FFTs in Graphics and Vision

Alignment, Invariance
and Pattern Matching

Outline

Alignment

Shape Matching

Invariance

Pattern Matching

Shape Representation

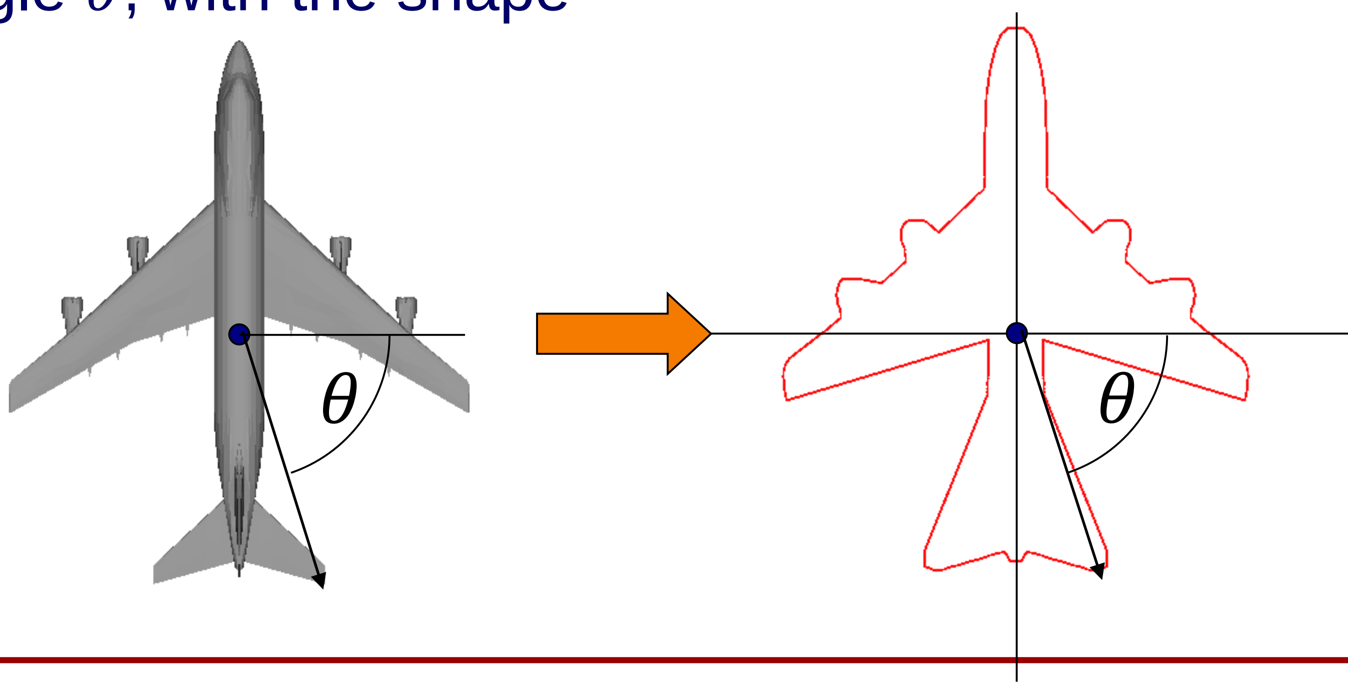
For 2D shape matching/analysis, it is common to represent the geometry of a shape by a circular array of real values.

Shape Representation

Example:

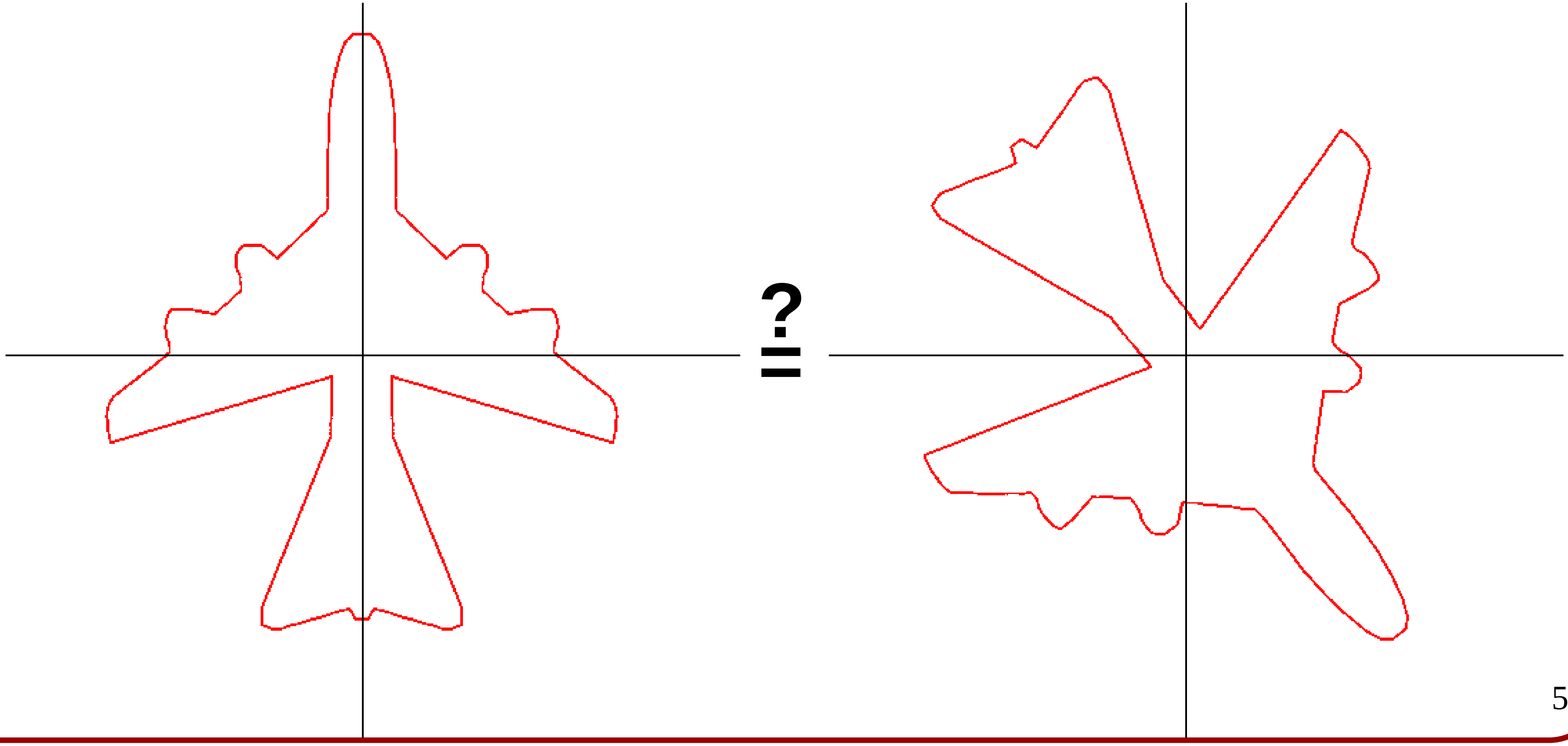
The <u>circular extent function</u> represents the extent of the shape about the center of mass:

- The value at an angle θ is the distance to the last point of intersection of the ray from the origin, with angle θ, with the shape



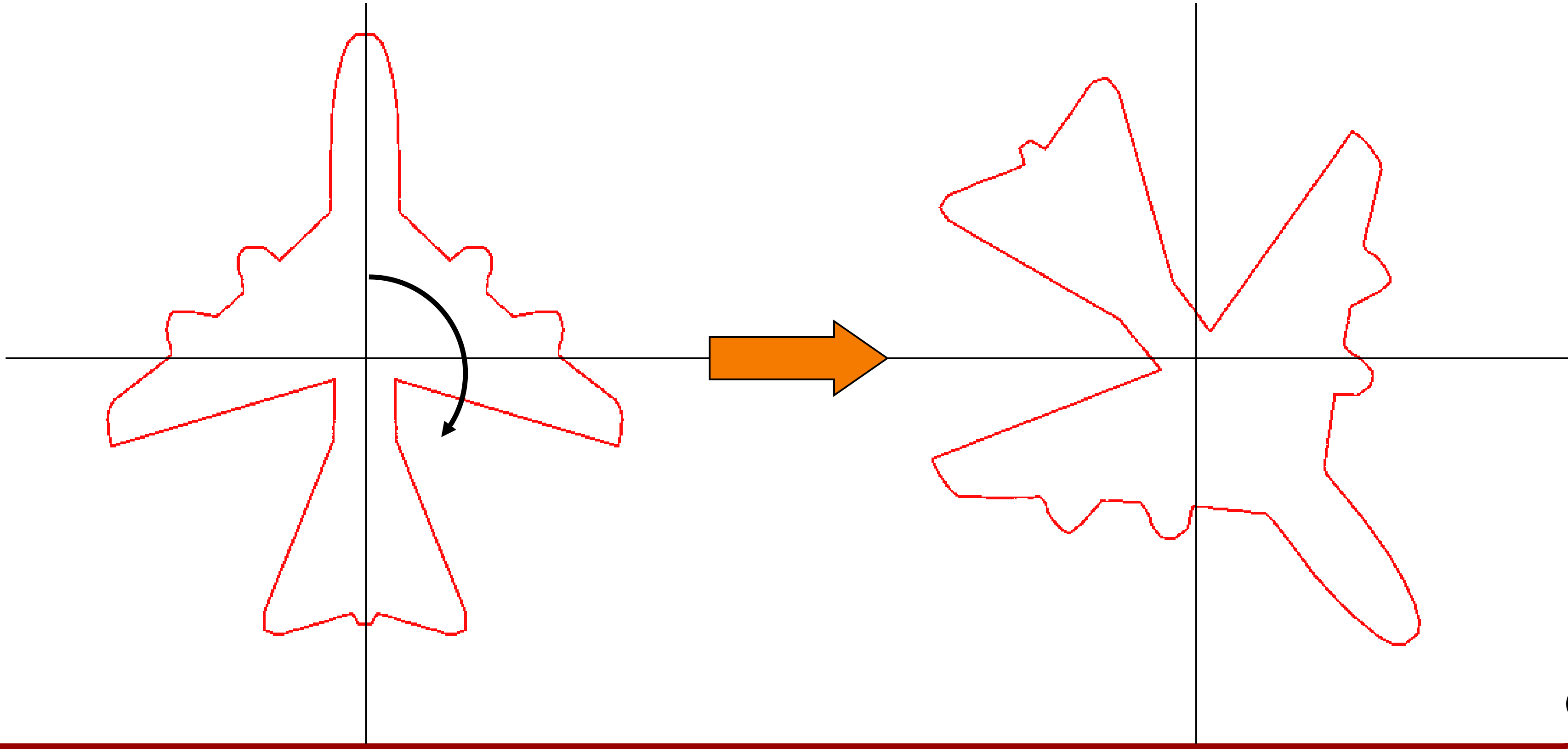
Alignment

Since the shape of an object doesn't change when we rotate it, we would like to know if the two arrays are equivalent, up to rotation.



Alignment

Is there a rotation that will rotate the first array into the second?



Alignment

Is there a rotation that will rotate the first array into the second?

Given the n-dimensional arrays $f[\cdot]$ and $g[\cdot]$, is there an index α such that:

$$g[\cdot] = \rho_\alpha(f[\cdot])$$
$$\Updownarrow$$
$$g[k] = f[k-\alpha] \quad \forall 0 \leq k < n$$

Semantics

In a continuous setting, asking the binary question “are the arrays equal” is not very meaningful, since

- Sampling
- Noise
- Etc

can cause “equal” arrays to have different values.

Semantics

Is there a rotation that will rotate the first array into the second?

For every rotation, how close is the rotation of the first array to the second array?

For every rotation α, what is the value of:

$$D(\rho_\alpha(f[\cdot]), g[\cdot])$$

Alignment

Since the space of functions on a circle is an inner product space, we have a metric:

$$D^2(f[\cdot],g[\cdot]) = \|f[\cdot]-g[\cdot]\|^2$$

We would like to evaluate:

$$\begin{aligned} D^2_{f,g}(\alpha) &= D^2(\rho_\alpha(f[\cdot]),g[\cdot]) \\ &= \|\rho_\alpha(f[\cdot])-g[\cdot]\|^2 \\ &= \langle \rho_\alpha(f[\cdot])-g[\cdot],\rho_\alpha(f[\cdot])-g[\cdot]\rangle \end{aligned}$$

at every α.

Alignment

$$D^2_{f,g}(\alpha) = \langle \rho_\alpha(f[\cdot]) - g[\cdot], \rho_\alpha(f[\cdot]) - g[\cdot] \rangle$$

Re-writing this equation gives:

$$D^2_{f,g}(\alpha) = \langle \rho_\alpha(f[\cdot]), \rho_\alpha(f[\cdot]) \rangle + \langle g[\cdot], g[\cdot] \rangle$$
$$-\langle \rho_\alpha(f[\cdot]), g[\cdot] \rangle - \overline{\langle \rho_\alpha(f[\cdot]), g[\cdot] \rangle}$$

Alignment

$$D^2_{f,g}(\alpha) = \langle \rho_\alpha(f[\cdot]), \rho_\alpha(f[\cdot])\rangle + \langle g[\cdot], g[\cdot]\rangle - \langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle - \overline{\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle}$$

Since the Hermitian dot-product of two real-valued arrays is also a real value:

$$D^2_{f,g}(\alpha) = \|\rho_\alpha(f[\cdot])\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

Alignment

$$D^2_{f,g}(\alpha) = \|\rho_\alpha(f[\cdot])\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

Since ρ_α is a unitary transformation:

$$D^2_{f,g}(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

Alignment

$$D^2_{f,g}(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

To compute the distance between a rotation of the circular array $f[\cdot]$ by α, and the circular array $g[\cdot]$, we need to compute:

- The magnitude of $f[\cdot]$: $\|f[\cdot]\|^2$,
- The magnitude of $g[\cdot]$: $\|g[\cdot]\|^2$,
- The value of the cross-correlation: $\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$

Alignment

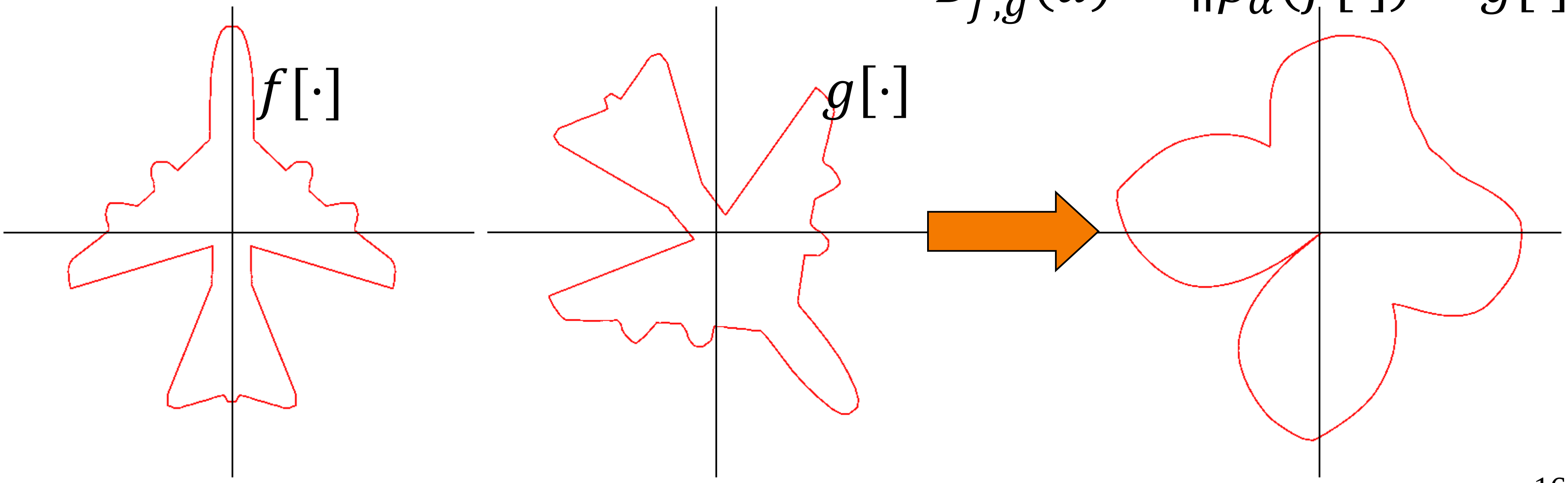
$$D_{f,g}^2(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

- The magnitude of $f[\cdot]$:
  - Constant independent of α: $O(n)$ time.
- The magnitude of $g[\cdot]$:
  - Constant independent of α: $O(n)$ time.
- The value of the cross-correlation:
  - With an FFT: $O(n \log n)$ time.

Alignment

$$D^2_{f,g}(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle$$

$$D^2_{f,g}(\alpha) = \|\rho_\alpha(f[\cdot]) - g[\cdot]\|^2$$

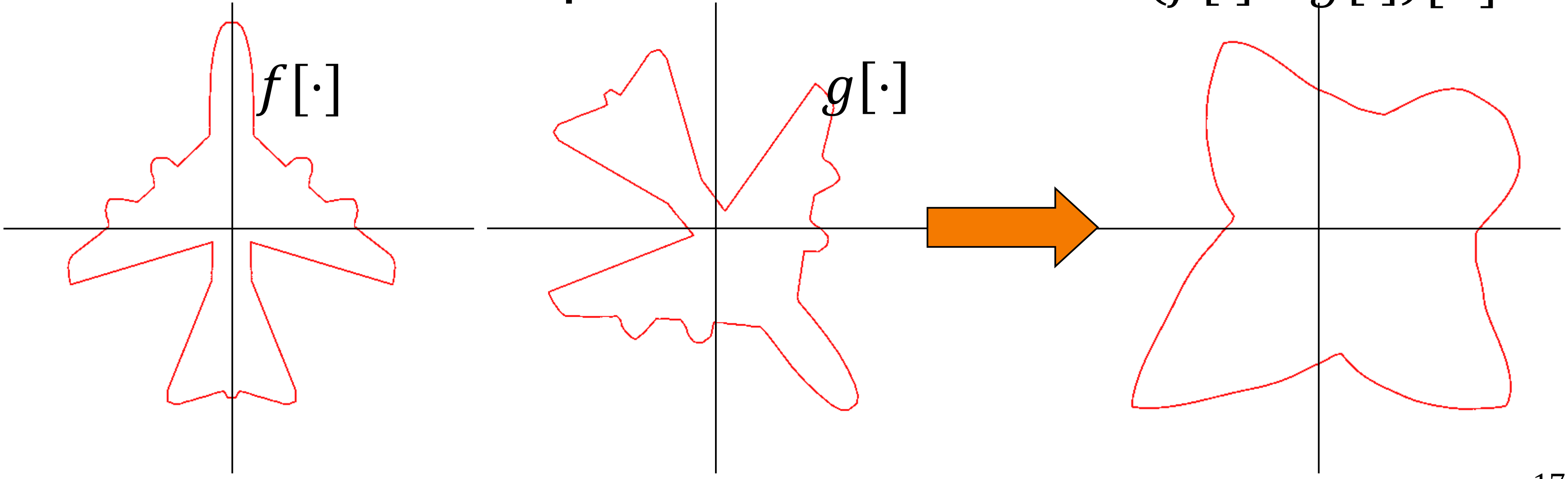


Alignment

$$D^2_{f,g}(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\boxed{\langle \rho_\alpha(f[\cdot]), g[\cdot]\rangle}$$

Because the norms are constant, instead of looking for the minimum distance, we can look for the maximum dot-product.

$(f[\cdot] \star g[\cdot])[\alpha]$

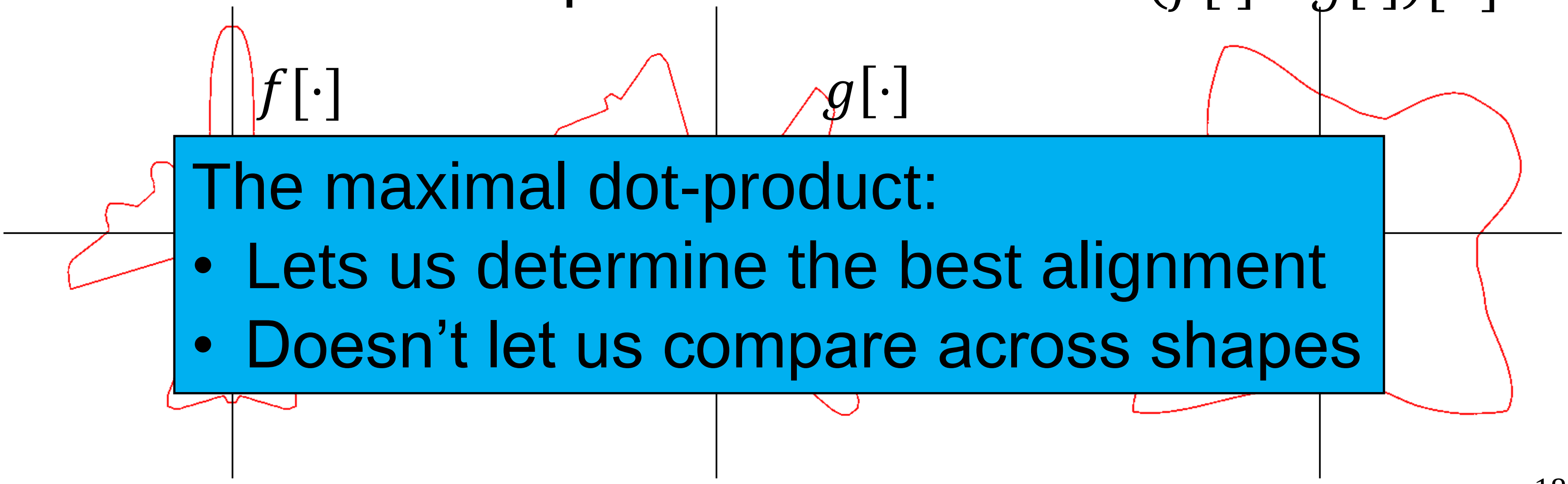


Alignment

$$D_{f,g}^2(\alpha) = \|f[\cdot]\|^2 + \|g[\cdot]\|^2 - 2\boxed{\langle \rho_\alpha(f[\cdot]), g[\cdot] \rangle}$$

Because the norms are constant, instead of looking for the minimum distance, we can look for the maximum dot-product.

$(f[\cdot] \star g[\cdot])[\alpha]$



The maximal dot-product:

- Lets us determine the best alignment
- Doesn't let us compare across shapes

Outline

Alignment

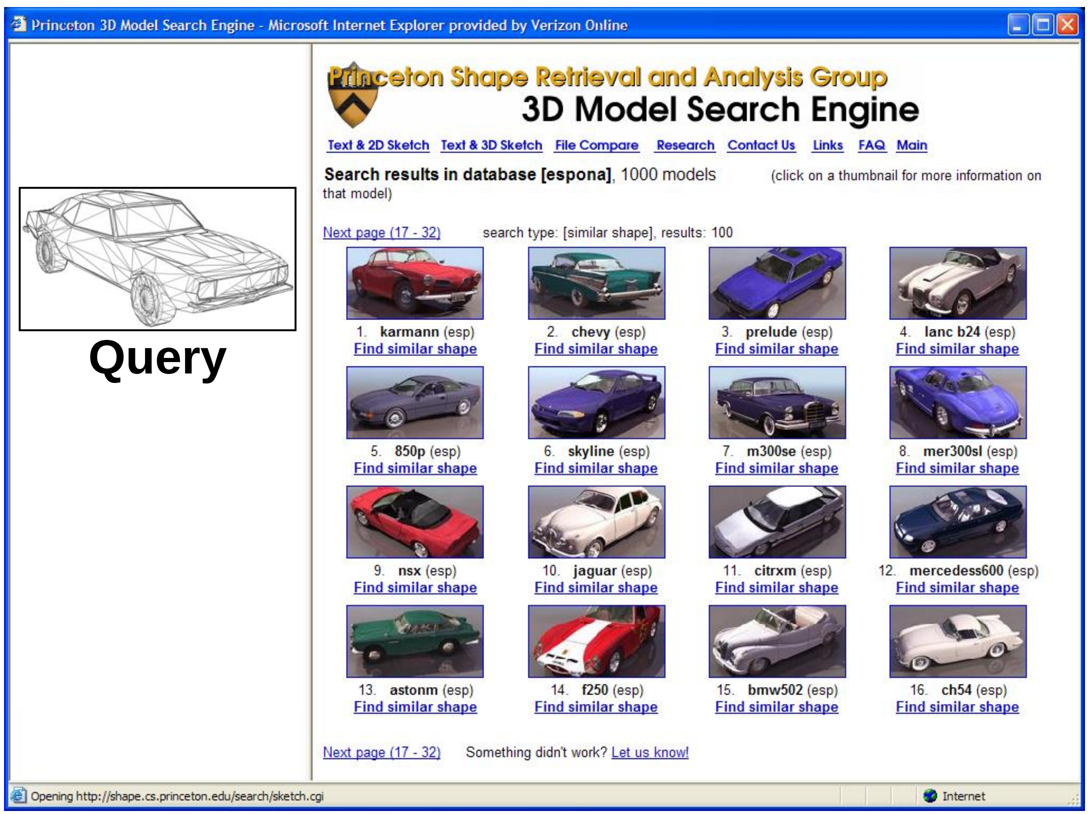
Shape Matching

Invariance

Pattern Matching

Shape Matching

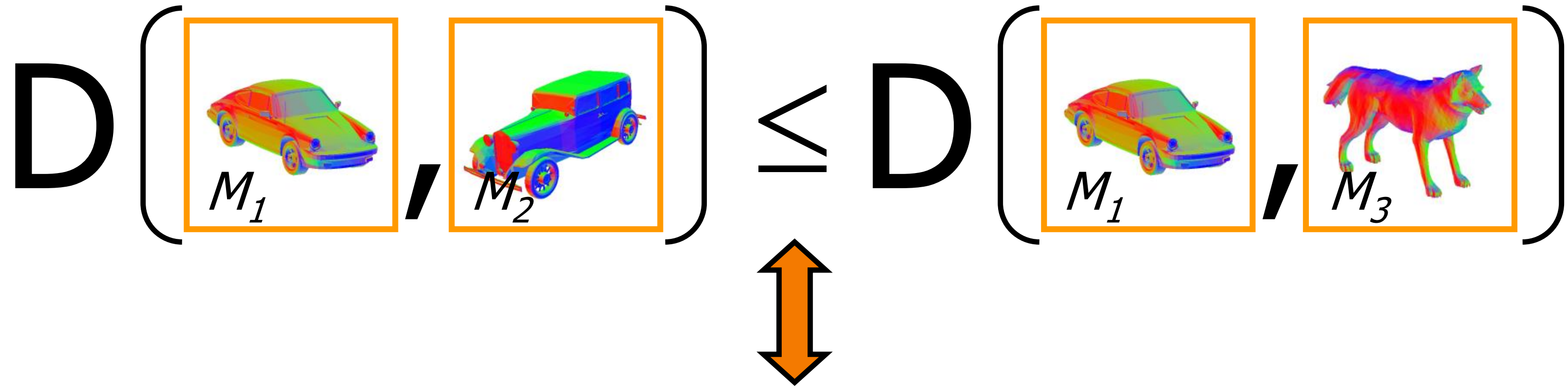
In shape matching applications, we would like to find the shapes in a database that are most similar to a given query.



Shape Matching

General approach:

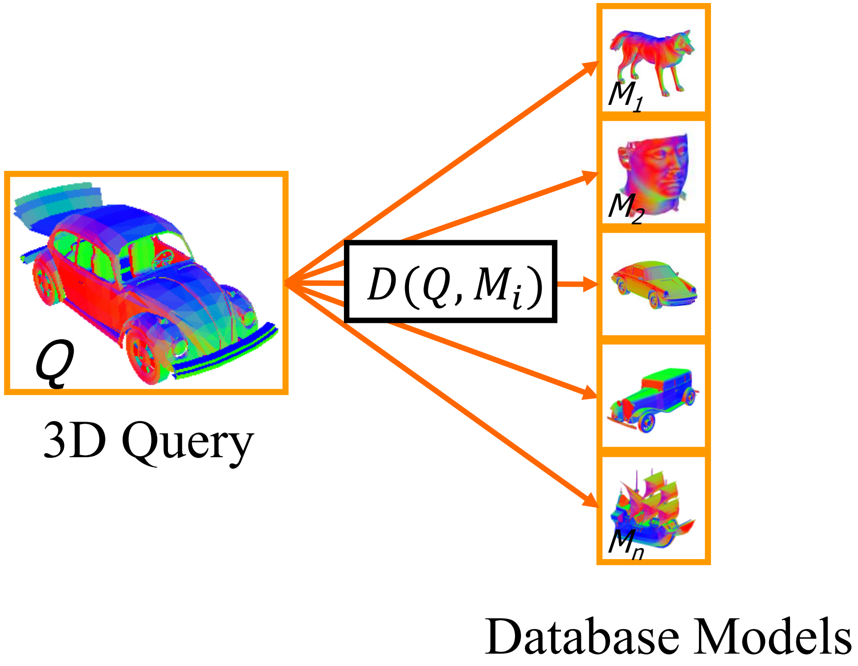
Define a function that takes in two models and returns a measure of their proximity.



M_1 is closer to M_2 than it is to M_3

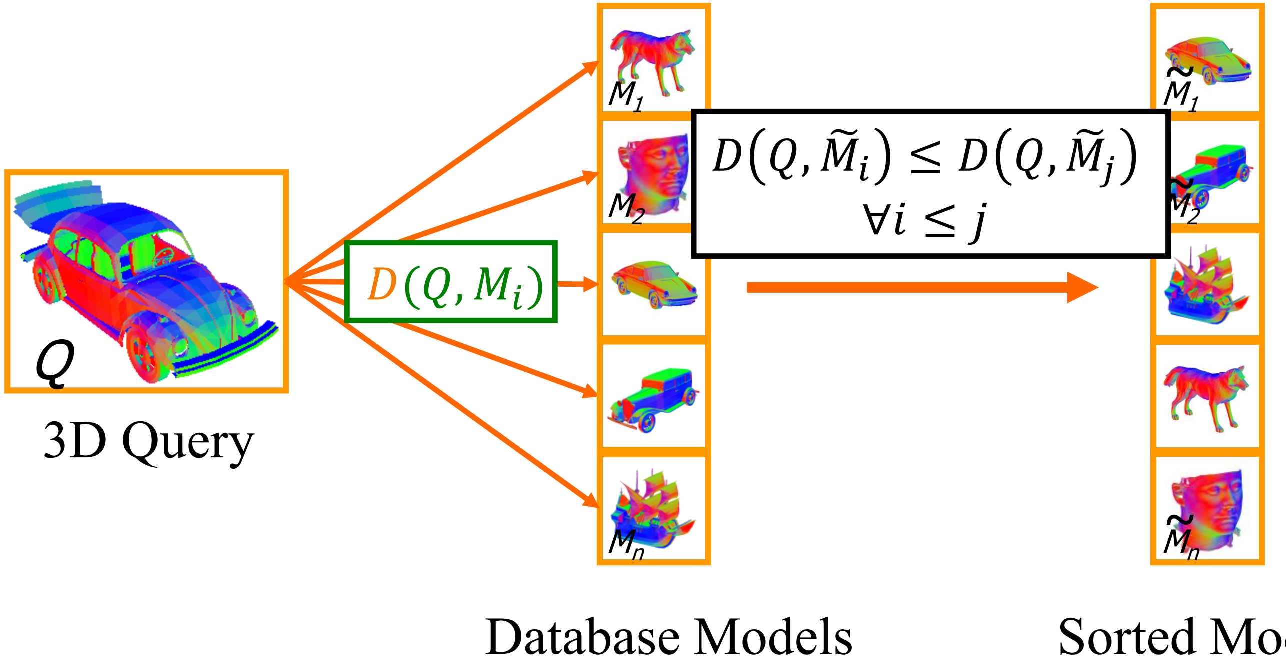
Database Retrieval

- Compute the distance from the query to each database model



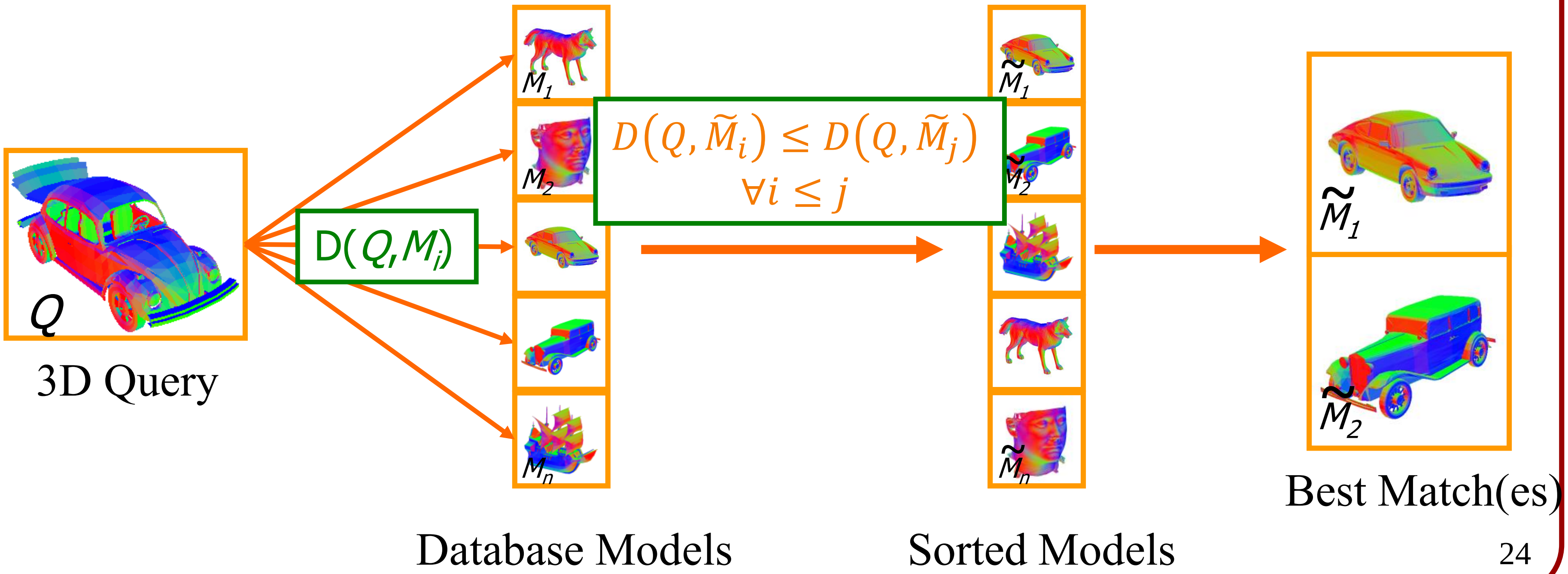
Database Retrieval

- Sort the database models by proximity



Database Retrieval

- Return the closest matches



Shape Matching

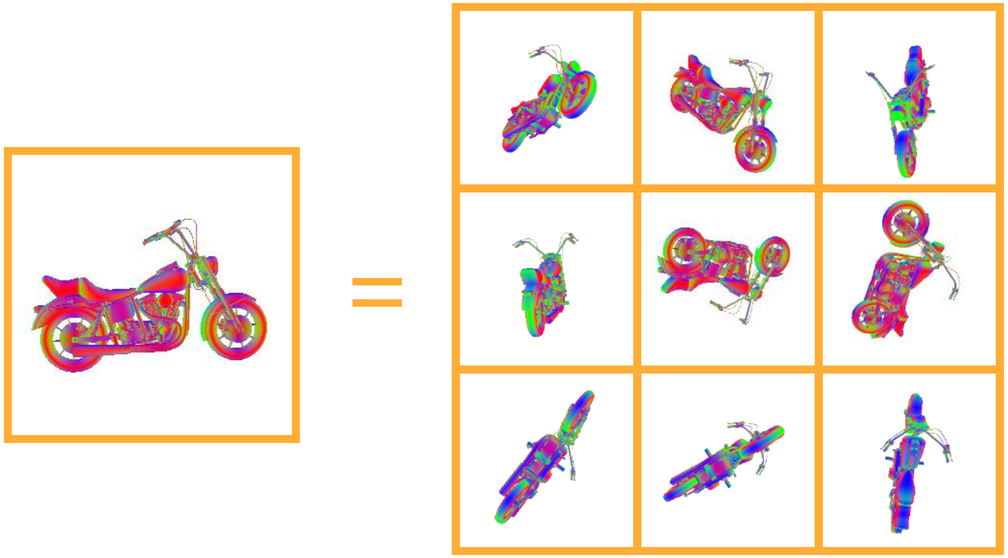
To do this efficiently, models are often represented by *Shape Descriptors*:

- Arrays of values encapsulating information about the shape of the model, such that
- The distance between the arrays gives a measure of proximity of the underlying shapes.

Shape Matching

Challenge:

Since the shape of the model doesn't change if we rotate it, we would like to match models across rotational poses.



Shape Matching

Challenge:

Since the shape of the model doesn't change if we rotate it, we would like to match models across rotational poses.

Solution 1:

Define the measure of similarity by using the FFT to find the distance between two models at the best possible alignment.

Shape Matching

Challenge:

Since the shape of the model doesn't change if we rotate it, we would like to match models across rotational poses.

Solu

Defi

to fi

best

This can be too slow for interactive applications that need to return the best match from very large databases.

Not quite true for 1D arrays, but becomes more true as the dimension increases.

Shape Matching

<u>Challenge</u>:

Since the shape of the model doesn't change if we rotate it, we would like to match models across rotational poses.

<u>Solution 2</u>:

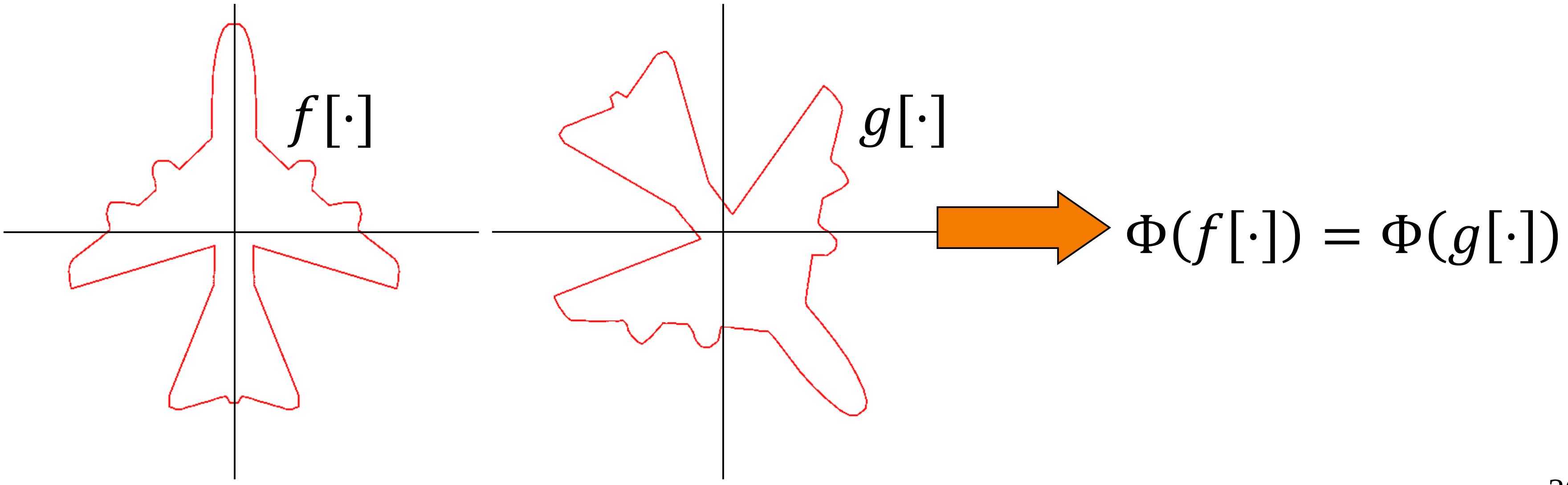
Design a descriptor that is rotation-invariant:

- Instances of the same shape in different poses will give the same shape descriptor.

Invariance

Given an array $f[\cdot]$, we would like to define a mapping Φ taking $f[\cdot]$ to some other array, s.t.:

$$\Phi(f[\cdot]) = \Phi\big(\rho_\alpha(f[\cdot])\big) \quad \forall\alpha$$



Invariance

Given an array $f[\cdot]$, we can express it in terms of its Fourier decomposition:

$$f[\cdot] = \sum_{k=0}^{n-1} \hat{f}[k] \cdot v_k[\cdot]$$

If we rotate $f[\cdot]$ by α we get:

$$\rho_\alpha(f[\cdot]) = \sum_{k=0}^{n-1} \hat{f}[k] \cdot \rho_\alpha(v_k[\cdot])$$

Invariance

$$\rho_\alpha(f[\cdot]) = \sum_{k=0}^{n-1} \hat{f}[k] \cdot \rho_\alpha(v_k[\cdot])$$

Since the $v_k[\cdot]$ are a basis for the one-dimensional irreducible representations:

$$\rho_\alpha(v_k[\cdot]) = \lambda_k(\alpha) \cdot v_k[\cdot]$$

where $\lambda_k(\alpha)$ is a complex number.

Since the representation is unitary, we know that $\lambda_k(\alpha)$ must have unit complex norm.

Invariance

$$\rho_\alpha(f[\cdot]) = \sum_{k=0}^{n-1} \hat{f}[k] \cdot \lambda_k(\alpha) \cdot v_k[\cdot] \quad \text{w/ } \|\lambda_k(\alpha)\| = 1$$

In particular, we have:

$$\left\|\widehat{\rho_\alpha(f[\cdot])}[k]\right\| = \left\|\lambda_k(\alpha) \cdot \hat{f}[k]\right\| = \left\|\hat{f}[k]\right\| \quad \forall k$$

Invariance

$$\left\|\widehat{\rho_\alpha(f[\cdot])}[k]\right\| = \left\|\hat{f}[k]\right\|$$

So, we can get a rotation invariant representation of $f[\cdot]$ by storing only the magnitudes of the Fourier coefficients (i.e. discarding phase):

$$\Phi(f[\cdot]) = \left\{\left\|\hat{f}[0]\right\|, \cdots, \left\|\hat{f}[n-1]\right\|\right\}$$

Invariance

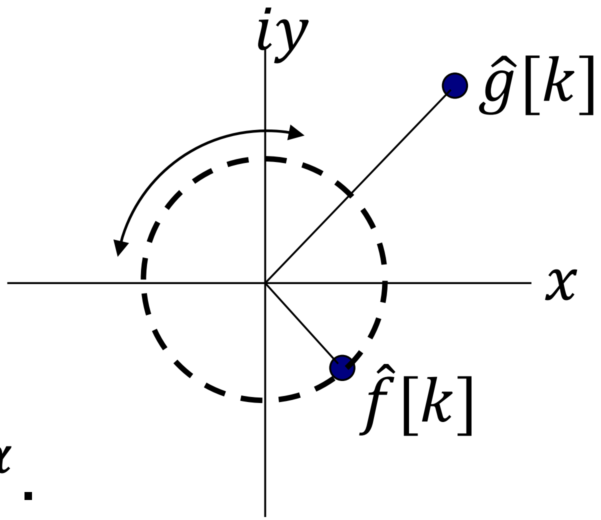
What kind of information do we get when we compare just the amplitudes of the Fourier coefficients?

Invariance

Suppose we are given two arrays $f[\cdot]$ and $g[\cdot]$ with only one non-zero Fourier coefficient:

$$f[\cdot] = \hat{f}[k] \cdot v_k[\cdot]$$
$$g[\cdot] = \hat{g}[k] \cdot v_k[\cdot]$$

what is the measure of similarity at the optimal alignment?



If we rotate $f[\cdot]$ by α, this amounts to multiplying the k-th Fourier coefficient by $e^{-ik\alpha}$.

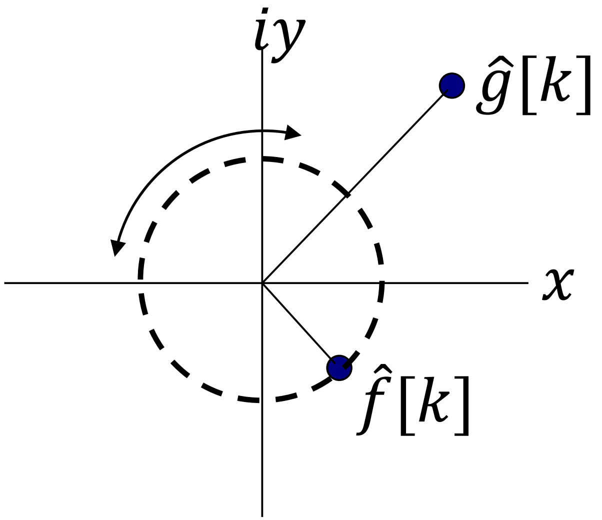
But this is just a rotation in the complex plane.

Invariance

Suppose we are given two arrays $f[\cdot]$ and $g[\cdot]$ with only one non-zero Fourier coefficient:

$$f[\cdot] = \hat{f}[k] \cdot v_k[\cdot]$$
$$g[\cdot] = \hat{g}[k] \cdot v_k[\cdot]$$

what is the measure of similarity at the optimal alignment?



At the optimal rotation, the Fourier coefficients are on the same line and the measure of similarity is the difference between the lengths.

Invariance

Storing only the amplitudes we get a shape representation $\Phi(f[\cdot])$ that:

- Is invariant to rotations
- Provides a measure of similarity that is the distance between $f[\cdot]$ and $g[\cdot]$ if we could optimally align the frequency components <u>independently</u>.

↓

- This is a lower bound for the distance between $f[\cdot]$ and $g[\cdot]$ at the optimal alignment.

Invariance

How good is the lower bound?

After discarding phase, what's left?

Invariance

How good is the lower bound?

After discarding phase, what's left?

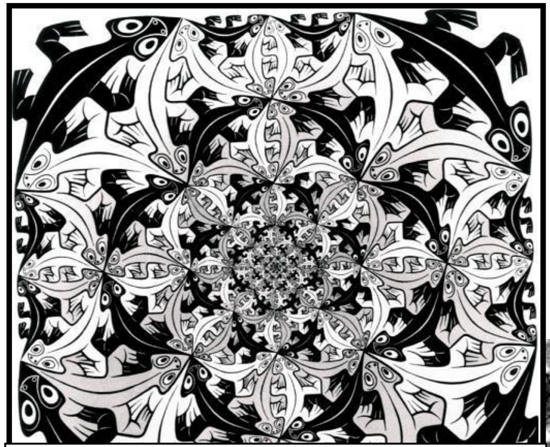
<u>Experiment</u>:

To test this, we can consider what happens when we take two arrays and swap the amplitudes of the Fourier coefficients:

$$f[\cdot] = \sum_{k=0}^{n-1} r_k e^{i\theta_k} v_k[\cdot] \qquad g[\cdot] = \sum_{k=0}^{n-1} s_k e^{i\phi_k} v_k[\cdot]$$

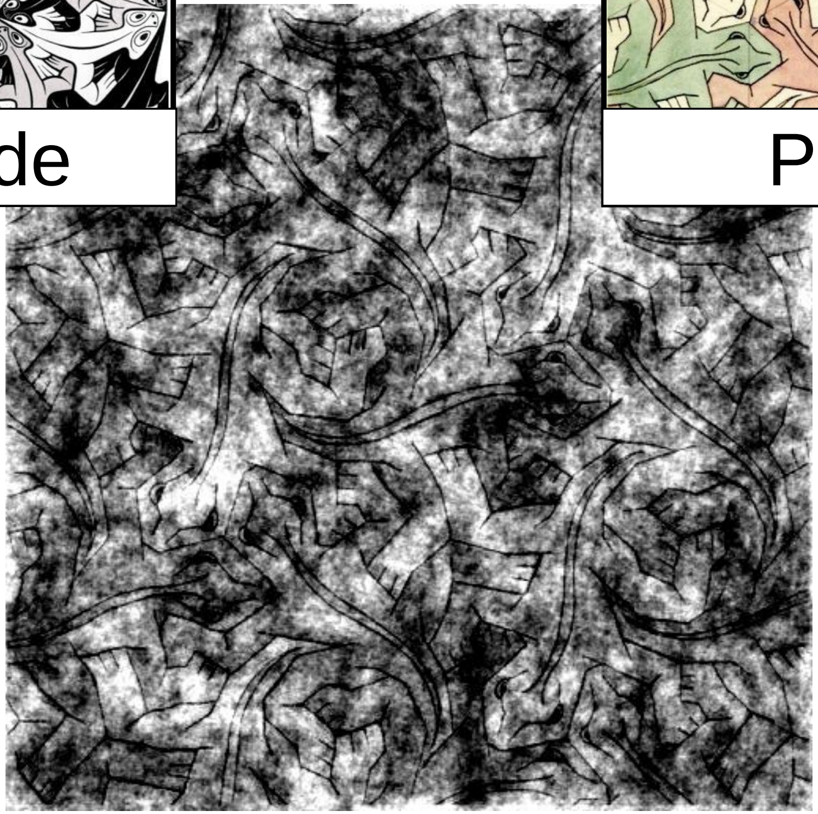
$$\text{ASwap}(f[\cdot], g[\cdot]) = \sum_{k=0}^{n-1} r_k e^{i\phi_k} v_k[\cdot]$$

Invariance



Amplitude

Phase



Invariance

For human perception, dominant information occurs at image boundaries.

These discontinuities arise when the phases are lined up so the occurrence of boundaries is strongly phase dependent.

If the grid encodes other type of information (non-visual) amplitude can be more important.

Outline

Alignment

Shape Matching

Invariance

Pattern Matching

Notation

If $f[\cdot]$ and $g[\cdot]$ are two n-dimensional arrays, then we can define $f[\cdot]\cdot g[\cdot]$ to be the entry-wise product of the two arrays:

$$(f[\cdot]\cdot g[\cdot])[j]\equiv f[j]\cdot g[j]$$

Note 1

If $f[\cdot]$ is a real-valued n-dimensional array, and $g[\cdot]$ and $h[\cdot]$ are complex-valued n-dimensional arrays, then:

$$
\begin{aligned}
\langle f[\cdot]\cdot g[\cdot], h[\cdot]\rangle &= \sum_{k=0}^{n-1} (f[\cdot]\cdot g[\cdot])[k]\cdot \overline{h[k]} \\
&= \sum_{k=0}^{n-1} f[k]\cdot g[k]\cdot \overline{h[k]} \\
&= \sum_{k=0}^{n-1} g[k]\cdot \overline{f[k]}\cdot \overline{h[k]} \\
&= \sum_{k=0}^{n-1} g[k]\cdot \overline{(f[\cdot]\cdot h[\cdot])[k]} \\
&= \langle g[\cdot], f[\cdot]\cdot h[\cdot]\rangle
\end{aligned}
$$

Note 2

If ρ_α is the unitary representation that shifts an array by α indices:

$$\rho_\alpha(f[\cdot])[k] \equiv f[k-\alpha]$$

then:

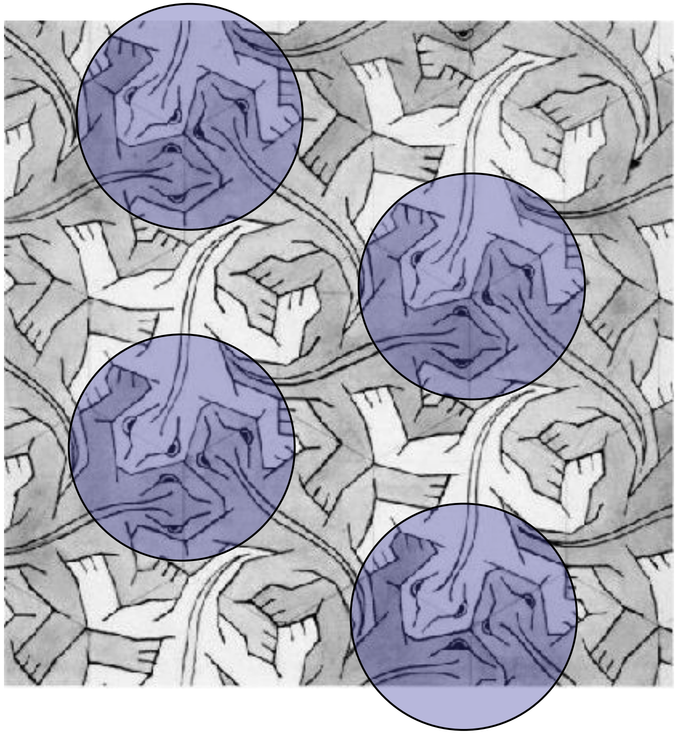
$$\big(\rho_\alpha(f[\cdot]\cdot g[\cdot])\big) = \rho_\alpha(f[\cdot])\cdot\rho_\alpha(g[\cdot])$$

Pattern Matching

Given an instance of a pattern, find all occurrences of the pattern within a target image:

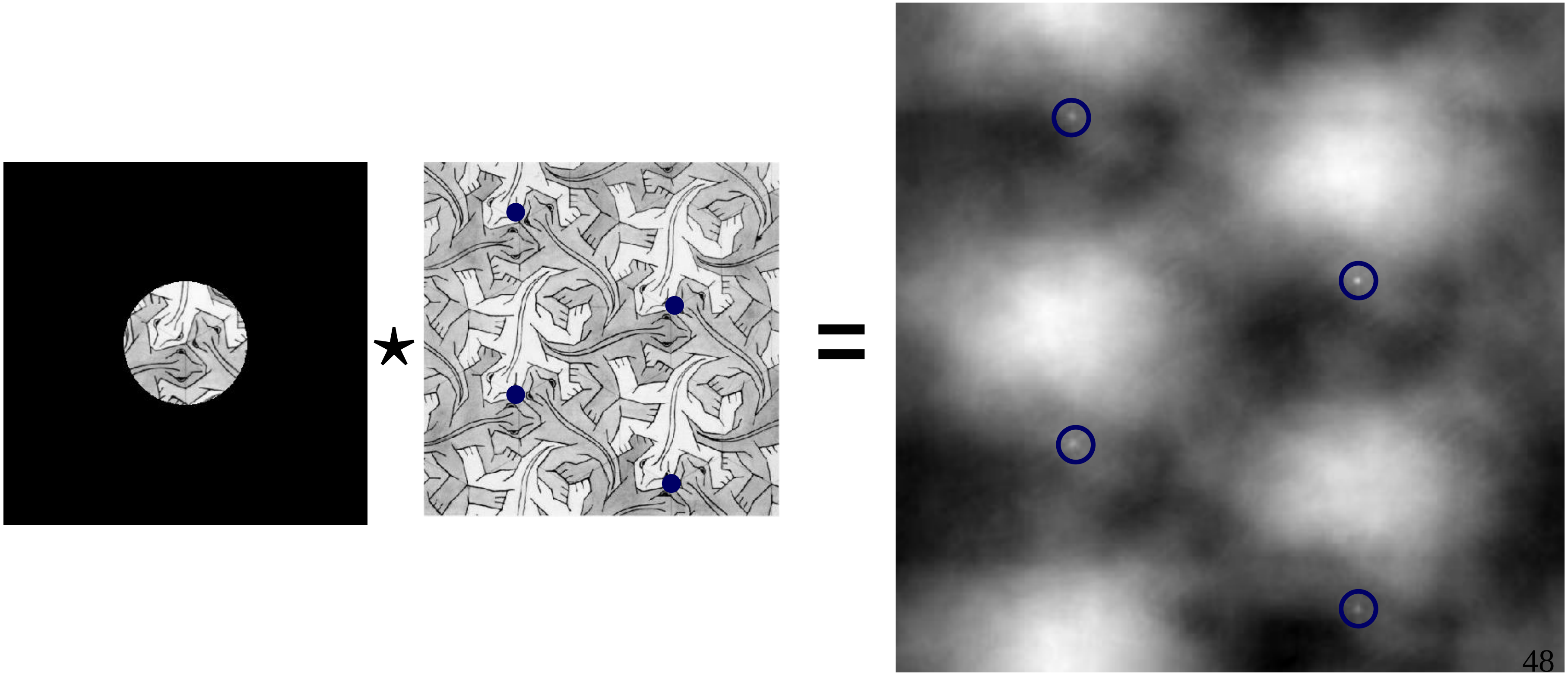
Pattern $f[\cdot][\cdot]$

Target Image $g[\cdot][\cdot]$



Pattern Matching

We could compute the cross-correlation of the pattern with the image and look for local maxima:



Pattern Matching

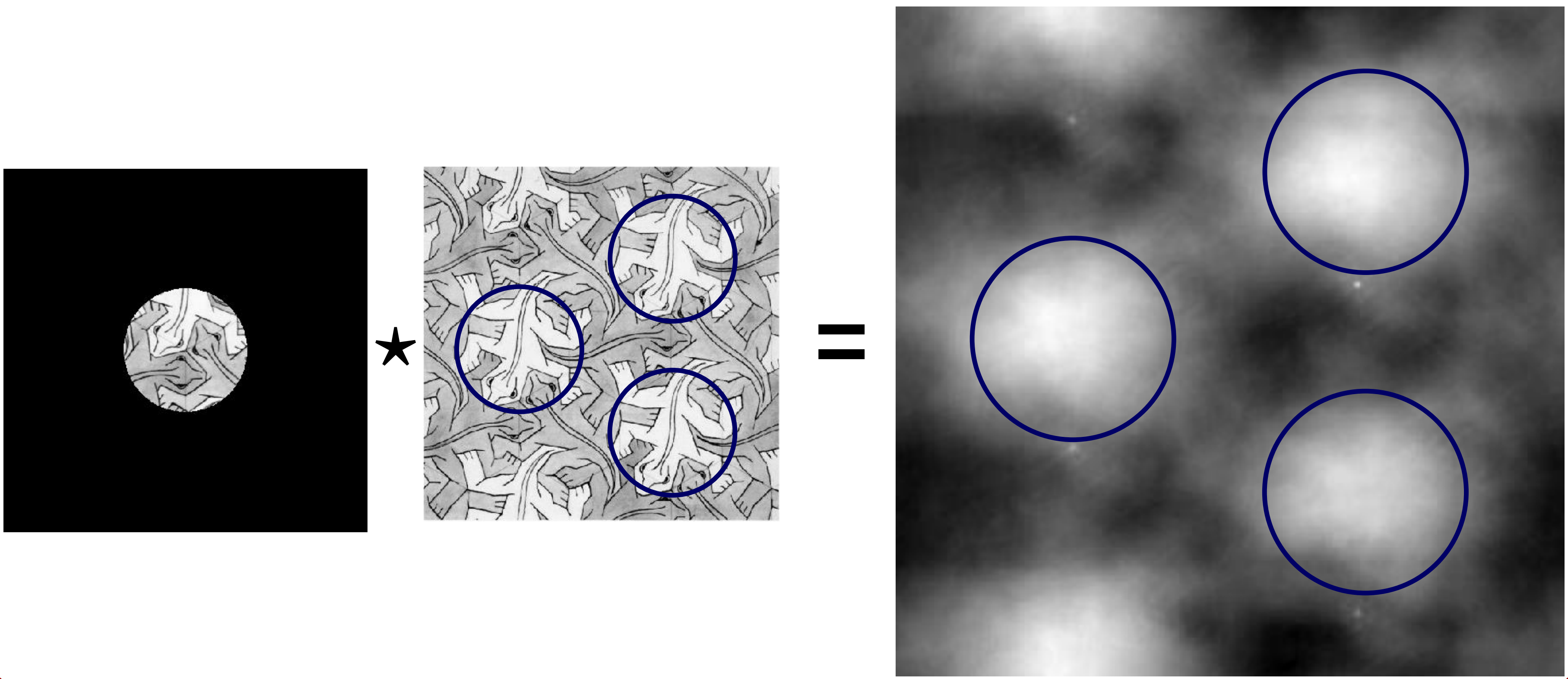
The cross-correlation has large values because the dot product of the image with the translated pattern instance is large.

What causes $\langle f[\cdot], g[\cdot]\rangle$ to be large?

Pattern Matching

What causes $\langle f[\cdot], g[\cdot]\rangle$ to be large?

- If the values of $f[\cdot]$ and $g[\cdot]$ are correlated
- If the values of $g[\cdot]$ are large



Pattern Matching

We don't want to measure:

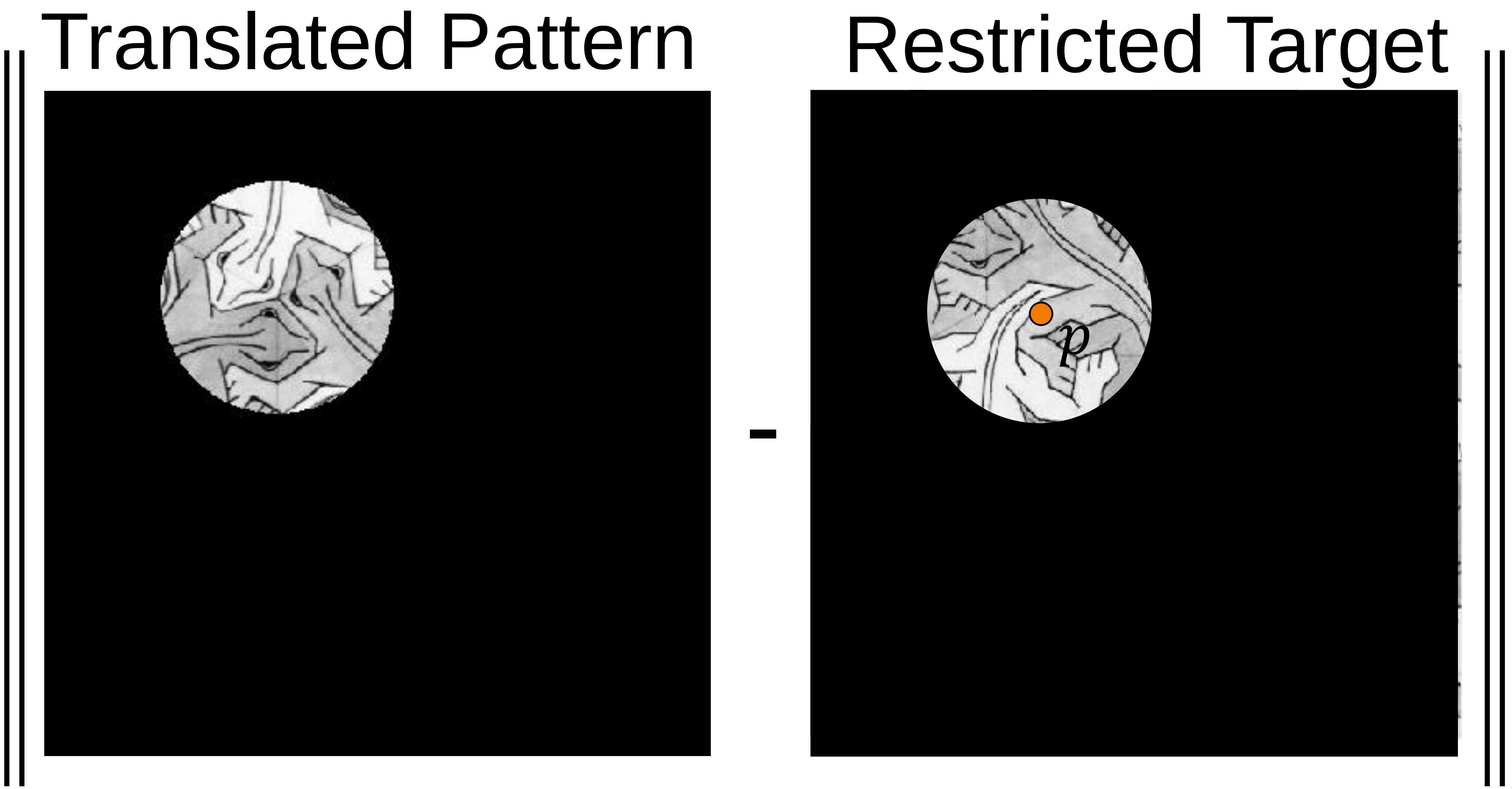
How correlated is the pattern instance with a region in the image?

What we want to measure is:

How similar is the pattern instance with a region in the image?

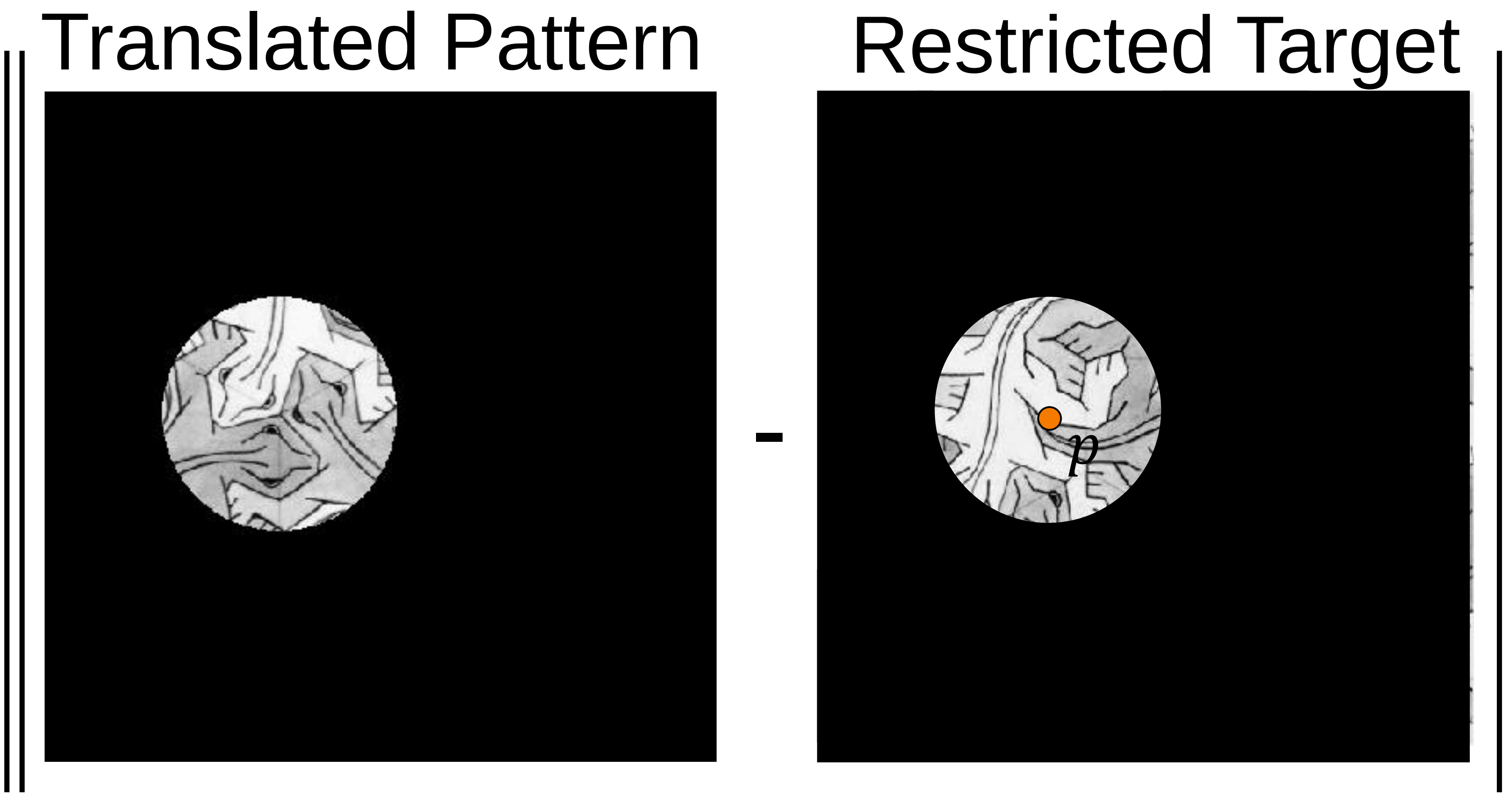
Pattern Matching

For every point in the image, we want to know how similar the region about the point is to the translated pattern.



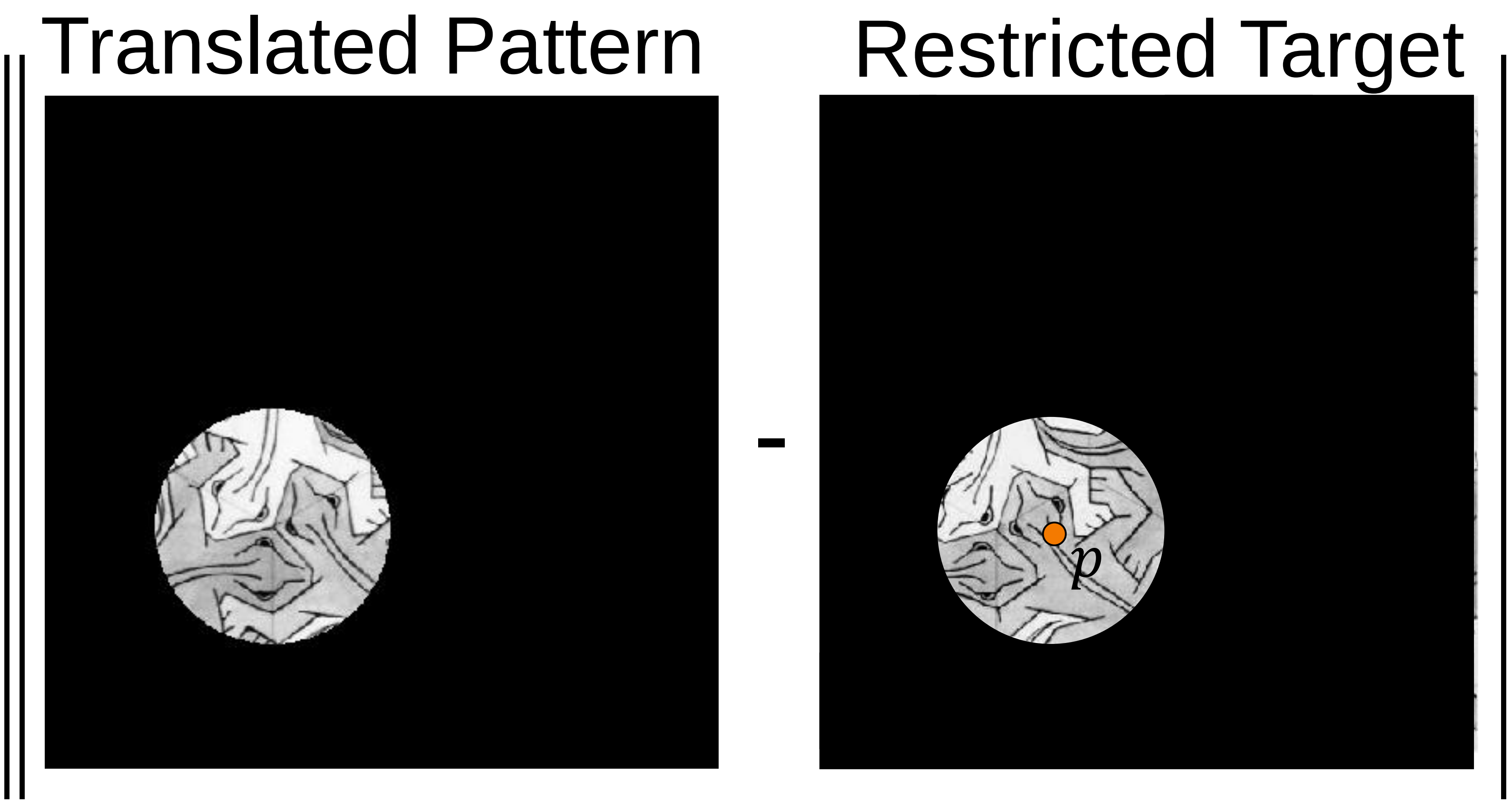
Pattern Matching

For every point in the image, we want to know how similar the region about the point is to the translated pattern.



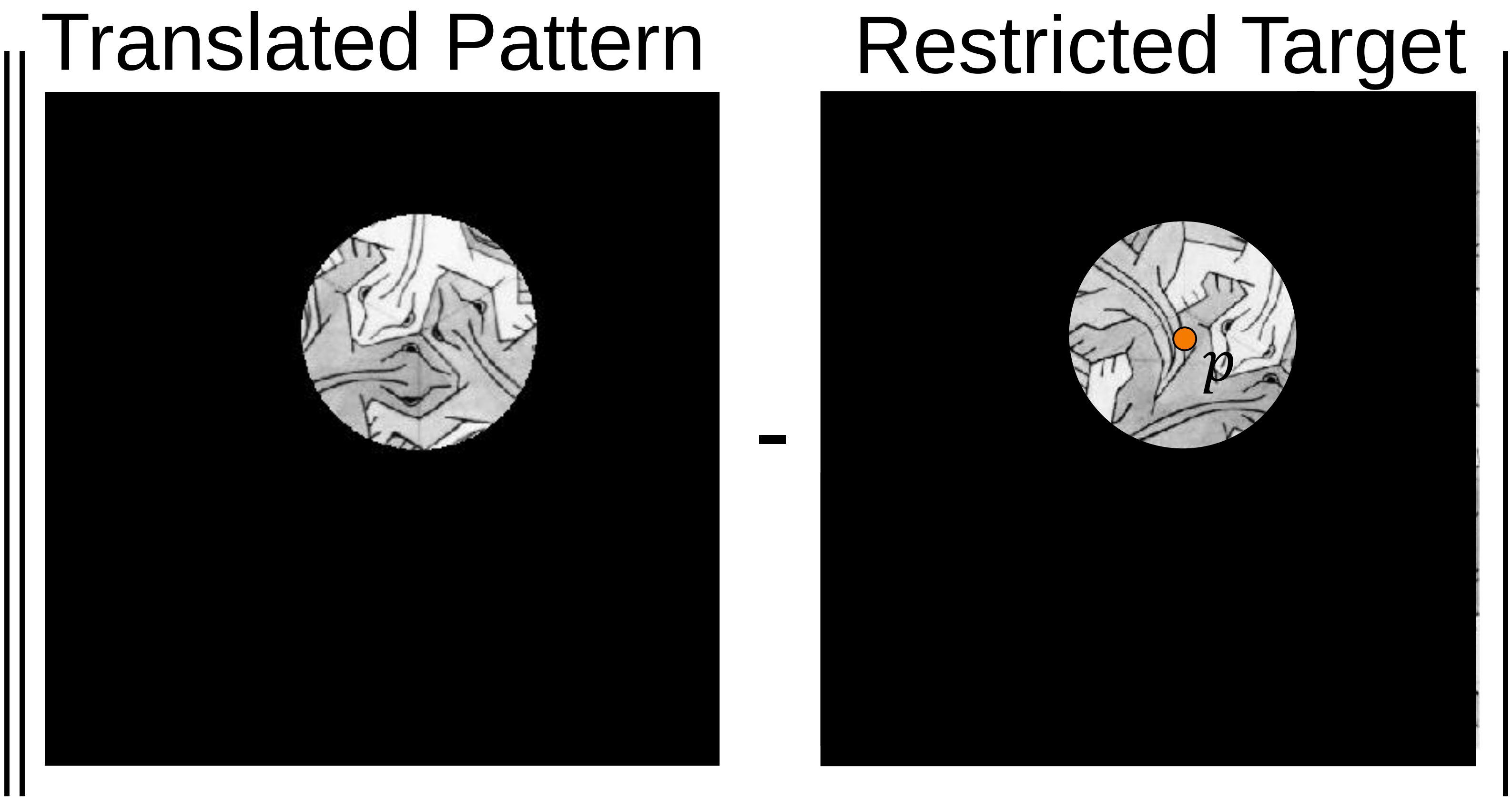
Pattern Matching

For every point in the image, we want to know how similar the region about the point is to the translated pattern.



Pattern Matching

For every point in the image, we want to know how similar the region about the point is to the translated pattern.



Pattern Matching

How do we express this formally?

Pattern Matching

How do we express this formally?

If we represent the pattern by $f[\cdot][\cdot]$, then the translation of $f[\cdot][\cdot]$ to the point p is written as:

$$\rho_p(f[\cdot][\cdot])$$

We want to restrict $g[\cdot][\cdot]$ to the region about p by zeroing out the part of $g[\cdot][\cdot]$ away from p.

Pattern Matching

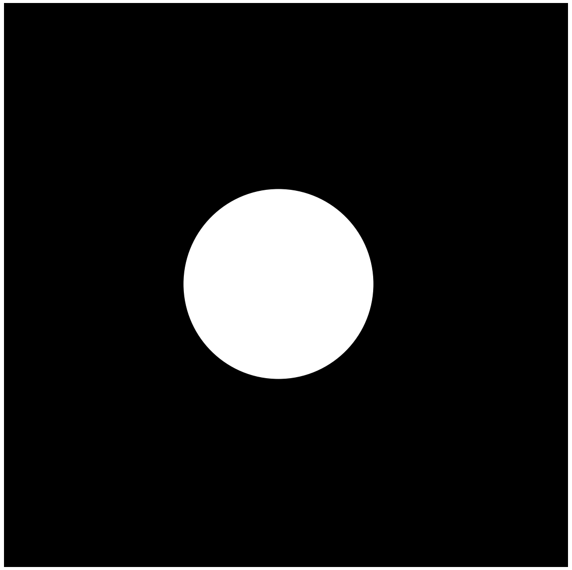
How do we express this formally?

Let $\chi[\cdot][\cdot]$ be the masking grid for the pattern:

$$\chi[j][k] = \begin{cases} 1 & \text{if } f[j][k] \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$f[\cdot][\cdot]$

$\chi[\cdot][\cdot]$



Pattern Matching

How do we express this formally?

The restriction of $g[\cdot][\cdot]$ to the region about p can be expressed as:

$$\rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot]$$

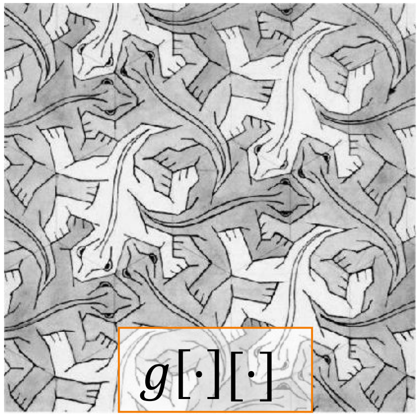
- $\rho_p(\chi[\cdot][\cdot])$ translates the mask so that it's centered on p.
- Multiplying by $\rho_p(\chi[\cdot][\cdot])$ zeros out everything except for the region about p.

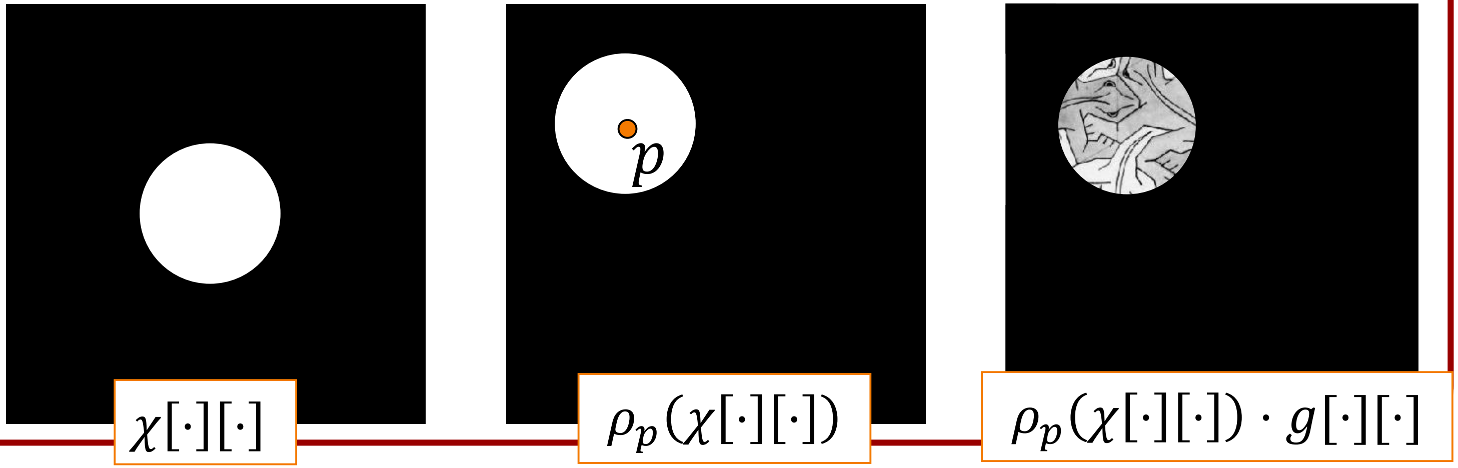
Pattern Matching

How do we express this formally?

The restriction of $g[\cdot][\cdot]$ to the region about p can be expressed as:

$$\rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot]$$



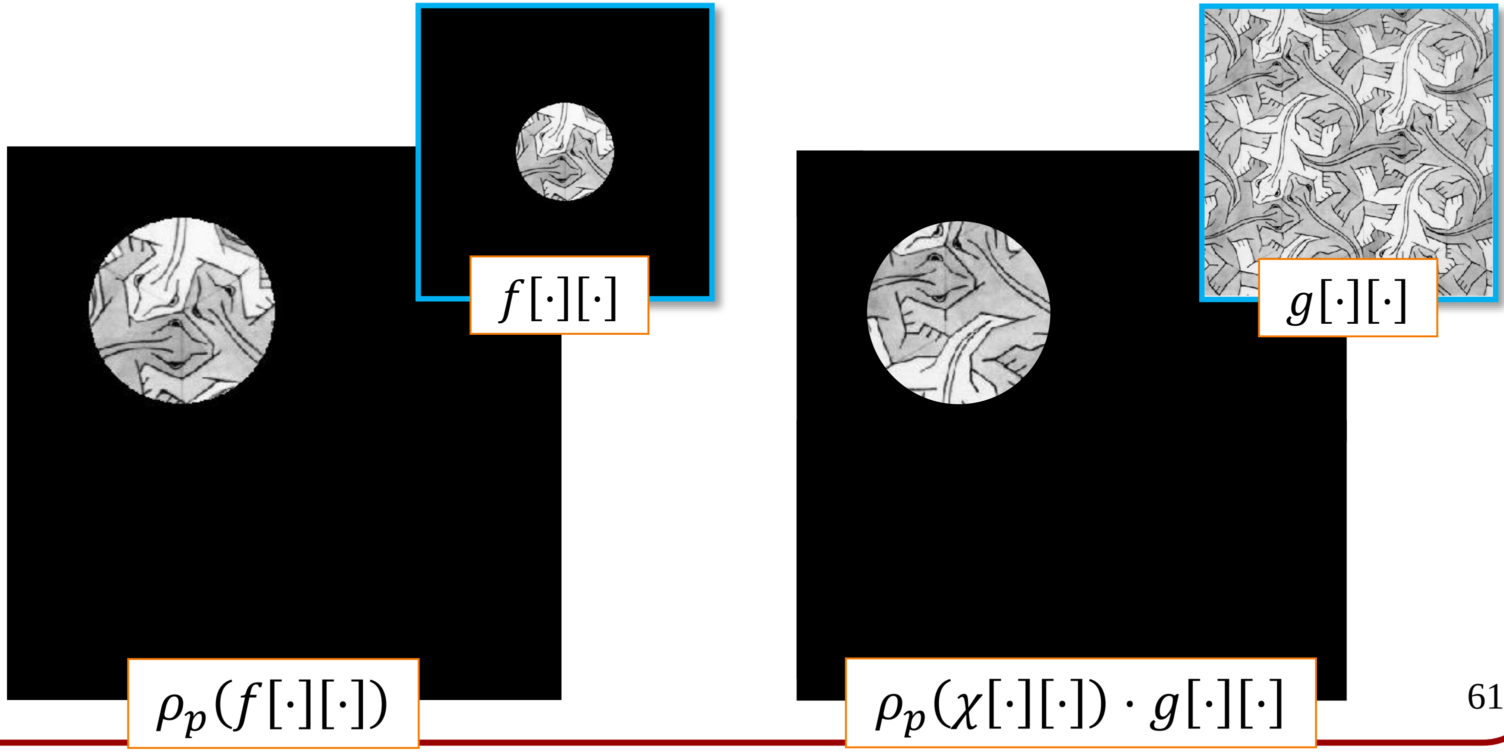


Pattern Matching

How do we express this formally?

For every p, we would like to compute:

$$\left\|\rho_p(f[\cdot][\cdot]) - \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot]\right\|^2$$



Pattern Matching

How do we express this formally?

For every p, we would like to compute:

$$\left\| \rho_p(f[\cdot][\cdot]) - \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \right\|^2$$

Writing this out in terms of dot-products gives three terms:

- $\langle \rho_p(f[\cdot][\cdot]), \rho_p(f[\cdot][\cdot]) \rangle$
- $-2\langle \rho_p(f[\cdot][\cdot]), \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle$
- $\langle \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot], \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle$

Pattern Matching (Term 1)

$$\langle \rho_p(f[\cdot][\cdot]), \rho_p(f[\cdot][\cdot]) \rangle$$

Since the representation is unitary:

$$\langle \rho_p(f[\cdot][\cdot]), \rho_p(f[\cdot][\cdot]) \rangle = \|f[\cdot][\cdot]\|^2$$

Pattern Matching (Term 2)

$$-2\langle \rho_p(f[\cdot][\cdot]), \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle$$

Since $\chi[\cdot][\cdot]$ is real-valued, we can move it to the other side of the dot-product:

$$\langle \rho_p(f[\cdot][\cdot]), \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle = \langle \rho_p(\chi[\cdot][\cdot]) \cdot \rho_p(f[\cdot][\cdot]), g[\cdot][\cdot] \rangle$$

Since the representation commutes with point-wise multiplication we get:

$$\langle \rho_p(f[\cdot][\cdot]), \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle = \langle \rho_p(\chi[\cdot][\cdot] \cdot f[\cdot][\cdot]), g[\cdot][\cdot] \rangle$$

And since $\chi[\cdot][\cdot]$ is equal to one whenever $f[\cdot][\cdot]$ is non-zero we get:

$$\langle \rho_p(f[\cdot][\cdot]), \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle = \langle \rho_p(f[\cdot][\cdot]), g[\cdot][\cdot] \rangle$$

Pattern Matching (Term 3)

$$\langle \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot], \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle$$

Since $\chi[\cdot][\cdot]$ and $g[\cdot][\cdot]$ are real-valued, we can move them to the other sides of the dot-product:

$$\begin{aligned} \langle \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot], \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle &= \left\langle \left(\rho_p(\chi[\cdot][\cdot])\right)^2, g^2[\cdot][\cdot] \right\rangle \\ &= \langle \rho_p(\chi^2[\cdot][\cdot]), g^2[\cdot][\cdot] \rangle \end{aligned}$$

Since $\chi[\cdot][\cdot]$ equals 0 or 1, we have:

$$\chi[\cdot][\cdot] = \chi^2[\cdot][\cdot]$$

So that:

$$\langle \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot], \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot] \rangle = \langle \rho_p(\chi[\cdot][\cdot]), g^2[\cdot][\cdot] \rangle$$

Pattern Matching

Combining all of this together, we get:

$$\left\|\rho_p(f[\cdot][\cdot]) - \rho_p(\chi[\cdot][\cdot]) \cdot g[\cdot][\cdot]\right\|^2 = \|f[\cdot][\cdot]\|^2 + \langle \rho_p(\chi[\cdot][\cdot]), g^2[\cdot][\cdot] \rangle - 2\langle \rho_p(f[\cdot][\cdot]), g[\cdot][\cdot] \rangle$$

Or somewhat more cleanly:

$$\|f[\cdot][\cdot]\|^2 + \boxed{\chi[\cdot][\cdot] \star g^2[\cdot][\cdot]} - 2\boxed{f[\cdot][\cdot] \star g[\cdot][\cdot]}$$

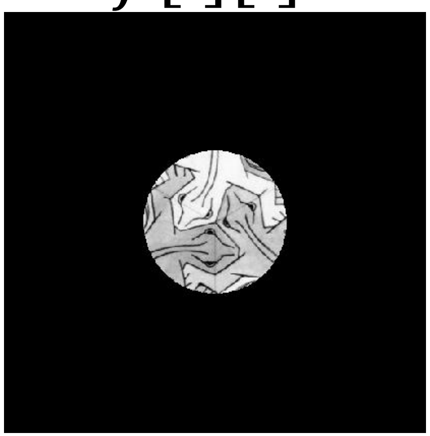
The windowed norm

The moving dot-product

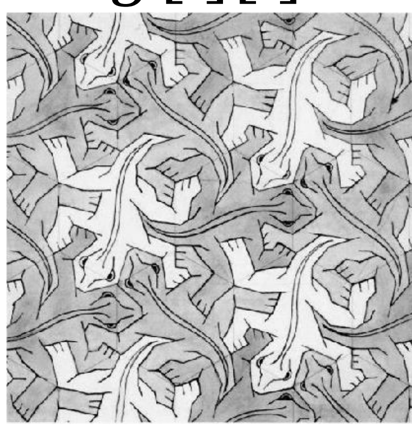
Pattern Matching (Term 1)

$$\|f[\cdot][\cdot]\|^2 + \chi[\cdot][\cdot] \star g^2[\cdot][\cdot] - 2f[\cdot][\cdot] \star g[\cdot][\cdot]$$

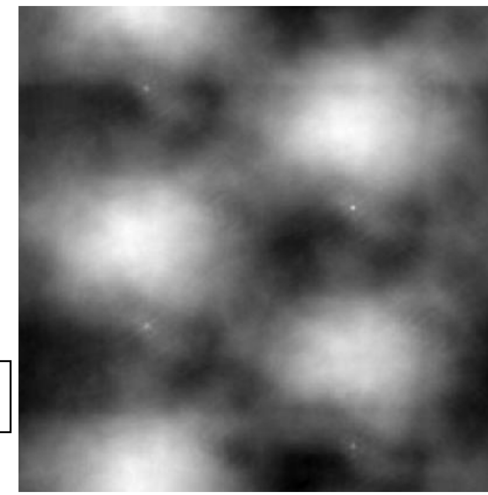
$f[\cdot][\cdot]$



$g[\cdot][\cdot]$



$f[\cdot][\cdot] \star g[\cdot][\cdot]$



$\|f[\cdot][\cdot]\|^2 + \cdots$