UNIVERSITY OF ALBERTA

EDMONTON·ALBERTA·CANADA

CMPUT 307: 2D TEMPLATE MATCHING

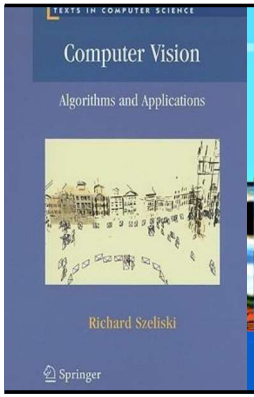
Instructor: Dr. Anup Basu

Notes by: Mehdi Faraji | Winter 2020

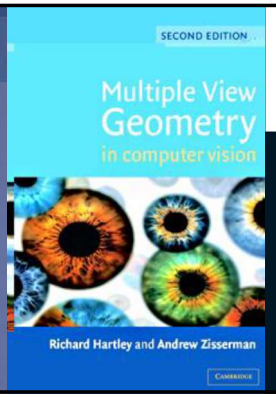
Department of Computing Science

References

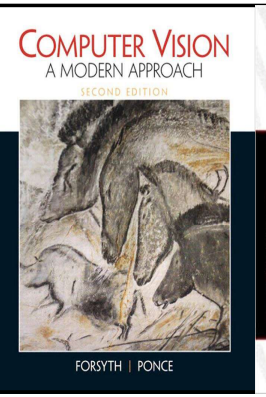
UNLESS OTHERWISE STATED, MATERIALS ON THIS SLIDE ARE TAKEN FROM THE FOLLOWING TEXTBOOKS



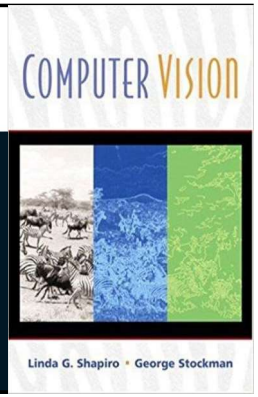
(Szeliski, 2010)



(Hartley & Zisserman, 2003)



(Forsyth & Ponce, 2011)



(Shapiro & Stockman, 2001)

Template Matching

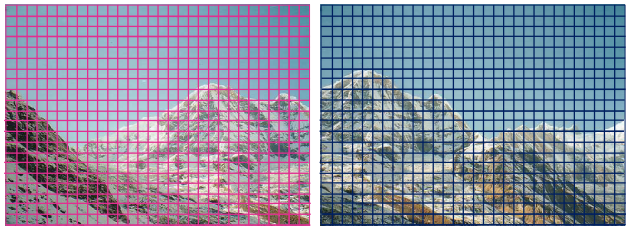
- Sometimes we need to match (align) images
- How can you match these images?

  Assuming that both images are available

Naïve Way:

1. Compare all pixels' intensity of the first image with all intensities of the second image

Will it work?



Template Matching

Naïve Way:

1. Compare all pixels' intensity of the first image with all intensities of the second image

Will it work?

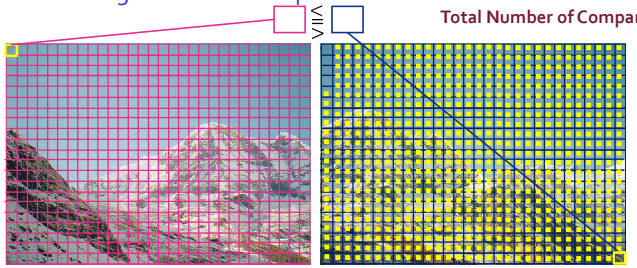
Let's see…

1. Assume that it is a good idea → is it practical?

- If left image contains N_1 pixels and the right image consists of N_2 pixels:

  For each pixel of the first image we need N_2 comparisons

Total Number of Comparisons= $N_1 N_2$



Template Matching

UNIVERSITY OF ALBERTA

Naïve Way:

1. Compare all pixels' intensity of the first image with all intensities of the second image

Will it work?

Let's see…

2. Let's figure out if we can correctly match two images.

For simplicity, we select a small patch of both images

Notice that two patches are visually very similar

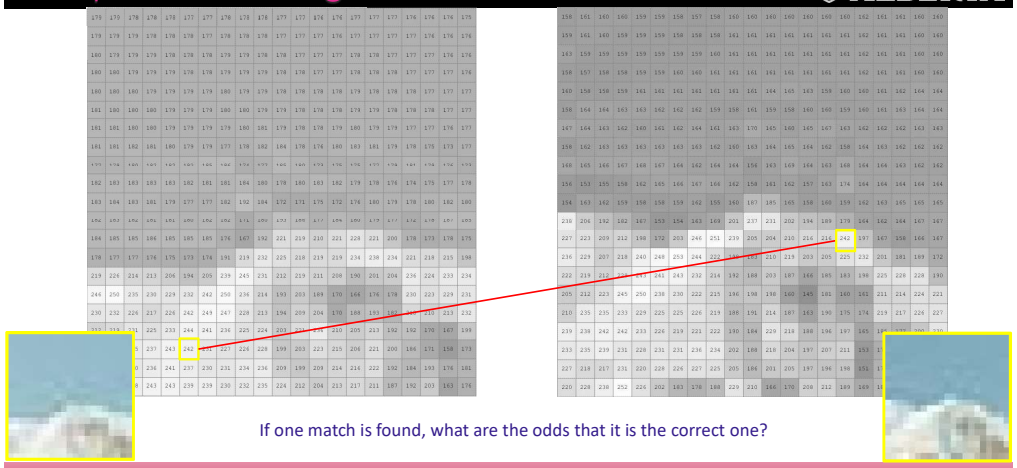
```
I1 = imread('Images\mountain1.png');
I2 = imread('Images\mountain2.png');

rad = 20;
croppedI1 = I1(123-rad:123+rad,325-rad:325+rad,:);
croppedI2 = I2(103-rad:103+rad,75-rad:75+rad,:);

imtool(rgb2gray(croppedI1));
imtool(rgb2gray(croppedI2));
```

Template Matching

UNIVERSITY OF ALBERTA



If one match is found, what are the odds that it is the correct one?

Template Matching

UNIVERSITY OF ALBERTA

Naïve Way:

1. Compare all pixels' intensity of the first image with all intensities of the second image

Will it work?

Let's see…

No, It won't work! Bad Idea!

2. Let's divide the images into blocks and compare each block to the blocks of the other image.

- This is a sample of exhaustive search for a specific block of the first image over all blocks of the second image
- For doing this, we need a strategy to compare blocks
- This approach sometimes is called Template Matching or Window Matching

Template Matching

UNIVERSITY OF ALBERTA

Block Comparisons

How can we find out which two blocks of pixels are similar?

Block (patch) Similarity Measures (Metrics)

To calculate a numerical value that indicates in what degree two blocks are similar, we can use:

1. Pixel-based distance

The simplest way is to directly calculate the distance between corresponding pixels of each block by subtracting them from each other. Assuming that both blocks have the same width (w) and height (h):

$$D(B_1, B_2) = \sum_{i=1}^{w}\sum_{j=1}^{h} |B_1(i,j) - B_2(i,j)|$$

Or Sum of Squared Differences

$$SSD(B_1, B_2) = \sum_{i=1}^{w}\sum_{j=1}^{h} (B_1(i,j) - B_2(i,j))^2$$

Having lower SSD score means higher similarity.

Block Comparisons

How can we find out which two blocks of pixels are similar?

Block (patch) Similarity Measures

To calculate a numerical value that indicates in what degree two blocks are similar, we can use:

2. Cross Correlation

Calculate the Correlation score as follows:

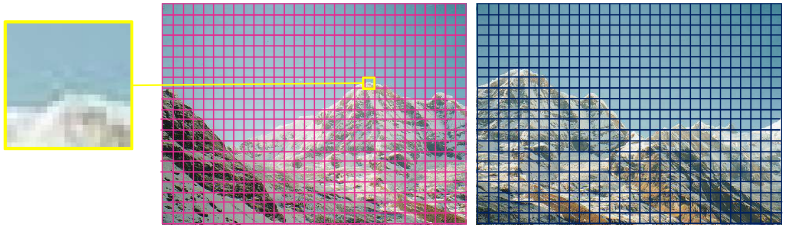
$$C(B_1, B_2) = \sum_{i=1}^{w} \sum_{j=1}^{h} B_1(i,j) B_2(i,j)$$

Having higher correlation score means higher similarity.

Block Comparisons

We search for the selected block from the left image in the right image using two similarity measures, namely the Pixel-based Distance and Correlation.

Let's code!



Block Comparisons

Let's code!

```
clc;
close all;
I1rgb = imread('Images\mountain1.png');
I2rgb = imread('Images\mountain2.png');

I1 = im2double(rgb2gray(I1rgb));
I2 = im2double(rgb2gray(I2rgb));

% rad is 10 so the width of the block is 2*10+1=21
rad = 10;
w = 2*rad+1; % here width = height
% Reading a block from the reference image
B1x = 325;
B1y = 123;
B1 = I1(B1y-rad:B1y+rad,B1x-rad:B1x+rad);
% Padding zeros to I2
I2 = padarray(I2,[rad,rad]);
[R2,C2] = size(I2);
C_Scores = zeros(R2,C2);
SSD_Scores = zeros(R2,C2);
```

Reading Images and converting to grayscale double matrices

Block Comparisons

Let's code!

```
clc;
close all;
I1rgb = imread('Images\mountain1.png');
I2rgb = imread('Images\mountain2.png');

I1 = im2double(rgb2gray(I1rgb));
I2 = im2double(rgb2gray(I2rgb));

% rad is 10 so the width of the block is 2*10+1=21
rad = 10;
w = 2*rad+1; % here width = height
% Reading a block from the reference image
B1x = 325;
B1y = 123;
B1 = I1(B1y-rad:B1y+rad,B1x-rad:B1x+rad);
% Padding zeros to I2
I2 = padarray(I2,[rad,rad]);
[R2,C2] = size(I2);
C_Scores = zeros(R2,C2);
SSD_Scores = zeros(R2,C2);
```

Selecting the a block from the reference image.
We know where the block is.
rad is the width of the block.

Template Matching

UNIVERSITY OF ALBERTA

Block Comparisons

Let's code!

```
clc;
close all;
I1rgb = imread('Images\mountain1.png');
I2rgb = imread('Images\mountain2.png');

I1 = im2double(rgb2gray(I1rgb));
I2 = im2double(rgb2gray(I2rgb));

% rad is 10 so the width of the block is 2*10+1=21
rad = 10;
w = 2*rad+1; % here width = height
% Reading a block from the reference image
B1x = 325;
B1y = 123;
B1 = I1(B1y-rad:B1y+rad,B1x-rad:B1x+rad);
% Padding zeros to I2
I2 = padarray(I2,[rad,rad]);
[R2,C2] = size(I2);
C_Scores = zeros(R2,C2);
SSD_Scores = zeros(R2,C2);
```

14	10	32	25
8	16	0	9
7	6	2	55
88	54	25	32

Image

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	14	10	32	25	0	0
0	0	8	16	0	9	0	0
0	0	7	6	2	55	0	0
0	0	88	54	25	32	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

After padding with 2 rows and 2 columns

To move a block on the second image, we need to pad the image with zeros.
Initialize two matrices for scores.

Template Matching

UNIVERSITY OF ALBERTA

Block Comparisons

Let's code!

```
% Searching for B1 in I2
for x = rad +1:C2-rad
    for y = rad+1:R2-rad
        B2 = I2(y-rad:y+rad,x-rad:x+rad);
        SSD_Scores(y,x) = SSDScore(B1,B2);
        C_Scores(y,x) = CorrelationScore(B1,B2);
    end
end
% Converting scores to the original coords
SSD_Scores = SSD_Scores(rad+1:R2-rad,rad+1:C2-rad);
C_Scores = C_Scores(rad+1:R2-rad,rad+1:C2-rad);
I2 = I2(rad+1:R2-rad,rad+1:C2-rad);
[R2,C2] = size(I2);
% Finding the maximum score
[SSDmin,SSDminInd] = min(SSD_Scores(:));
[Cmx,CmxInd] = max(C_Scores(:));
[ySSD,xSSD] = ind2sub([R2,C2],SSDminInd);
[yC,xC] = ind2sub([R2,C2],CmxInd);
```

```
function SSD = SSDScore(B1,B2)
   SSD = sum((B1(:)-B2(:)).^2);
end
function C = CorrelationScore(B1,B2)
   C = sum(B1(:).*B2(:));
end
```

The main loop of the code that goes over all blocks of the second image and calculates the scores based on the SSD and Correlation metrics.

Template Matching

UNIVERSITY OF ALBERTA

Block Comparisons

Let's code!

```
% Searching for B1 in I2
for x = rad +1:C2-rad
    for y = rad+1:R2-rad
        B2 = I2(y-rad:y+rad,x-rad:x+rad);
        SSD_Scores(y,x) = SSDScore(B1,B2);
        C_Scores(y,x) = CorrelationScore(B1,B2);
    end
end
% Converting scores to the original coords
SSD_Scores = SSD_Scores(rad+1:R2-rad,rad+1:C2-rad);
C_Scores = C_Scores(rad+1:R2-rad,rad+1:C2-rad);
I2 = I2(rad+1:R2-rad,rad+1:C2-rad);
[R2,C2] = size(I2);
% Finding the maximum score
[SSDmin,SSDminInd] = min(SSD_Scores(:));
[Cmx,CmxInd] = max(C_Scores(:));
[ySSD,xSSD] = ind2sub([R2,C2],SSDminInd);
[yC,xC] = ind2sub([R2,C2],CmxInd);
```

```
function SSD = SSDScore(B1,B2)
   SSD = sum((B1(:)-B2(:)).^2);
end
function C = CorrelationScore(B1,B2)
   C = sum(B1(:).*B2(:));
end
```

Converting the score matrices back to the original dimensions.

Template Matching

UNIVERSITY OF ALBERTA

Block Comparisons

Let's code!

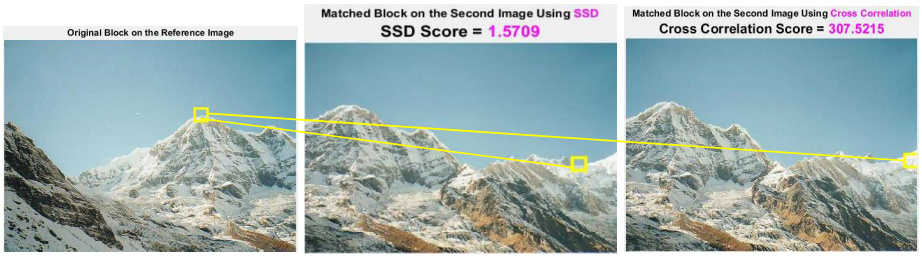
```
% Searching for B1 in I2
for x = rad +1:C2-rad
    for y = rad+1:R2-rad
        B2 = I2(y-rad:y+rad,x-rad:x+rad);
        SSD_Scores(y,x) = SSDScore(B1,B2);
        C_Scores(y,x) = CorrelationScore(B1,B2);
    end
end
% Converting scores to the original coords
SSD_Scores = SSD_Scores(rad+1:R2-rad,rad+1:C2-rad);
C_Scores = C_Scores(rad+1:R2-rad,rad+1:C2-rad);
I2 = I2(rad+1:R2-rad,rad+1:C2-rad);
[R2,C2] = size(I2);
% Finding the maximum score
[SSDmin,SSDminInd] = min(SSD_Scores(:));
[Cmx,CmxInd] = max(C_Scores(:));
[ySSD,xSSD] = ind2sub([R2,C2],SSDminInd);
[yC,xC] = ind2sub([R2,C2],CmxInd);
```

```
function SSD = SSDScore(B1,B2)
   SSD = sum((B1(:)-B2(:)).^2);
end
function C = CorrelationScore(B1,B2)
   C = sum(B1(:).*B2(:));
end
```

Finding the most similar block by finding the maximum correlation score and minimum SSD score.

Block Comparisons

Let's run it!



Both measures failed! ☹

Block Comparisons

Why did it fail?

1. Even if the images have been taken by the same model of camera, the gain and sensitivity still varies.
2. Overall radiance of surfaces varies in time due to air molecules, wind, clouds, … that creates images with different illumination level.
3. Noise plays a crucial role.

MATH REVIEW: *Statistics – Standard Score*

- Normalization in Statistics conveys a range of meanings.
- Usually, we use normalization when we want to adjust vectors of values that are measured on different scales to a common scale.
- *Standard Score or z-score is a way of normalization when the populations are available.*
- *Z-score is a dimensionless quantity.*
- *We need to know the mean and standard deviation of the complete population.*
- For a raw vector x we can calculate the *z-score* as follows:

$$z = \frac{\mathbf{x} - \mu}{\sigma}$$

- $\mu = \frac{1}{n}(\sum_{i=1}^{n} x(i))$ is the mean of the population
- $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x(i) - \mu)^2}$ is the standard deviation of the population that quantifies the amount of variation of elements of the vector x

Block Comparisons

What is the solution?

Intensity Normalization

If B is a block of image (matrix of intensities), we calculate the z-scores for each pixels as:

$$\mathrm{z}(x, y) = \frac{\mathbf{B}(x,y) - \mu}{\sigma}$$

The normalized scores are then calculated as:

Normalized Sum of Squared Differences (NSSD):

$$NSSD(\mathbf{B}_1, \mathbf{B}_2) = \frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h}(\mathrm{z}_1(i,j) - \mathrm{z}_2(i,j))^2$$

Normalized Cross Correlation (NCC):

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h}\mathrm{z}_1(i,j)\ \mathrm{z}_2(i,j)$$

Block Comparisons

- For simplicity of writing equations, we convert the matrices to vector, so we need only one summation notation.
- In Mathematics specially in Linear Algebra, this process is called Vectorization which converts a matrix into a column vector.

$$\begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix} = (a \quad b \quad c \quad d \quad e \quad f \quad g \quad h \quad i)^T$$

Therefore, we can use only one $\sum$ in our equations to traverse all the element of the matrix.

$$\sum_{i=1}^{w}\sum_{j=1}^{h}\mathbf{B}(i,j) \rightarrow \sum_{i}\mathbf{B}(i)$$

Block Comparisons

Normalized Sum of Squared Differences (NSSD):

$$NSSD(\mathbf{B}_1,\mathbf{B}_2) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sigma_1} - \frac{\mathbf{B}_2(i)-\mu_2}{\sigma_2}\right)^2$$

Bring n inside the parenthesis:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sqrt{n}\sigma_1} - \frac{\mathbf{B}_2(i)-\mu_2}{\sqrt{n}\sigma_2}\right)^2$$

Insert the standard deviation equation:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sqrt{n}\sqrt{\frac{1}{n}\sum_{j=1}^{n}(\mathbf{B}_1(j)-\mu_1)^2}} - \frac{\mathbf{B}_2(i)-\mu_2}{\sqrt{n}\sqrt{\frac{1}{n}\sum_{j=1}^{n}(\mathbf{B}_2(j)-\mu_2)^2}}\right)^2$$

ns in the denominators are cancelled out.

Block Comparisons

Normalized Sum of Squared Differences (NSSD):

$$NSSD(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sqrt{\sum_{j=1}^{n}(\mathbf{B}_1(j)-\mu_1)^2}} - \frac{\mathbf{B}_2(i)-\mu_2}{\sqrt{\sum_{j=1}^{n}(\mathbf{B}_2(j)-\mu_2)^2}}\right)^2$$

Substituting $\overline{\mathbf{B}} = \mathbf{B} - \mu$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_1(j)^2}} - \frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_2(j)^2}}\right)^2$$

Block Comparisons

Normalized Cross Correlation (NCC):

$$NCC(\mathbf{B}_1,\mathbf{B}_2) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sigma_1}\right)\left(\frac{\mathbf{B}_2(i)-\mu_2}{\sigma_2}\right)$$

Bring n inside the parenthesis:

$$NCC(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sqrt{n}\sigma_1}\right)\left(\frac{\mathbf{B}_2(i)-\mu_2}{\sqrt{n}\sigma_2}\right)$$

Insert the standard deviation equation:

$$NCC(\mathbf{B}_1,\mathbf{B}_2) = \sum_{i=1}^{n}\left(\frac{\mathbf{B}_1(i)-\mu_1}{\sqrt{n}\sqrt{\frac{1}{n}\sum_{j=1}^{n}(\mathbf{B}_1(j)-\mu_1)^2}}\right)\left(\frac{\mathbf{B}_2(i)-\mu_2}{\sqrt{n}\sqrt{\frac{1}{n}\sum_{j=1}^{n}(\mathbf{B}_2(j)-\mu_2)^2}}\right)$$

ns in the denominators are cancelled out.

Block Comparisons

Normalized Cross Correlation (NCC):

$$NCC(\mathbf{B_1}, \mathbf{B_2}) = \sum_{i=1}^{n} \left(\frac{\mathbf{B}_1(i) - \mu_1}{\sqrt{\sum_{j=1}^{n} (\mathbf{B}_1(j) - \mu_1)^2}} \right) \left(\frac{\mathbf{B}_2(i) - \mu_2}{\sqrt{\sum_{j=1}^{n} (\mathbf{B}_2(j) - \mu_2)^2}} \right)$$

Substituting $\overline{\mathbf{B}} = \mathbf{B} - \mu$

$$NCC(\mathbf{B_1}, \mathbf{B_2}) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_1(j)^2}} \right) \left(\frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_2(j)^2}} \right)$$

MATH REVIEW: *Linear Algebra*

Vector

Vector is often used in Geometry and Physics to represent various quantities, such as force, speed, movement, acceleration and etc.

A vector has a:

- Length
- Direction

We show a vector by a **lower case boldfaced** character.

Examples.

$\mathbf{v_1} = \begin{pmatrix} 2 \\ -3 \end{pmatrix}$ is a vector in $\mathfrak{R}^2$

$\mathbf{v_2} = \begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix}$ is a vector in $\mathfrak{R}^3$

MATH REVIEW: *Linear Algebra*

Vector

$\mathbf{v_n} = \begin{pmatrix} 2 \\ \vdots \\ -4 \end{pmatrix}_{1 \times n}$ is a column vector in $\mathfrak{R}^n$

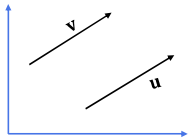
Transpose of a Vector

A column vector can be expressed as a row vector by using the transpose:

$$\mathbf{v} = \begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix} \rightarrow \mathbf{v}^T = (2 \quad 0 \quad -4)$$

MATH REVIEW: *Linear Algebra* from: (Hughes & Foley, 2014)

Vector – How to think about vectors



- Usually students see vectors as arrows.
- What if we ask about these two vectors? Are these vectors equal?
- Although these arrows are in different places and are clearly different, in terms of vector representation, they are representing the same vector.
- Therefore, to have a better understanding about vectors, it is better to see them as a **displacement:**

  It represents an amount by which you must move to get from one place to another.

For example, to get from the point $(3,1)$ to point $(5,0)$ you must move by 2 in x direction and by -1 in y direction. So, this displacement is represented by vector $[2 \quad -1]^T$

- Now, we can see that both vectors depicted in the figure are representing the same displacement, so they are equivalent.

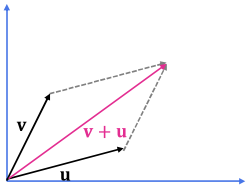
Vector Addition

$$\mathbf{v} + \mathbf{u} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 + u_1 \\ v_2 + u_2 \end{pmatrix}$$

Geometric representation

If the vectors are not attached:

- Note that for vectors only length and direction properties are important.
- Therefore, we can shift vectors in a way that both start from the same location → Origin



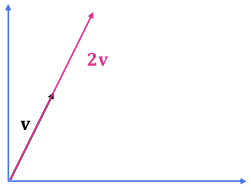
- Using the so-called parallelogram law we obtain the sum of two vectors.

Scalar multiplication of Vectors

Vectors can be multiplied by a number:

$$s\mathbf{v} = s\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} sv_1 \\ sv_2 \end{pmatrix}$$

Geometric representation

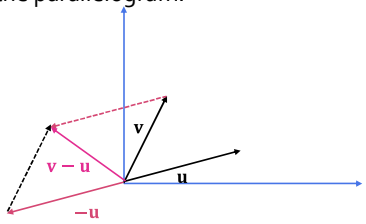


Vector Subtraction

$$\mathbf{v} - \mathbf{u} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} - \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 - u_1 \\ v_2 - u_2 \end{pmatrix}$$

Geometric representation

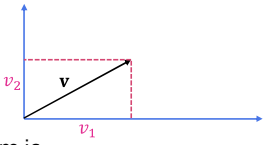
We can add $(-\mathbf{u})$ to $\mathbf{v}$ using the parallelogram:



Length of a Vector

The length of the vector $\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ is calculated by the Pythagorean Theorem:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2}$$



- The length of the vector is also called the *norm* of the vector.
- It is denoted by double absolute values.
- If the vector has three components (vectors in 3D space) the norm is:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

- For n dimensional vectors:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

Length of a Vector

By definition, the norm satisfies the following conditions:

1. If $\mathbf{v} \neq 0$, $\|\mathbf{0}\| = 0 \rightarrow \|\mathbf{v}\| > 0$
2. For all scalar c and vectors $\mathbf{v}$, $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

Length of a Vector - Recap

- There are numerous norms that are used in practice.
- In our work, the norm most often used is the so-called 2-norm, which, for a vector $\mathbf{v}$ in real $\mathfrak{R}^n$, space is defined as:

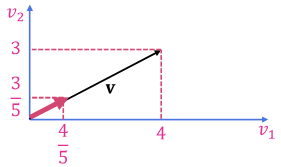
$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Which is recognized as the Euclidean distance from the origin to point $\mathbf{v}$
- This gives the expression the familiar name Euclidean norm.
- The expression also is recognized as the length of a vector $\mathbf{v}$, with origin at point O.
- The norm also can be written as

$$\|\mathbf{v}\| = (\mathbf{v}^T\mathbf{v})^{\frac{1}{2}}$$

Unit Vectors

- Unit vectors are vectors that their length is 1.
- We denote the unit vector by a boldfaced lowercase letter with a hat $\hat{\mathbf{v}}$ (pronounced v hat)
- All non-zero vector can be decomposed into unit vectors
- For example, if we have a vector $\mathbf{v} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$



- We can say that $\mathbf{v} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} = 5\begin{pmatrix} \frac{4}{5} \\ \frac{3}{5} \end{pmatrix} \rightarrow \hat{\mathbf{v}} = \begin{pmatrix} \frac{4}{5} \\ \frac{3}{5} \end{pmatrix}$
- What is the length of length vector $\hat{\mathbf{v}}$?

$$\|\hat{\mathbf{v}}\| = \sqrt{\left(\frac{4}{5}\right)^2 + \left(\frac{3}{5}\right)^2} = \sqrt{\left(\frac{16}{25}\right) + \left(\frac{9}{25}\right)} = 1$$

Unit Vectors

$$\mathbf{v} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} \rightarrow \|\mathbf{v}\| = \sqrt{4^2 + 3^2} = \sqrt{25} = 5$$

$$\mathbf{v} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} = 5\begin{pmatrix} \frac{4}{5} \\ \frac{3}{5} \end{pmatrix} \rightarrow \hat{\mathbf{v}} = \begin{pmatrix} \frac{4}{5} \\ \frac{3}{5} \end{pmatrix}$$

- We can write the vector as a scalar product of its length and its unit vector
- Therefore, the unit vector can be obtained by:

$$\forall\, \mathbf{v} \neq 0 : \hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

Dot Product

- The Inner Product or Dot Product of two vectors is defined as:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}_{1\times n} \quad \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}_{1\times n}$$

$$\mathbf{v}\cdot\mathbf{u} = v_1u_1 + v_2u_2 + \cdots + v_nu_n = \sum_{i=1}^{n} v_iu_i$$

- The result of dot product of two vectors is a scalar.
- Dot Product is not defined, if the vectors have different number of components

Dot Product - Properties

- For all vectors $\mathbf{v}$, $\mathbf{u}$, and $\mathbf{w}$ of the same dimension and for all numbers c, we have:

Commutative:

$$\mathbf{v}\cdot\mathbf{u} = \mathbf{u}\cdot\mathbf{v}$$

Distributive over vector addition:

$$\mathbf{v}\cdot(\mathbf{u}+\mathbf{w}) = \mathbf{v}\cdot\mathbf{u} + \mathbf{v}\cdot\mathbf{w}$$

Associative with respect to scalar multiplication:

$$\mathbf{v}\cdot(c\mathbf{u}) = (c\mathbf{v})\cdot\mathbf{u} = c(\mathbf{u}\cdot\mathbf{v})$$

No cancellation:

if $\mathbf{v}\cdot\mathbf{u} = \mathbf{v}\cdot\mathbf{w}$ and $\mathbf{v} \neq \mathbf{0}$, you cannot cancel out $\mathbf{v}$ from both sides of the equation.

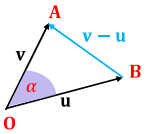
- Also, note that: $\|\mathbf{v}\|^2 = \mathbf{v}\cdot\mathbf{v}$

Dot Product – Geometric Point of View

- The Distance between two vectors is obtained by finding the norm of their subtraction:
- Subtraction of the two vectors:
- **Cosine Law:**

$$AB^2 = OA^2 + OB^2 - 2\,.OA.OB.\cos(\alpha)$$

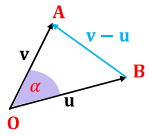
- $AB = \|\mathrm{v}-\mathrm{u}\|$
- $OA = \|\mathrm{v}\|$
- $OB = \|\mathrm{u}\|$



- Therefore:

$$\|\mathbf{v}-\mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\|\mathbf{v}\|\|\mathbf{u}\|cos(\alpha)$$

Dot Product – Geometric Point of View



$$\|\mathbf{v}-\mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\|\mathbf{v}\|\|\mathbf{u}\|cos(\alpha) \qquad (1)$$

- From the properties of dot product, we know that $\|\mathbf{v}\|^2 = \mathbf{v}\cdot\mathbf{v}$

$$\rightarrow \|\mathbf{v}-\mathbf{u}\|^2 = (\mathbf{v}-\mathbf{u})\cdot(\mathbf{v}-\mathbf{u}) = \mathbf{v}\cdot\mathbf{v} - 2\mathbf{v}\cdot\mathbf{u} + \mathbf{u}\cdot\mathbf{u}$$

$$\rightarrow \|\mathbf{v}-\mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\mathbf{v}\cdot\mathbf{u} \qquad (2)$$

- From (1) and (2) we have:

$$\|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\mathbf{v}\cdot\mathbf{u} = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\|\mathbf{v}\|\|\mathbf{u}\|cos(\alpha)$$

- Therefore:

$$\mathbf{v}\cdot\mathbf{u} = \|\mathbf{v}\|\|\mathbf{u}\|cos(\alpha)$$

Block Comparisons

Normalized Sum of Squared Differences (NSSD):

$$NSSD(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_1(j)^2}} - \frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_2(j)^2}} \right)^2$$

Normalized Cross Correlation (NCC):

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_1(j)^2}} \right) \left(\frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_2(j)^2}} \right)$$

Where $\overline{\mathbf{B}} = \mathbf{B} - \mu$

Based on what we just learned in the Math Review:

$$\sqrt{\sum_{j=1}^{n} \overline{\mathbf{B}}_1(j)^2} = \|\overline{\mathbf{B}}_1\|$$

Block Comparisons

Replacing $\|\overline{\mathbf{B}}_1\|$ into the equations:

Normalized Sum of Squared Differences (NSSD):

$$NSSD(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\|\overline{\mathbf{B}}_1\|} - \frac{\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_2\|} \right)^2$$

Normalized Cross Correlation (NCC):

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\|\overline{\mathbf{B}}_1\|} \right) \left(\frac{\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_2\|} \right)$$

Block Comparisons

A Geometric Interpretation:

$$NSSD(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\|\overline{\mathbf{B}}_1\|} - \frac{\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_2\|} \right)^2$$

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \sum_{i=1}^{n} \left(\frac{\overline{\mathbf{B}}_1(i)}{\|\overline{\mathbf{B}}_1\|} \right) \left(\frac{\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_2\|} \right)$$

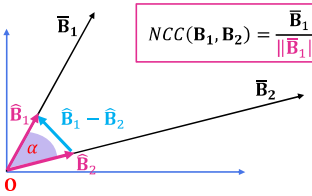
$$\widehat{\mathbf{B}}_1 = \frac{\overline{\mathbf{B}}_1}{\|\overline{\mathbf{B}}_1\|}$$ Is the unit vector of $\overline{\mathbf{B}}$

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \frac{\sum_{i=1}^{n} \overline{\mathbf{B}}_1(i) \overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_1\| \|\overline{\mathbf{B}}_2\|}$$

Based on the definition of dot product:

$$NSSD(\mathbf{B}_1, \mathbf{B}_2) = \|\widehat{\mathbf{B}}_1 - \widehat{\mathbf{B}}_2\|^2$$

$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \frac{\overline{\mathbf{B}}_1 \cdot \overline{\mathbf{B}}_2}{\|\overline{\mathbf{B}}_1\| \|\overline{\mathbf{B}}_2\|} = \widehat{\mathbf{B}}_1 \cdot \widehat{\mathbf{B}}_2 = \cos(\alpha)$$



Suppose we have 2D vectors:

- What is the NCC?
- Remember $\mathbf{v} \cdot \mathbf{u} = \|\mathbf{v}\| \|\mathbf{u}\| cos(\alpha)$?

Therefore, NCC is the cosine of the angle between the two vectors

Block Comparisons

A Geometric Interpretation:

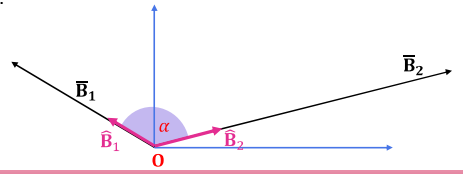
$$NCC(\mathbf{B}_1, \mathbf{B}_2) = \frac{\overline{\mathbf{B}}_1 \cdot \overline{\mathbf{B}}_2}{\|\overline{\mathbf{B}}_1\| \|\overline{\mathbf{B}}_2\|} = \widehat{\mathbf{B}}_1 \cdot \widehat{\mathbf{B}}_2 = \cos(\alpha)$$

Therefore, NCC is the cosine of the angle between the two vectors

$$-\mathbf{1} \leq NCC(\mathbf{B}_1, \mathbf{B}_2) = \cos(\alpha) \leq \mathbf{1}$$

- When NCC is close to **-1**, the angle between two vectors is large which means that the two vectors are different and far from each other (negatively correlated).
- When NCC is close to ZERO,

the vectors are Orthogonal (uncorrelated).



Block Comparisons

Algebraic relationship between NSSD and NCC:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)}{\|\overline{\mathbf{B}}_1\|}-\frac{\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_2\|}\right)^2 \qquad NCC(\mathbf{B}_1,\mathbf{B}_2)=\sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|}\right)$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)^2}{\|\overline{\mathbf{B}}_1\|^2}-2\frac{\overline{\mathbf{B}}_1(i)\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|}+\frac{\overline{\mathbf{B}}_2(i)^2}{\|\overline{\mathbf{B}}_2\|^2}\right)$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\frac{\sum_{i=1}^{n}\overline{\mathbf{B}}_1(i)^2}{\|\overline{\mathbf{B}}_1\|^2}+\frac{\sum_{i=1}^{n}\overline{\mathbf{B}}_2(i)^2}{\|\overline{\mathbf{B}}_2\|^2}-2\frac{\sum_{i=1}^{n}\overline{\mathbf{B}}_1(i)\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|}$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\frac{\|\overline{\mathbf{B}}_1\|^2}{\|\overline{\mathbf{B}}_1\|^2}+\frac{\|\overline{\mathbf{B}}_2\|^2}{\|\overline{\mathbf{B}}_2\|^2}-2\frac{\sum_{i=1}^{n}\overline{\mathbf{B}}_1(i)\overline{\mathbf{B}}_2(i)}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|}$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=2\big(1-NCC(\mathbf{B}_1,\mathbf{B}_2)\big)$$

Block Comparisons – Similarity Metrics Summary

- Standard Score or z-score ($z=\frac{x-\mu}{\sigma}$) is a way of normalization when the populations are available.
- Z-score is a dimensionless quantity.
- We showed that the normalized scores can be obtained using z-score normalization:

$$NSSD(B_1,B_2)=\frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h}(z_1(i,j)-z_2(i,j))^2$$

$$NCC(B_1,B_2)=\frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h}z_1(i,j)\ z_2(i,j)$$

- Using a geometric interoperation, we proved that both NSSD and NCC can be viewed as some basic operations by the unit vectors of each image block.

Block Comparisons – Similarity Metrics Summary

- In fact, NSSD can be calculated by the squared length (norm) of the difference between the unit vector of each block $\widehat{\mathbf{B}}_1-\widehat{\mathbf{B}}_2$:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\left\|\widehat{\mathbf{B}}_1-\widehat{\mathbf{B}}_2\right\|^2$$

- NCC can also be obtained by dot product of the unit vectors of each block which equals the cosine of the angle between two vectors:

$$NCC(\mathbf{B}_1,\mathbf{B}_2)=\frac{\overline{\mathbf{B}}_1\cdot\overline{\mathbf{B}}_2}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|}=\widehat{\mathbf{B}}_1\cdot\widehat{\mathbf{B}}_2=\cos(\alpha)$$

- An inevitable corollary of this equivalence is that NCC is bounded by:
  - `-1` : meaning that the two vectors are negatively correlated (o: Orthogonal/Uncorrelated)
  - `1` : meaning that the two vectors are equal and highly correlate
- We learned that there is a relationship between NSSD and NCC:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=2\big(1-NCC(\mathbf{B}_1,\mathbf{B}_2)\big)$$

Block Comparisons – Similarity Metrics Summary - Let's code!

- We can write Matlab code in various ways:

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_1(j)^2}}-\frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_2(j)^2}}\right)^2$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\left\|\widehat{\mathbf{B}}_1-\widehat{\mathbf{B}}_2\right\|^2$$

$$NSSD(\mathbf{B}_1,\mathbf{B}_2)=\frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h}(z_1(i,j)-z_2(i,j))^2$$

```
function NSSD = NSSDScore(B1,B2)
% Calculates NSSD metric without any built-in function
    f = (B1(:)-mean(B1(:)));
    g = (B2(:)-mean(B2(:)));
    f_norm = sqrt(sum(f.*f));
    g_norm = sqrt(sum(g.*g));
    NSSD = sum( (f./f_norm - g./g_norm).^2);
end
```

```
function NSSD = NSSDScore_UV(B1,B2)
% Calculates NSSD metric based on the unit vector formulation
    f = (B1(:)-mean(B1(:)));
    g = (B2(:)-mean(B2(:)));
    NSSD = norm((f./norm(f)) - (g./norm(g)))^2;
end
```

```
function NSSD = NSSDScore_ZS(B1,B2)
% Calculates NCC metric based on the z-scores
    z1 = zscore(B1(:));
    z2 = zscore(B2(:));
    % Since Matlab's std used the Corrected equations of
    %Standard Deviation, we should divide our summation by n-1
    NSSD = sum( (z1 - z2).^2)./(numel(B1)-1);
end
```

Block Comparisons – Similarity Metrics Summary - Let's code!

- We can write Matlab code in various ways:

$$NCC(\mathbf{B_1},\mathbf{B_2}) = \sum_{i=1}^{n}\left(\frac{\overline{\mathbf{B}}_1(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_1(j)^2}}\right)\left(\frac{\overline{\mathbf{B}}_2(i)}{\sqrt{\sum_{j=1}^{n}\overline{\mathbf{B}}_2(j)^2}}\right)$$

$$NCC(\mathbf{B_1},\mathbf{B_2}) = \frac{\overline{\mathbf{B}}_1 \cdot \overline{\mathbf{B}}_2}{\|\overline{\mathbf{B}}_1\|\|\overline{\mathbf{B}}_2\|} = \widehat{\mathbf{B}}_1 \cdot \widehat{\mathbf{B}}_2 = \cos(\alpha)$$

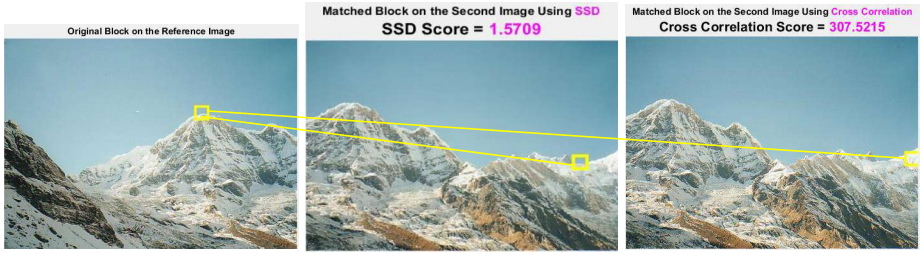
$$NCC(\mathbf{B_1},\mathbf{B_2}) = \frac{1}{n}\sum_{i=1}^{w}\sum_{j=1}^{h} z_1(i,j)\ z_2(i,j)$$

```
function NCC = NCCScore(B1,B2)
% Calculates NCC metric without any built-in function
   f = (B1(:)-mean(B1(:)));
   g = (B2(:)-mean(B2(:)));
   f = f ./ sqrt(sum(f.*f));
   g = g ./ sqrt(sum(g.*g));
   NCC = sum(f.*g);
end
```

```
function NCC = NCCScore_UV(B1,B2)
% Calculates NCC metric based on the unit vector formulation
   f = (B1(:)-mean(B1(:)));
   g = (B2(:)-mean(B2(:)));
   NCC = dot(f./norm(f),g./norm(g));
end
```

```
function NCC = NCCScore_ZS(B1,B2)
% Calculates NCC metric based on the z-scores
   z1 = zscore(B1(:));
   z2 = zscore(B2(:));
   % Since Matlab's std used the Corrected equations of
   Standard Deviation, we should divide our summation by n-1
   NCC = sum(z1.*z2)./(numel(B1)-1);
end
```

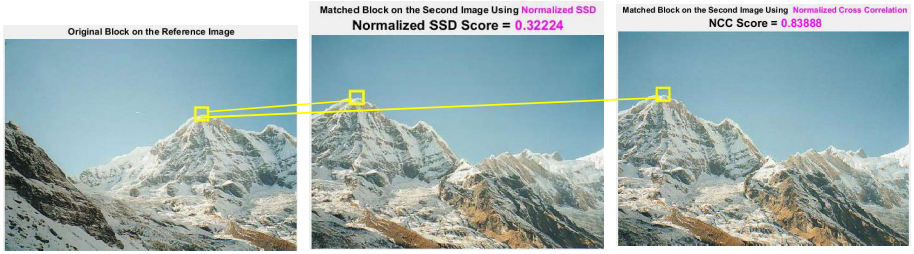

Block Comparisons



We saw that without normalization both methods failed!

Block Comparisons

Let's check the normalized versions! (m-file name: Unit_3_Block_Comparisons_Normalized_Scores.m)

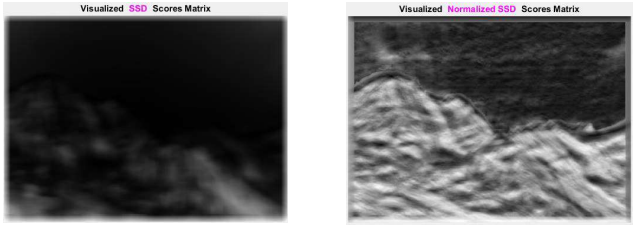


- In comparing blocks with NSSD we looked for minimum NSSD.
- However, when we use NCC we should find blocks with maximum NCC (higher correlation)

Block Comparisons

What do the score matrices looks like?

- Note that we have compared a block around every pixels in the second image with the template block of the first image, and store the NSSD and NCC in two different matrices.
- If we adjust the intensity range of the score matrices and display them, we obtain:

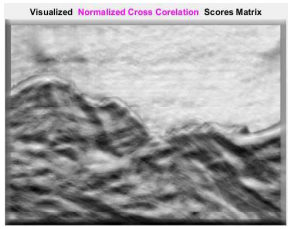


- We can see that scores obtained without doing normalizations (left image) are very blurry which demonstrate high uncertainty in the process of comparisons.

Block Comparisons

What do the score matrices looks like?

- Note that we have compared a block around every pixels in the second image with the template block of the first image, and store the NSSD and NCC in two different matrices.
- If we adjust the intensity range of the score matrices and display them, we obtain:

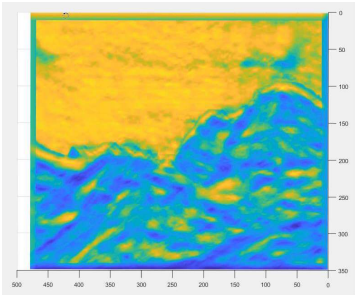


- We can see that scores obtained without doing normalizations (left image) are very blurry which demonstrate high uncertainty in the process of comparisons.

Block Comparisons

What do the score matrices looks like?

- We can also display the scores as a 3D surfaces:



Block Comparisons – Highest Correlated Blocks

What are the highest correlated blocks?

- We see the 6 highest correlated blocks.
- Since there is a large portion of sky in the template block, the last two blocks obtained a high correlation score.