# Geometric Adaptations of PDE-G-CNNs

Gijs Bellaard*, Gautam Pai*, Javier Olivan Bescos†, Remco Duits*

* Eindhoven University of Technology, Eindhoven, The Netherlands
† Philips, Eindhoven, The Netherlands

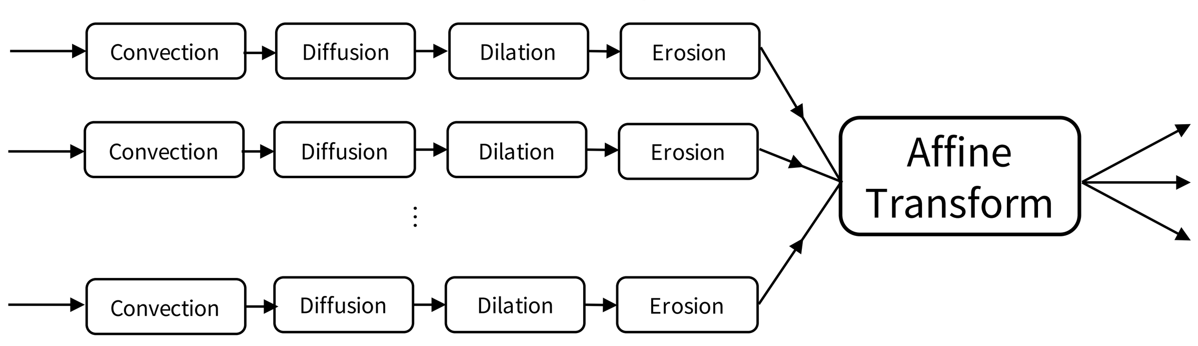
## What are PDE-G-CNNs?

In a **PDE-based group equivariant convolution neural network (PDE-G-CNN)** the usual components that make up **CNNs**, that being convolutions, max pooling, and activation functions, are replaced by solvers for **geometrically interpretable partial differential equations (PDEs)**. The parameters that describe the PDEs are learned during training. Moreover, just like **G-CNNs**, PDE-G-CNNs are **group equivariant**. This means, for example, that when an input image is roto-translated one can be **certain** that the output of the network roto-translated accordingly.

| | CNN | G-CNN | PDE-G-CNN |
|---|---|---|---|
| Equivariant[1] | ✗ | ✓ | ✓ |
| Interpretable | ✗ | ✗ | ✓ |

Specifically, the PDEs are **convection, diffusion, dilation, and erosion**. These PDEs respectively correspond to shifting, smoothing, max pooling, and min pooling. They are solved by resampling, linear convolutions, and morphological convolutions.

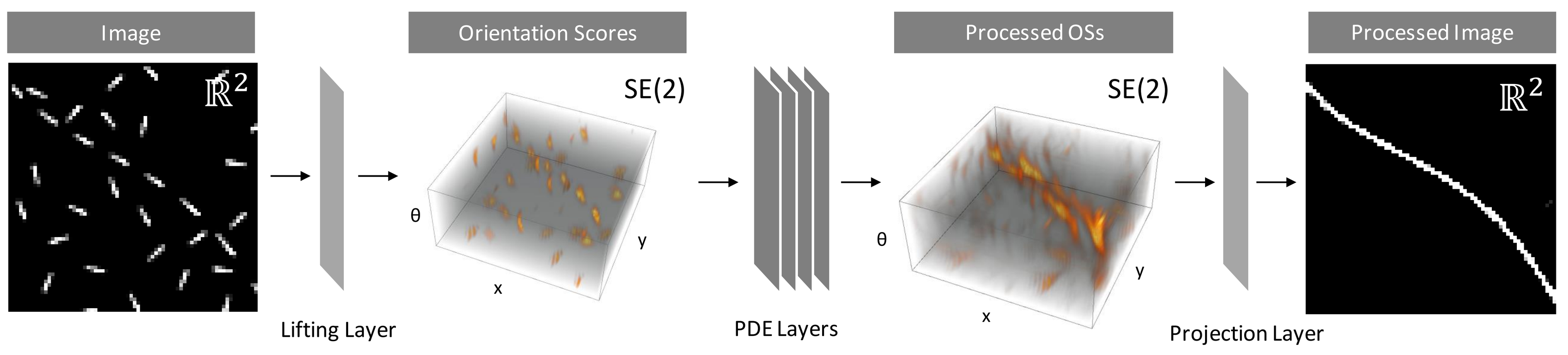
PDE Layer




| Name | Convection | Diffusion | Dilation | Erosion |
|---|---|---|---|---|
| Solver | Resampling | Linear Conv | Morphological Conv | Morphological Conv |
| PDE | $\frac{\partial W}{\partial t} = -v \cdot \nabla W$ | $\frac{\partial W}{\partial t} = \frac{1}{2}\Delta W$ | $\frac{\partial W}{\partial t} = \frac{1}{2}\lVert\nabla W\rVert^2$ | $\frac{\partial W}{\partial t} = -\frac{1}{2}\lVert\nabla W\rVert^2$ |

However, in stark contrast to classical CNNs the feature maps of PDE-G-CNNs **do not live on** $\mathbb{R}^2$. Instead the feature maps are defined on $SE(2)$**: the group of rigid body motions of the plane**. Consequently, the PDEs are in actuality defined on $SE(2)$. This change in architecture is necessary to achieve roto-translation equivariance, while not sacrificing expressivity. We represent points in $SE(2)$ by $(x, y, \theta) \in \mathbb{R}^2 \times [0, 2\pi)$ where $(x, y)$ represent the translation and $\theta$ the rotation.

Given that the input and output of the network are images on $\mathbb{R}^2$, but our feature maps are on $SE(2)$, we need a way to translate between the two. The process of going (in an equivariant manner) from $\mathbb{R}^2$ to $SE(2)$ is called **lifting**, and vice versa **projecting**.

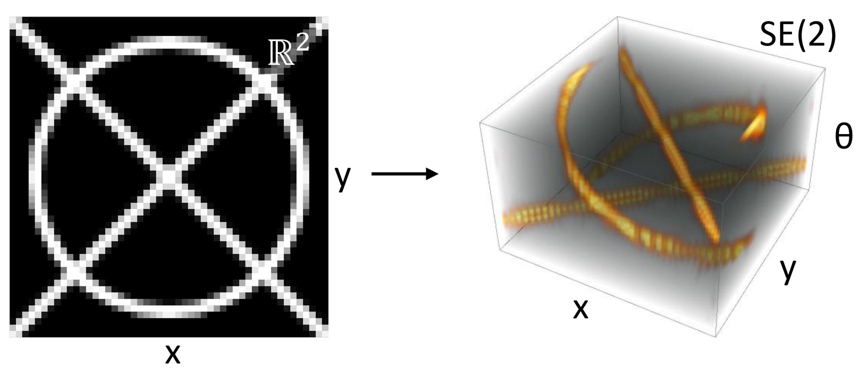



## Lifting, Orientations Scores & Cake Wavelets

There is one very natural way to **lift** images using **orientation score transforms**. Intuitively, the orientation score transform $\mathcal{W}$ takes a **kernel** $\psi$, roto-translates it across the input image $f$, and "saves" the response as a function on $SE(2)$.

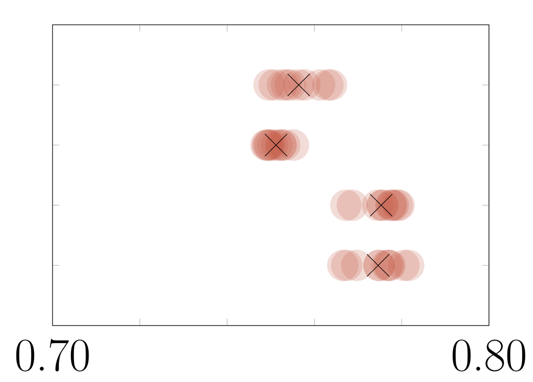
$$(\mathcal{W}_\psi f)(x, y, \theta) := \int_{\mathbb{R}^2} \psi(R_\theta^{-1}(\boldsymbol{y} - [\begin{smallmatrix} x \\ y \end{smallmatrix}]))\ f(\boldsymbol{y})\ d\boldsymbol{y},$$

If an appropriate kernel is chosen to perform the lifting, such as **cake wavelets**[2], the lifted image becomes exceedingly **interpretable**: if the image contains a local orientation $\theta$ at position $x, y$ the response at $(x, y, \theta) \in SE(2)$ is high.




In the original PDE-G-CNN paper[3] lifting was performed with **learned kernels**. The interpretability of PDE-G-CNNs would improve drastically if we used the cake wavelets, but by doing this we might lose some performance. Experiments on a vessel segmentation dataset (DCA1) tell us the following: **lifting with cake wavelets increases interpretability while not sacrificing performance**.

| Type | Lifting | Params. | Dice Coeff. |
|---|---|---|---|
| CNN | - | 25662 | 0.756 |
| G-CNN | Trained | 24632 | 0.751 |
| PDE-G-CNN | Trained | 2560 | 0.775 |
| PDE-G-CNN | Fixed | 2536 | 0.775 |




[1]Technically CNNs are translation equivariant, but we are discussing $SE(2)$ equivariance.
[2]An online visualization of cake wavelets can be found at `www.bellaard.com/apps/Cake Wavelets/`
[3]`https://link.springer.com/article/10.1007/s10851-022-01114-x`
[4]The metrics on $SE(2)$ we consider are left-invariant and can therefor completely be determined by a single $3 \times 3$ matrix.
[5]The actual distance $d$ is not visualized here. Instead we see an approximation of $d$ called the logarithmic distance approximation $\rho_c$.
[6]In the paper one more geometric adaptation is considered.

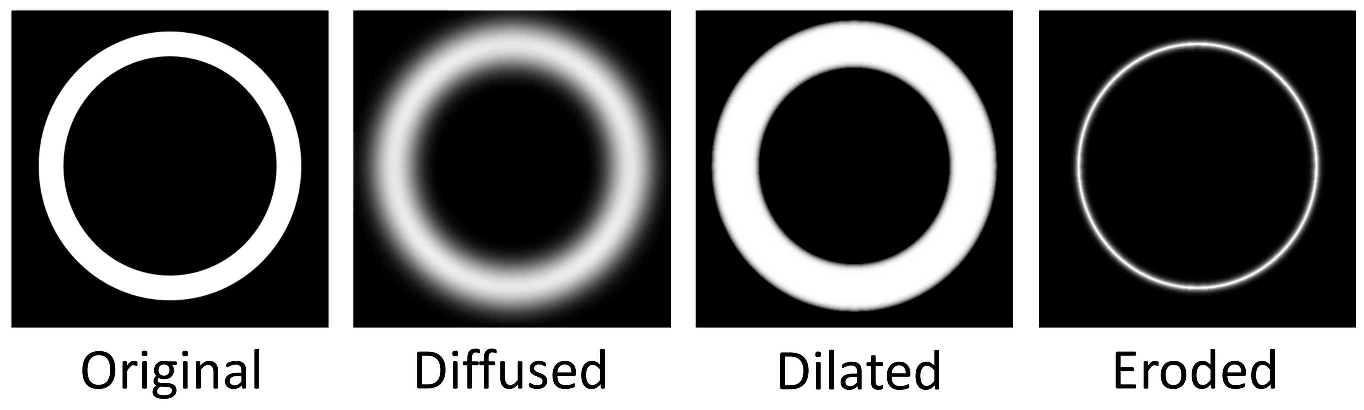
## Diffusion, Dilation, Erosion & Convolutions

**Diffusion** with initial condition $W|_{t=0} = U$ in $\mathbb{R}^2$ can be solved using a **linear convolution** $*$ with a **kernel** $k_t$. Similarly, **dilation & erosion** in $\mathbb{R}^2$ are solved using a **morphological convolution** $\square$ with a different kernel.

| Name | PDE | Solution | Kernel |
|---|---|---|---|
| Diffusion | $\frac{\partial W}{\partial t} = \frac{1}{2}\Delta W$ | $W = k_t * U$ | $k_t(\boldsymbol{x}) \propto \exp(-\frac{1}{2t}\lVert\boldsymbol{x}\rVert^2)$ |
| Dilation | $\frac{\partial W}{\partial t} = +\frac{1}{2}\lVert\nabla W\rVert^2$ | $W = -(k_t \,\square\, -U)$ | $k_t(\boldsymbol{x}) = \frac{1}{2t}\lVert\boldsymbol{x}\rVert^2$ |
| Erosion | $\frac{\partial W}{\partial t} = -\frac{1}{2}\lVert\nabla W\rVert^2$ | $W = k_t \,\square\, U$ | $k_t(\boldsymbol{x}) = \frac{1}{2t}\lVert\boldsymbol{x}\rVert^2$ |

$$(k * f)(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\mathbb{R}^2} k(\boldsymbol{x} - \boldsymbol{y}) \cdot f(\boldsymbol{y})\ d\boldsymbol{y}$$

$$(k \,\square\, f)(\boldsymbol{x}) = \inf_{\boldsymbol{y}\in\mathbb{R}^2} k(\boldsymbol{x} - \boldsymbol{y}) + f(\boldsymbol{y})$$




However, remember that in a PDE-G-CNN these PDEs take place on $SE(2)$. Luckily the solution method generalizes straightforwardly to $SE(2)$: we replace the convolutions with **group convolutions** and replace $\lVert\boldsymbol{x}\rVert = d(\boldsymbol{0}, \boldsymbol{x})$ in the kernels $k_t$ with the **Riemannian distance** $d(e, g)$ to the identity element $e$ on $SE(2)$.

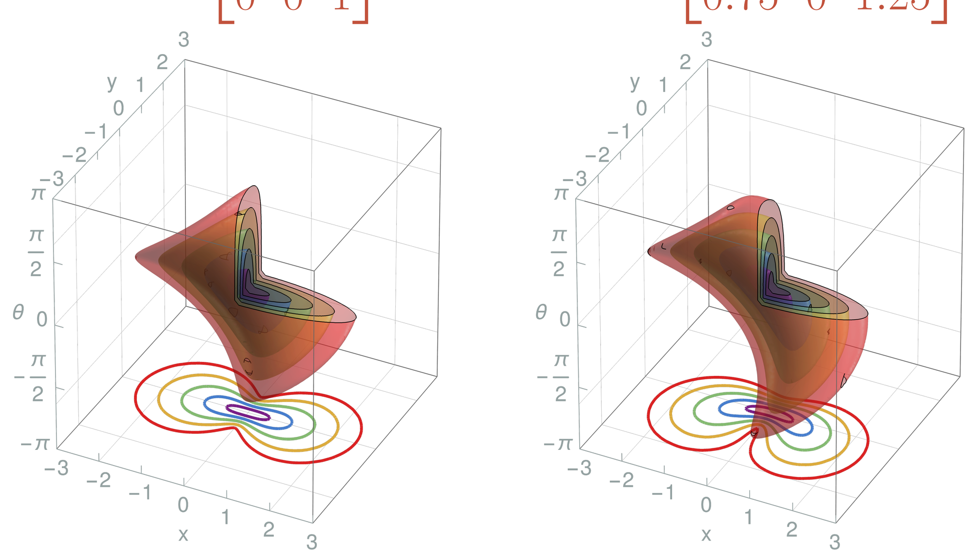
## (Non)Diagonal Metrics

But to turn $SE(2)$ into a **Riemannian manifold** we need to endow it with a **metric** $\mathcal{G}$. The group $SE(2)$ is a 3-dimensional manifold, and to fully describe the metric we need a single $3 \times 3$ **symmetric positive definite** (SPD) matrix[4]. This can be compared with the definition of an **inner product space**.

The effect of diffusion, dilation and erosion are determined completely by the metric and thus, in turn, determined completely by this matrix. In fact, the **trainable parameters** in a PDE-G-CNN are exactly these matrices. In the original PDE-G-CNN paper only **diagonal** matrices were considered. Allowing for the more general **non-diagonal** matrices might improve performance of the PDE-G-CNNs as the effect of the PDEs is changed considerably.

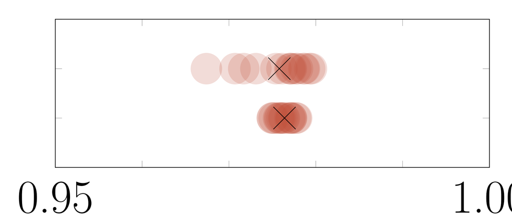
Given that the diffusion, dilation, and erosion kernels $k_t$ are solely functions of the distance $d$, we can get an idea of how the metric affects the PDEs by visualizing the distance $d$. We can visualize the distance $d$ by plotting **Riemannian spheres**: i.e. isocontours of the distance $d$ to the identity $e$[5]. This is analogous to the standard spheres in $\mathbb{R}^3$, that being isocontours of the distance to the origin.

$$\mathcal{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathcal{G} = \begin{bmatrix} 1.25 & 0 & 0.75 \\ 0 & 16 & 0 \\ 0.75 & 0 & 1.25 \end{bmatrix}$$




Experiments on a synthetic line completion dataset (Lines) tell us the following: **Allowing for non-diagonal metrics has no significant impact on the average performance.**

| Metric | Params. | Dice Coeff. |
|---|---|---|
| Diagonal | 6018 | 0.9758 |
| Nondiagonal | 7458 | 0.9764 |




## Conclusion

We have investigated two[6] geometric adaptations of PDE-G-CNNs.

- We showed on the DCA1 dataset that one does not need to lift with learned kernels in PDE-G-CNNs. A fixed lifting layer using cake wavelets gives identical performance, while considerably increasing the geometric interpretation of PDE-G-CNNs.
- We showed on the Lines dataset that using nondiagonal metrics gives no significant increase in performance. We hypothesize that the effect of a nondiagonal metric can be (sufficiently) emulated by a diagonal metric combined with convection.

**Acknowledgements:** We gratefully acknowledge the Dutch Foundation of Science (NWO) for funding of VICI 2020 Exact Sciences (Duits, Geometric Learning for Image Analysis, VI.C. 202-031).