

Visual Chirality—Supplemental Material: Commutativity and the Chirality of Imaging Processes

Zhiqiu Lin¹ Jin Sun^{1,2} Abe Davis^{1,2} Noah Snavely^{1,2}
Cornell University¹ Cornell Tech²

Abstract

In this document we explore how commutativity of different image processes can be used to predict the chirality resulting from the interactions between these imaging operators, including Bayer demosaicing, JPEG compression, and random cropping.

1. Introduction

A key goal of our work is to understand how reflection changes what we learn from image data. We can think of this change as the difference between two distributions: one represented by a data set, and the other represented by its reflection. We suspect that when training a network to distinguish between samples from these two different distributions, it can often accomplish this task by looking at low-level artifacts left by various imaging processes. This leads us to ask, when can we attribute visual chirality to the content being imaged, and when might it instead be the result of the imaging process? To answer this question, we develop theory relating whether an imaging process preserves the achirality (i.e., symmetry) of a distribution to the commutativity between the imaging process and reflection. Based on the theory we developed in Section 2, we introduce a simple technique to quickly examine this commutativity on random samples in order to predict whether an imaging process might introduce visual chirality into this distribution. Furthermore, our analysis can be generalized to other image transformation, such as random cropping.

Section 2 reviews the definitions and introduces the basic connection between the commutativity of an imaging process \mathbf{J} with respect to an image transformation \mathbf{T} —as in our main paper, \mathbf{T} is reflection—and whether \mathbf{J} can preserve the symmetry with respect to \mathbf{T} . Section 3 applies our theory to analyzing the chirality introduced by common digital imaging processes—Bayer demosaicing and JPEG compression—for a specific image transformation reviewed by our main paper that is mirror reflection. Section 4 examines how translation invariance, as incorporated through random cropping, can influence the chirality of imaging processes. A surpris-

ing empirical finding is that by taking a collection of images that are initially achiral, and passing them through Bayer demosaicing and JPEG compression, this processed collection can become *chiral*, even when looking at random crops of these images. This holds even though Bayer demosaicing and JPEG compression alone are insufficient to introduce chirality to an achiral distribution when random cropping is applied. Our theoretical and empirical results altogether suggest that imperceptible chiral traces may be left in photos by non-commutative imaging pipelines, which has implications on self-supervised learning, image forensics, data augmentation, etc.

2. Commutativity and chirality

We begin by reviewing the derivation from our paper. Consider a distribution \mathbf{D} over images. For our purposes, \mathbf{D} can be thought of as a discrete probability distribution over images from which one has a number of samples (e.g., to form a dataset), in which case $\mathbf{D}(\mathbf{x})$ denotes the probability of image \mathbf{x} .

Our main derivation shows that an imaging process \mathbf{J} (e.g., JPEG compression) preserves the achirality of a distribution \mathbf{D} with respect to some transformation \mathbf{T} when \mathbf{J} and \mathbf{T} commute under \mathbf{D} . We first derive this generally for any bijective transformation \mathbf{T} (thus not just mirror reflection).

To clarify our problem setting, let \mathbf{x} denote an image from an arbitrary distribution \mathbf{D} . Let \mathbf{T} be any bijective transformation from one image to another—a mirror flip being a specific example of such a transformation \mathbf{T} . Let \mathbf{J} be an image processing operator, such as JPEG compression. \mathbf{J} also maps images to other images, but is not necessarily a bijection. For instance, in the case of a lossy operation like JPEG compression, multiple input images may compress to the same output compressed image. That is to say, given some input image \mathbf{x} and its processed version $\mathbf{y} = \mathbf{J}(\mathbf{x})$, the preimage of \mathbf{y} , $\mathbf{J}^{-1}(\mathbf{y})$, is a set of images of cardinality $|\mathbf{J}^{-1}(\mathbf{y})| \geq 1$.

We are interested in image distributions that are (or are not) symmetric with respect to the bijective transformation \mathbf{T} . We call a distribution \mathbf{D} symmetric with respect to \mathbf{T} if,

for all \mathbf{x} ,

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})). \quad (1)$$

We also refer to such a symmetric distribution as *achiral*, and conversely, we refer to an asymmetric distribution as *chiral*.

We consider the effect of \mathbf{J} on the distribution \mathbf{D} and its symmetries. For simplicity, We will use $\mathbf{D}_{\mathbf{J}}$ to refer to the distribution that results from applying \mathbf{J} to the domain of \mathbf{D} . In other words, if \mathbf{y} is an output processed image, then

$$\mathbf{D}_{\mathbf{J}}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{J}^{-1}(\mathbf{y})} \mathbf{D}(\mathbf{x}). \quad (2)$$

Finally, we are interested in proving symmetry properties of distributions of processed images when the transformation of symmetry \mathbf{T} commutes with the image processing operator \mathbf{J} , in other words, when, for all \mathbf{x}

$$\mathbf{T}(\mathbf{J}(\mathbf{x})) = \mathbf{J}(\mathbf{T}(\mathbf{x})) \quad (3)$$

Now, we can state our proposition as follows:

Proposition 1. *Given a distribution \mathbf{D} that is symmetric with respect to a bijective transformation \mathbf{T} , if operation \mathbf{J} and \mathbf{T} are commutative, then $\mathbf{D}_{\mathbf{J}}$ is also symmetric with respect to \mathbf{T} (aka \mathbf{J} preserves the symmetry of \mathbf{D}).*

Proof. For all images \mathbf{x} in distribution \mathbf{D} , given \mathbf{D} is symmetric w.r.t \mathbf{T} we have:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})) \quad (4)$$

And the commutativity of \mathbf{T} and \mathbf{J} implies:

$$\mathbf{J}(\mathbf{T}(\mathbf{x})) = \mathbf{T}(\mathbf{J}(\mathbf{x})) \quad (5)$$

To prove that $\mathbf{D}_{\mathbf{J}}$ is also symmetric w.r.t \mathbf{T} , we need to show:

$$\mathbf{D}_{\mathbf{J}}(\mathbf{J}(\mathbf{x})) = \mathbf{D}_{\mathbf{J}}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) \quad (6)$$

The above Equation 6, using the definition of $\mathbf{D}_{\mathbf{J}}$ in Equation 2, can be expanded to:

$$\sum_{\mathbf{x}' \in \mathbf{J}^{-1}(\mathbf{J}(\mathbf{x}))} \mathbf{D}(\mathbf{x}') = \sum_{\mathbf{x}'' \in \mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x})))} \mathbf{D}(\mathbf{x}'') \quad (7)$$

By symmetry of \mathbf{D} w.r.t \mathbf{T} , Equation 7 is equivalent to:

$$\sum_{\mathbf{x}' \in \mathbf{J}^{-1}(\mathbf{J}(\mathbf{x}))} \mathbf{D}(\mathbf{T}(\mathbf{x}')) = \sum_{\mathbf{x}'' \in \mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x})))} \mathbf{D}(\mathbf{x}'') \quad (8)$$

The key to showing that this equality holds is to establish that we are summing over the same set of images on both sides. To do this, we first define $\mathbf{T}_{\mathbf{S}}$ to be the image under \mathbf{T} of a set of images $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, i.e., the set resulting from applying \mathbf{T} to each image in set \mathbf{S} :

$\mathbf{T}_{\mathbf{S}}(\mathbf{S}) = \{\mathbf{T}(\mathbf{x}_1), \mathbf{T}(\mathbf{x}_2), \dots\}$. With this definition, Equation 8 becomes:

$$\sum_{\mathbf{x}' \in \mathbf{T}_{\mathbf{S}}(\mathbf{J}^{-1}(\mathbf{J}(\mathbf{x})))} \mathbf{D}(\mathbf{x}') = \sum_{\mathbf{x}'' \in \mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x})))} \mathbf{D}(\mathbf{x}'') \quad (9)$$

For the above equality to hold, it suffices to show that $\mathbf{T}_{\mathbf{S}}(\mathbf{J}^{-1}(\mathbf{J}(\mathbf{x})))$ and $\mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x})))$ are the same set. We use the fact that \mathbf{T} is a bijection, and hence has an inverse \mathbf{T}^{-1} . Note that since we assume \mathbf{T} and \mathbf{J} are commutative, it follows that \mathbf{T}^{-1} and \mathbf{J} are also commutative. Therefore, for any $\mathbf{x}' \in \mathbf{T}_{\mathbf{S}}(\mathbf{J}^{-1}(\mathbf{J}(\mathbf{x})))$, we have that:

$$\mathbf{J}(\mathbf{T}^{-1}(\mathbf{x}')) = \mathbf{J}(\mathbf{x}) \quad (10)$$

The above Equation 10 is equivalent to the following equations:

$$\mathbf{T}^{-1}(\mathbf{J}(\mathbf{x}')) = \mathbf{J}(\mathbf{x}) \quad (11)$$

$$\mathbf{J}(\mathbf{x}') = \mathbf{T}(\mathbf{J}(\mathbf{x})) \quad (12)$$

Note that Equation 12 is equivalent to the statement that $\mathbf{x}' \in \mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x})))$. Hence, because of the equivalence of Equations 10 and 12, we show that

$$\mathbf{J}^{-1}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) = \mathbf{T}_{\mathbf{S}}(\mathbf{J}^{-1}(\mathbf{J}(\mathbf{x}))) \quad (13)$$

thus completing the proof. \square

Preserving achirality vs. chirality. Note that we have not accounted for the scenario where \mathbf{J} removes asymmetries. This means that while achirality is preserved, chirality may not be. In fact, loss of chirality is almost certain to happen, as imaging is necessarily lossy and therefore trivially creates symmetry. However, our primary concern is determining whether the asymmetries we learn from data are the result of content or an artifact of processing. This question does not apply to asymmetries we never observe. The design of a chirality-preserving imaging system could be an interesting problem related to computational photography, but we leave this to future work.

2.1. Proof Based on Group Theory

Here we offer a more general proof based on group theory. Our approach is to show that, when \mathbf{J} commutes with \mathbf{T} , \mathbf{J} maps between cyclic groups generated by \mathbf{T} that partition our domain. The advantage of this proof over our previous proof is that it shows \mathbf{T} only needs to be associative and have an inverse (the conditions on a group operation), and it better explains what happens when \mathbf{J} is non-injective.

Consider a transformation $\mathbf{T} : \mathbb{R}^n \mapsto \mathbb{R}^n$ that is associative and invertible, and a second transformation $\mathbf{J} : \mathbb{R}^n \mapsto \mathbb{R}^n$ that commutes with \mathbf{T} , so:

$$\mathbf{T}\mathbf{J}\mathbf{x} = \mathbf{J}\mathbf{T}\mathbf{x} \quad (14)$$

We call a distribution $\mathbf{D} : \mathbb{R}^n \mapsto \mathbb{R}$ symmetric with respect to \mathbf{T} if, for all \mathbf{x} , we have

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})). \quad (15)$$

and we define $\mathbf{D}_{\mathbf{J}}$, the transformation of \mathbf{D} by \mathbf{J} , as

$$\mathbf{D}_{\mathbf{J}}(\mathbf{x}) = \sum_{\mathbf{x}_i: \mathbf{J}\mathbf{x}_i = \mathbf{x}} \mathbf{D}(\mathbf{x}_i) \quad (16)$$

Proposition 2. *If \mathbf{J} commutes with \mathbf{T} and a distribution \mathbf{D} is symmetric with respect to \mathbf{T} , then the transformed distribution $\mathbf{D}_{\mathbf{J}}$ will also be symmetric with respect to \mathbf{T} .*

Proof. We first show that \mathbf{J} defines a mapping between cyclic subgroups of our domain. We then show that this map is a homomorphism, which we use to relate $\mathbf{D}_{\mathbf{J}}(\mathbf{x})$ to $\mathbf{D}_{\mathbf{J}}(\mathbf{T}(\mathbf{x}))$.

Since \mathbf{T} is associative and invertible, we can use it to partition our domain into non-overlapping cyclic subgroups $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ generated by \mathbf{T} :

$$\langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{ \dots, \mathbf{T}^{-1}\mathbf{x}_i, \mathbf{x}_i, \mathbf{T}\mathbf{x}_i, \mathbf{T}^2\mathbf{x}_i, \mathbf{T}^3\mathbf{x}_i, \dots \} \quad (17)$$

where the identity element of each group can be chosen as any arbitrary element within the group. The group operation \cdot can be thought of as a permutation of our domain relative to the identity:

$$\mathbf{T}^a \mathbf{x}_i \cdot \mathbf{T}^b \mathbf{x}_i = \mathbf{T}^{a+b} \mathbf{x}_i \quad (18)$$

As each such subgroup shares the same group operation and is closed under that operation, any two $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ must either be equivalent or disjoint. The order $|\langle \mathbf{x}_i \rangle^{\mathbf{T}}|$ of each subgroup depends on the symmetries of \mathbf{x}_i with respect to \mathbf{T} . For example, if \mathbf{T} is simple reflection about a particular axis then $|\langle \mathbf{x}_i \rangle^{\mathbf{T}}| = 1$ for images \mathbf{x}_i that are symmetric about that axis, and $|\langle \mathbf{x}_i \rangle^{\mathbf{T}}| = 2$ for images that are asymmetric about that axis.

Now consider how \mathbf{J} transforms each of the subgroups $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$:

$$\mathbf{J}\langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{ \dots, \mathbf{J}\mathbf{T}^{-1}\mathbf{x}_i, \mathbf{J}\mathbf{x}_i, \mathbf{J}\mathbf{T}\mathbf{x}_i, \mathbf{J}\mathbf{T}^2\mathbf{x}_i, \dots \} \quad (19)$$

If \mathbf{J} commutes with \mathbf{T} , we can rewrite the above as

$$\mathbf{J}\langle \mathbf{x}_i \rangle^{\mathbf{T}} = \{ \dots, \mathbf{T}^{-1}\mathbf{J}\mathbf{x}_i, \mathbf{J}\mathbf{x}_i, \mathbf{T}\mathbf{J}\mathbf{x}_i, \mathbf{T}^2\mathbf{J}\mathbf{x}_i, \dots \} \quad (20)$$

giving us

$$\mathbf{J}\langle \mathbf{x}_i \rangle^{\mathbf{T}} = \langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}} \quad (21)$$

This shows that \mathbf{J} maps cyclic subgroups generated by \mathbf{T} to cyclic subgroups that can be generated by \mathbf{T} .

Symmetry with respect to \mathbf{T} can be restated as the condition that all elements within common cyclic subgroups generated by \mathbf{T} should share the same probability. It is therefore sufficient for us to show that the map $\mathbf{J} : \langle \mathbf{x}_i \rangle^{\mathbf{T}} \mapsto \langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$ is

a homomorphism, as the first isomorphism theorem ensures the same number of equal-probability elements from $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ will map to each element of $\langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$.

Recall that a homomorphism $h : G \mapsto H$ is defined by the relation $h(u \cdot v) = h(u) \cdot h(v)$. It is simple to show that this holds for \mathbf{J} and our cyclic subgroups when \mathbf{J} commutes with \mathbf{T} :

$$\begin{aligned} \mathbf{J}(\mathbf{T}^a \mathbf{x}_i \cdot \mathbf{T}^b \mathbf{x}_i) &= \mathbf{J}(\mathbf{T}^{a+b} \mathbf{x}_i) \\ &= \mathbf{T}^{a+b} \mathbf{J}(\mathbf{x}_i) \\ &= \mathbf{T}^a \mathbf{J}(\mathbf{x}_i) \cdot \mathbf{T}^b \mathbf{J}(\mathbf{x}_i) \end{aligned} \quad (22)$$

This is sufficient to prove our proposition. For completeness, we also reformulate $\mathbf{D}_{\mathbf{J}}$ in terms of the cyclic subgroups $\langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$. We will use the notation $\langle \mathbf{x}_i \rangle_D^{\mathbf{T}}$ to denote an indicator distribution that maps every element of $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ to 1, and every other element to 0. Note that any distribution we can represent as the weighted sum of $\langle \mathbf{x}_i \rangle_D^{\mathbf{T}}$ must preserve symmetry with respect to \mathbf{T} . We can express \mathbf{D} as:

$$\mathbf{D} = \sum_i \mathbf{D}(\mathbf{x}_i) \langle \mathbf{x}_i \rangle_D^{\mathbf{T}} \quad (23)$$

Now, using the first isomorphism theorem to account for the case where \mathbf{J} is non-injective, we can combine Equation 16 and 23 to write $\mathbf{D}_{\mathbf{J}}$ as

$$\mathbf{D}_{\mathbf{J}} = \sum_i (\mathbf{D}(\mathbf{x}_i) |k_{\mathbf{J}\mathbf{x}_i}|) \langle \mathbf{J}\mathbf{x}_i \rangle_D^{\mathbf{T}} \quad (24)$$

where $k_{\mathbf{J}\mathbf{x}_i}$ is the kernel of $\mathbf{J} : \langle \mathbf{x}_i \rangle^{\mathbf{T}} \mapsto \langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$. This concludes our proof. \square

2.2. Symmetries and Groups of Transformations

In Section 2.1 we proved Proposition 2 by showing that \mathbf{J} formed a homomorphism between cyclic subgroups $\langle \mathbf{x}_i \rangle^{\mathbf{T}}$ and $\langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$. We now consider the case where $\mathbf{D}_{\mathbf{J}}$ is the sum of multiple such homomorphisms \mathbf{J}_j

$$\mathbf{D}_{\mathbf{J}}(\mathbf{x}) = \sum_j \sum_{\mathbf{x}_i: \mathbf{J}_j \mathbf{x}_i = \mathbf{x}} \mathbf{D}(\mathbf{x}_i) \quad (25)$$

In this case, the sum of symmetric distributions is a symmetric distribution, which tells us that $\mathbf{D}_{\mathbf{J}}$ will still be symmetric. Now note that by permuting the elements on the right side of Equation 25, we can define a new set of transformations that sum to the same $\mathbf{D}_{\mathbf{J}}$, but have a different kind of commutativity, which we call *glide commutativity*, demonstrated in Figure 1.

$$\mathbf{J} : \langle \mathbf{x}_i \rangle^{\mathbf{T}} \mapsto \langle \mathbf{J}\mathbf{x}_i \rangle^{\mathbf{T}}$$

2.3. Non-Commutativity and Chirality

What happens when \mathbf{J} does not commute with \mathbf{T} ? Even when this is the case, it is simple to find transformed distributions $\mathbf{D}_{\mathbf{J}}$ that are symmetric with respect to \mathbf{T} , indicating

Commutativity Residual for Horizontal Reflection Over Different Image Dimensions

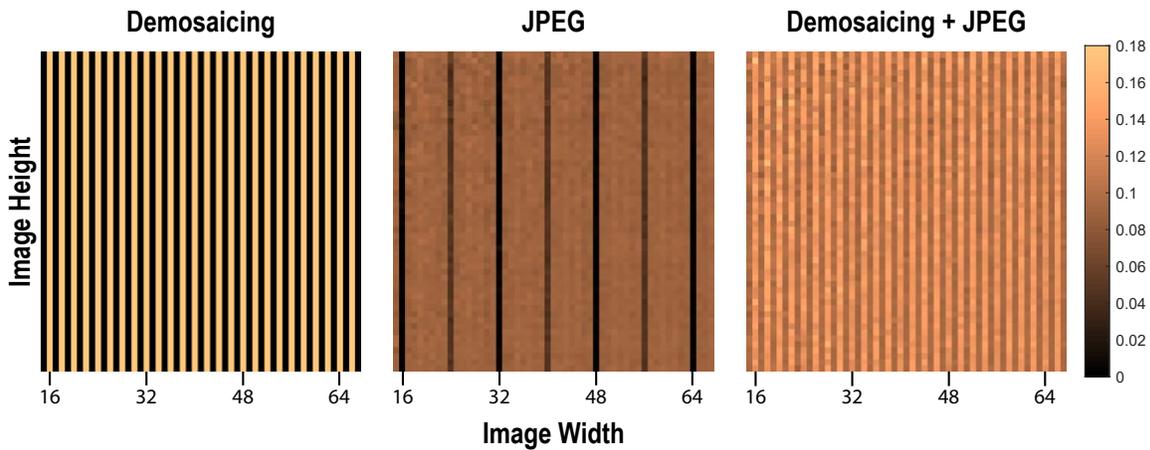


Figure 3. **Commutativity residuals for demosaicing (left), JPEG compression (middle) and their composition (right):** Each image shows how commutativity residual, measured in absolute average percent error per pixel, varies with different image sizes. For integers n we see commutativity in demosaicing at image widths of $2n - 1$ (i.e., odd widths), and in JPEG compression at widths of $16n$. We do not see commutativity when both are applied.

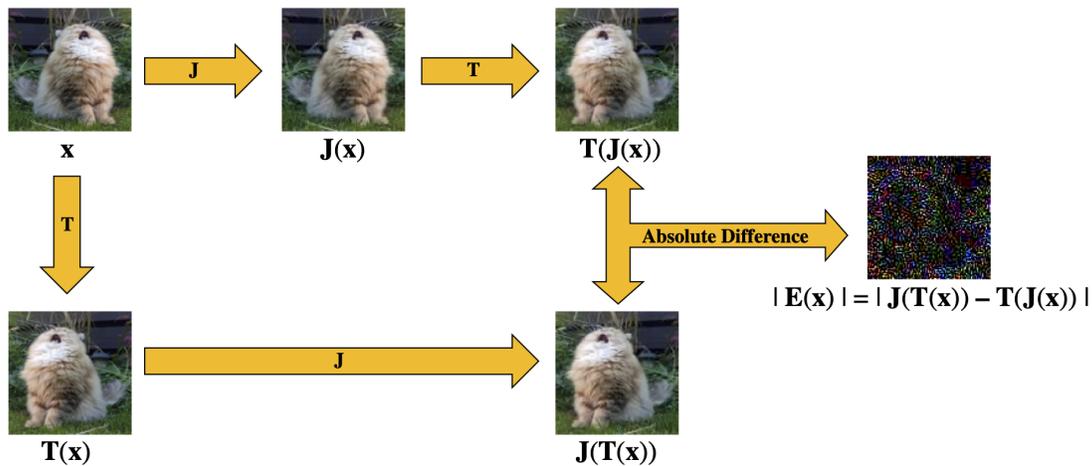


Figure 4. **A visual example of the commutative residual method.** This figure illustrates how we apply the commutative residual method to a natural image. Here T is the horizontal reflection operation, and J is the composition of Bayer demosaicing and JPEG compression. The image used here has a width of 100px. For better visualization of the imperceptible differences shown in the residual image, we scale the resulting residual by a factor of 10. Consistent with the results in Figure 3, the residual image is not zero (which would be all black), i.e., the commutative residual is non-zero.

JPEG compression. We start with a brief summary of these two operations.

Bayer filters and demosaicing. Many modern digital cameras (including cellphone cameras) capture color by means of a square grid of colored filters that lies atop of the grid

of photosensors in the camera. An 8×8 example of such a color filter grid, known as a Bayer filter mosaic, is shown in Figure 2. In such cameras, each pixel’s sensor measures intensity for a single color channel (red, green, or blue), and so to produce a full color image at full resolution, we must interpolate each color channel such that each pixel uni-

mately has an R, G, and B value. This interpolation process is known as *demosaicing*. For our analysis we assume, as is typical, that a Bayer filter mosaic pattern consists of a tiled 2×2 element (GRBG in the case of Figure 2).

Note that the 8×8 Bayer filter mosaic in Figure 2 has interesting symmetry properties. The 8×8 pattern as a whole is asymmetric—flipping it horizontally will result in a red pixel in the upper-left corner, rather than a green pixel. The same is true for any even-sized Bayer filter mosaic. However, from the perspective of the center of any pixel, the pattern is locally symmetric. Moreover, if we imagine a 9×9 version of this mosaic (or indeed any odd-sized pattern), that mosaic would be symmetric.

JPEG compression. JPEG is one of the most common (lossy) compression schemes for images. There are two main ways that JPEG compresses image data. First, it converts images into the $Y' C_b C_r$ colorspace and downsamples the chroma channels (C_b and C_r), typically by a factor of two. Then it splits each channel into a grid of 8×8 pixel blocks and computes the discrete cosine transform (DCT) of each block. In the luminance (Y') channel, each block covers an 8×8 pixel region of the original image, while for the chroma channels, each block corresponds to a 16×16 pixel region in the original image, due to the $2 \times$ downsampling. Finally, the DCT of each block is strategically quantized to further compress the data at low perceptual cost.

For the purposes of our analysis, one noteworthy aspect of JPEG compression is that for images with dimensions that are not a multiple of 16, there will be boundary blocks that do not have a full 8×8 complement of pixels. These are handled specially by the JPEG algorithm.

3.1. Commutative Residuals and Image Size

If we evaluate commutative residuals on arbitrarily random images for demosaicing, they will be nonzero about half of the time. For JPEG, they will be nonzero over 90% of the time. But if we sample over different image sizes more systematically, a pattern begins to emerge. Figure 3 visualizes the commutative residuals for random noise images as a function of image width and height. To illustrate the commutative residual visually, Figure 4 shows an example of the commutative residual image for a natural image with a width of 100px. We can see that demosaicing appears to preserve the symmetry for images with odd widths, while JPEG compression seems to preserve the symmetry for image with widths that are divisible by 16. We can explain this result by considering the geometry of Bayer patterns and JPEG block grids. Bayer patterns (Figure 2) have horizontal symmetry when reflected about any line centered on a pixel column, while the JPEG block grid, which consists of 8×8 blocks that correspond to 8×8 or 16×16 blocks of the original image, is horizontally symmetric only around grid lines, which rest between columns at 16-pixel intervals.

Figure 3 shows that our black-box analysis of commutative residuals is able to reveal the grid structures underlying both algorithms and show how each grid structure impacts preservation of achirality. Notably, when the commutative residual for any transformation for a given image width is zero, we know that this transformation preserves the achirality of the original distribution with such width. We also notice that the combination of demosaicing followed by JPEG compression can never be commutative with respect to flipping because these two imaging processes do not have zero commutative residual for the same image width. Note further that the combination of these two operations is very standard in imaging pipelines, and so we can expect results on synthetic data to apply to real images as well. One of the great plus of our commutative residual method is that all of our analysis involved in generating Figure 3 can be performed in just a few minutes on a laptop using unoptimized MATLAB code.

When the commutative residual is non-zero, we further hypothesize that such scenarios will often result in loss of achirality, i.e., a non-commutative imaging process will make an originally achiral distribution chiral. To verify this, we trained deep network models on three achiral distributions of Gaussian noise images, corresponding to three different square images sizes: one with odd width (99×99), one with even width that is not an integer multiple of 8 (100×100), and one that is a multiple of 16 (112×112). The results predicted by our hypothesis for demosaicing, JPEG, and their composition are summarized in Table 1. Also, at each pixel of each sample image, the value of each channel was sampled from a different Gaussian distribution. For each channel we used a different mean (R:0.6, G:0.5, B:0.9) and standard deviation (R:0.3, G:0.25, B:0.4) to reduce the number of symmetries present other than **T**. A sample image, before and after processing, is shown in Figure 5.

We used the same ResNet model as in main paper (with randomly initialized weights) on the chirality (flip/no-flip) task for each of these nine datasets, performing a grid search for an optimal learning rate using a log scale. As expected, our network model can never achieve more than 50% test accuracy on distributions that were predicted by our analysis to be achiral (i.e., commutative residual is zero); surprisingly, our network achieved near perfect accuracy on the output processed distributions resulted from non-commutative imaging processes. This gave a strong empirical evidence that non-commutativity results in loss of achirality.

Note that this analysis assumes that we use the whole images after Bayer demosaicing and/or JPEG compression, i.e., no cropping has taken place on the images. Interestingly, these results mirror the situation of training networks on *real* images with no random cropping, as described in the main paper. Figure 8 shows that networks trained to classify chirality on resized (but not cropped) Instagram images often

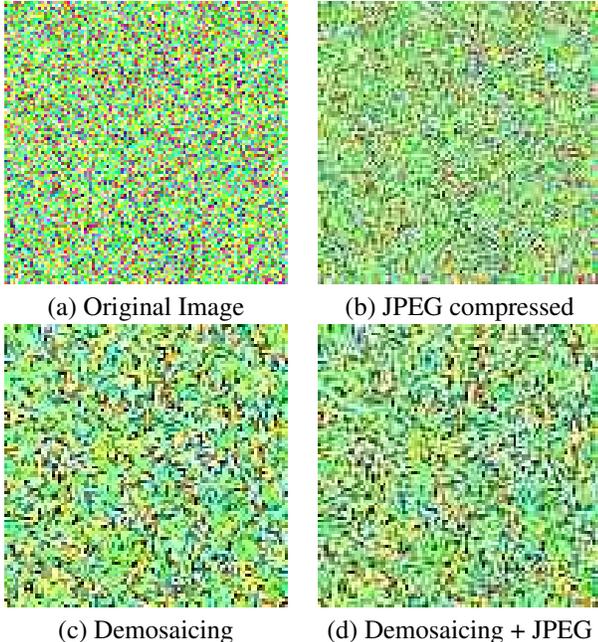


Figure 5. **A sample image from our Gaussian noise image distribution after different imaging operations.** This image is of size (100,100) and is generated using the Gaussian noise method described in Section 3.1

Imaging Operation	Image size		
	99	100	112
Demosaicing	A	C	C
JPEG	C	C	A
Demosaicing+JPEG	C	C	C

Table 1. **Predicted chirality of three (initially achiral) Gaussian noise image distributions (corresponding to three different square image sizes) under each of three processing schemes.** ‘C’ means chiral, and ‘A’ means achiral. Explanation: 99px images should remain achiral under demosaicing, since it is odd size. 112 should remain achiral under JPEG compression since it is divisible by 16. Everything else becomes chiral as discussed earlier. We verify this table empirically by training network models on the nine distributions resulting from these transformations.

seem to focus on image evidence near boundaries (first row), which we hypothesis is due exactly to the kinds of chiral artifacts discussed in this section. However, training with random cropping data augmentation yields networks that appear to focus on much more high-level features (second row). In the next section, we discuss the interaction of processing with random cropping (or image translation) and how the addition of random cropping can make a chiral imaging process achiral.

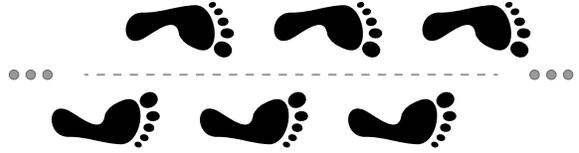


Figure 6. **Glide Symmetry:** Human footprints often exhibit glide symmetry. The infinitely repeating footprint pattern shown here is equivalent to the reflection of a shifted version of itself.

4. Random Cropping and Glide Symmetry

As our analysis makes few assumptions about \mathbf{T} , \mathbf{J} and \mathbf{D} , we can apply it to other symmetries and data augmentation strategies used in learning. For example, translational invariance is a common and useful prior that is often applied to data through the use of random crops as a type of data augmentation. Doersch *et al.* [1] found that when they trained a network to predict the relative position of different regions in an image, it would “cheat” by utilizing chromatic aberration for prediction. We can use our observation about commutativity to explain this behavior by considering a family of transformations in the plane. The self-supervision task used in Doersch *et al.* requires the network to distinguish between different translations, which is only possible when the following symmetry does not hold:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}_v(\mathbf{x})) \quad (28)$$

where \mathbf{T}_v is translation by some vector $v \in \mathbb{R}^2$. Our commutativity analysis tells us that this symmetry can be broken by any \mathbf{J} that does not commute (or glide commute) with translation. This agrees with the findings of Doersch *et al.* that the network was able to “cheat” using artifacts caused by chromatic aberration, which is not translation-invariant, as its effect is spatially varying.

4.1. Detecting Glide Symmetries

If we revisit our analysis of commutative residuals under an assumption of translation invariance, we can draw new conclusions about the chirality of demosaicing and JPEG compression. In particular, we find that by incorporating translation invariance in the form of random cropping, we can change the chirality of these operations.

To test for glide-commutativity, we must look for the permutation pattern described in Section 2.2. To do this, we first define a way of phase-shifting $\mathbf{T}(\mathbf{J}(\mathbf{x}))$ and $\mathbf{J}(\mathbf{T}(\mathbf{x}))$. For this, we define $\mathbf{JT}_\phi(\mathbf{x})$ and $\mathbf{TJ}_\phi(\mathbf{x})$ as the process of:

1. Padding \mathbf{x} with a large, constant number of pixels on all sides.
2. Translating the padded image by ϕ .
3. Applying \mathbf{T} then \mathbf{J} for $\mathbf{JT}_\phi(\mathbf{x})$, or \mathbf{J} then \mathbf{T} for $\mathbf{TJ}_\phi(\mathbf{x})$.
4. Translating by $\mathbf{T}(-\phi)$.

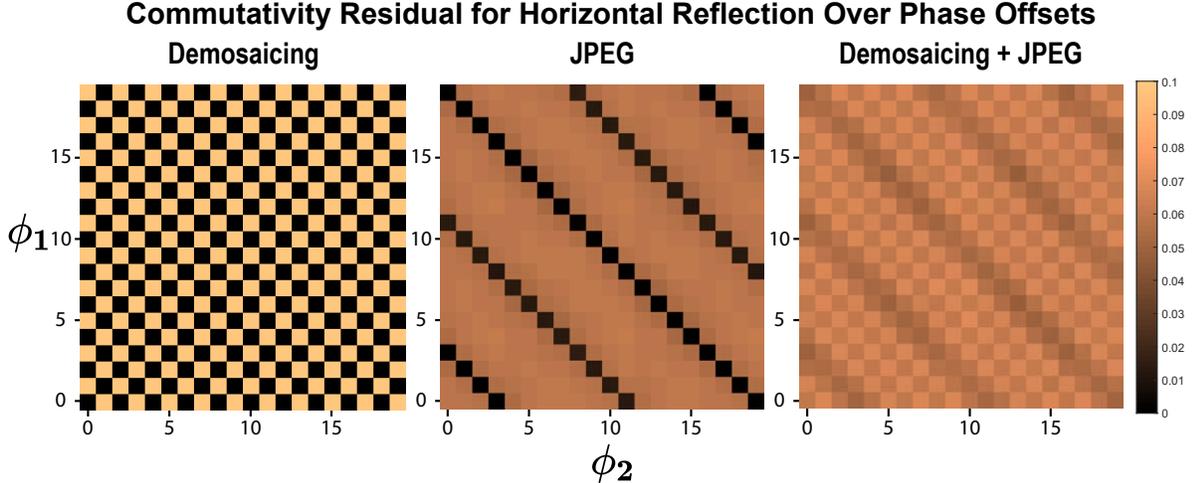


Figure 7. **Glide Commutativity Residuals for demosaicing (left), JPEG compression (middle) and their composition (right)**: Each image shows the glide commutativity residual, measured in absolute average percent error per pixel, measured over different phase shifts. For certain ϕ_1 and ϕ_2 we see commutativity in demosaicing and in jpeg compression alone. We do not see commutativity when both are applied.

5. Cropping out the previously padded pixels.

This has the effect of performing \mathbf{J} and \mathbf{T} as if the image had occurred at a translation of ϕ from its original position. For grid-based algorithms like demosaicing and JPEG compression, this effectively phase-shifts the grid structure used in the algorithm.

To test for glide-commutativity we simply look for some repeating pattern of zeros in residuals of the form:

$$\mathbf{e}_{\mathbf{J}}(\mathbf{x}, \phi_1, \phi_2) = \frac{1}{k} \sum_{\text{pixels}} |\mathbf{J}\mathbf{T}_{\phi_1}(\mathbf{x}) - \mathbf{T}\mathbf{J}_{\phi_2}(\mathbf{x})| \quad (29)$$

This pattern of zeros describes the permutation pattern described in Section 2.2. As the results in Figure 3 show, we verified that the vertical components of ϕ_1 and ϕ_2 do not matter. We therefore set them only to vary in the x dimension of the image. Figure 7 shows the residuals calculated for a range of phase shifts. We see that both demosaicing and JPEG compression appear to be glide-commutative due to the regular repeating pattern of zeros. However, the combination of demosaicing and JPEG compression does not appear to be glide-commutative, and we can see this is because zeros always occur at different phase shifts for each of the two operations.

4.2. Empirical chirality in the presence of random crops

The analysis from the previous section has simple implications (in terms of random cropping on images): (1) The distribution of random crops (while avoiding cropping from the boundary of 16 pixels) from an originally achiral distribution of images that has undergone either demosaicing or

JPEG compression (but not both) should remain achiral. (2) On the other hand, surprisingly, random crops (avoiding a 16-pixel margin around the boundary in the cropped image) on that achiral distribution of images after both demosaicing and JPEG compression may likely become chiral.

To verify this analysis empirically, we again train ResNet models on the same achiral Gaussian distributions as introduced in Section 3.1. Specifically, we take random crops of size (512, 512) from the center (560, 560) of the (576, 576) Gaussian noise images to avoid possible boundary effects from a 16-pixel margin. We train separate networks on each of the three output image distributions obtained from applying each of the three imaging operations (demosaicing, JPEG compression, and composition of demosaicing followed by JPEG compression) on the initial Gaussian noise image distribution. Note that, as before, we perform a log-scale grid search over learning rates.

The network training results show that neither demosaicing nor JPEG compression alone is sufficient to produce a chiral distribution under random cropping: models trained with such images fail to achieve more than 50% accuracy. This suggests that chirality is preserved when those operations are applied in isolation. But when both operations are applied, surprisingly, the image distribution becomes chiral: the trained network achieves 100% training and test accuracy. This accords with our theoretical analysis of the glide-commutativity under these operations. Together, our analysis and empirical study suggest that chiral traces are left in photographs via the Bayer demosaicing and JPEG compression imaging processes.



Figure 8. Class Activation Maps (CAM) resulting from two preprocessing procedures used in training ImageNet-pretrained models on the chirality task: (top row) simple bilinear resizing and (bottom row) random cropping. Recall from the main paper that the CAM tends to fire on discriminative regions for classification. Note the heavy focus on edge and corner regions on bilinear resized images, likely due to edge artifacts caused by JPEG compression or demosaicing (or both). These artifacts disappear when random cropping is applied.

References

- [1] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 7
- [2] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, 14(3):360–369, March 2005. 4