathbf{a}$ ($\mathbf{a} \in \mathcal{L}$ and $\tau \in \mathbb{T}$). We write $\mathbf{a}$ in boldface to stress that we are now talking about a vector in the Hilbert lattice $\mathcal{L}$.

Now suppose that $\mathbb{T}$ is a group of *linear* transformations on $\mathcal{L}$. We then have (with $\mathbf{a} \in \mathcal{L}$, $\tau \in \mathbb{T}$ and $k \in \mathcal{K}$)

$$\Lambda(\mathbf{a})_{\tau,k} = (\tau\,\mathbf{a})_k = \mathbf{e}^k \cdot (\tau\,\mathbf{a}) = (\tau^*\,\mathbf{e}^k) \cdot \mathbf{a}.$$

So $\Lambda(\mathbf{a})_i = \mathbf{f}_i \cdot \mathbf{a}$ for some set of vectors $\{\mathbf{f}_i\}_{i\in\mathcal{I}} \subseteq \mathcal{L}$, with $\mathcal{I} = \mathbb{T} \times \mathcal{K}$ and $\mathbf{f}_i = \tau^*\,\mathbf{e}^k$. Depending on the group $\mathbb{T}$, this set of vectors may or may not form a frame. If it does form a frame, then it is invariant to $\mathbb{T}$, by the considerations in Section III-C. We can now state Theorem 2, showing that there is always at least one dual frame that allows us to go back to $\mathcal{L}$ in a group-invariant way.

Theorem 2. *If $\mathbb{T}$ is a transformation group acting on a lattice $\mathcal{L}$, $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ is a $\mathbb{T}$-invariant frame for $\mathcal{L}$, and the inner product on $\mathcal{M} = \mathcal{L}^{\mathbb{T}}$ is invariant to $\rho(\tau)$ for all $\tau \in \mathbb{T}$ (given by $(\rho(\tau)\,\mathbf{u})_\sigma = \mathbf{u}_{\sigma\tau}$ for all $\mathbf{u} \in \mathcal{M}$), then the frame $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ forms a $\mathbb{T}$-invariant frame-dual pair with its canonical dual.*

Proof. We use the property that the canonical dual minimizes $\|\mathbf{u} - F\hat{F}^*\,\mathbf{u}\|^2$ for all $\mathbf{u} \in \mathcal{M}$. Suppose that $\rho : \mathbb{T} \to \mathrm{Aut}(\mathcal{M})$ is a representation, $\tau \in \mathbb{T}$, $\mathbf{u} \in \mathcal{M}$, F is the analysis operator corresponding to the frame $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$, and $\hat{F}^*$ is the synthesis operator corresponding to its canonical dual. Define $\mathbf{a} = \hat{F}^*\,\mathbf{u}$ and $\mathbf{b} = \hat{F}^*\,\rho(\tau)\,\mathbf{u}$. The theorem now effectively states that $\mathbf{b} = \hat{F}^*\,\rho(\tau)\,\mathbf{u} = \tau\,\hat{F}^*\,\mathbf{u} = \tau\,\mathbf{a}$, for all $\mathbf{u}$ and τ.

To prove the theorem, note that $\mathbf{a}$ is the unique minimizer of $\|\mathbf{u} - F\mathbf{a}\|^2$, while $\mathbf{b}$ is the unique minimizer of $\|\rho(\tau)\,\mathbf{u} - F\mathbf{b}\|^2$ (by the definition of the canonical dual). Since we assume that the inner product on $\mathcal{M}$ is invariant to $\rho(\tau)$ for all $\tau \in \mathbb{T}$, $\mathbf{b}$ is also the (unique) minimizer of $\|\mathbf{u} - \rho(\tau^{-1})\,F\mathbf{b}\|^2$. By the invariance of the original frame, this cost function is equal to $\|\mathbf{u} - F\tau^{-1}\mathbf{b}\|^2$. Finally, since τ is a bijection on $\mathcal{L}$, $\tau^{-1}\,\mathbf{b}$ and $\mathbf{a}$ must be equal and $\mathbf{b}$ equals $\tau\,\mathbf{a}$. This proves that the canonical dual satisfies the second half of Eq. (2), and since the statement is conditional on the original frame satisfying the first half, this concludes the proof. □

In Example 8 we already showed one example of a group-invariant frame-dual pair, we will now show a more colour related example of a group-invariant frame-dual pair.

Example 9. *(Hue-invariant frame-dual pair).* As alluded to before, the problem with “false colours” is likely not that they are different from the original colours, but rather that they do not make any sense. Why should the infimum of two colours that are both (perceptually) close to magenta be black or grey, while the infimum of two colours similarly close to green is green(ish)? Here we show how to construct a frame-dual pair that is invariant to hue rotations, resulting in a qualitatively similar result in both cases, see Fig. 6.

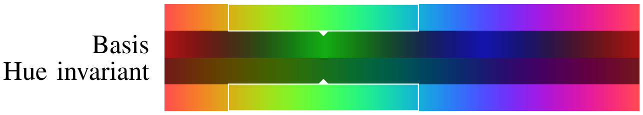


Figure 6. The top and bottom bars cycle through all possible hues. The middle two bars compute the infimum over a range of hues at each position (indicated by the white rectangles). The top-middle bar uses a product order on the RGB basis, giving a brighter and more saturated response at the primary colours than elsewhere. In contrast, the bottom-middle bar uses the hue-invariant frame developed in Example 9, giving a similar result for all hues.

Example 3 showed that the set of all hue rotations can be considered a transformation group $\mathbb{T}$ on the RGB colour space (here considered to be $\mathbb{R}^3$). The group $\mathbb{T}$ is (group) isomorphic to the two-dimensional rotation group $SO(2)$. A rotation of ϕ degrees around the grey axis will be denoted by $r_{g,\phi}$, but instead of writing $\mathbf{u}_{r_{g,\phi}}$ we will use the more readable $\mathbf{u}_\phi$. The index set $\mathcal{I}$ can then be considered to be $[0, 2\pi) \times \{1, 2, 3\}$ and $\mathbf{f}_{\phi,k} = r^*_{g,\phi}\,\mathbf{e}^k$. Taking $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{\mathcal{I}}$, we now define the inner product on $\mathbb{R}^{\mathcal{I}}$ by using the inner product on $\mathbb{R}^3$:

$$\mathbf{u} \cdot \mathbf{v} = \frac{1}{2\pi} \int_0^{2\pi} \mathbf{u}_\phi \cdot \mathbf{v}_\phi \,\mathrm{d}\phi.$$

Clearly, this inner product is invariant to the group $\mathbb{T}$.

The set of vectors $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ as defined above forms a tight frame with frame constants $A = B = 1$:

$$\begin{aligned}\|F\mathbf{a}\|^2 = F\mathbf{a} \cdot F\mathbf{a} &= \frac{1}{2\pi} \int_0^{2\pi} \|r_{g,\phi}\,\mathbf{a}\|^2 \,\mathrm{d}\phi \\ &= \frac{1}{2\pi} \int_0^{2\pi} \|\mathbf{a}\|^2 \,\mathrm{d}\phi \qquad \text{(rotations are orthogonal)} \\ &= \|\mathbf{a}\|^2.\end{aligned}$$

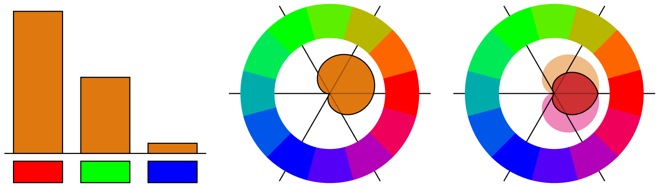
This means that the frame is its own (canonical) dual. Also, since the frame itself is invariant to $\mathbb{T}$ (as illustrated in Fig. 7), it forms a $\mathbb{T}$-invariant frame-dual pair (with itself), by Theorem 2.

The synthesis operator corresponding to the hue-invariant frame found above can be given explicitly. By the definitions above and the linearity of the inner product, we must have:

$$\begin{aligned}F^*\mathbf{u} \cdot \mathbf{a} = \mathbf{u} \cdot F\mathbf{a} &= \frac{1}{2\pi} \int_0^{2\pi} \mathbf{u}_\phi \cdot (r_{g,\phi}\,\mathbf{a}) \,\mathrm{d}\phi \\ &= \frac{1}{2\pi} \int_0^{2\pi} (r^*_{g,\phi}\,\mathbf{u}_\phi) \cdot \mathbf{a} \,\mathrm{d}\phi \\ &= \left(\frac{1}{2\pi} \int_0^{2\pi} r^*_{g,\phi}\,\mathbf{u}_\phi \,\mathrm{d}\phi\right) \cdot \mathbf{a}.\end{aligned}$$

So $F^*\mathbf{u} = \frac{1}{2\pi} \int_0^{2\pi} r^*_{g,\phi}\,\mathbf{u}_\phi \,\mathrm{d}\phi$.

The above example can be easily extended to colour *images*, rather than single colours. A colour image can be considered



(a) $\mathbf{v} : (224, 120, 16)$ (b) $\bar{\mathbf{v}} = F\mathbf{v}$ (c) $\bar{\mathbf{v}} \wedge \bar{\mathbf{w}}$

Figure 7. Illustration of taking the infimum in a hue-invariant representation. (a) Colours are often represented using their red, green and blue components. As established before, taking the infimum is not invariant to changes in hue in that representation. (b) The same colour as in (a), but now in the hue-invariant frame developed in Example 9. The colour is shown using a polar plot of the inner product with the frame vectors. (c) The infimum of two colours (indicated using orange and magenta polar region plots) in a hue-invariant frame, with $\bar{\mathbf{w}} = F r_{g,-60^\circ} \mathbf{v}$.

$$\begin{tikzcd}
\mathcal{L} \arrow[r, "\Lambda = F"] \arrow[d, "\phi"'] & \mathcal{M} \arrow[d, "\psi"] \\
\mathcal{L} & \mathcal{M} \arrow[l, "\hat{F}^*"']
\end{tikzcd}$$

Figure 8. If the lattice $\mathcal{L}$ is (also) a Hilbert space, and $\mathbb{T}$ consists purely of linear transformations, then Λ can be seen to be the analysis operator F corresponding to a frame. This means we can easily define a mapping back to $\mathcal{L}$ using the synthesis operator corresponding to the canonical dual of this frame. If ψ is then invariant to $\mathbb{T}$ on $\mathcal{L}$, then $\phi = \hat{F}^* \circ \psi \circ F$ is $\mathbb{T}$-invariant.

an element of the lattice $\mathrm{Fun}(E, C)$, where C is the colour space and E the image domain. Assuming that C is a Hilbert lattice, $\mathrm{Fun}(E, C)$ is a Hilbert lattice as well, with "scalars" being elements of $\mathrm{Fun}(E, \mathbb{R})$. Now assume $\mathbb{T}$ is a group of transformations on C. This group can then be used to construct an isomorphic group $\mathbb{T}'$ on $\mathrm{Fun}(E, C)$, such that for every $\tau \in \mathbb{T}$ there is a $\tau' \in \mathbb{T}'$, defined by $\tau'(\mathbf{a})(x) = \tau(\mathbf{a}(x))$, for all $x \in E$. It can then be seen that $\mathrm{Fun}(E, C)^{\mathbb{T}'}$ is isomorphic to $\mathrm{Fun}(E, C^{\mathbb{T}})$. So whenever we can find a useful frame for a colour space, we can easily extend this result to colour images.

To recap, in Section III-C we showed how to construct a mapping Λ that can lift a lattice $\mathcal{L}$ to a lattice $\mathcal{M}$ that is invariant to $\mathbb{T}$ on $\mathcal{L}$. We motivated that this allowed the easy definition of a morphological operator ψ on $\mathcal{M}$ that is invariant to $\mathbb{T}$ on $\mathcal{L}$. However, we could not easily get back to $\mathcal{L}$. In this section we showed how this can be done in one common case. If $\mathcal{L}$ is (also) a Hilbert space spanned by the basis $\{\mathbf{e}_k\}_{k \in \mathcal{K}}$, and $\mathbb{T}$ contains only linear transformations, then Λ (under certain conditions on the group) is the analysis operator for the frame $\{\mathbf{f}_i\}_{i \in \mathcal{I}} \subseteq \mathcal{L}$, with $i \in \mathbb{T} \times \mathcal{K}$ and $\mathbf{f}_i = \tau^* \mathbf{e}^k$. We have shown that the constructed frame always leads to a $\mathbb{T}$-invariant frame-dual pair with its canonical dual. This allows us to go back to $\mathcal{L}$ in a such a way that the compound operator ϕ, defined by $\phi = \hat{F}^* \circ \psi \circ F$, is a $\mathbb{T}$-invariant operator on $\mathcal{L}$. This is illustrated in Fig. 8.

It should be noted that, for example, an erosion defined on $\mathcal{M}$ does not (necessarily) result in an erosion after mapping back to $\mathcal{L}$. In fact, it typically will not. However, as we use the canonical dual, mapping back to $\mathcal{L}$ and then to $\mathcal{M}$ again does lose as little as possible (in a least squares sense). As it is often simply not possible to define a certain kind of operator in a group invariant manner on the original space (at least in a non-trivial way), this seems like a reasonable compromise.

Example 10. *(A simple hue-invariant dilation).* Suppose we wish to create a hue-invariant simple dilation by a flat structuring element B. On a greyscale image $p \in \mathrm{Fun}(E, \mathbb{R})$, where E is some Euclidean space, we would have (with $x \in E$)

$$\delta_B(p)(x) = \bigvee_{b \in B} p(x - b).$$

If we have lifted an RGB colour image $\mathbf{a} \in \mathrm{Fun}(E, \mathbb{R}^3)$ to $\mathbf{u} \in \mathrm{Fun}(E, \mathbb{R}^{\mathcal{I}})$ using Example 9, then we can define a hue-invariant dilation on the lifted lattice using the exact same formula as for the greyscale case. So for every $i \in \mathcal{I}$

$$\delta_B(\mathbf{u})(x)_i = \bigvee_{b \in B} \mathbf{u}(x - b)_i.$$

On colour images we could thus define a hue-invariant dilation-like operator using $\tilde{\delta}_B(\mathbf{a}) = \hat{F}^* \, \delta_B(F\mathbf{a})$.

Example 11. *(A dilation using group-valued structuring elements).* Continuing the previous example, rather than writing $\mathbf{u} \in \mathrm{Fun}(E, \mathbb{R}^{\mathcal{I}})$ we could also write $\mathbf{u} \in \mathrm{Fun}(E \times \mathbb{T}, \mathbb{R}^3)$ (recall that in this case $\mathcal{I} = \mathbb{T} \times \{1, 2, 3\}$, with $\mathbb{T}$ the group of hue rotations). Identifying E with the Euclidean translation group we can see that $E \times \mathbb{T}$ is a group: the direct product of the Euclidean translation group and the group of hue rotations. We could thus have a structuring element B that is a subset of the group $E \times \mathbb{T}$ and use ($\xi \in E \times \mathbb{T}$, $i \in \{1, 2, 3\}$)

$$\delta_B(\mathbf{u})(\xi)_i = \bigvee_{\beta \in B} \mathbf{u}(\beta^{-1} \, \xi)_i.$$

This operation is invariant to the representation of hue rotations on the lattice $\mathrm{Fun}(E, \mathbb{R}^{\mathcal{I}})$, defined by $[\rho(\tau) \, \mathbf{u}](x, \sigma) = \mathbf{u}(x, \sigma \, \tau)$, with $x \in E$ and $\sigma, \tau \in \mathbb{T}$:

$$\begin{aligned}
\delta_B(\rho(\tau) \, \mathbf{u})(x, \sigma) &= \bigvee_{\beta \in B} [\rho(\tau) \, \mathbf{u}](\beta^{-1} \, (x, \sigma)) \\
&= \bigvee_{(b, \omega) \in B} \mathbf{u}(b^{-1} \, x, \omega^{-1} \, \sigma \, \tau) = \delta_B(\rho(\tau) \, \mathbf{u})(x, \sigma \, \tau) \\
&= [\rho(\tau) \, \delta_B(\mathbf{u})](x, \sigma)
\end{aligned}$$

This example is closely related to group morphology [2, 26].

At this point it is interesting to examine what would happen if we would simply order colours by their lightness or luminance (essentially their projection on the grey axis). Clearly, such an order would be invariant to hue rotations. However, it would also not give rise to well-defined infima and suprema (as the order says nothing about what to do with the colour information). To use such a preorder for morphological filters we thus need something extra. For example, we could simply keep the original hue and saturation, or take the hue and saturation of the least/greatest element according to the preorder. In both cases we essentially forego filtering the colour information. To avoid that one could order colours using a basis (or frame) of which one of the vectors is the grey axis. In that case hue (and possibly saturation) invariance could become an issue, and we can apply our method [32].

C. Invariance of the inner product

Theorem 2 explicitly requires the inner product on $\mathcal{M}$ to be invariant to the chosen representation of $\mathbb{T}$. Is this a sensible restriction? In our example this posed no great difficulty, but what about other kinds of transformations?

As a first step to getting a grip on the scope of the restriction to group-invariant inner products, consider Lemma 3.

Lemma 3. *A linear transformation $\rho(\tau)$ on $\mathcal{M}$ (with $\tau \in \mathbb{T}$) is orthogonal if and only if the inner product on $\mathcal{M}$ is invariant to $\rho(\tau)$.*

Proof. The transformation $\rho(\tau)$ is orthogonal if and only if $\rho(\tau)^* = \rho(\tau)^{-1}$. Here $\rho(\tau)^*$ is defined to be the linear transformation that satisfies $(\rho(\tau)\,\mathbf{u})\cdot\mathbf{v} = \mathbf{u}\cdot(\rho(\tau)^*\,\mathbf{v})$, with $u, v \in \mathcal{M}$. So if $\rho(\tau)$ is orthogonal, then the inner product on $\mathcal{M}$ is invariant to $\rho(\tau)$:

$$\begin{aligned}(\rho(\tau)\,\mathbf{u})\cdot(\rho(\tau)\,\mathbf{v}) &= (\rho(\tau)^*\,\rho(\tau)\,\mathbf{u})\cdot\mathbf{v}\\ &= (\rho(\tau)^{-1}\,\rho(\tau)\,\mathbf{u})\cdot\mathbf{v} = \mathbf{u}\cdot\mathbf{v}.\end{aligned}$$

Conversely, if the inner product is invariant to $\rho(\tau)$, then for $\mathbf{u}, \mathbf{v} \in \mathcal{M}$ and $\tau \in \mathbb{T}$:

$$(\rho(\tau)\,\mathbf{u})\cdot\mathbf{v} = \mathbf{u}\cdot(\rho(\tau)^*\,\mathbf{v}) = (\rho(\tau)\,\mathbf{u})\cdot(\rho(\tau)\,\rho(\tau)^*\,\mathbf{v}).$$

In other words, $(\rho(\tau)\,\mathbf{u})\cdot(\mathbf{v} - \rho(\tau)\,\rho(\tau)^*\,\mathbf{v}) = 0$ for *all* $\mathbf{u}$ and $\mathbf{v}$ in $\mathcal{M}$. Considering that the inner product is positive definite, and taking $\mathbf{r} = \mathbf{v} - \rho(\tau)\,\rho(\tau)^*\,\mathbf{v}$ and $\mathbf{u} = \rho(\tau)^{-1}\,\mathbf{r}$, we get $\mathbf{r} = 0$. Thus $\mathbf{v} = \rho(\tau)\,\rho(\tau)^*\,\mathbf{v}$ and $\rho(\tau)^* = \rho(\tau)^{-1}$. This leads us to conclude that if the inner product is invariant to $\rho(\tau)$, that $\rho(\tau)$ is orthogonal. Since we also showed the converse, this concludes the proof. □

Based on the above, Proposition 1 rules out a whole class of transformations. In particular, any group that includes some sort of scaling transformation will not work with the construction developed above.

Proposition 1. *If there is an eigenvector $\mathbf{e} \in \mathcal{L}$ of $\tau \in \mathbb{T}$ with associated eigenvalue λ such that $|\lambda| \neq 1$, then $\rho(\tau)$ cannot be orthogonal.*

Proof. Consider the frame condition stated in Eq. (1). There must be finite positive frame constants A and B, such that the squared norm of the frame vector $F\mathbf{a}$ is between $A\,\|\mathbf{a}\|^2$ and $B\,\|\mathbf{a}\|^2$ (for *any* $\mathbf{a} \in \mathcal{L}$). Without loss of generality, assume that $\|\mathbf{e}\|^2 = 1$, so that $A \leq \|F\mathbf{e}\|^2 \leq B$ and $\|\tau^n\,\mathbf{e}\| = |\lambda|^n$. Here τ^n is used to denote the n times repeated application of τ.

We can now use a proof by contradiction. If $\rho(\tau)$ would be orthogonal, then $\|\rho(\tau)^n F\mathbf{e}\|^2 = \|F\mathbf{e}\|^2$ and we would thus have $A \leq \|\rho(\tau)^n F\mathbf{e}\|^2 \leq B$ for all (integer) $n \geq 0$. On the other hand, since $\rho(\tau)^n F\mathbf{e} = F\tau^n\mathbf{e}$, we should also have $A\,|\lambda|^n \leq \|\rho(\tau)^n F\mathbf{e}\|^2 \leq B\,|\lambda|^n$. If $|\lambda| < 1$, we can choose an n such that $B\,|\lambda|^n < A$, leading to the contradiction $A \leq \|\rho(\tau)^n F\mathbf{e}\|^2 \leq B\,|\lambda|^n < A$. Similarly, if $|\lambda| > 1$, then we can choose an n such that $A\,|\lambda|^n > B$, again leading to a contradiction. This leads us to conclude that if $|\lambda| \neq 1$, $\rho(\tau)$ cannot be orthogonal. □

V. Results

Aptoula and Lefèvre [4] evaluated different orders for use with mathematical morphology. We use the same noise reduction and texture classification tasks to demonstrate the objective benefits of group-invariant frames.

We use the RGB basis, the hue-invariant frame developed in Example 9 and a similar one that is invariant to *all* rotations (not just the ones around the grey axis). The latter effectively consists of all unit vectors in $\mathbb{R}^3$ (weighted equally), and is tight, just like the hue-invariant frame [32].

Our current implementation[4] is a very basic one that simply uses a fairly large sampling of frame vectors to approximate the exact result. Because of this the running time of the operators based on frames is simply a (large) constant factor more than that of the operators based on bases. However, we expect that in practice many optimizations could be made to decrease the runtime impact of our method. For example, the current implementation uses a particularly huge number of frame vectors to ensure a very accurate answer, but in our experience a much small number of vectors often already give quite acceptable results. Also, we expect that the choice of frame vectors could be optimized for specific images to further reduce the number of directions to sample (in the degenerate case of greyscale images encoded as RGB images one or two directions would suffice for example). Finally, some frame-based filters can probably be computed directly on the original representation.

A. Noise reduction

In the noise reduction task Gaussian noise was added to an image (Lena), the noisy image was lifted to one of three different frames, filtered, and then projected back to the original colour space. Only the RGB colour space was used, as this makes the most sense with this particular test (after all, the noise is added in this space), and led to best performance in the original experiments. Also, the aim is primarily to demonstrate the potential impact of using group-invariant frames.

The noise was Gaussian, with the covariance matrix (using ρ to denote the correlation coefficient)

$$\Sigma = \sigma^2 \begin{pmatrix}1 & \rho & \rho\\ \rho & 1 & \rho\\ \rho & \rho & 1\end{pmatrix}.$$

To generate Gaussian random numbers with the above covariance matrix: first find a matrix $\Sigma^{\frac{1}{2}}$ such that $\Sigma^{\frac{1}{2}}(\Sigma^{\frac{1}{2}})^T = \Sigma$; the correlated noise is then generated by generating normally distributed samples and multiplying them by $\Sigma^{\frac{1}{2}}$.

The noise reduction filter was the OCCO filter. It is built from a structural opening γ_b and a structural closing ϕ_b, both using the same structuring element b. In this case a 3 × 3 cross was used (the origin and its four nearest neighbours). Both filters operate on $\mathcal{L} = \mathrm{Fun}(E, C)$ using a product order, with C the (RGB) colour space. Following Aptoula and Lefèvre [4], the operator is then defined by ($\mathbf{a} \in \mathcal{L}$)

$$\mathrm{OCCO}_b(\mathbf{a}) = \frac{1}{2}\left(\gamma_b(\phi_b(\mathbf{a})) + \phi_b(\gamma_b(\mathbf{a}))\right). \tag{3}$$

[4] Our code is available as supplemental material.

ρ	RGB	Hue	Rot.	Lex.
0	0.78	0.68	**0.64**	2.22
0.9	0.78	0.76	**0.74**	1.09
		Normalised MSE (×100)		
0	40.3	37.7	**36.5**	68.1
0.9	40.2	39.8	**39.2**	47.8
		RMSE remaining (%)		

Figure 9. The “Lena” image, with image values in the range $[0, 255]$, was corrupted with Gaussian noise ($\sigma = 32$), with different correlation coefficients (ρ). The OCCO operator was applied to the corrupted images in the original RGB basis, the hue-invariant frame and the rotation-invariant frame. For comparison purposes, the same filter was also applied using a lexicographical order (ordering first on red, then on green, then on blue). The values in the top part of the table show the normalised mean squared error (multiplied by 100), while the bottom part uses the RMSE remaining. As can be seen, using a group-invariant frame can give a clear improvement.

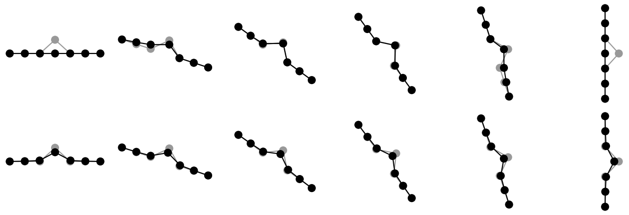


Figure 10. The effect of rotation invariance on the OCCO filter. Each diagram shows part of a one-dimensional signal in which each value is a 2D vector. The signals are plotted as in Fig. 2. The error-free signals are not shown, but assumed to be linear ramps of the form $u[i] = (1/2, 1/2) + i\,(\cos(a), \sin(a))$. Shown in grey are the perturbed signals, where the value at $i = 0$ is offset at a right angle to the original signal. Shown in black are the OCCO filtered signals (the structuring element has a width of 3). The top row shows the results for using the OCCO filter directly on the 2D basis, the bottom row shows the results for the rotation-invariant frame (in 2D). As can be seen, the OCCO filter on the basis performs perfectly for signals that are aligned with one of the basis directions, but quite badly for signals that are even slightly misaligned. In contrast, the OCCO filter on the rotation-invariant frame gives a much more consistent result, effectively the average behaviour of the filter on the basis, averaged over all rotations of the space (this is better than the RGB basis result in most cases).

The OCCO operator was lifted to the group-invariant frames by taking $\mathrm{OCCO}_b(\mathbf{u})_\tau = \mathrm{OCCO}_b(\mathbf{u}_\tau)$. An image was then filtered by lifting it to $\mathcal{M} = \mathrm{Fun}(E, C^\mathbb{T})$, applying the (lifted) OCCO operator, and then projecting back to $\mathcal{L} = \mathrm{Fun}(E, C)$ using the canonical dual frame.

The results of the noise reduction task are summarized in Fig. 9. The error metric used originally is the *normalised mean squared error (MSE)*, which is just the squared L_2-norm of the error in the filtered image divided by the squared L_2-norm of the original image (see [4] for details). Unfortunately this error metric is quite difficult to interpret when comparing results for different images. Therefore, we also use the *root mean squared error (RMSE) remaining*: the RMSE in the filtered image divided by the RMSE in the noisy image. For example, looking at the table in Fig. 9 we see that noise is reduced to approximately 40% (or less) of its original strength.

The group-invariant frames clearly lead to better noise reduction[5]. This makes sense: The OCCO filter essentially removes bumps. When using the RGB basis it only removes bumps in the red, green and blue directions. So “yellow” noise is less likely to be suppressed than red or green noise. Using the hue-invariant frame remedies this, but still not covers all possible directions (for example the direction $(1, -1, 0)$). This could explain why the rotation-invariant frame (whose frame vectors point in all directions) performs best. This phenomenon is illustrated in 2D in Fig. 10.

The effect of correlation is also interesting. As can be seen, increasing correlation decreases the benefit of using group-invariant frames. This likely has to do with the fact that in many images the main variation is along (or close to) the grey axis. As the correlated noise is strongest along the grey axis, noise and data values “line up”. As points on a line can only be ordered in two ways, rotational invariance does not help here. This is corroborated by repeating the experiment using a version of Lenna where the green channel has been inverted. In that case the predominant gradient directions are no longer aligned with the grey axis, and the effect is less pronounced (see Fig. 11). This shows the ability of group-invariant frames to do better than a basis, while gracefully degrading to similar performance when it cannot do better.

B. Texture classification

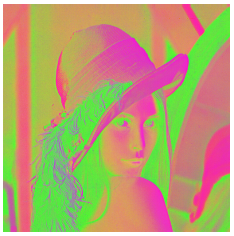
In this task a number of textures are grouped into classes. Part of the textures serve as training data and the rest is used for testing. The textures are taken from Outex13 [33]. Nearest-neighbour classification is used, based on the Euclidean distance between feature vectors. Again we limit ourselves to the RGB colour space (and derived group-invariant frames), as using different colour spaces had hardly any effect on the performance in the original tests using a product order [4].

The feature vectors are based on morphological covariance. To compute the morphological covariance of an image (for a specific shift), a point-wise infimum is computed of the image and a shifted version of itself. The result is then summed over all positions (but independently for each “channel”), and divided by the sum of the pixel values of the original image. This is done for a fixed set of shifts, and when using group-invariant frames, for a finite (fixed) subset of the frame.

Shifts in four different directions (0°, 45°, 90° and 135°), and in each direction by L_∞ distances of 1 to 49 pixels (in step sizes of two), were used in both cases. In total this gives 25 feature values per direction and 100 per channel. In the RGB case 3 channels were used, for the group-invariant frames subsets of 150 frame vectors were used to create 150 channels. (Note that in this task we do not need to project back to the original colour space.) This follows the description given by Aptoula and Lefèvre [4].

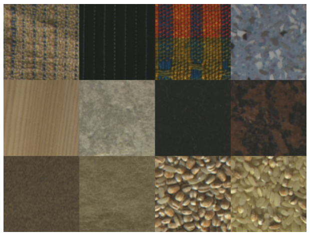
The results summarized in Fig. 12 again show improved performance using the group-invariant frames. However, there are some complications. First of all, the classification results are better than in the original work, even using the RGB basis. This is probably due to slight differences in implementation. Secondly, we need some tricks to make sure that the features can be computed in a sensible manner when using the group-invariant frames. Specifically, for each direction we make sure

[5] The experiment was repeated using a set of 20 images, with similar results.



ρ	RGB	Rot.
0	40.2 (40.3)	**36.5** (36.5)
0.9	40.3 (40.2)	**38.0** (39.2)
		RMSE remaining (%)

Figure 11. For the rotation-invariant frame, inverting the green channel of Lena leads to improved performance when filtering images corrupted with correlated noise. Presumably this effect is a result of having less of the data aligned with the noise. The performance for the RGB basis remains essentially the same (as the filter is self-dual and applied per channel). For comparison purposes, the values from Fig. 9 are shown between parentheses.



RGB	Hue	Rot.
81.8	84.1	**85.4**
Classification rate (%)		

Figure 12. The classification rates for a simple nearest-neighbour scheme based on features computed using different frames. Clearly, the group-invariant frames again lead to better performance than the original RGB basis. On the left a couple of the textures used in the test are shown.

that the projection of the entire RGB cube is non-negative. Otherwise some directions give much less stable results than others (due to division by numbers close to zero).

Still, that the group-invariant frames lead to improved performance is not surprising. As an extreme case, consider a texture that simply oscillates between pure red and blue. When analysing this using just the original RGB basis such a texture is indistinguishable from a texture that oscillates between black and magenta (a combination of red and blue), as we effectively have no information about the relative phase of the red and blue components. On the other hand, using the rotation-invariant frame we also look at the projection onto the vector $(-1, 0, 1)$, giving excellent discrimination.

VI. Conclusions and Future Work

Mathematical morphology can (and often should) be generalized to colour images using a product order. However, it is important to use a colour space that is invariant to certain transformations, as ignoring such invariances leads to unintuitive results. Explicitly building invariant colour representations and applying morphological filters to them gives the best performance to date in several tasks that were previously used to compare different orders.

A fresh look at the history of the notorious “false colour problem” reveals that it stems from a lack of invariance to certain transformations, rather than, as is often assumed, the lack of a total order. This suggests building lattices and operators that are invariant to such transformations. This work shows a systematic way of doing just that, for groups of linear transformations on Hilbert (vector) spaces.

In future work it might make sense to look at groups containing non-linear transformations. In this light, it might also make sense to look more closely at what kind of transformation group is actually meaningful. For example, in this work a very crude approximation to a hue rotation was used (it does not fully preserve perceptual brightness and saturation). So what transformation groups would make sense from a perceptual point of view? And what about transformation groups that are optimal for a given task and data set?

Also, the method developed here was derived from the viewpoint of mathematical morphology, but could in principle apply to other settings as well. For example, the argument in favour of rotation invariance for computing the morphological covariance applies equally well to computing the autocorrelation. It would be interesting to explore such (generalised) approaches in more detail.

As far as performance goes, the current approach effectively boils down to filtering a greyscale image 150 times, which is pretty slow. This is the result of trying to approximate filtering an infinite number of “channels” in the frame representation. In future research it might pay to look closer at how best to select such a subset. Or even to find a completely different way to achieve the same end result.

It could also be interesting to compare the rotation-invariant approach developed here with an approach based on principal component analysis (PCA). A PCA-based approach might be able to mitigate some of the issues discussed above (like not being able to differentiate a red-blue oscillation from a black-magenta oscillation), but necessarily cannot do so for all such cases at the same time (as it still relies on finding a basis). Also, it is not immediately clear how a PCA-based approach could relate to using a frame like the hue-invariant frame.

Finally, this work focuses on colour, but really any type of multi-variate data could be treated in a similar manner, for example (diffusion) tensor data [34]. Burgeth et al. [35] suggested that morphological operators on tensors should be “rotationally invariant”, and taking this as a basis for mathematical morphology on diffusion tensor data seems promising.

References

[1] J. Serra, Ed., *Theoretical Advances*, ser. Image Analysis and Mathematical Morphology. Academic Press, 1988, vol. 2.

[2] H. J. A. M. Heijmans, *Morphological image operators*. Academic Press, 1994.

[3] G. Birkhoff, *Lattice theory*, ser. American Mathematical Society Colloquium Publications. American Mathematical Society, 1961, vol. 25.

[4] E. Aptoula and S. Lefèvre, “A comparative study on multivariate mathematical morphology,” *Pattern Recognit.*, vol. 40, no. 11, pp. 2914–2929, Nov. 2007.

[5] B. Naegel and N. Passat, “Component-Trees and Multivalue Images: A Comparative Study,” in *Mathematical Morphology and Its Application to Signal and Image Processing*, ser. LNCS, M. H. F. Wilkinson and J. B. T. M. Roerdink, Eds. Springer Berlin Heidelberg, 2009, vol. 5720, ch. 24, pp. 261–271.

[6] J. Chanussot and P. Lambert, "Total ordering based on space filling curves for multivalued morphology," in *Mathematical Morphology and its Applications to Image and Signal Processing*, H. J. A. M. Heijmans and J. B. T. M. Roerdink, Eds. Kluwer Academic Publishers, 1998, pp. 51–58.

[7] H. Talbot, C. Evans, and R. Jones, "Complete Ordering and Multivariate Mathematical Morphology," in *Mathematical Morphology and its Applications to Image and Signal Processing*, H. J. A. M. Heijmans and J. B. T. M. Roerdink, Eds. Kluwer Academic Publishers, 1998, pp. 27–34.

[8] F. G. Ortiz, F. Torres, J. Angulo, and S. T. Puente, "Comparative study of vectorial morphological operations in different color spaces," in *Intelligent Robots and Computer Vision XX*, ser. SPIE Proceedings, D. P. Casasent and E. L. Hall, Eds., vol. 4572, no. 1, Oct. 2001, pp. 259+.

[9] A. Hanbury and J. Serra, "Mathematical Morphology in the HLS Colour Space," in *BMVC*, T. F. Cootes and C. J. Taylor, Eds. British Machine Vision Association, 2001.

[10] ——, "Mathematical Morphology in the CIELAB Space," *Image Analysis & Stereology*, vol. 21, no. 3, pp. 201–206, Nov. 2002.

[11] G. Louverdis, M. I. Vardavoulia, I. Andreadis, and P. Tsalides, "A new approach to morphological color image processing," *Pattern Recognit.*, vol. 35, no. 8, pp. 1733–1741, Aug. 2002.

[12] V. De Witte, S. Schulte, M. Nachtegael, D. Van der Weken, and E. Kerre, "Vector Morphological Operators for Colour Images," in *Image Analysis and Recognition*, ser. LNCS, M. Kamel and A. Campilho, Eds. Springer Berlin Heidelberg, 2005, vol. 3656, ch. 82, pp. 667–675.

[13] F. Flórez-Revuelta, "Ordering of the RGB Space with a Growing Self-organizing Network. Application to Color Mathematical Morphology," in *Artificial Neural Networks: Biological Inspirations — ICANN 2005*, ser. LNCS, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, Eds. Springer Berlin Heidelberg, 2005, vol. 3696, ch. 60, pp. 385–390.

[14] J. Angulo, "Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis," *Comput. Vis. Image Underst.*, vol. 107, no. 1-2, pp. 56–73, Jul. 2007.

[15] V. De Witte, S. Schulte, M. Nachtegael, T. Mélange, and E. E. Kerre, "A Lattice-Based Approach to Mathematical Morphology for Greyscale and Colour Images," in *Computational Intelligence Based on Lattice Theory*, ser. Studies in Computational Intelligence, V. Kaburlasos and G. Ritter, Eds. Springer Berlin Heidelberg, 2007, vol. 67, ch. 7, pp. 129–148.

[16] E. Aptoula and S. Lefèvre, "On lexicographical ordering in multivariate mathematical morphology," *Pattern Recognit. Lett.*, vol. 29, no. 2, pp. 109–118, Jan. 2008.

[17] J. Serra, "The "False Colour" Problem," in *Mathematical Morphology and Its Application to Signal and Image Processing*, ser. LNCS, M. H. F. Wilkinson and J. B. T. M. Roerdink, Eds. Springer Berlin Heidelberg, 2009, vol. 5720, ch. 2, pp. 13–23.

[18] J. Goutsias, H. J. A. M. Heijmans, and K. Sivakumar, "Morphological Operators for Image Sequences," *Comput. Vis. Image Underst.*, vol. 62, no. 3, pp. 326–346, Nov. 1995.

[19] E. Aptoula and S. Lefèvre, "α-Trimmed lexicographical extrema for pseudo-morphological image analysis," *J. Vis. Commun. Image Represent.*, vol. 19, no. 3, pp. 165–174, Apr. 2008.

[20] M. D'Zmura, "Colour and the Processing of Chromatic Information," in *Colour Perception: Mind and the physical world*, R. Mausfeld and D. Heyer, Eds. Oxford University Press, Nov. 2003, pp. 142–153.

[21] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, no. 4, pp. 678–689, Apr. 1990.

[22] N. Bouaynaya and D. Schonfeld, "Theoretical Foundations of Spatially-Variant Mathematical Morphology Part II: Gray-Level Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 837–850, May 2008.

[23] J. Serra, "Anamorphoses and function lattices," in *Image Algebra and Morphological Image Processing IV*, ser. SPIE Proceedings, E. R. Dougherty, P. D. Gader, and J. C. Serra, Eds., vol. 2030, no. 1, Jun. 1993, pp. 2–11.

[24] P. Soille, "Beyond self-duality in morphological image analysis," *Image Vis. Comput.*, vol. 23, no. 2, pp. 249–257, Feb. 2005.

[25] R. A. Peters, "A new algorithm for image noise reduction using mathematical morphology," *IEEE Trans. Image Process.*, vol. 4, no. 5, pp. 554–568, May 1995.

[26] J. B. T. M. Roerdink, "Group morphology," *Pattern Recognit.*, vol. 33, no. 6, pp. 877–895, Jun. 2000.

[27] D. J. Robinson, *A course in the theory of groups*, ser. Graduate Texts in Mathematics. Springer-Verlag, 1982, vol. 80.

[28] M. Suzuki, *Group theory, 1*. Springer, 1982.

[29] R. J. Duffin and A. C. Schaeffer, "A Class of Nonharmonic Fourier Series," *Trans. Am. Math. Soc.*, vol. 72, no. 2, pp. 341+, Mar. 1952.

[30] O. Christensen, *Frames and Bases: An Introductory Course*. Birkhäuser ; Springer e-books, 2008.

[31] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applicatons*. New York: Wiley, 1974.

[32] J. J. van de Gronde and J. B. T. M. Roerdink, "Group-invariant frames for colour morphology," in *Mathematical Morphology and Its Applications to Signal and Image Processing*, ser. LNCS, C. L. Luengo Hendriks, G. Borgefors, and R. Strand, Eds., vol. 7883. Springer Berlin Heidelberg, 2013, pp. 267–278.

[33] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex - New framework for empirical evaluation of texture analysis algorithms," in *Int. Conf. Pattern Recognit.*, 2002, pp. 701–706.

[34] J. J. van de Gronde and J. B. T. M. Roerdink, "Frames for tensor field morphology," in *Geometric Science of Information*, ser. LNCS, F. Nielsen and F. Barbaresco, Eds., vol. 8085. Springer Berlin Heidelberg, 2013, pp. 527–534.

[35] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk, “Morphology for matrix data: Ordering versus PDE-based approach,” *Image Vis. Comput.*, vol. 25, no. 4, pp. 496–511, Apr. 2007.

Jasper van de Gronde studied computing science at the University of Groningen, the Netherlands, where he obtained his M.Sc. in 2011. He is currently working towards his Ph.D. as a member of the Scientific Visualization and Computer Graphics group at the Johann Bernoulli Institute for Mathematics and Computer Science of the University of Groningen. His research interests include mathematical morphology, compressed sensing, and signal/image processing in general.

Jos Roerdink studied biology and physics at the University of Nijmegen, the Netherlands, where he obtained his M.Sc. in theoretical physics in 1979. Following his Ph.D. (1983) from the University of Utrecht and a two-year position (1983-1985) as a Postdoctoral Fellow at the University of California, San Diego, both in the area of stochastic processes, he joined the Centre for Mathematics and Computer Science in Amsterdam, where he worked from 1986-1992 on image processing and tomographic reconstruction. He was appointed associate professor (1992) and full professor (2003), respectively, at the Johann Bernoulli Institute for Mathematics and Computer Science of the University of Groningen, where he currently holds a chair in Scientific Visualization and Computer Graphics. His research interests include mathematical morphology, biomedical visualization, neuroimaging and bioinformatics.