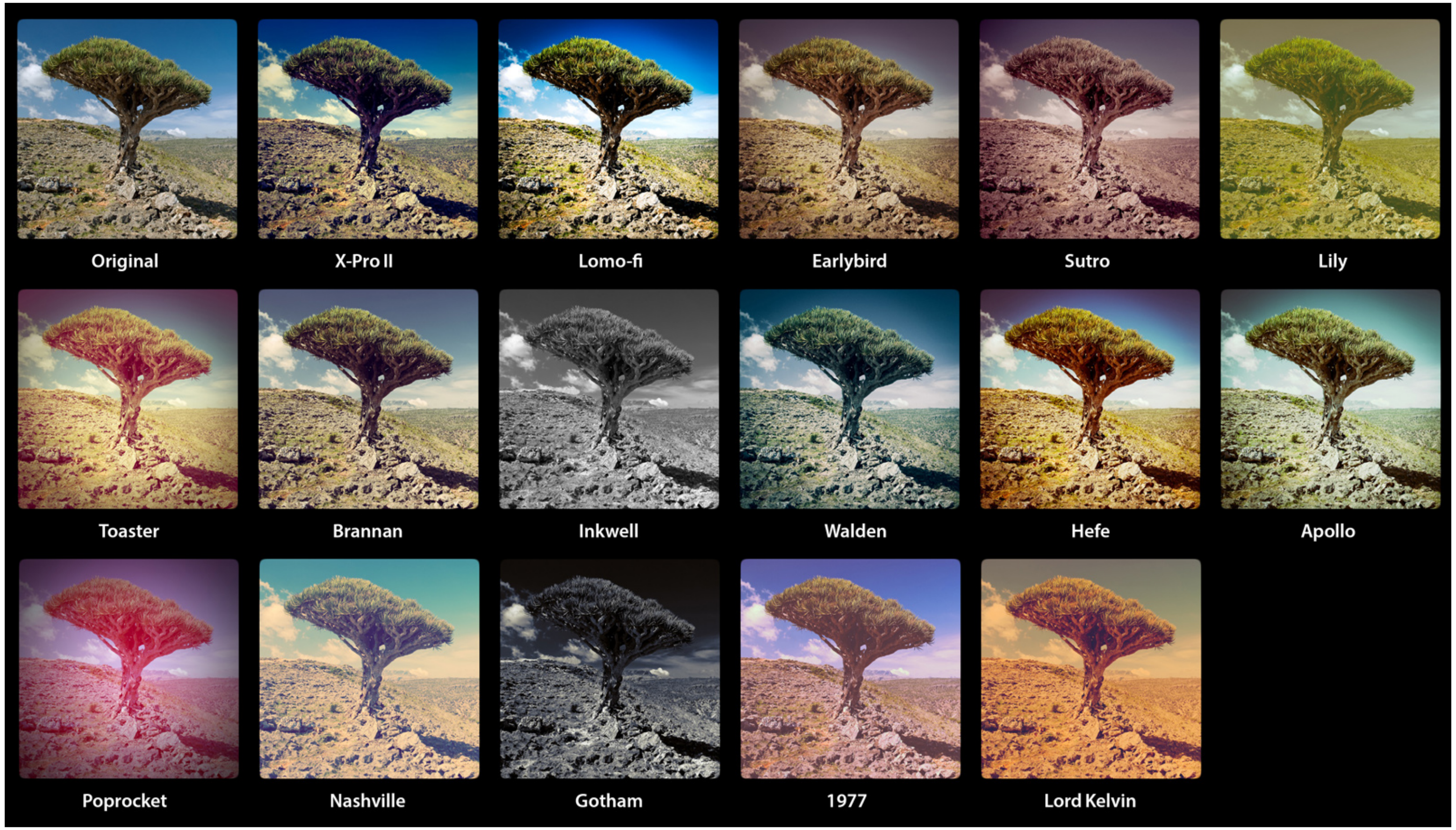
Functors, Comonads, and Digital Image Processing

JUSTIN LE, CHAPMAN UNIVERSITY SCHMID COLLEGE OF SCIENCE AND TECHNOLOGY

Let’s talk about Filters

The problem with filters



getglasses.com.au

Categories

$\mathcal{S}$
Set

Objects

$\mathbb{Z}$ (integers)

$\mathbb{R}$ (real numbers)

$\mathbb{P}$ (people)

Morphisms

$round : \mathbb{R} \to \mathbb{Z}$

$age : \mathbb{P} \to \mathbb{R}$

Composition

$round \circ age : \mathbb{P} \to \mathbb{Z}$

$(round \circ age)(p) = round(age(p))$

Functors

- Objects to Objects
- Morphisms to Morphisms

Ex: Infinite List Functor

$$L(X) = X^{\mathbb{N}}$$

X to infinite lists of things in X

92 4, 9, 8, 75, -3, ...

$\in \mathbb{Z}$ $\in L(\mathbb{Z})$

$$f : X \to Y$$

$$L(f) : L(X) \to L(Y)$$

Comonads

1. Extract $\epsilon : W(X) \to X$
2. Duplicate $\delta : W(X) \to W(W(X))$
3. Laws $\epsilon \circ \delta = id$, etc.

Infinite List Functor

$\epsilon([4, 9, 8, 75, -3..]) = 4$ $\delta([4, 9, 8, 75, -3..]) = ???$

Cokleisli Arrows

$$f : W(X) \to Y \qquad g : W(Y) \to Z$$

$$g \circ f : W(X) \to Z$$

Comonads only!

Extension

$$f : W(X) \to Y$$

$$f^*: W(X) \to W(Y)$$

Functor 1: "Image with Focus"

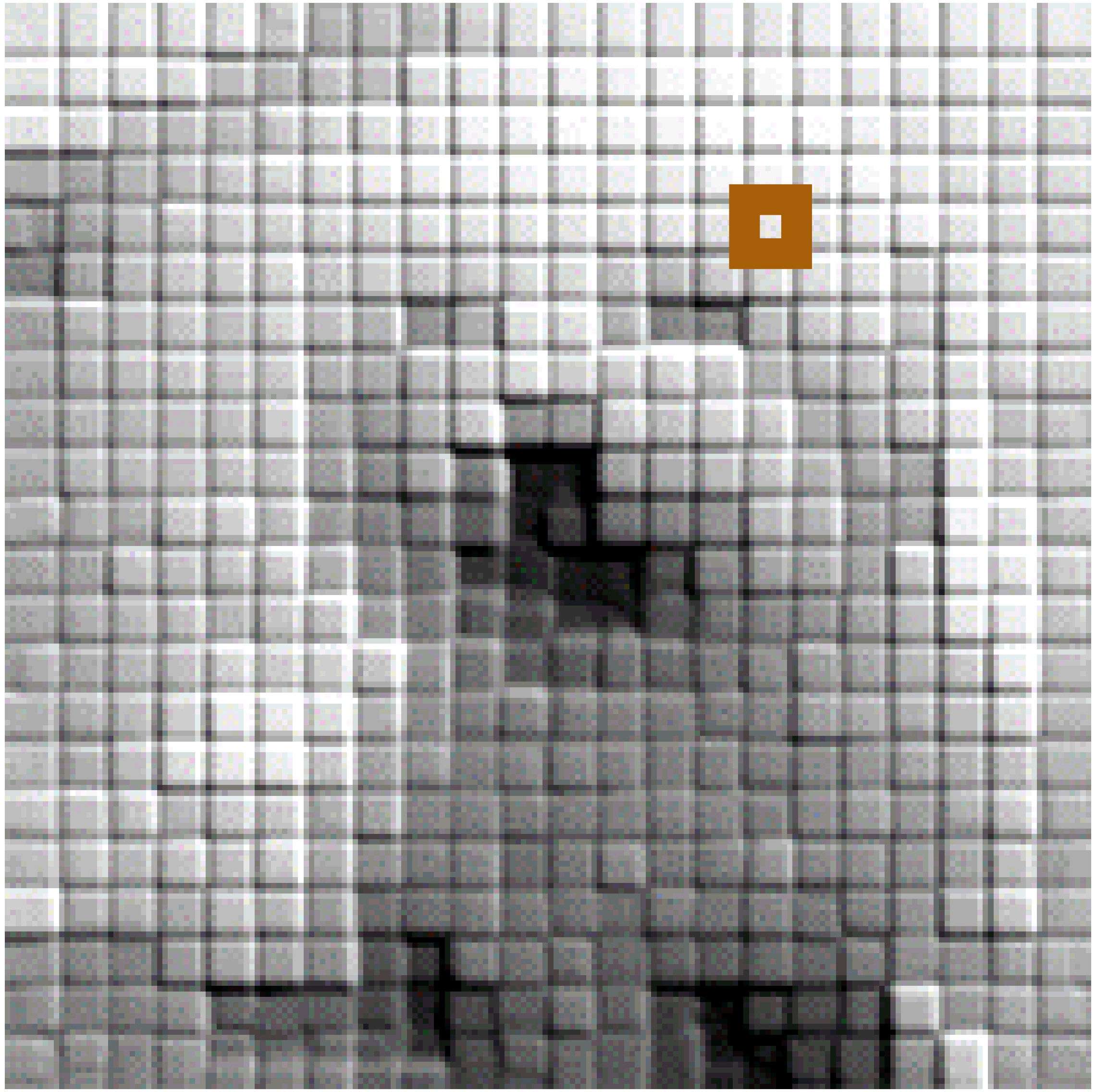
$$I(X) = \mathbb{Z}^2 \times X^{\mathbb{Z}^2}$$

$$5$$

$$\left((5,2), \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & ⋰ \\ \cdots & 7 & 3 & 19 & \cdots \\ \cdots & 22 & 4 & 120 & \cdots \\ \cdots & 8 & 79 & 1 & \cdots \\ ⋰ & \vdots & \vdots & \vdots & \ddots \end{bmatrix}\right)$$

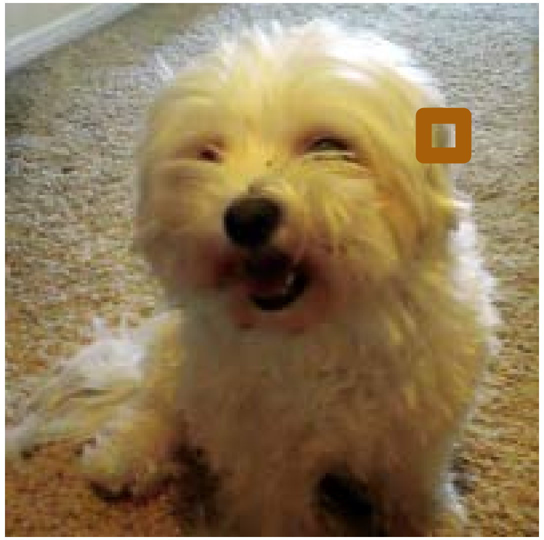
$\in \mathbb{N}$

$\in I(\mathbb{N})$



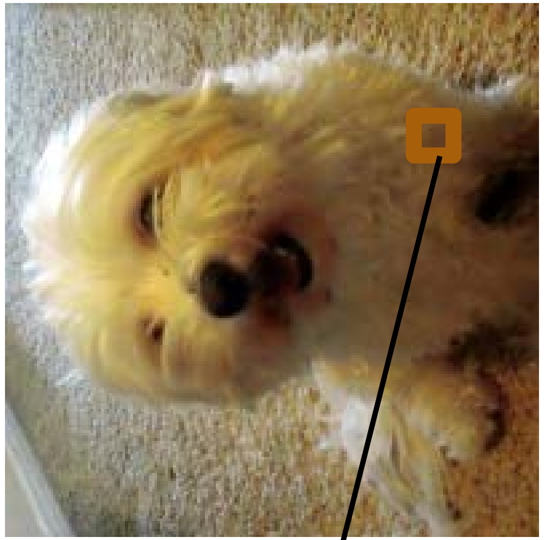
$\in I([0, N] \subset \mathbb{N})$

Position-Aware Transformation

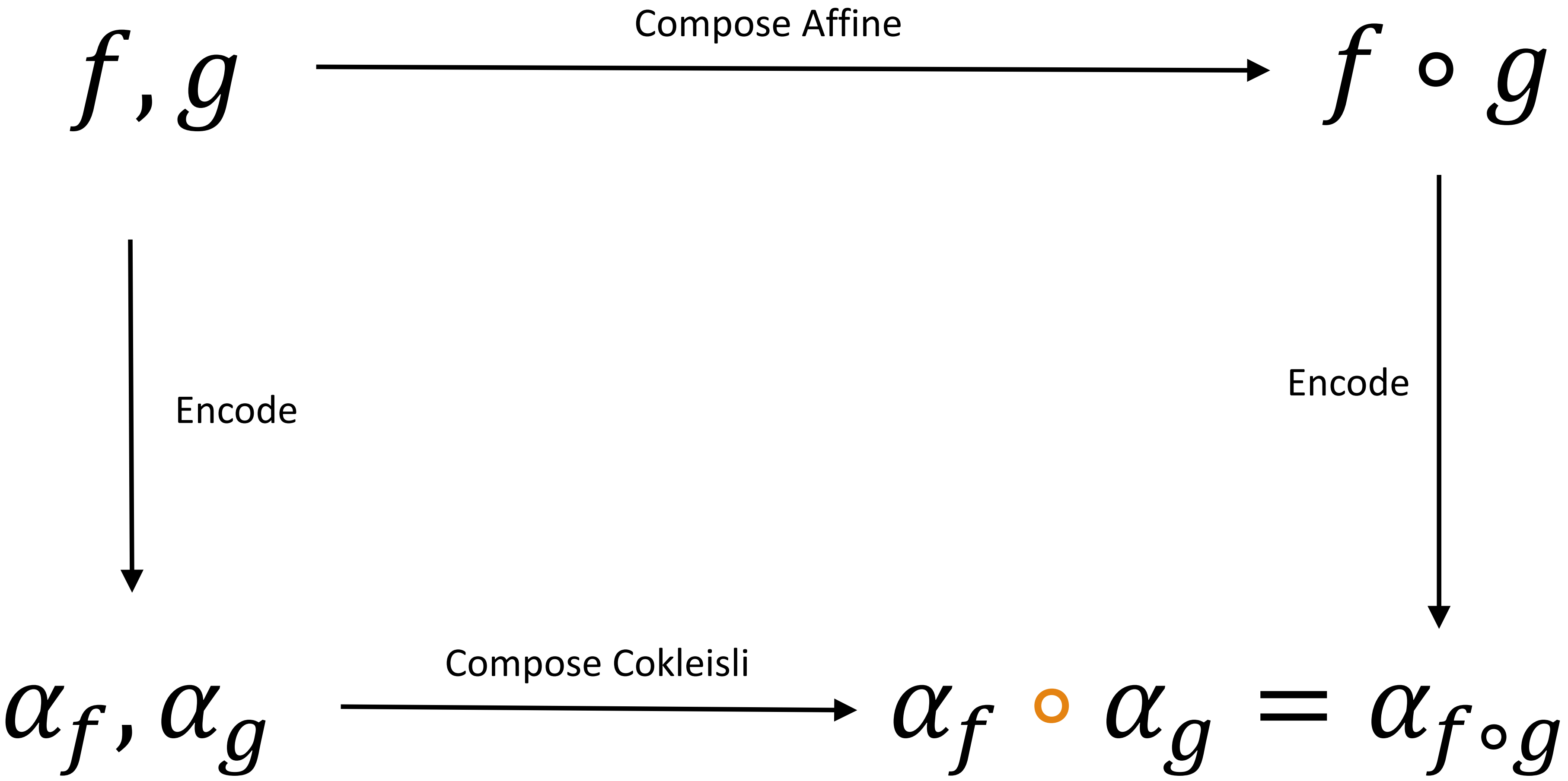


$$\begin{bmatrix} 0 & -1 & w \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Affine Transformation Matrix



$$I(\mathbb{N}) \rightarrow \mathbb{N}$$


f, g
Compose Affine
$f \circ g$
Encode
Encode
α_f, α_g
Compose Cokleisli
$\alpha_f \circ \alpha_g = \alpha_{f \circ g}$

Functor 2: "Local Neighborhood"

$$\mathsf{G}(X) = X^{\mathbb{Z}^2}$$

$$5$$

$$\in \mathbb{N}$$

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & ⋰ \\ \cdots & 7 & 3 & 19 & \cdots \\ \cdots & 22 & 4 & 120 & \cdots \\ \cdots & 8 & 79 & 1 & \cdots \\ ⋰ & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$\in G(\mathbb{N})$$

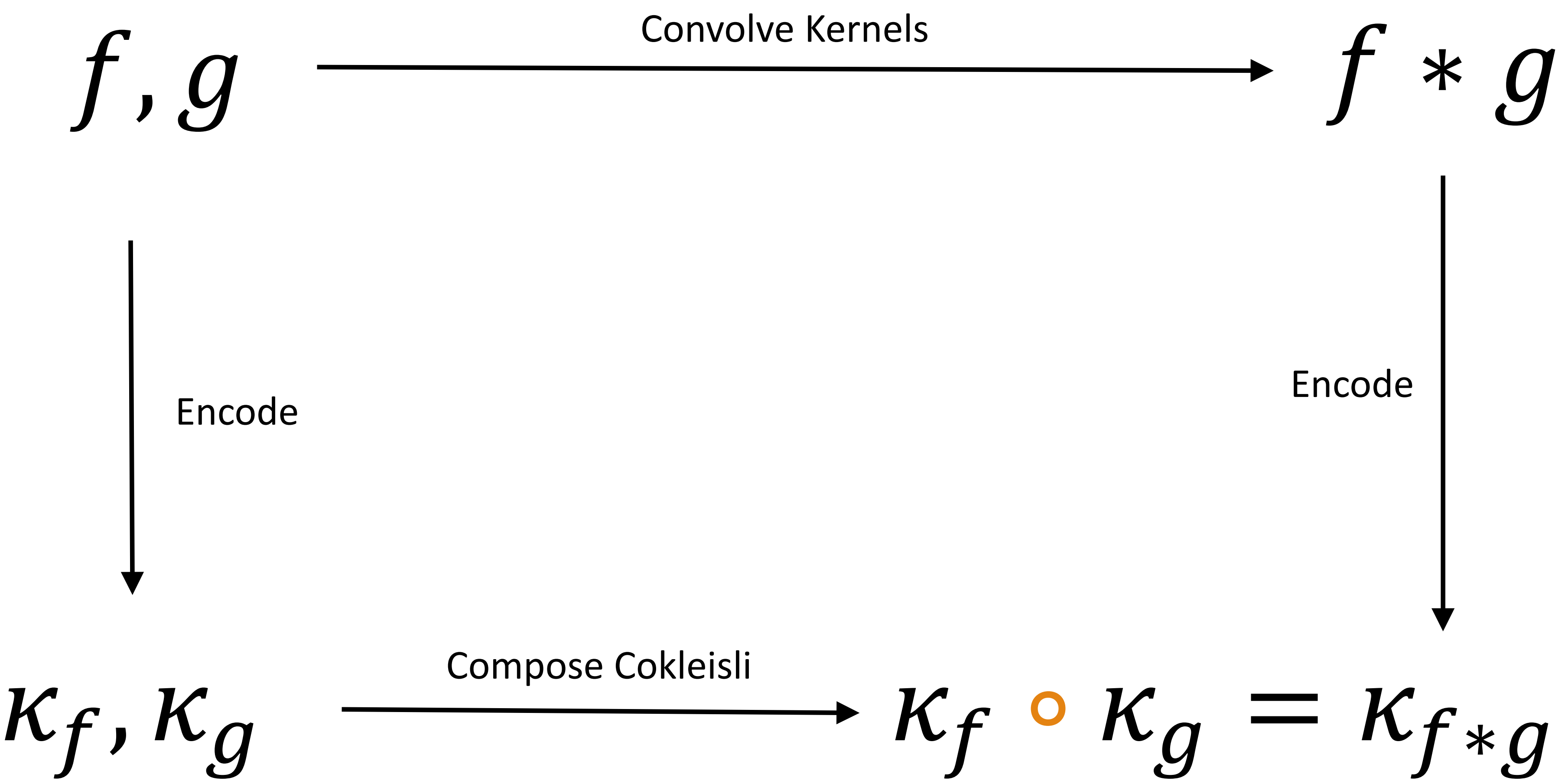
Local/Relative Transformations

$$\longrightarrow \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \longrightarrow$$

Kernel/Convolution Matrix

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \iddots \\ \cdots & 7 & 3 & 19 & \cdots \\ \cdots & 2 & 5 & 6 & \cdots \\ \cdots & 8 & 1 & 9 & \cdots \\ \iddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \xrightarrow{\qquad\qquad} 13$$

$G(\mathbb{Z})$ $\qquad\qquad$ $\mathbb{Z}$



f, g
Convolve Kernels
$f * g$
Encode
Encode
Compose Cokleisli
κ_f, κ_g
$\kappa_f \circ \kappa_g = \kappa_{f*g}$

Extensions of I are Decoded Filters

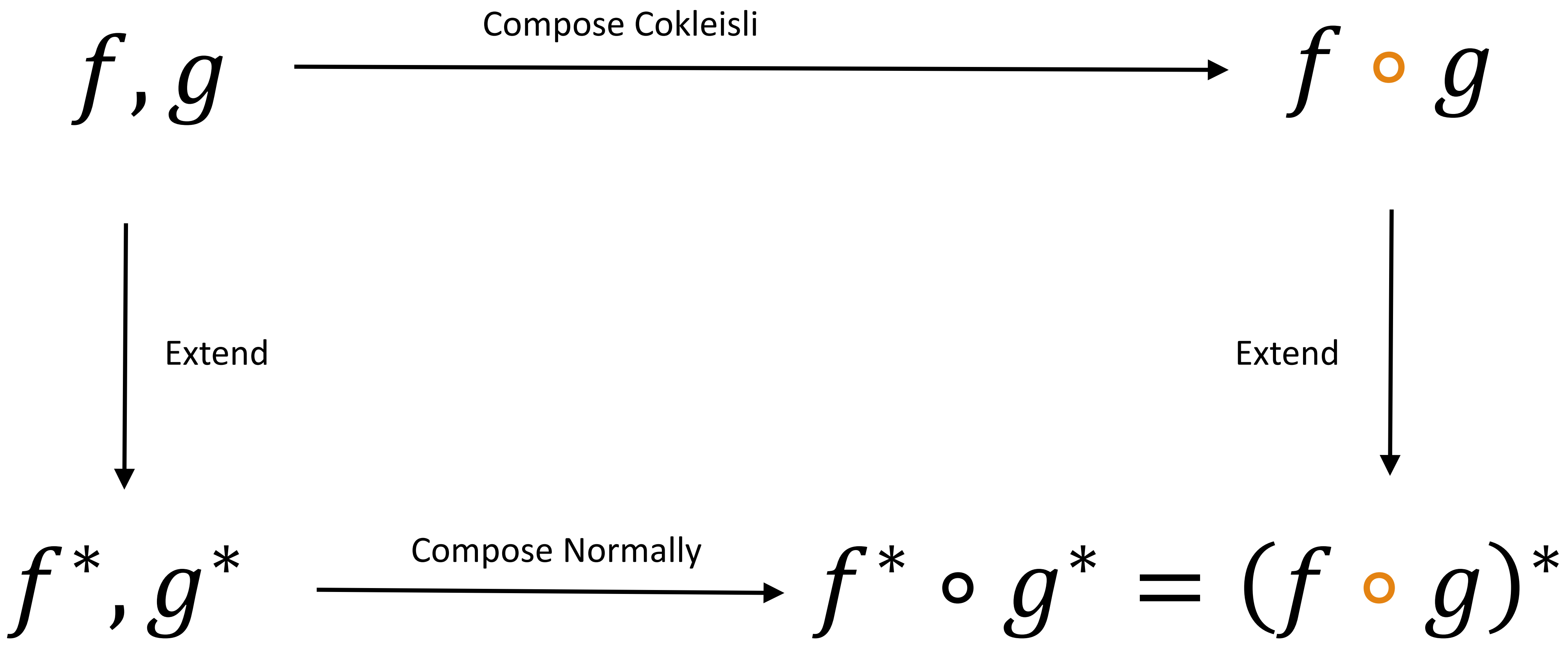
$$f : I(X) \to Y \longrightarrow f^* : I(X) \to I(Y)$$

focus stays same

$$f^* : X^{\mathbb{N}^2} \to Y^{\mathbb{N}^2}$$

Classical image filter

Commutation Abounds



Decoded Neighborhoods

$$f : G(X) \to Y$$

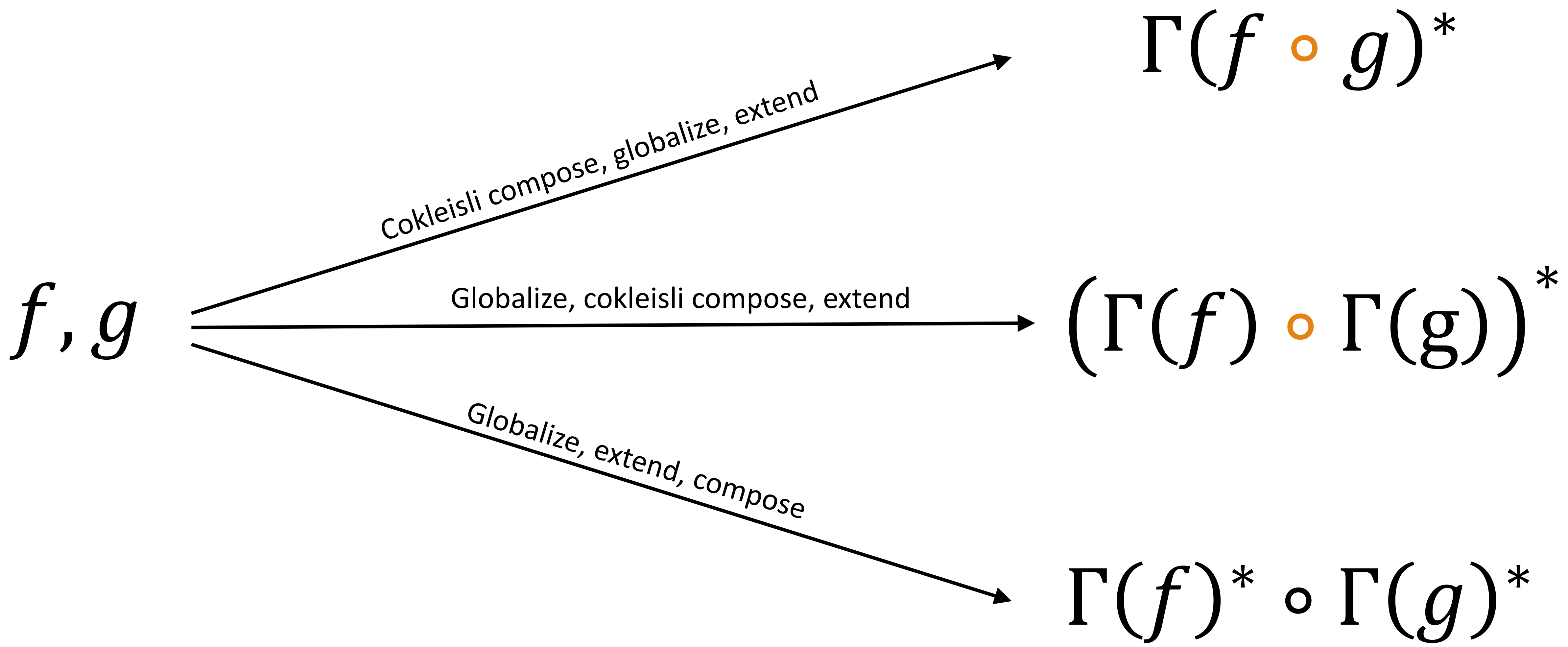
↓ "Globalization", Γ

$$\Gamma(f) : I(X) \to Y$$

→

$$\Gamma(f)^* : X^{\mathbb{N}^2} \to Y^{\mathbb{N}^2}$$

Classical image filter



f, g
Cokleisli compose, globalize, extend
$\Gamma(f \circ g)^*$
Globalize, cokleisli compose, extend
$\left(\Gamma(f) \circ \Gamma(g)\right)^*$
Globalize, extend, compose
$\Gamma(f)^* \circ \Gamma(g)^*$

Extension, Globalization are Cheap

- Written once: **less bugs**
- Optimize once, **unlimited return** on performance
- Trivially **parallelizable**
- Globalization can handle boundary conditions, low-level

Generalization

$$I_n(X) = \mathbb{Z}^n \times X^{\mathbb{Z}^n}$$

$$G_n(X) = X^{\mathbb{Z}^n}$$

n	Application
1	Audio, Time signals
2	Images
3	Video
1000+	Difference Equations

In Conclusion

- Better **math**
- Better **engineering**
- Better **development process**
- Better **world**