# Image Registration:

- Reference vs sensed / test Image
- Multimodal
- Temporal
- Multiview

## Transformation Estimation

| Reference | Sensed Image |
| --- | --- |
| lowercase | uppercase |
| $(x, y)$ | $(X, Y)$ |

$\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} \to p \to$ sensed pixels

$\begin{pmatrix} x \\ y \end{pmatrix} \to p \to$ ref. pixels

# Day 1

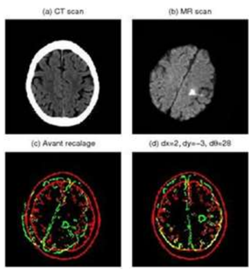
UNIVERSIDAD DE GRANADA

COSI Computational Colour and Spectral Imaging

# Chapter 1.

## Image Registration.

## General work-flow and basic registration techniques using Matlab






Eva M. Valero

COLOUR IMAGING LAB
University of Granada
SPAIN

# Outline



(1) What is image registration and what is it for?

(2) General Image Registration Workflow

(3) Basic image transformation functions

(4) Control Point and Feature Extraction

(5) Control Point Matching

(6) Obtaining the Transformation Function

(7) Image resampling

(8) Evaluation metrics for image registration quality

(9) Lab session: Basic image registration algorithms. Solving image registration in Matlab

## 1. What is image registration and what is it for?

UNIVERSIDAD DE GRANADA

Computational Colour and Spectral Imaging

→ Definition: aligning two images so that the pixels are corresponding [1]

(a) (b)

(a) Sensed or test Image (b) Reference Image

We register the sensed image onto the reference image

01/16/2023 ACIP. Image registration 4/74

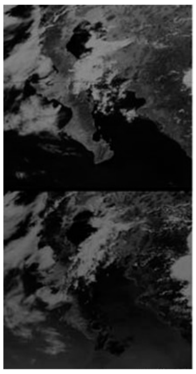
Definition of registration. The basic elements to work with are two images (if there are more, then the process is repeated using two images at a time). The two images are called reference image and sensed or test image. The reference image is the one we want to keep as it is, and the sensed image is the one we want to align with the reference. We can see the sensed image as representing a transformation of the reference image (in geometrical terms). The registration problem is to find this transformation, that will allow us to properly map the reference pixels onto the sensed image coordinate system. In the figure, the reference image is (b) and the sensed or test image is (a).

UNIVERSIDAD DE GRANADA

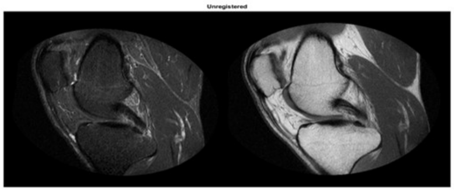
## 1. What is image registration and what is it for?

Computational Colour and Spectral Imaging

→ Some instances of application:



(a) Satellite Imaging
(images from LeMoigne et al. 2011)[2]



(b) Medical Imaging (images from Mathworks)[3]

**Multimodal** images: images obtained from different devices or with signifficant differences in intensity



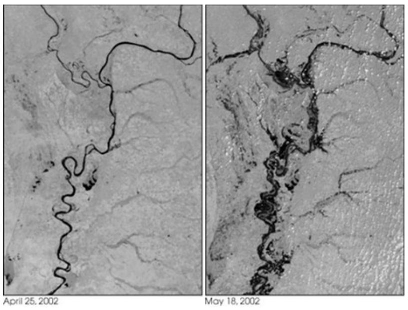
We see two examples of registration challenges in the satellite imaging domain (Baja California at different times of the day and night) and medical imaging domains (MRI images of a knee with different techniques). These two are instances of what is called Multimodal images, because the different acquisition techniques or the different illumination conditions result in noticeable intensity differences between the reference and sensed images. The registration problem is harder for multimodal images because there is usually a huge intensity difference between reference and sensed images at the starting point of the registration process, and even when the two images are aligned, they will not be similar in intensity.

UNIVERSIDAD DE GRANADA

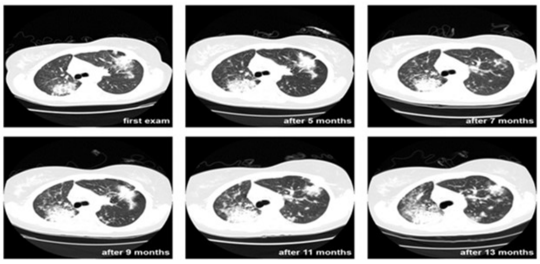
## 1. What is image registration and what is it for?

Computational Colour and Spectral Imaging

→ Some instances of application:




(c) Satellite Imaging
(images from LeMoigne et al. 2011)[2]




(d) Medical Imaging (images from [4])

**Temporal** images: images obtained with the same capture device at different times

01/16/2023 ACIP. Image registration 6/74

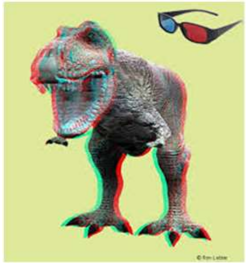
Sometimes the conditions of the scene change between capturing the reference and sensed images, even if they are captured with the same device. This is the case of temporal image registration. In the medical imaging field, it might correspond as well to the monitoring of tumor growth in a given subject. In this last case, we need to be very careful to avoid transforming the reference image in a way that makes us unable to monitor the growth of the tumor, which is restricted to a small area of the image.

UNIVERSIDAD DE GRANADA

## 1. What is image registration and what is it for?

Computational Colour and Spectral Imaging

→ Some instances of application:



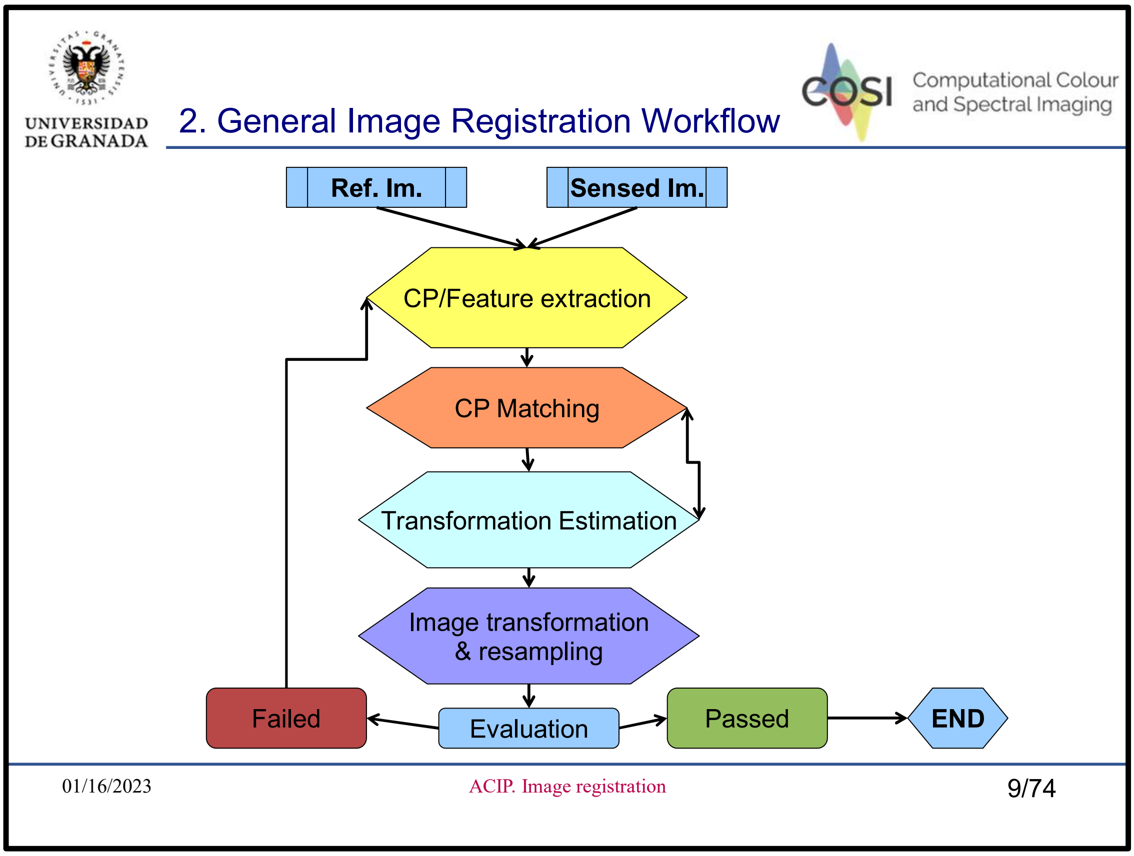
(e)3D view generation or panorama imaging (images from google and Parklab webpage)[5]

**Multiview** images: images obtained with the same capture device but with perspective changes induced either by capture device or scene elements movement

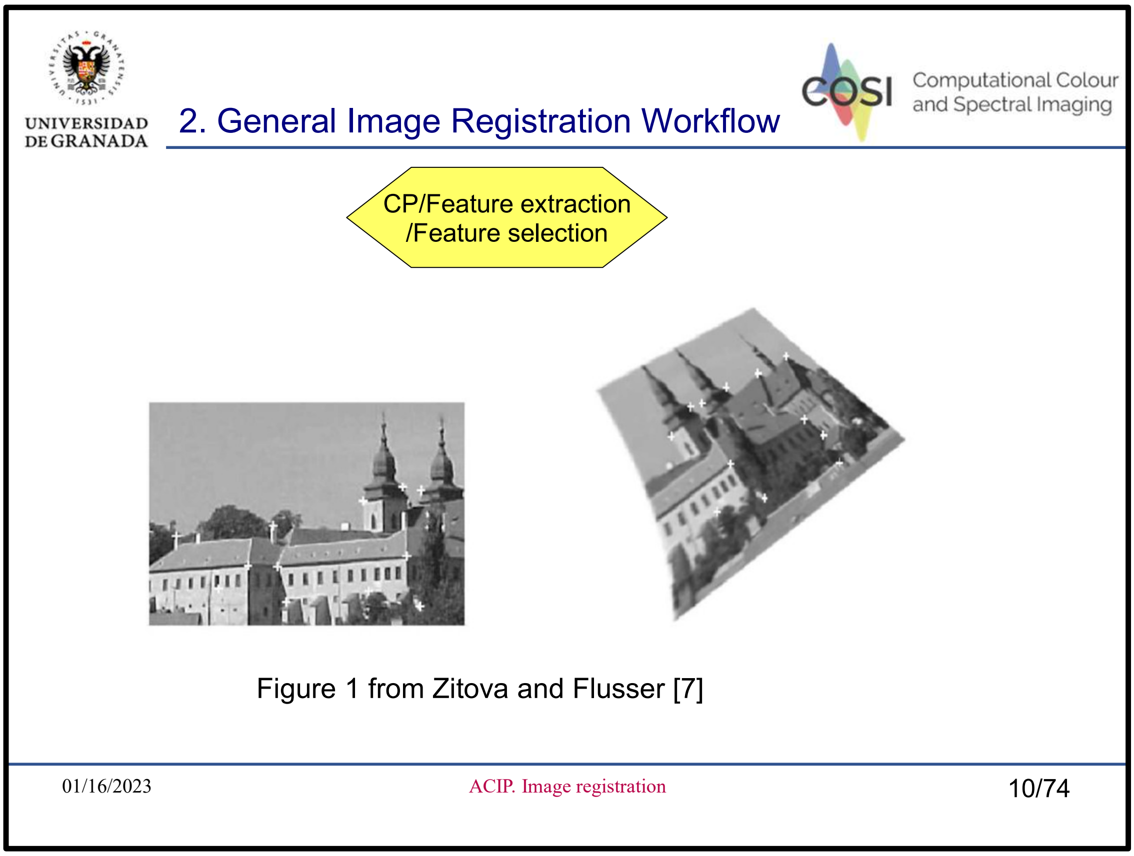
01/16/2023 ACIP. Image registration 7/74

If we want to build a panoramic image out of several pictures of a scene taken from different perspectives, we need registration to be able to correctly align the different components of the scene. Also, we need it to build correctly 3D images out of two pictures of the same object captured from different points of view (anagliph images, or polarized 3D images).

## Quick question




We now present the general registration workflow, which we will be referring to quite often in the course of the chapter. This workflow is valid for most image registration techniques, but not all; some methods will skip some of the steps. For instance, area methods skip the Control Point (CP) extraction, or some methods can skip the matching step if the CP extraction ensures correspondence (for instance, when we use a regular pattern or checkerboard for CP extraction, and the two images to be registered are not extremely different).




For most registration methods, a key step is the extraction of the so-called control points, which will be later on used to learn the transformation function. As we see here, they are extracted independently in the reference and sensed images. CPs can be extracted manually or automatically, based on many different criteria related to spatial image features. To identify a relevant point in the image,  many different algorithms can be used. Most of them involve the extraction of features from the image. If the feature information is used later on in the workflow to help for finding corresponding CPs, then the features are preserved in this step for future use.

UNIVERSIDAD DE GRANADA

## 2. General Image Registration Workflow

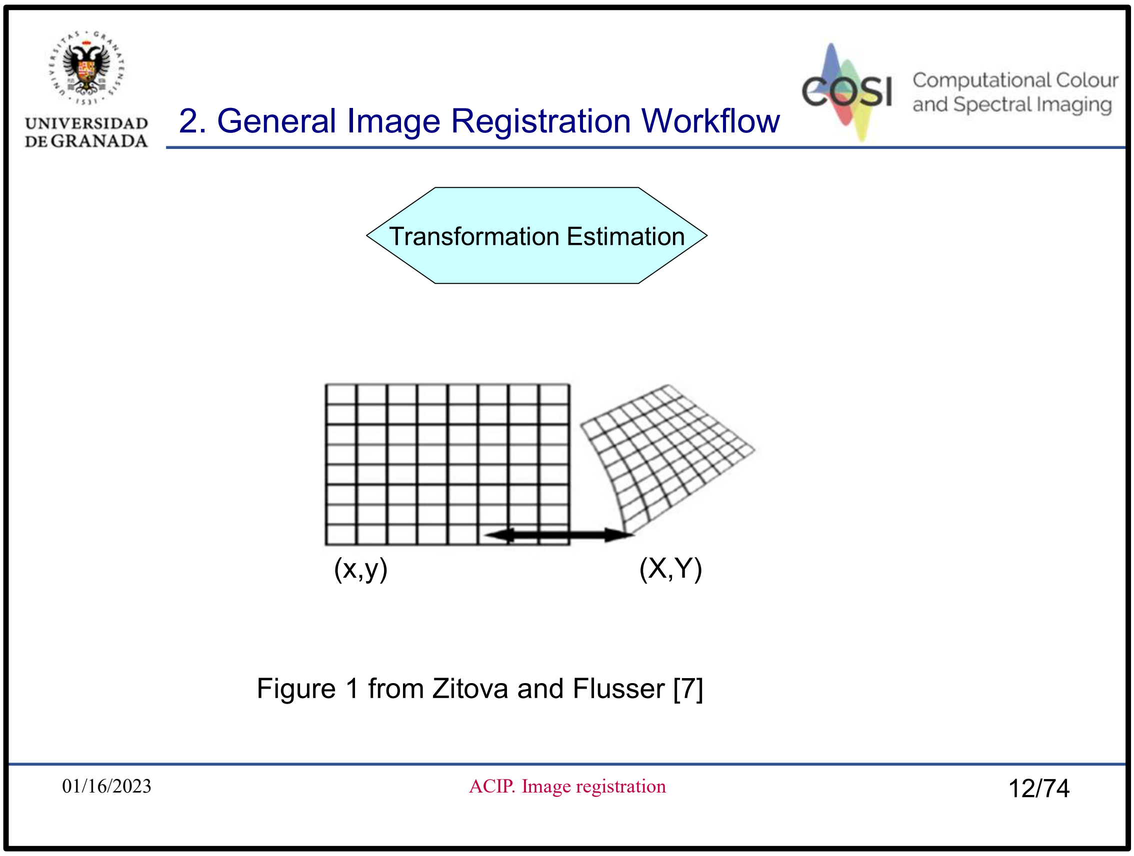
Computational Colour and Spectral Imaging

CP Matching

Figure 1 from Zitova and Flusser [7]



If the CPs are extracted automatically, then the extraction algorithm will not necessarily find exactly the same CP sets in both images. So, it becomes necessary to find which pairs of CPs are corresponding, and this step of the process requires additional techniques. We see in the pictures how after the matching, the corresponding CP pairs are labeled with the same number. Quite often, the steps of CP matching and learning the transformation function are quite connected. The basic idea is that if we get close to the real transformation function, then when we apply it to the reference image CPs, we will get pixels that are very close to the corresponding CPs in the sensed image.




As explained previously, some times we can obtain a first estimation of the transformation function from the matching step. Afterwards, some refinement is needed to make the original estimation as precise as possible. This process depends very critically on if we know a priori the kind of transformation that we are searching for, or not entirely, or not at all. If we know about this transformation and if it is the same for every pixel in the reference image, then we can use regression techinques to learn the correct parameters. If we know there are some local variations that we want to model, then more sophisticated estimators will be required. In the end, the goal is always to fix the transformation parameters starting from a predefined search grid or an initial guessing.

UNIVERSIDAD DE GRANADA

## 2. General Image Registration Workflow

Computational Colour and Spectral Imaging

Image transformation & resampling

Figure 1 from Zitova and Flusser [7]



Once the transformation is known, then we can proceed in different ways to perform the registration. One possibility is to start from the reference pixel, then map it onto the sensed image, and take the corresponding sensed intensity to build a new "registered" image by placing this intensity in the spatial position corresponding to the reference pixel. Or we can just model a transformation that predicts the pixel displacement needed to move the sensed pixel onto its corresponding reference pixel. Other possibility is to apply the inverse transform of the one mapping reference onto sensed images, but this is not very common because of some unstabilities which might come up due to the inversion process. Also, some transformations will not be clearly invertible.

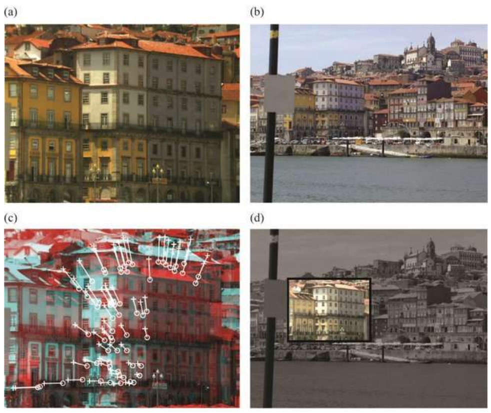
# Quick questions

Which step is represented in figure c)?

A) CP or Key point extraction
B) Feature-based CP Matching
C) Learning the transformation
D) Image transformation and re-sampling

And in figure d)?

B




D

UNIVERSIDAD DE GRANADA

3. Basic Image Transformation Functions

Computational Colour and Spectral Imaging

→ **Global** and **local** transformation functions:

Global: same transformation for all pixels.

Local: the transformation changes across the scene (stitching problem)

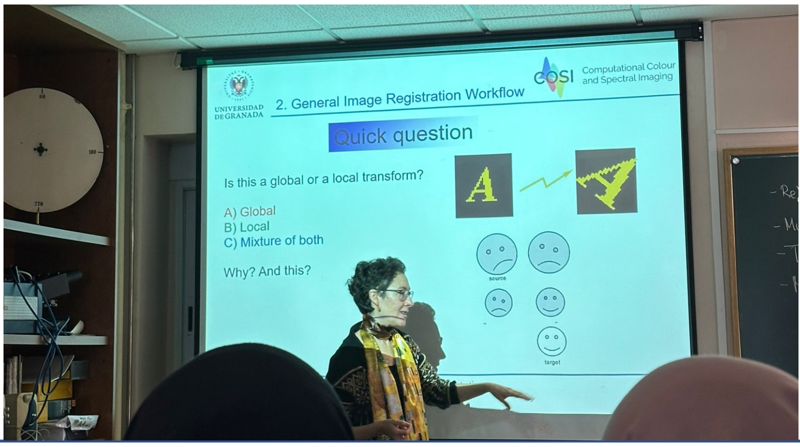
(Images from Google)

01/16/2023 ACIP. Image registration 15/74

As we pointed out before, it is important to know if the transformation we are searching for is the same for all pixels of the image (global) or it changes with pixel location (local). When we learn local transformations, we need to make sure that they are stitched together seamlessly to perform the registration process without creating discontinuities in the registered image.

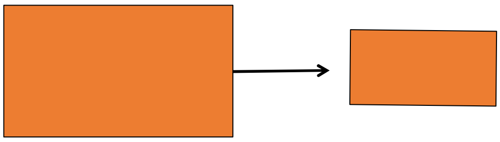
# Quick question

1) Local
2) Mixture



→ What we need to know about a global transformation function:

a) What are its effects on the image?: e.g. it scales the image (change size in the same proportion in horizontal and vertical direction)



Scaled transformation

b) How can we express it mathematically? (obtain the transformation matrix)

$$\mathbf{P} = T\mathbf{p}$$

Sensed

$$X = sx$$
$$Y = sy$$

ref

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



Let's learn about how to characterize a global transform. We need to know, in practical terms, the answer to four different questions. The first one is what is the transformation doing to the image (let's use the scaling transform as an example: a scaling transform changes the size of the image by the same amount in the horizontal and vertical axes; it either reduces or increases the image size, while maintaining the image shape). The second question is how can we express the global transform in a convenient mathematical way. We start from the basic equations, where uppercase P represents the sensed image pixels, and lowercase p represents the reference pixels. The transformation T can be expressed in matricial form for further convenience. The '1' is added to point out that it is a 2D transform in our case.

UNIVERSIDAD DE GRANADA



Computational Colour and Spectral Imaging

→ What we need to know about a global transformation function:

c) How many unknown parameters will you have to estimate? (e.g. only 1 (s) for the scaling transform)

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

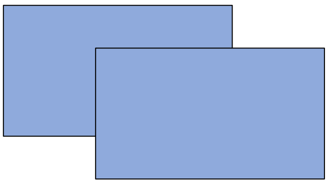
d) How many pairs of Control Points will you need for the parameter estimation? (e.g. only one for the scaling transform)



The third question is about the number of parameters that we will need to estimate to fully characterize our transform. In the case of the scaling transform, we just need to estimate one parameter (s, scaling factor). Finally, it is very straightforward to know how many pairs of corresponding CPs we will need to solve our global transform estimation problem. This depends on the number of parameters to estimate, taking into account that each pair of corresponding CP allows for two equations from which to estimate the parameters (x and y components). So for the scaling transform, it is enough (in fact, more than enough) to have one pair of corresponding CPs to solve the registration problem.

- 1 parameter
- 1 CP pair needed

3.1 **Translation**

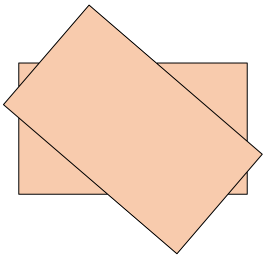


$$X = x + h$$
$$Y = y + k$$

2 parameters

1 CP

3.2 **Rotation**



$$X = x \cdot \cos\theta - y \cdot \sin\theta$$
$$Y = x \cdot \sin\theta + y \cdot \cos\theta$$

* How does theta corresponds to clock/counter clockwise



The most simple global transformation is translation. The image is just displaced (moved) without changing its shape. In terms of spatial coordinates, it is described by a fixed additive change in x and y position.

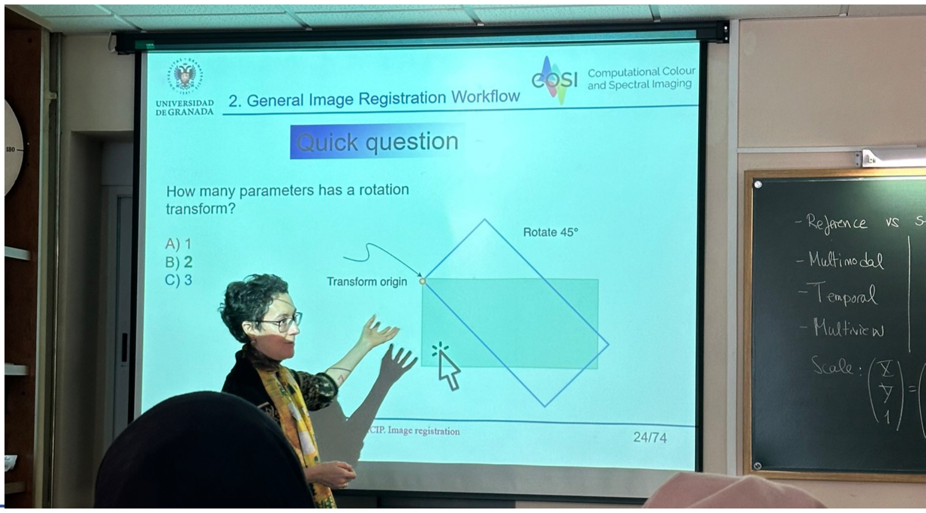
For the rotation, we can see the effect on the reference rectangle of an instance of this transform, and the standard way of expressing it in mathematical terms. The parameter theta is the angle of rotation. Its sign changes for clockwise anad counter-clockwise rotation.

UNIVERSIDAD DE GRANADA

## 2. General Image Registration Workflow

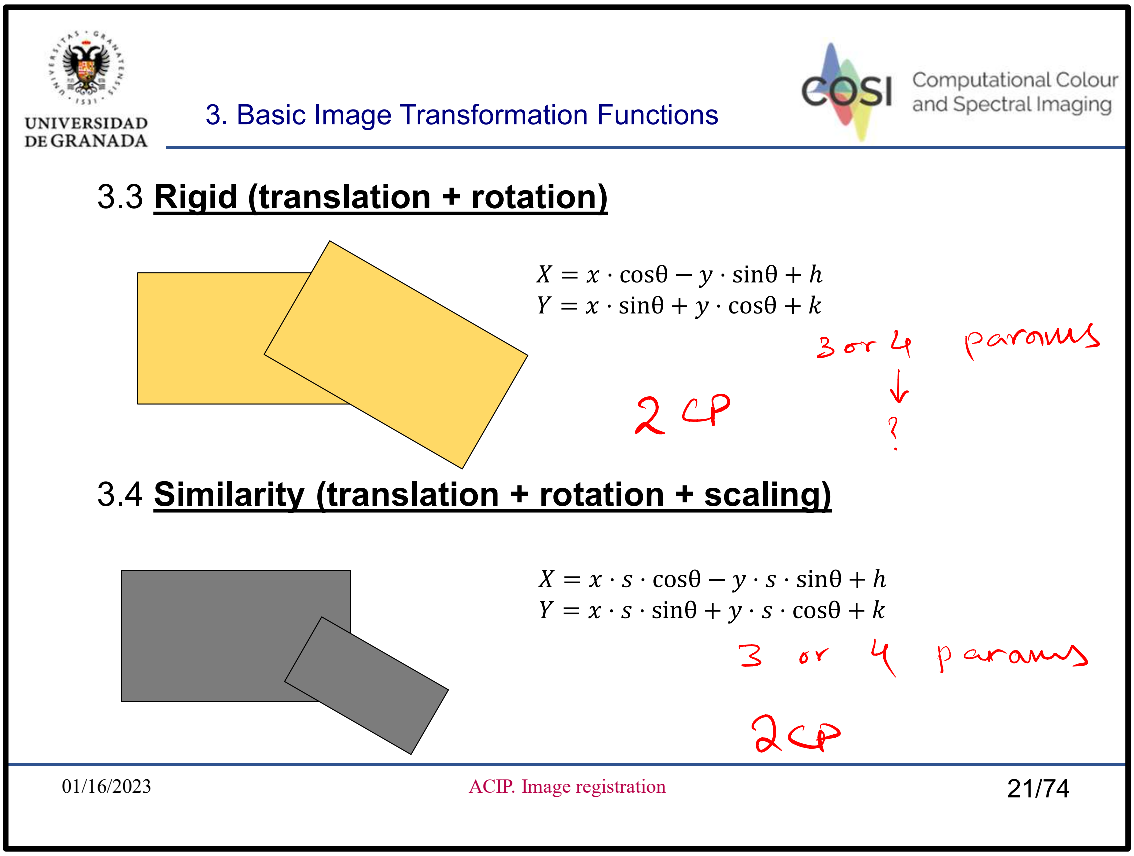
Computational Colour and Spectral Imaging

### Quick question






sin & cos are related
but still 2 things



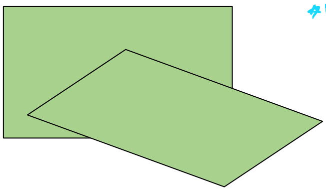

If we apply a translation followed by a rotation (or viceversa), then we get a rigid transform. It preserves the size of each object in the image. When we join a translation, a rotation and a scaling transformation, then we have a similarity transform as the final result. The variation from the rigid case is that the image changes size (it is scaled by a constant factor s). In this transformation, the angles between each pair of segments in the reference image are preserved in the sensed image. In the example shown in the slide, the s factor is less than unity.

UNIVERSIDAD DE GRANADA

3. Basic Image Transformation Functions

COSI Computational Colour and Spectral Imaging

### 3.5 Affine (translation + rotation + scaling + shearing)



*parallel

$$X = a_1 \cdot x + a_2 \cdot y + a_3$$
$$Y = a_4 \cdot x + a_5 \cdot y + a_6$$

6 Params
3 CP

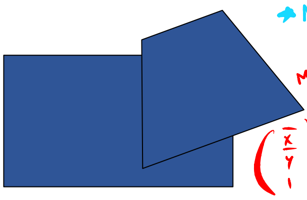
General shear

$$T = \begin{pmatrix} 1 & S_y & 0 \\ S_x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Horizontal Shear

$$\bar{x} = x + S_y y$$
$$\bar{y} = y$$

### 3.6 Projective (homography)



*Not parallel any more

Matrix form:-

$$\begin{pmatrix} \bar{x} \\ \bar{y} \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$X = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1}$$
$$Y = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}$$

8 params
4 CP

HW: Consult the book

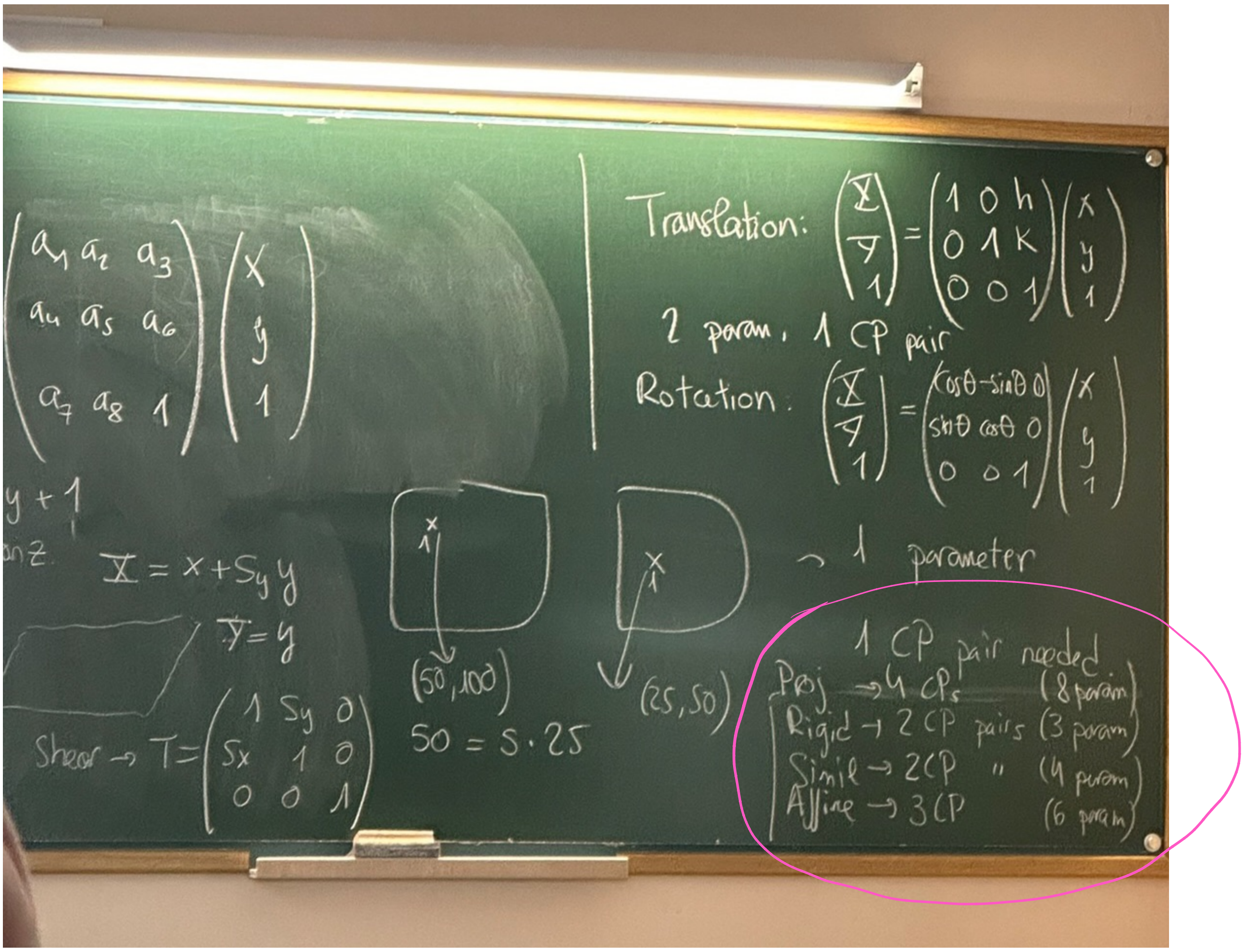
Change of perspective. The straight lines are still straight after the transformation.

01/16/2023 ACIP. Image registration 22/74

The affine transform adds the shearing to the similarity transform. It is usually expressed using non-explicit parameters or matrix coefficients, because the mathematical explicit form is rather long. This transforms does not preserve the angles between segments, but it preservers parallelity of segments.

Finally, the most complex of the common global geometric transformations that we are going to describe here is the projective transform or homography. It is adequate to represent the change of perspective happening when the capturing device sees the scene from another point of view, or when the object plane is tilted with respect to the optical axis of the camera.

This is illustrated in Brown (1992, ref. [6], section 2.4). Neither angles between lines nor parallelism are preserved.



UNIVERSIDAD DE GRANADA

## 3. Basic Image Transformation Functions

Computational Colour and Spectral Imaging

- Tips for using spatial transformations in Matlab:

1) Get the spatial reference framework of the image, using *imref2d* function.

2) The matrix transforms used in Matlab are slightly different than the standard ones: [x y 1] = [u v 1] * T
T is then the transposed matrix of the standard transform (for rotation, there is no need to do this when using imwarp or imrotate, but it is needed for single points transformation with transformPointsForward).

3) For most transforms, you can manage well using the function *imwarp*, using tform to define the transformation beforehand.

4) Use 'OutputView' as parameter for convenient visualization

not projective = affine transform

$\boxed{\text{imwarp}}$

Reference frame?

01/16/2023 ACIP. Image registration 23/74

The registration framework in Matlab involves the estimation of the transformation from sensed to reference images (opposite view from the one in our reference book, Goshtasby). Also, the transformations in Matlab are applied using row vectors as base, so the matrices are the transposed versions of the standard way of writing global transforms. In the slide, we get some tips for applying transforms to images in Matlab, which we will use for tackling class Activity 1.

# Activity 1

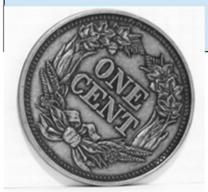
UNIVERSIDAD DE GRANADA

COSI Computational Colour and Spectral Imaging

**Global transforms. 3 class points. Team work (3 teams)**

Use any image you like as a starting point. Implement in Matlab two transformations among the ones described, with free parameters. Use either pre-defined functions or *affine2d* where applicable. Use the tips provided in slide 23.

Discuss the results obtained.
Does the transformed image look as sharp as the original image? Why?
Send the code you used to solve this class activity to the instructor, along with a brief report showing results and answering the questions.
DEADLINE: April 24th 14 h



UNIVERSIDAD DE GRANADA

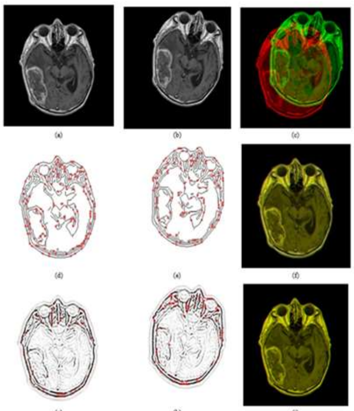
COSI Computational Colour and Spectral Imaging

# Day 2

Missed this class

Covered in Kimia's

CPs can be extracted manually (visual interactive process) or automatically (usually by feature computation searching local extrema of the feature image or corners/blobs)



→ Interest in automatic CP extraction

→ If the method requires it, the feature vectors are built from the CPs and used in the CP matching step afterwards

→ Generally, two steps: Feature detection and Feature extraction

Image from http://article.sapub.org/image/10.5923.j.ajbe.20120203.02_039.gif



In this section, we will describe the basics of one of the key steps in image registration, the CP extraction. Most registration algorithms use automatic CP extraction based on image features, or else search for uniqueness in the image distribution, like centers of blobs, edges, or corners.
In the figure, we can see an example of feature detection for monomodal images, applied to compensate for subject motion during brain scan acquisition. Esentially the edge features are extracted by two different procedures (standard Canny edge detector and othe technique based on shift-invariant contourlet transforms. In the right column, we can see the

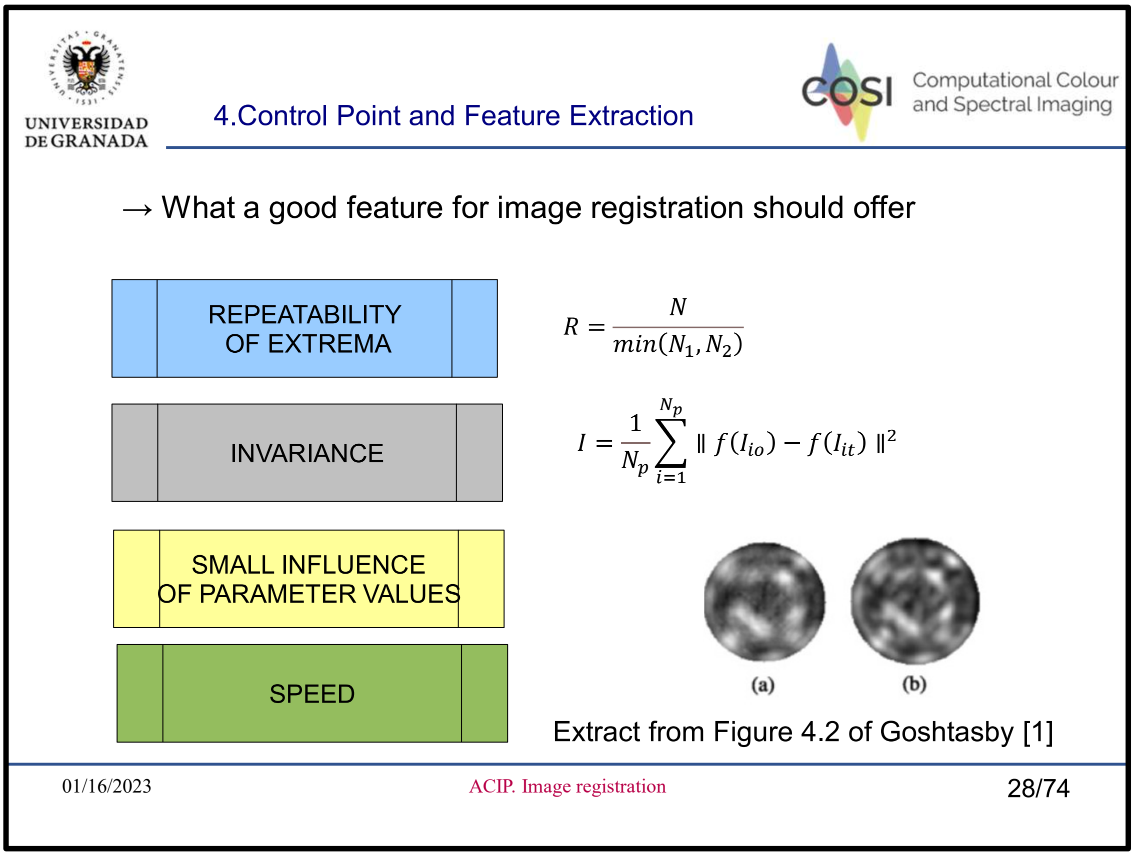
results of the registration in color overlay (reference image in red, and registered image in green). The yellower the overlay image, the better the quality of the registration obtained, especially if there are no red or green fringes).

4.1 **Feature extraction**

| **Feature classification[1]** | |
|---|---|
| Statistical *(e.g. histogram mean)* | Differential *(e.g. Laplacian, Gaussian derivatives)* |
| Geometrical *(e.g. edges, Hu moments)* | Spatial domain (*e.g. deviation from Mean, contrast)* |
| Algebraic *(singular value decomposition, Harris and Stephens corner detector)* | Color (*e.g. relative color, dominant color)* |
| Frequency domain (*DFT, Haar transform*) | Fractal Dimension |
| Filter responses *(e.g. Gabor/steerable filters/Laws masks)* | Information theoretic *(e.g. local entropy)* |
| Combined (*e.g. SURF, BRISK, MinEigen, FAST, ORB*) | |



CP detection is very closely related to feature extraction, because in many CP detection algorithms features are used to identify the CP candidates. After the CPs are extracted, we can decide if to use their features to perform the next step in the registration workflow, or else just keep the CP locations as features and proceed to the matching step. In the table, we offer a tentative classification of image features. We will not describe these methods in detail, because of lacking of time. Complete information can be found in [1], and in the Matlab reference (see links in the reference section) for the combined methods.




We can evaluate the quality of the features used for registration based on repeatability (proportion of CP in common found for the two images), and also invariance, which compares directly the feature values of the reference and sensed image in corresponding pixels. We can also consider that it is convenient that the feature does not pose a parametric optimization problem, and that it is fast to compute. In the remainder of this section, we will evaluate features of four different categories by assessing their corresponding "feature images" obtained from the reference and sensed images using a pre-defined neighbourhood around each pixel.

### Feature Detection with Matlab built-in features

| Detector | Feature Type | Function | Scale Independent |
|---|---|---|---|
| FAST [1] | Corner | detectFASTFeatures | No |
| Minimum eigenvalue algorithm [4] | Corner | detectMinEigenFeatures | No |
| Corner detector [3] | Corner | detectHarrisFeatures | No |
| SURF [11] | Blob | detectSURFFeatures | Yes |
| KAZE [12] | Blob | detectKAZEFeatures | Yes |
| BRISK [6] | Corner | detectBRISKFeatures | Yes |
| MSER [8] | Region with uniform intensity | detectMSERFeatures | Yes |
| SIFT | Blobs and corners | detectSIFTFeatures | Yes |
| ORB [13] | Corner | detectORBFeatures | No |

https://es.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html



We present here a summary of detector methods in Matlab, and the type of features they detect.

## Quick question

## Feature Extraction with Matlab built-in features

| Descriptor | Binary | Function and Method | Invariance | | Typical Use | |
|---|---|---|---|---|---|---|
| | | | Scale | Rotation | Finding Point Correspondences | Classification |
| HOG | No | extractHOGFeatures(I, ...) | No | No | No | Yes |
| LBP | No | extractLBPFeatures(I, ...) | No | Yes | No | Yes |
| SURF | No | extractFeatures(I,points,'Method','SURF') | Yes | Yes | Yes | Yes |
| KAZE | No | extractFeatures(I,points,'Method','KAZE') | Yes | Yes | Yes | Yes |
| FREAK | Yes | extractFeatures(I,points,'Method','FREAK') | Yes | Yes | Yes | No |
| BRISK | Yes | extractFeatures(I,points,'Method','BRISK') | Yes | Yes | Yes | No |
| ORB | Yes | extractFeatures(I,points,'Method','ORB') | No | Yes | Yes | No |
| •Block<br>•Simple pixel neighborhood around a keypoint | No | extractFeatures(I,points,'Method','Block') | No | No | Yes | Yes |



We see in this table a summary of the main features that Matlab incorporates for image processing in general, along with the transformations for which they are invariant and their main application domains. The function to call for its extraction is also listed. You will find this table useful when tackling Activity 2.

## 4.2.11 Example experiment comparing several features

| Feature | Metric | Blur | Noise | Intensity | Rotation | Scaling |
|---|---|---|---|---|---|---|
| $F_1$ hist_mean | I | 0,97 | **0,98** | **0,98** | **0,99** | **0,93** |
| | R | **1** | 0,75 | 0,86 | **0,88** | **0,12** |
| $F_2$ Hu moment | I | **0,98** | **0,98** | **0,98** | **0,99** | **0,93** |
| | R | 0,88 | 0,67 | **0,81** | 0,81 | 0,40 |
| $F_3$ Laws $B_{33}$ | I | 0,93 | 0,92 | **0,98** | 0,92 | 0,91 |
| | R | 0,83 | **0,85** | **0,91** | 0,87 | **0,93** |
| $F_4$ Entropy | I | **0,82** | **0,89** | **0,97** | **0,88** | **0,90** |
| | R | **0,25** | **0,51** | 0,84 | **0,46** | 0,32 |

Invariance values are very high for these features. Hardest transformation: scaling



We now can see the results of Goshtasby's experiment with four selected features (in the book, he uses a much more extensive set of features, out of which we selected the invariance and repeatability values corresponding to four representative instances of features). The best features (globally speaking) seem to be F2 and F3. F1 also offers surprisingly good values save for repeatability to scaling and noise.

**Step by Step.** *Example 1)* Feature Image and CP extraction with Histogram Mean

1) Implementation of the feature function (if needed)

2) Selecting neighbourhood type and size

3) Obtaining the feature image using either *imfilter,* or *nlfilter,* depending on the linearity of the feature function.

4) Don't forget to normalize adequately the feature image if *imshow* is used for showing the results

5) CP extraction as extrema of the feature image (things to decide: how many CPs? How to rank the extrema? the N strongest? The evenly-spaced N?). Extract and store the (x,y) coordinates or Location values)



We give here an exmple (which will be run in the class) about how to implement feature detection and CP extraction manually. It will not be asked of you to be able to do this, but it might help you understand the general process involved.

- Tips for CP extraction with Matlab's built-in functions:

1) **Matlab features:** use the functions detectFeatures and extractFeatures. Call extractFeatures with the validPoints output. See how to use the selectstrongest parameter.

Example:
```
ptsOriginal = detectSURFFeatures(original);
extractFeatures(original, 'Method','SURF')
```

2) **Plot** the locations of the extracted CP along with your image.

3) Analyze the **distribution of locations** within the image, and which regions or objects tend to correspond to extracted CPs.



Here you can find some tips about the next class activity. Use them along with your blank code.

UNIVERSIDAD DE GRANADA

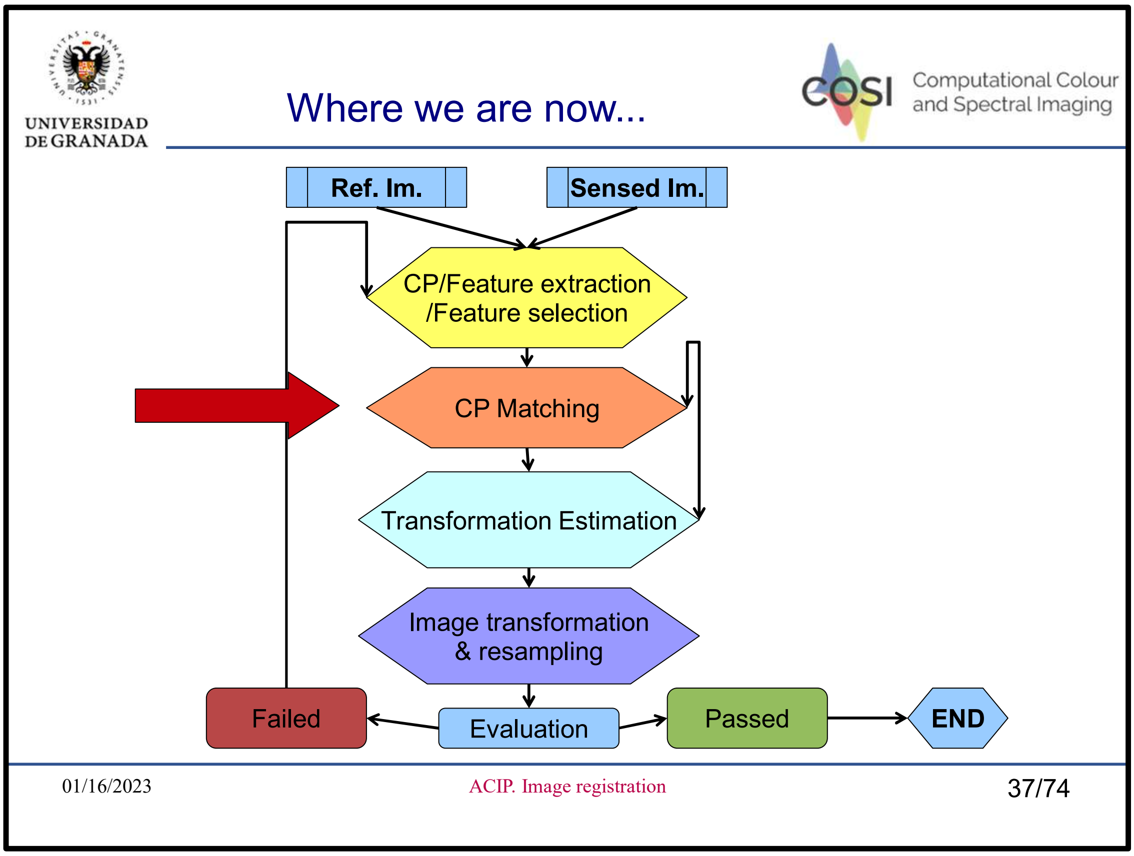
# Activity 2

COSI Computational Colour and Spectral Imaging

**Automatic CP extraction with Matlab feature functions.**
**4 class points.**

**What you need to do:**.
Complete the blank code if it helps, otherwise implement your own code.
**1)** Use an alternative feature already implemented in Matlab (SURF, BRISK, MSER, FAST, MinEigen, Harris, FAZE, ORB, see https://es.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html) to extract the CP from the image you used in activity 1. Take care to select a ood feature for the kind of transformation you implemented.
**2)** Now, use the transformed image from activity 1 and extract the CP from this image with the same Matlab feature.
**3)** Compare the two CP distributions in the original and transformed image.
Is the feature you have used sensitive to the changes introduced by the transform?
Write a small report with the discussion and results obtained, including a brief summary of the original reference proposing the feature.
DEADLINE: April 28th 14 h.

# Day 3




After considering some useful tools for CP and feature extraction, we are now in a position to describe some ways to perform the next step, which is finding the corresponding CPs in both reference and sensed image. If we use calibration images consisting of fixed grids (like checkerboards) for solving the registration problem in a particular capture device configuration, then this step might not be needed.

UNIVERSIDAD DE GRANADA

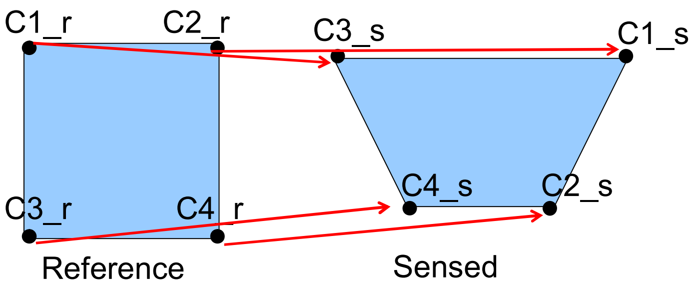
5. Control Point/Feature Matching

COSI Computational Colour and Spectral Imaging

## 5.1. **General problem statement**

→ Finding correspondences between two sets of CPs

→ Indexing between CPs or feature descriptors.




In case smoothing goes wrong:
- use transformation estimation to improve
- combine matches from different features

01/16/2023 ACIP. Image registration 38/74

In general, however, we will have two independent sets of control points, each one belonging to one of the images, and the order of the CPs will not be corresponding, because they were extracted and sorted separately and the two images might be very dissimilar. So our problem is indexing one set of control points to the other so that they form corresponding pairs. In the figure, we see this illustrated for a projective transformation. As we will see, the matching problem is very often imbricated into the finding transformation parameters problem, and both can be solved in one step for simple registration cases.

5.2. **Some common approaches to CP/Feature matching**

| **CP Matching**[[1]] | |
|---|---|
| **Random sample and consensus** | Axis of Minimum Inertia-based matching |
| Graph-based matching | Geometric invariance-based matching |
| **Feature-based matching** | Relaxation-based matching |
| Clustering-based matching | **Coarse-to-fine matching** |



There are many methods for solving the matching problem. We will briefly discuss the foundations of three of the most commonly used which are also representative of different strategies. More details about these algorithms and others not cited here can be found in [1].

5.2.1. **Random sample and consensus (RANSAC)** → for global transforms

→ Estimate global transformations (p parameters) for random subsets of q CPs taken from reference and sensed images. Only Location used.

$$q = \frac{p}{2}\, if\ p \text{ is } even$$

$$q = \frac{p+1}{2}\, if\ p \text{ is odd}$$

→ Transform the sensed image onto the reference image

→ Two CPs correspond if they meet a threshold criteria

$$\| f(P_i) - p_i \| < \varepsilon$$

→ The performance of the set: t (accuracy)

$$r = \frac{t}{n}$$

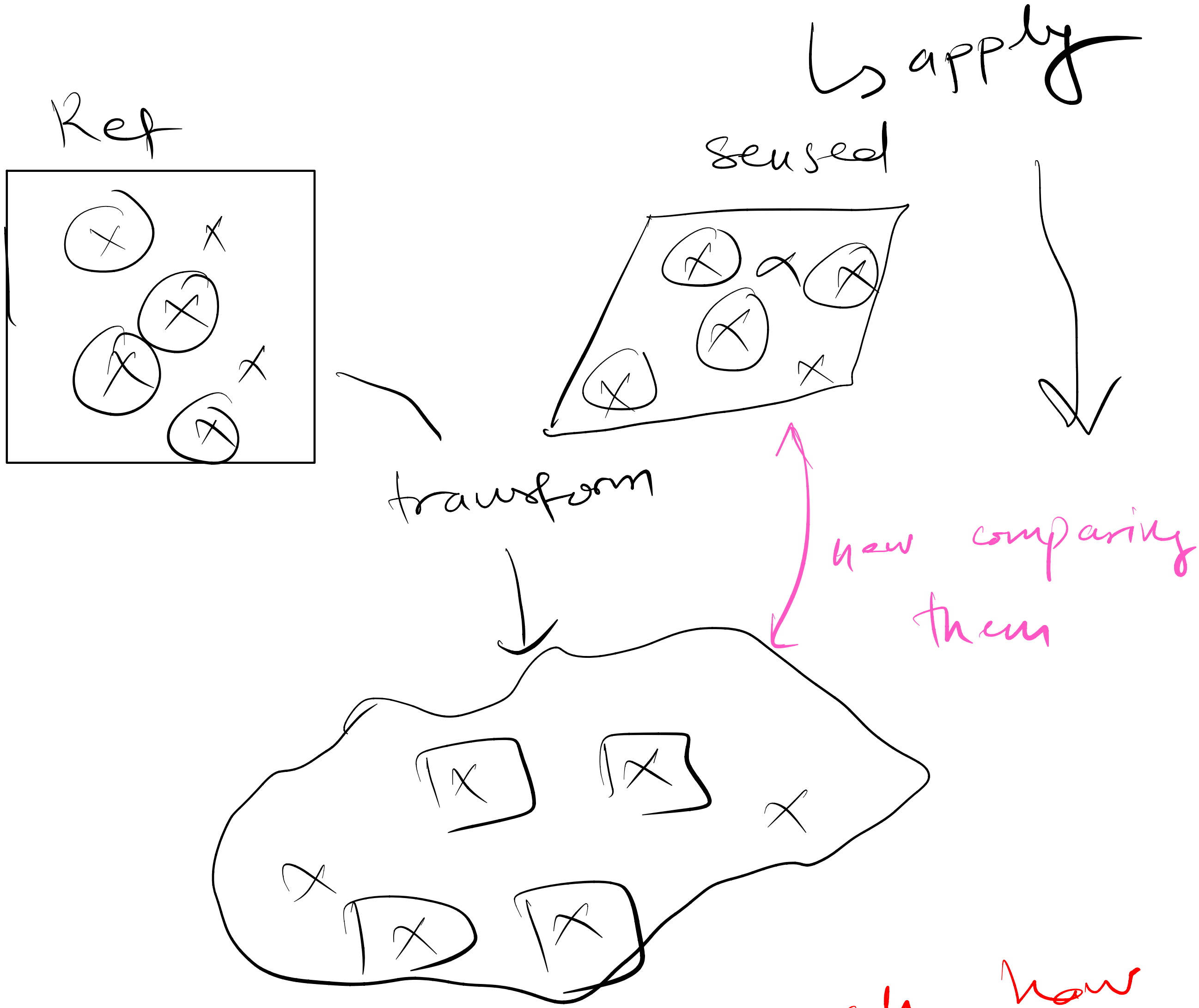
→ Algorithm parameters: ε, r (threshold for parameter set acceptability), N (maximum number of iterations)



RANSAC is maybe the most simple approach for solving the matching problem. The basic idea is to estimate the transformation parameters (assuming a known global transformation type) for many random subsets of CPs. Only the subsets that are really matching will provide good enough parameters, so when the sensed image is transformed, the transformed CPs will be reasonably close to the reference image CPs. The algorithm needs to fix three parameters: number of iterations N, threshold ratio for considering a random set acceptable (r) and threshold in accuracy for considering a pair of CPs as matched (ε). Once the matched set is found, then the transformation parameters are refined by least square estimation. This algorithm then provides not only the matching sets but also a reasonable estimation of the transformation parameters. It is a good example of how to use learning parameters to obtain some feedback into the matching process.

2 of CP subsets → estimate $tr_i$ → apply it to ref CP set

4 pairs → $tr_i$ ← projective




compute distances, check how many are below threshold.

$$r_i = \frac{t}{n}$$

UNIVERSIDAD DE GRANADA

5. Control Point/Feature Matching

COSI Computational Colour and Spectral Imaging

- RANSAC: toy example. r=0.8, N=1, ε=0.5 pixels

- Estimate ***Projective*** transform.
- 10 CPs in Original image, 8 CPs in test image.

Q1) Which is the value of the q parameter? 4 (4 pairs)

- Locations of first subset (CP1-CP4) and transformed first subset (CPt1-CPt4) after the first iteration

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| CP | (20,32) | (75,40) | (55,70) | (85,65) |
| CPt | (21.3, 31.5) | (40.2,-2.1) | (54.6, 3.3) | (4.4,25.6) |
| $\parallel f(P_i) - p_i \parallel$ | 1,39 | 54,87 | 66,7 | 89,71 |

Q2) Which is the value of r for this subset? 0 because none is below threshold (0.5)

Q3) Next step in the algorithm?

01/16/2023 ACIP. Image registration 41/74

We can see the RANSAC process in this toy example, to make more clear the different steps of the procedure. Matlab uses a variation of RANSAC called MSAC for feature-based matching within the function matchfeatures.

UNIVERSIDAD DE GRANADA

5. Control Point/Feature Matching

Computational Colour and Spectral Imaging

5.2.2. **Feature based matching**

→ Use feature values of CPs to estimate correspondence. Organize the corresponding candidates in a list ordered by increasing distance

→ In the computation of the list, the CPs are organized using graph theory (MST or full set)

Image from Google

→ The best subset of corresponding candidates used to estimate the transformation, and the quality of the matching is evaluated

01/16/2023 ACIP. Image registration 42/74

In this instance of feature-based matching, the feature vectors are used to compute distances between one subset of reference CPs and some sensed CPs, and the lists are ordered and updated for different subsets in the sensed image. To make the process faster, graph theory can be used to find related CP subsets in each image, by building the minimum spanning tree (MST). If this approach does not yield good results, then the full set of selected CPs can be used. As it happened with RANSAC, after the matching the parameters are estimated and the quality of the matching is computed based on the accuracy of the transformed CP set. In the figure, we see an example of applying feature-based matching to an image with local distortions.

HW

* Minimum Spanning Tree.

To be able to save computation

* To estimate transform so RANSAC can use it we use MST.

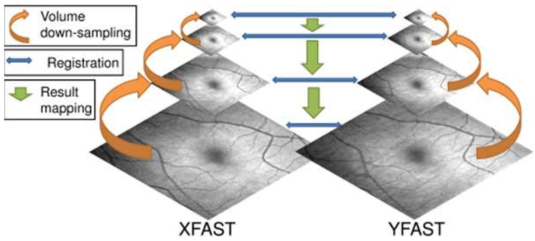

• and then estimate transform

more sophisticated than picking up random things.

5.2.3. **Coarse-to-fine matching**

→ Limitations of previous algorithms: work well mostly for global geometric transformations, slow if m (nr. of CPs) is high

→ Basic idea: reduce resolution of images. In low resolution, use affine transform. Find parameters, transform and increase resolution




→ Also known as pyramidal approach.

http://www.opticsinfobase.org/boe/fulltext.cfm?uri=boe-3-6-1182&id=233031



A quite interesting approach especially if the images have high spatial resolution and contain huge sets of control points, is the coarse-to-fine matching. We reduce the resolution of the images until only a few CPs are found. In this very low resolution, very likely an affine transformation will be enough to solve the registration problem. We do it and then increase the spatial resolution (usually in steps of 2) and apply a suitable technique to find again the parameters (which is easier because of the previous step). Then, we go on until the problem is solved for the original image resolution (again, simplified by the previous steps). Sometimes, tesellation is used to find corresponding regions instead of corresponding points. The pyramidal approach can also be used to speed up other registration approaches.

UNIVERSIDAD DE GRANADA

# Activity 3

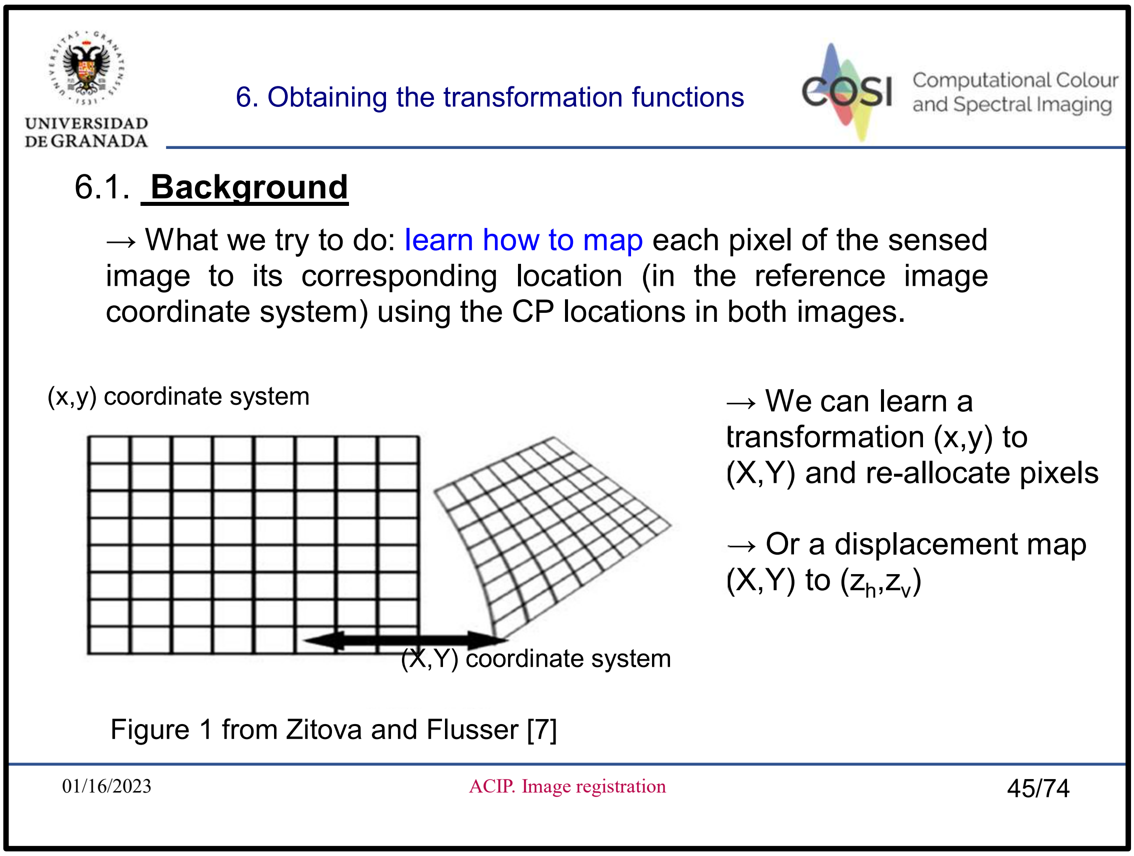
Computational Colour and Spectral Imaging

**Activity 3. 2 class points.**
Using feature-based matching methods in Matlab.

Start from the original and transformed coin images and your selected Matlab feature from **activity 2**. Match the extracted CP in both images using the *matchFeatures* function (based on comparing Feature vectors using a pre-defined distance and threshold), and plot all matches using the *showMatchedFeatures* function. Analyze how to tune adequately the matching function in Matlab to obtain a better result, if you are not satisfied with the results of the matching process.
You can complete the blank code if it makes it easier for you.
Send the code and a discussion of results (with figures) to the instructor.
DEADLINE: May 2nd 14 h

01/16/2023 ACIP. Image registration 44/74

In this activity, we will see how to perform feature-based matching in Matlab. The matching algorithm is pretty simple and it is based in evaluating the distances between feature vectors between the reference and sensed image CP sets. This basic procedure can be further refined if we implement the transformation estimation step and use the estimated transform to see if the CP sets are reasonably close. But we will see how to implement this step in the next class activity.




In case we did not obtain the transformation parameters in the previous steps, or in case we just obtained a rough estimation and we want to refine it, there are several ways to learn the correct values for these parameters, and some of them will be presented in this section. The optimal method to apply in terms of accuracy and efficiency will depend very much on the kind of transformation (global, local) that will solve the registration problem. We call lowercase x,y the coordinate system of the reference image, and X,Y the coordinate system of the sensed image. We will assume as a starting point that we want to learn a transformation from the reference to the sensed image, so that we can locate the pixel in the sensed image that must be allocated to each pixel position of the reference image, to register both correctly.

## 6.1. **Background**

→ Interpolation vs approximation

$F = f(x, y)$ Where F=X or F=Y, for interpolation

$F \approx f(x, y)$ For approximation


http://www.keremcaliskan.com/wp-content/uploads/2009/06/image082.png



The transfomation that relates both images can be obtained by two main procedures: interpolation from the data available (obtained in the CP positions) or approximation, especially interesting if we are not very sure about the accuracy of our CP locations. In the figure, we can see an illustration about the difference between interpolation and approximation: in interpolation, we will obtain a curve that passes effectively through all the measured points. In approximation, we can obtain a curve that is "acceptably close" to the measured points, but does not yield their exact values at their corresponding positions.

6.1. **Background**

| TF learning methods[1] | |
|---|---|
| **Robust estimators (global classic transformations or at least transformations of known functional form)** | **Adaptive transformation functions (transformation can vary locally)** |
| **Ordinary Least Squares variations: WLS, LMS, LTS** | Multiquadrics (standard or compactly supported) |
| Other robust estimators: ML, RM, Rank, Scale | Surface splines (TPS, B-splines) |
| | Moving Least Squares |
| | **Piecewise polynomials** |
| | Parametric Shepard Interpolation |



We can roughly classify the procedures for learning the transformation function into robust estimators (commonly used for global transformations, although not necessarily) and adaptive methods, which are more suitable for local distortion patterns. The most basic estimation method for adjusting the parameters is Ordinary Least Squares, and there are some variations that can be introduced in this method to get more resistance to outliers, like weighted LS, least Median squares, and Least Trimmed squares. There are also other methods not based on Least Squares that can be used for global transformation functions like Maximum Likelihood, Repeated Median, Rank or Scale estimators. We show as well in the table some representative methods corresponding to the adaptive group, which essentially covers different forms of approximating or interpolating functions that can vary locally.

**6.2. OLS estimation for global transformations of known functional form**

→ In OLS, we solve for the transformation parameters by minimizing the cost function of squared residuals:

$F = x^t a$ $\quad x = \{x_{1,} \dots x_m\}$ $\quad x_i$ , functions of (x,y)

$a = \{a_{1,} \dots a_m\}$ $\quad a_i$ , parameters

$$R = \sum_{i=1}^{n} (X_i - F_X)^2 + (Y_i - F_Y)^2$$

Very sensitive to outliers (incorrect correspondences)

→ Normal solution: $A = (XX^t)^{-1} X^t F$

X and F are obtained from the CPs



We can describe the parameter estimation process as a linear regression problem if we express the transformation function as a linear combination of the components of a vector of functions of (x,y) coordinates. The simplest solution to this problem would be ordinary least squares (OLS) regression, which uses a cost function which is very sensitive to outliers, since all the CP data contribute equally to the solution with no restrictions.

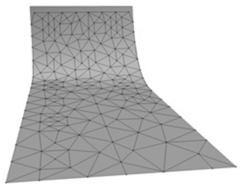
It can well be that for the particular problem we are trying to solve, we cannot ensure that CP locations are accurate enough. In that case, one possible alternative to OLS would be to use variations of this method which are designed to be less sensitive to outliers, like weighted least squares (WLS), or least median squares (LMS).

Also, we can turn to other standard regression solving approaches that use different cost functions, based on probability distributions or rank ordering of the data.

## 6.3. Adaptive estimation

### 6.3.1. Piecewise polynomials

→ Basic idea: triangulate reference and sensed sets, find corresponding triangles, and fit a polynomial transformation using the vertex points



Delaunay triangulation (image from Google)

→ The triangulation can be done as well using the intensity or feature values (3D triangulation)

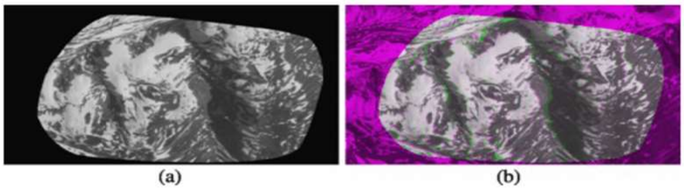



Figure 9.13 of Goshtasby [1], Loop triangulation



Other alternative to consider whenever there are local distortions is the piecewise polynomial approach. In this method, the images are triangulated (either in 2D or 3D using the intensity or feature information) and a polynomial regression problem is solved for each triangle, imposing additional boundary conditions to ensure that the different polynomials are stitched seamlessly. The results tend to restrict the area that is actually registered, and depend on the accuracy of the triangulation technique used. In the pictures shown, the Loop triangulation was applied. When the polynomials that are locally obtained are of degree one, the method is called "piecewise linear". This last method is implemented in Matlab's image processing toolbox as an instance of a local model.

## Quick question



① local transform to match faces b/w two people.

UNIVERSIDAD DE GRANADA

# Activity 4

COSI Computational Colour and Spectral Imaging

**Activity 4. 3 class points. Transformation estimation.**

Estimate the transform using corresponding pairs of CP, and use the transform to refine the matching process.

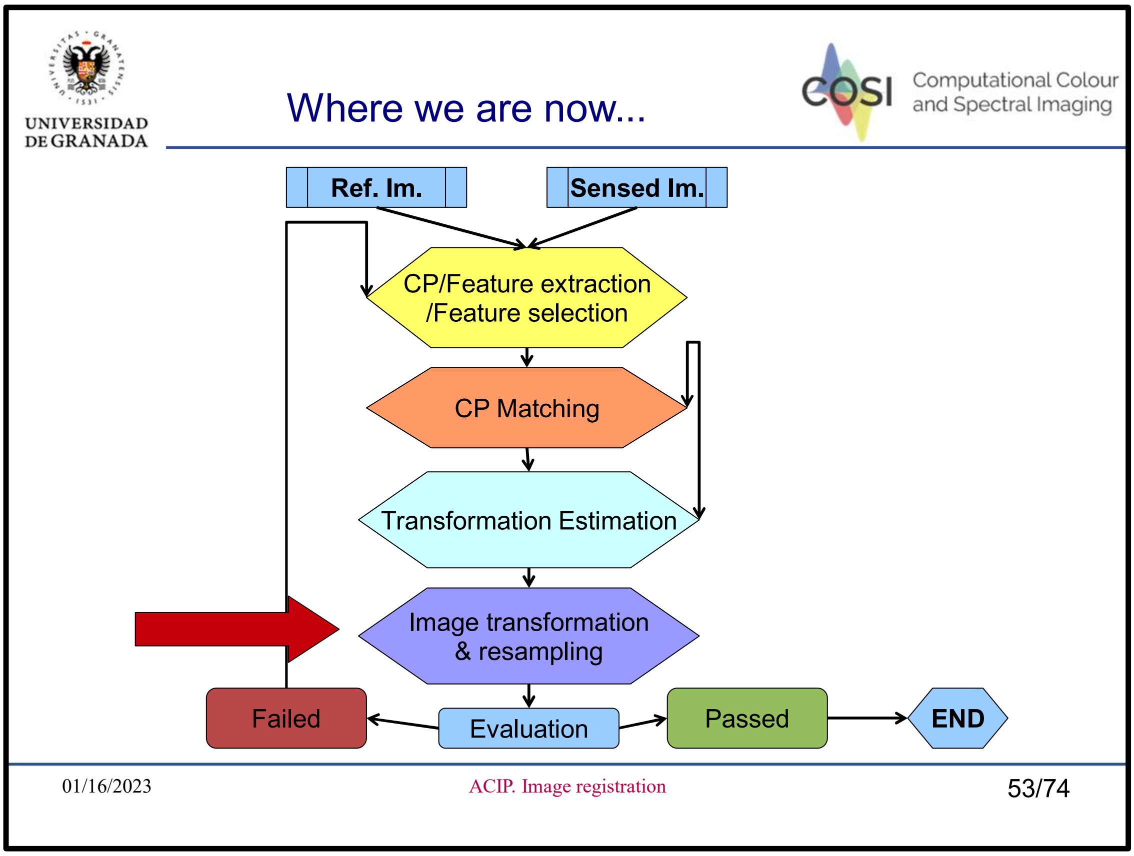
Complete the blank code (if it is easier for you this way) to estimate the transform that you applied in Activity 1, from pairs of matched CP in the original and transformed image (which you got from activity 3). Register the two images and build the overlap plot with the *imshowpair* function. Discuss the results and show them in the discussion. Feel free to try different approaches (changing features and matching parameters using the previous activities, changing the estimation function parameters) if you find unsatisfactory results.
DEADLINE: May 2nd 14 h

Sensed to Reference → Matlab



In this activity, we finally get to the point of being able to estimate the initial transform between reference and sensed images, and even using this estimation process to further refine the matching procedure and getting rid of the outliers in the matched CP sets. We also see some options for changing the parameters in the transformation estimation function.

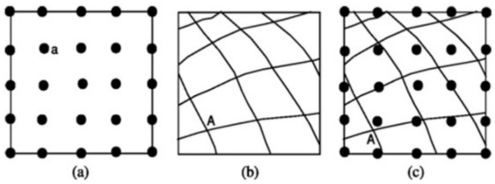
# Day 4




We have arrived now to the final stage of the registration work-flow previous to the evaluation of results: the application of the modelled transformation to the image that will be registered, and the resampling of the intensity values obtained.

### 7.1. **Why resampling?**

→ Case 1: we learn the function to obtain (X,Y) from (x,y)




→ We might obtain (X,Y) values that are not integers

Figure 10.1 of Goshtasby [1]

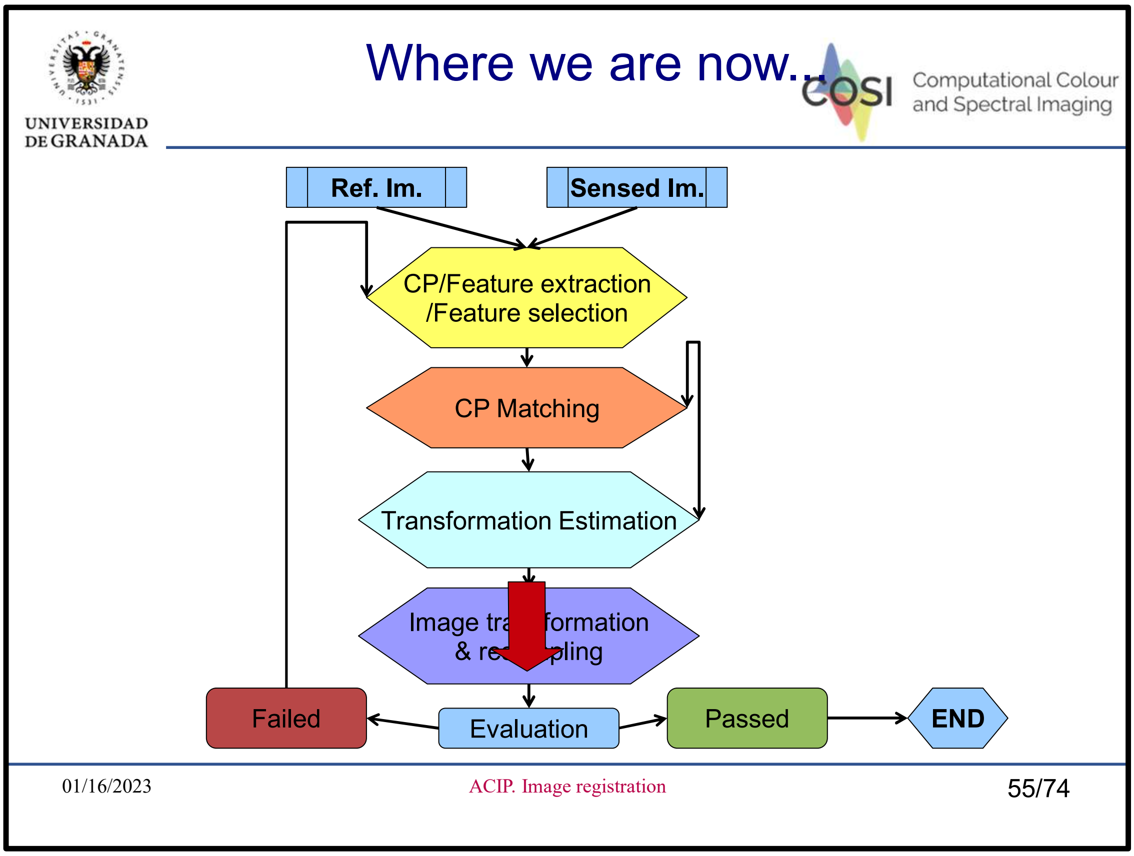
→ Case 2: we learn (zh, zv) to move (X,Y) into the reference image

We might obtain (zh,zv) values that are not integers

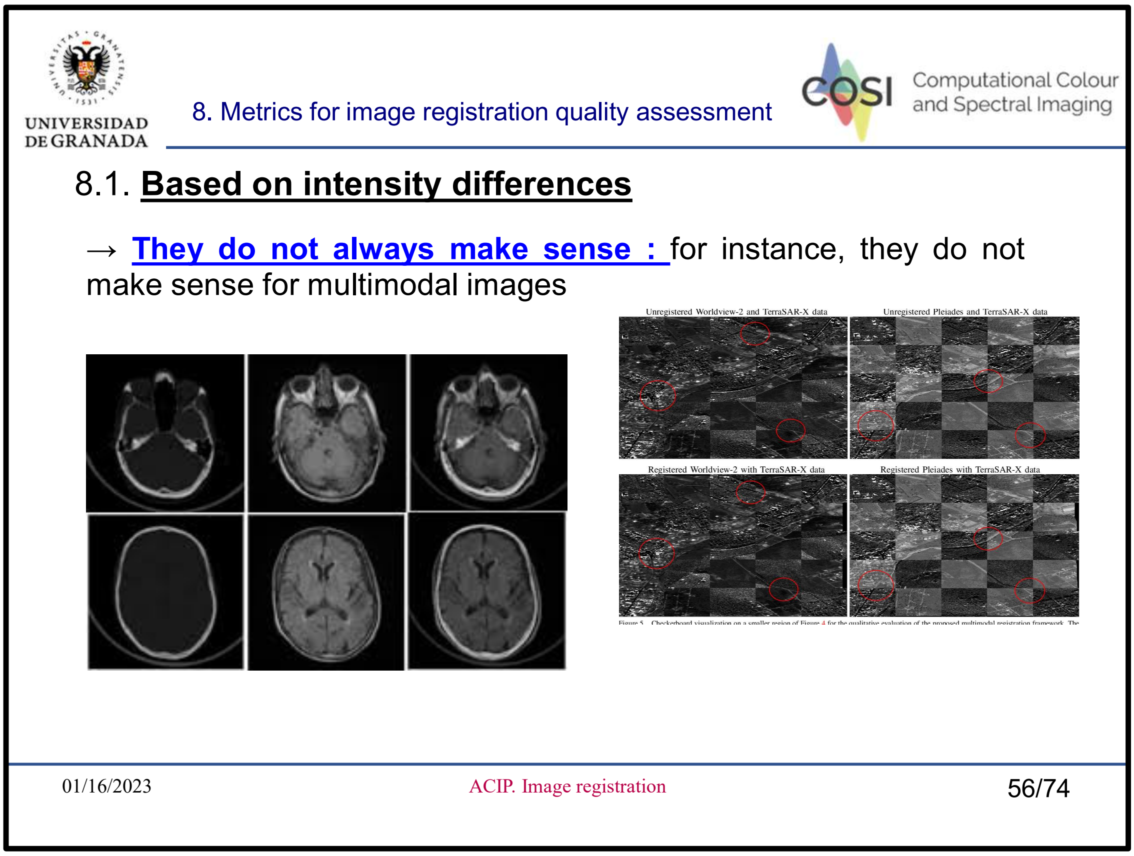
→ Methods: NN, bilinear interpolation, cubic interpolation



Let's first make clear why we need to resample the registered image. The main reason is that when we map the reference onto the sensed image (or else, when we learn the sensed image displacements on a pixel-by-pixel basis), it is very common that we don't obtain integer values for the (X,Y) or ($z_h$, $z_v$) functions. So in the end we come up with a registered image for which we know the intensities only at non-integer positions. In order to evaluate our registration results, we need to use the same coordinate grid for reference and sensed images, and so we need to obtain the registered image intensity values for this common grid. We see this idea illustrated in the figure. There are different methods for resampling the image, like Nearest neighbour (the simples but least accuarte one), bilinear interpolation (mostly the method of choice) and cubic interpolation (in case we need to place a special emphasis on accuracy at the cost of adding computaiton time).




We have arrived now to the final stage of the registration work-flow, which is the evaluation of the quality of the registration solution found with the previous steps.




The metrics based on intensity differences will not be used for multimodal image registration, because of the intrinsic intensity differences between reference and sensed images, which we can appreciate in the two instances selected in the figures. These differences would cause the metric to return a high value even for perfectly registered image pairs.

We can use many different metrics (all based on finding image differences) to assess registration quality. The most common metric is not quantitative but qualitative: visual comparison. Among the quantitative metrics, it is common to try to evaluate the registration quality in terms of intensity differences between the registered and reference images. However, note that this metric does not make sense if the two images are multimodal.

UNIVERSIDAD DE GRANADA

8. Metrics for image registration quality assessment

COSI Computational Colour and Spectral Imaging

8.1. **Based on intensity differences**

→ Example: **RMSE,** scaled between 0 and 255 for 8-bit images. Perfect result: zero RMSE.

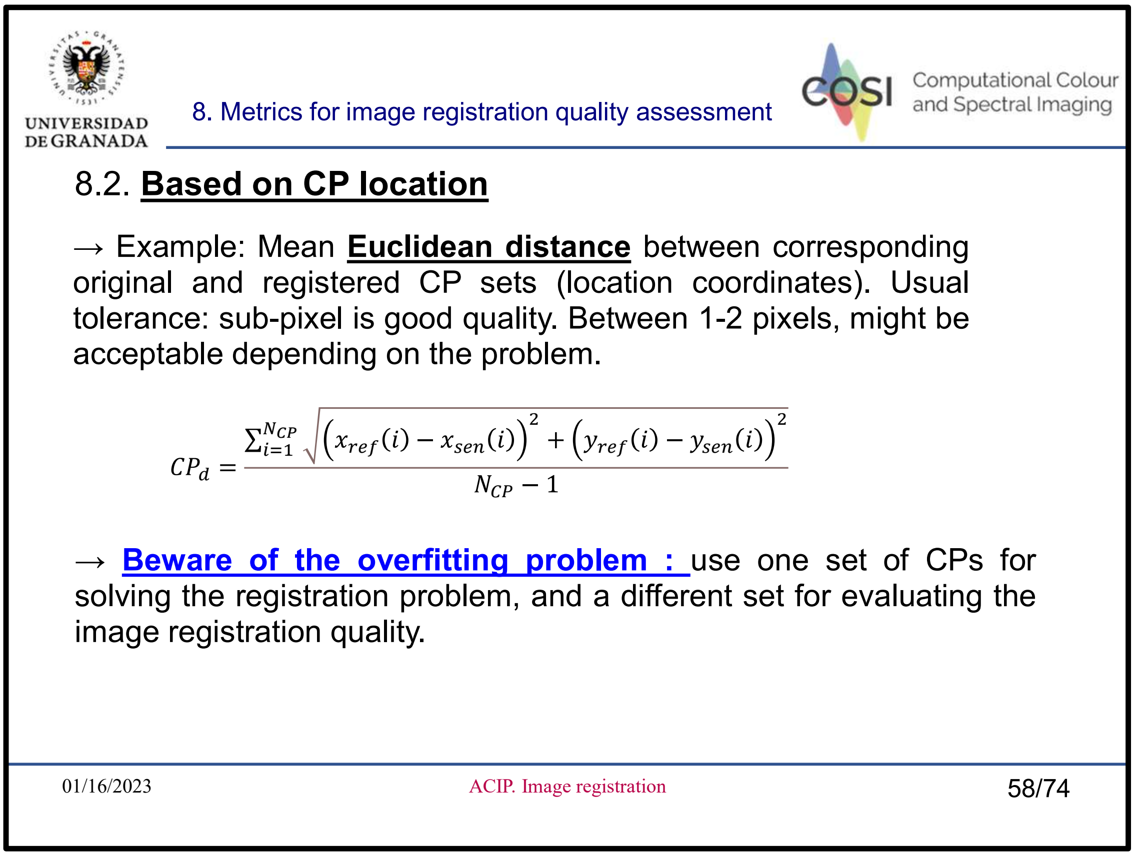
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_{pix}} \left(I_{ref}(i) - I_{sen}(i)\right)^2}{N_{pix} - 1}}$$

→ Normalized to the max intensity value in the reference image or at least to $2^N$ where N is the number of bits of the image.

→ When the two images have only one portion in common, the RMSE needs to be applied only within that region!

01/16/2023 ACIP. Image registration 57/74

When we have monomodal images and we want to assess the registration quality based on intensity differences, one common metric is the Root Mean Square Error, which accumulates the squared differences in intensity pixel by pixel, and divides the sum by the number of pixels minus one; then, it computes the square root of this quantity. For an 8-bit image, the maximum value would be 255. Sometimes, it is convenient to normalize the RMSE  so that its maximum value is one, to allow for a better comparison between different experiments. The RMSE must always be computed using the set of pixels that the reference and registered images have in common, in case they cover partially different regions of the scene, which is many times the case for monomodal images.




However, registration quality metrics based on image intensity tend to accumulate the errors resulting from the application of the transformation to the sensed or registered image (for instance, a slight blur might result in intensity differences that are not a direct consequence of errors in the registration algorithm, but just of the fact that we need to apply a spatial transform to the image).
So overall the most reliable metrics tend to be the ones based on CP location error, for instance the average Euclidean distance in two independently extracted sets of CPs from the registered and sensed images. It is important to extract anew the CP for evaluation, since the

algorithm might have a tendency to overfit and give very low difference in location for the CP sets used to estimate the transform.

8.2. **Based on CP location**

→ Example: Mean **Residual error** between corresponding original and registered CP sets.

$$CP_{resx} = \frac{\sum_{i=1}^{N_{CP}} |x_{ref}(i) - x_{sen}(i)|}{N_{CP} - 1}$$

$$CP_{resy} = \frac{\sum_{i=1}^{N_{CP}} |y_{ref}(i) - y_{sen}(i)|}{N_{CP} - 1}$$

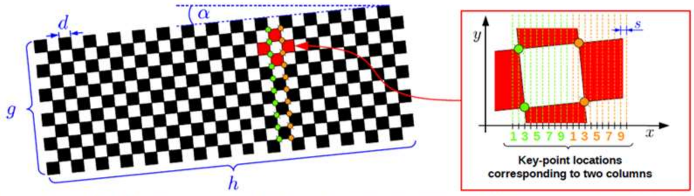


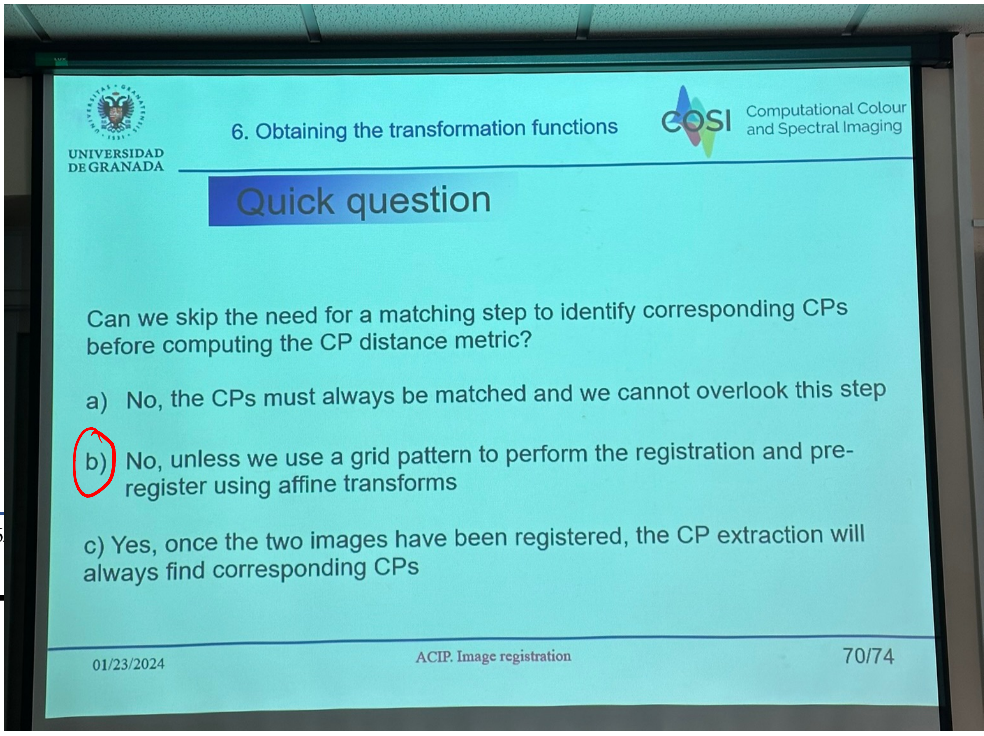

Fig 3: Design of a checkerboard pattern for geometrical calibration. Illustrated in the red box is the resulting horizontal key-point spacing $s$, scalars $h$ and $g$ are the number of black and white checkerboard pattern in horizontal and vertical direction respectively, $d$ is the side length of one square.

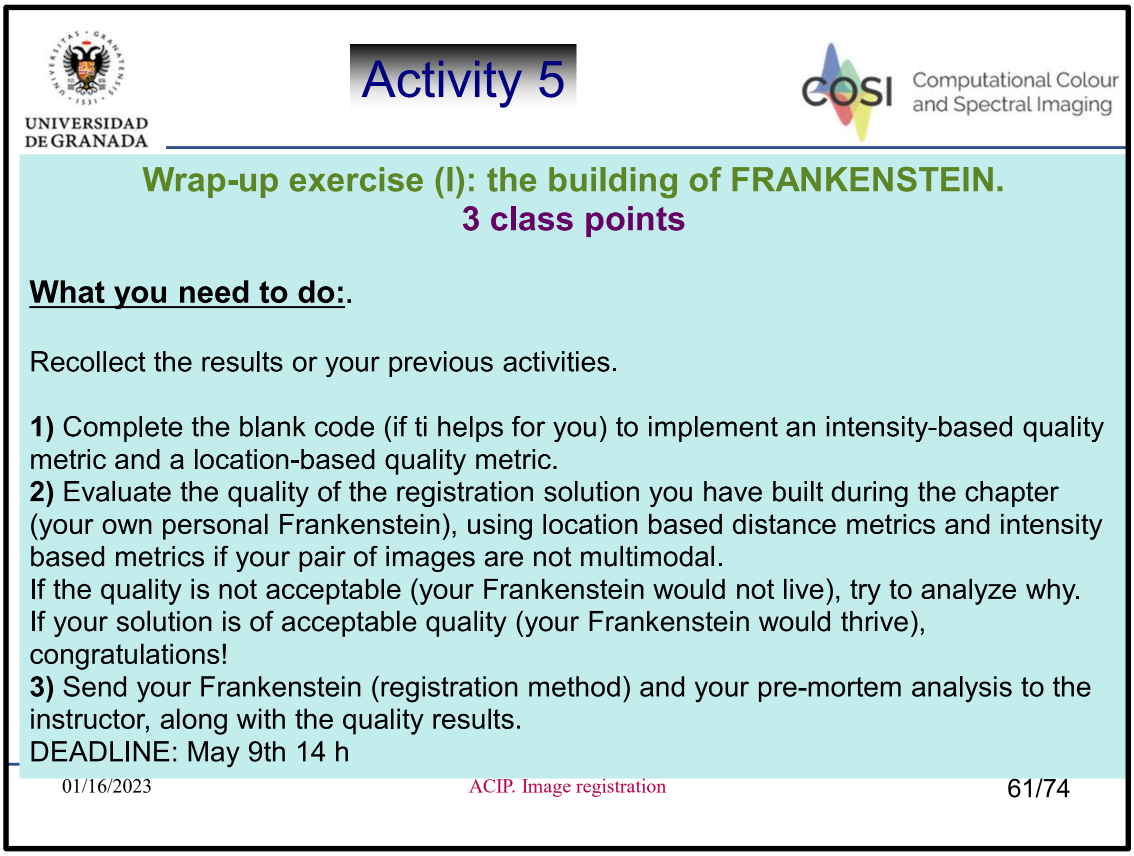
Figure from Eckhard et al. 2017 [12]



Other possibility is to compute the mean residual error (mean of the distribution of differences in x or y coordinates between the CP extracted from the reference and the registered images). This is the metric used in one of the registration publications of the Color Imaging Lab, based on the work of Timo and Jia Eckhard, former CIMET students and research associates in our group. For this work, we used a specifically designed calibration chart for solving the registration problem in a line-scan multispectral capture device. The algortihm was based on local transforms (B-splines) and resulted in a maximum residual error of around 0.32 pixels, with an average residual error of less than 0.1 pixels.

# Quick question






In this class activity, we will implement the last step of the registration algorithm, and evaluate our results.

HW: Go for a bit harder transformation than rotation.

### Representative Registration methods[7,1]

Medical Images

| Area-based methods (only for rigid transformations or images with not well defined objects; no feature extraction step) | Feature-based methods (global or local transformations using CP locations or features in the matching step) |
|---|---|
| Correlation-based For Global Transforms | Multiresolution techniques |
| **Phase correlation (FFT-based)** | Optimization-based registration |
| **MI-based** | **Adaptive registration** |
| Principal axes-based | **Deep Learning registration** |

→ Tentative classification based on most common uses. The cells are not watertight containers, strategies can be combined



In this final section about the global registration framework, we will describe the basic ideas of some representative registration methods. In a first tentative classification, we can distinguish between methods that skip the CP extraction and matching steps, called area methods, and methods that use CP locations or other features to aid in the registration (feature-based methods). We aim to provide a global vision about several representative techniques, but of course we cannot cover in the short time available the many instances of registration methods (or more generally speaking, registration strategies) developed so far, not even if we focus on the most recent tecniques.

So we will briefly review some classifcal instances of algorithms and others a bit more advanced.

$F_2(w_x, w_y) = e^{-i(w_x d_x + w_y d_y)} F_1(w_x, w_y)$

$F_2^*(w_x, w_y) = e^{i(w_x d_x + w_y d_y)} F_1^*(w_x, w_y)$

UNIVERSIDAD DE GRANADA

9. Some representative image registration methods

COSI Computational Colour and Spectral Imaging

9.1. **Area-based methods**

9.1.1. **Phase correlation registration[6]**

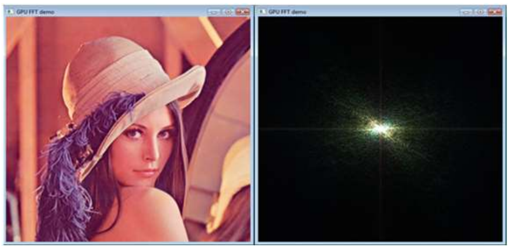
→ Valid for the simplest distortion (**translation**). Based on the properties of the FT

$$I_2(x, y) = I_1(x - d_x, y - d_y)$$

Interesting property of FT

$$F_2(\omega_x, \omega_y) = e^{(-i(\omega_x d_x + \omega_y d_y))} F_1(\omega_x, \omega_y)$$

$$\frac{F_2^c(\omega_x, \omega_y) F_1(\omega_x, \omega_y)}{| F_2^c(\omega_x, \omega_y) F_1(\omega_x, \omega_y) |} = e^{i(\omega_x d_x + \omega_y d_y)}$$



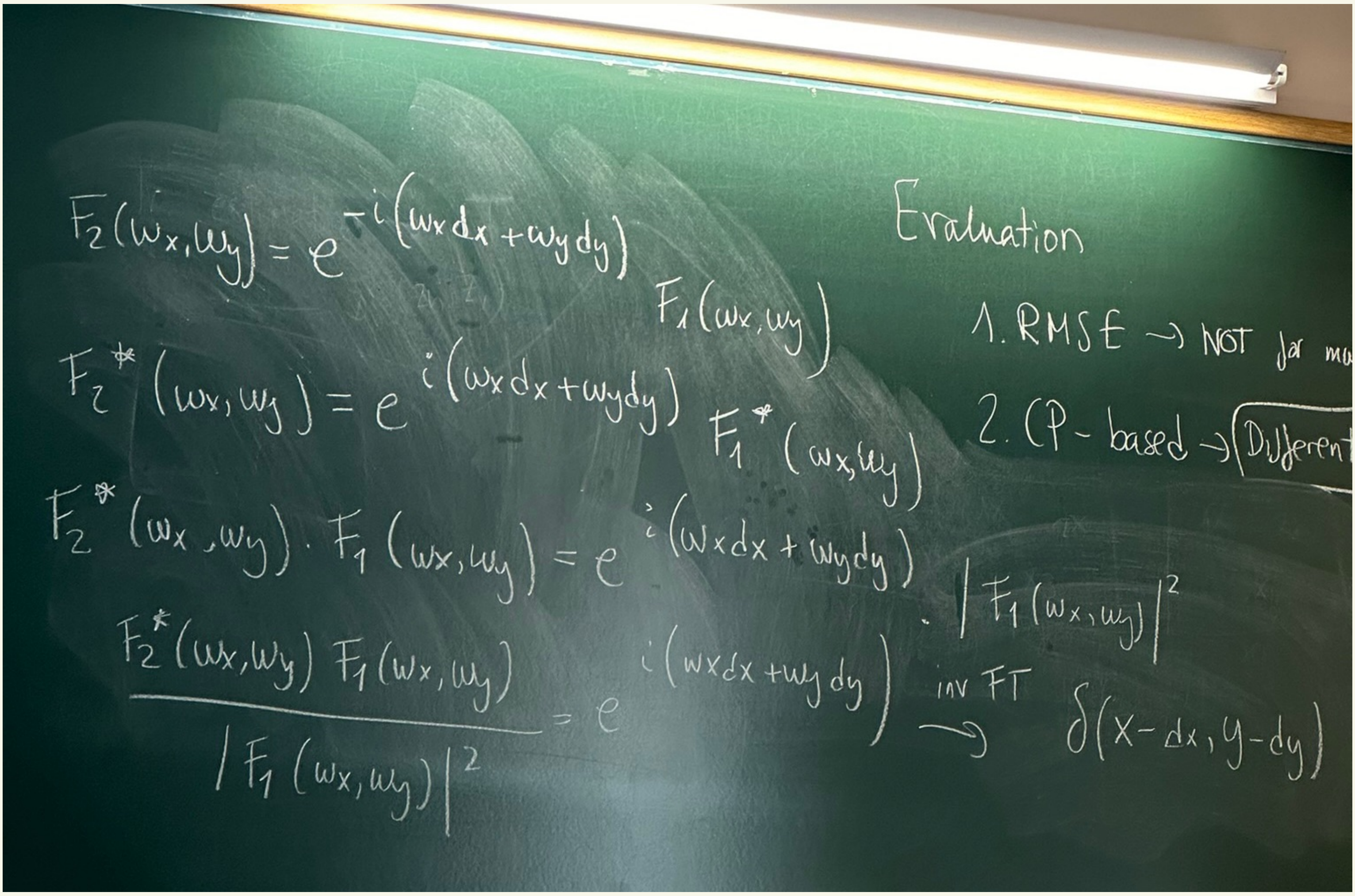
http://gpulab.compute.dtu.dk/research.html

→ The inverse FT of the phase correlation will be a delta centered at ($d_x$, $d_y$)

01/16/2023 ACIP. Image registration 63/74

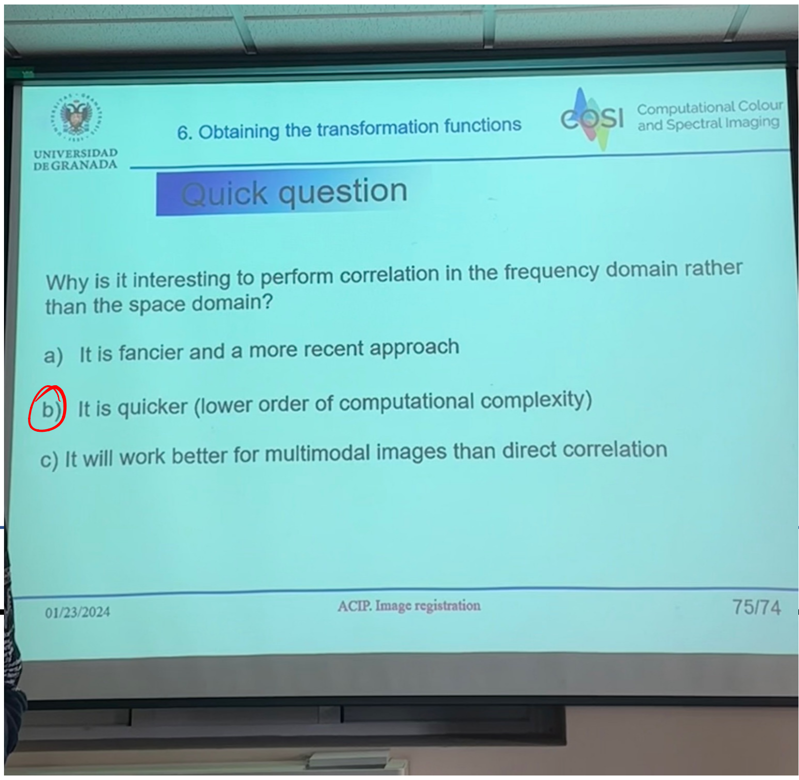
If we need to solve a simple translation, other alternative which is less dependent on intensity values is the phase correlation, which is based on the properties of the FT of an image when a translation is applied. The translation results in differences only in the phase of the FT, so by using multiplication by the complex conjugate of both images (normalized by the product of its modules), we get a complex exponential function. The IFT of this function will be a delta function centered at the location corresponding to the translation parameters (horizontal and vertical shifts). We will use phase correlation in the lab session.

# FT Properties:-




* equivalent for finding a correlation but in Fourier space
* very fast and efficient
* won't work for Multi Modal images
* difficult to find correlation in Multi Modal

## Quick question ?



## 9.1. **Area-based methods**

### 9.1.2. **Mutual Information registration**[7]

→ Especially indicated for multimodal images. Computes MI features over the whole image and finds the transformation by optimization

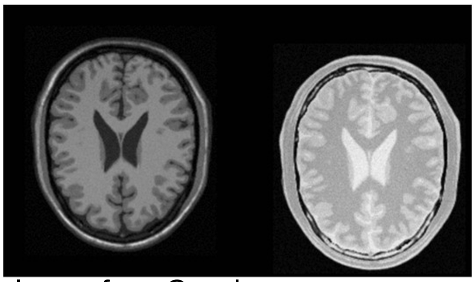


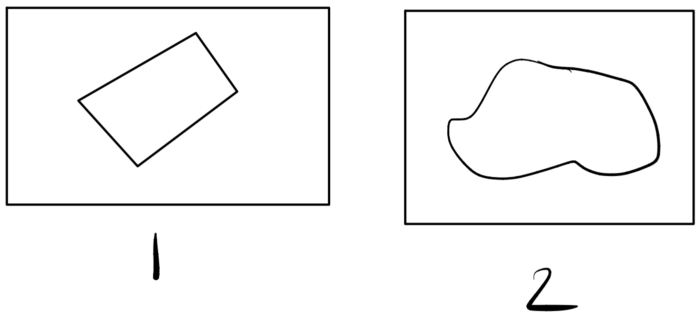
Image from Google

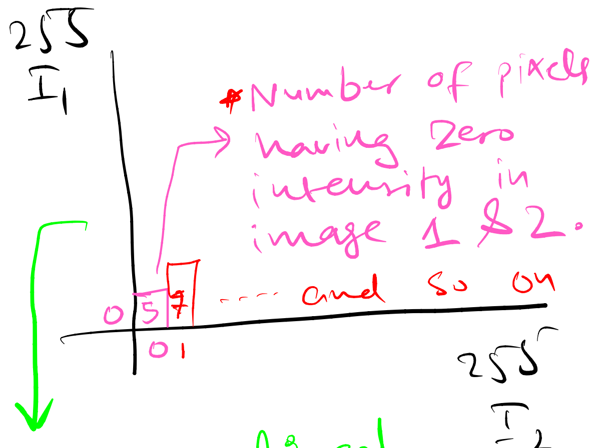
$$M_I = \sum_{i=1}^{255}\sum_{j=1}^{255} P_{12}(i,j)\log_2 \frac{P_{12}(i,j)}{P_1(i)P_2(j)}$$



Mutual information-based area methods were first introduced in the field of medical imaging registration because they are quite suitable for dealing with multimodal images. The basic idea of these methods is computing MI (mutual information) or E (entropy), or else any information theoretic feature for transformed sensed images, and finding the set of parameters that result in maximizing MI or minimizing cross Entropy by some optimization procedure. The Joint Probabilitty Distribution Function (JPDF or $P_{12}$ in the slide) are estimated directly from the reference and sensed images, and then used to compute the MI or E. The JPDF is a 2D histogram that shows the probability of a pixel with intensity value i for image 1 and intensity value 2 for image 2 to occur.

Joint histogram → 2D function



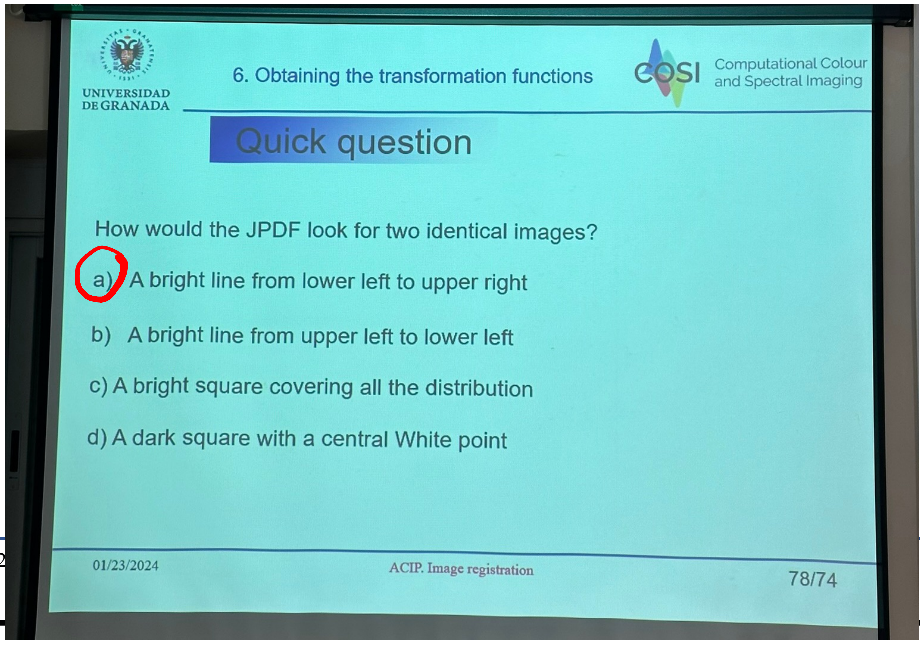




higher, if common

once normalized this would be $\boxed{P_{12}(i,j)}$

Joint probability

# Quick question



→ Lets go local now

UNIVERSIDAD DE GRANADA

9. Some representative image registration methods

COSI Computational Colour and Spectral Imaging

9.2. **Feature- based methods**

9.2.1. **Adaptive registration[1]**

→ Use a sequence of increasingly powerful but decreasingly efficient tools for registration

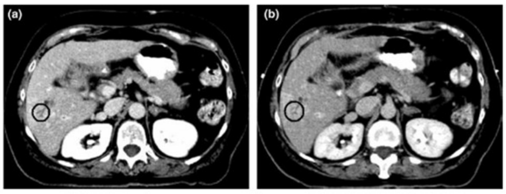



Figure 6 from Park et al. 2004[11]

→ E.g initial registration with few CPs and TPS. Evaluate error locally, increase the number of CPs, use more sophisticated methods

→ The different TFs must be joined seamlessly by imposing adequate boundary conditions

01/16/2023 ACIP. Image registration 67/74

An interesting general strategy that can be applied for challenging problems is the adaptive registration. It involves the use of a sequence of increasingly complex methods to progress in the registration solution. The registration error is evaluated locally, and more CPs are introduced in the areas which result in higher errors. After, all the local transforms must be joined with care to ensure that there are not any discontinuities created.

UNIVERSIDAD DE GRANADA

## 9. Some representative image registration methods

Computational Colour and Spectral Imaging

· more CPs here

increasing

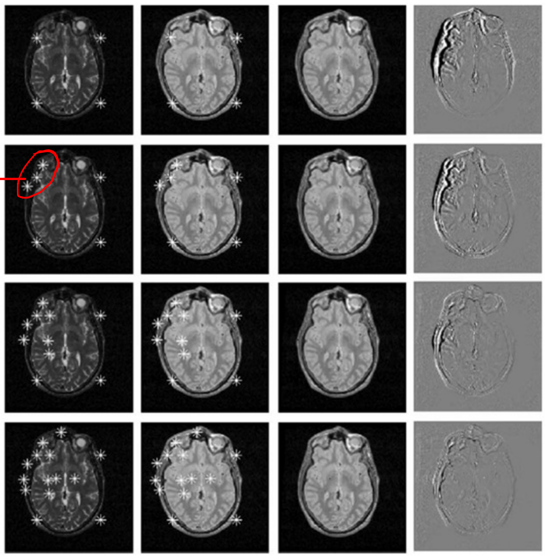


Figure 5 from Park et al. 2004[11]

**Do** affine registration
**Initialize** with largest grid and largest scale*
**For** I = large scale to small scale
  **Do**
    **Identify** location with maximum mismatch under the minimum distance constraint
    **Add** one control point at the identified location
    **Optimize** control points to maximize global MI
  **While** global MI increase > threshold
**End For**
*Scale parameters are the subblock size and the minimum distance constraint.

Fig. 1. Overall algorithm.

$$M = 1 - \frac{\mathrm{MI}(a,b)}{\min(H(a), H(b))}.$$



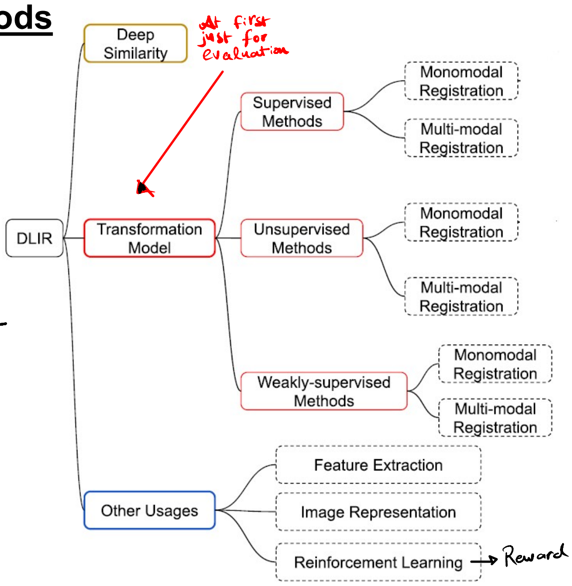
We can see an instance of an adaptive registration method in this figure. At each iteration, observe how the CP number in the areas with highest intensity difference is increased, and how the final registration quality gets progressively improved.

## 9.2. Feature- based methods

### 9.2.2. Deep Learning registration (DLIR) [13]

Important:-
→ cost functions
→ Defining the Net

• less costing to extract CP
• mostly remote sensing and medical imaging -



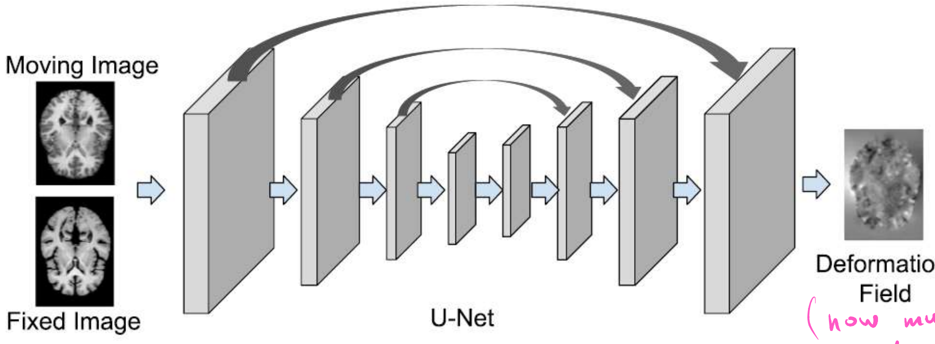

→ Bird'eye View [13]



Deep Learning based techniques have been increasingly used in the domain of image registration since 2015. They initially were auxiliary tools in conventional algorithms (Deep similarity: learning how to find a suitable metric to compare the reference and sensed images). The main usage of DL has been as part of the transformation models (in a wide sense, i.e. how to directly solve the registration problem using either know pairs of reference image and transformation (supervised methods), or using only the reference and sensed images as input to try to learn the transformation from them (unsupervised methods) or else using some information about CP or features in both images as aid for the transformation estimation (weakly supervised methods). Also, DL has been used for other purposes like learning useful features for a conventional registration algorithm, or as part of a chain of consecutive registration steps in a mixture of conventional and DL models.

9.2. **Feature- based methods**

9.2.2. **Deep Learning registration (DLIR)** [13]




(how much I need to move to match the image)

→ Example 1: encoder-decoder

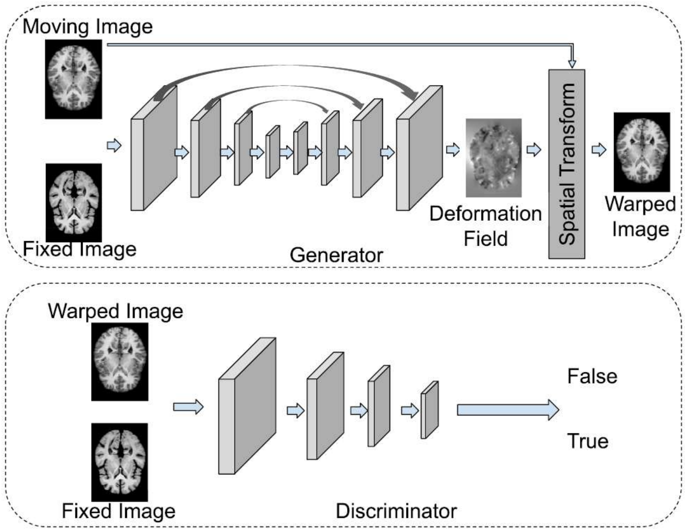


One of the least sophisticated approaches of Transformation Model DLIR would be to use an encoder-decoder structure (mostly U-Net architectures have been used, although there are other possibilities) to try to learn the Deformation Field from the Fixed (reference) and Moving (sensed) images. The cost function can be related to similarity metrics between the registered image and the reference image, or else (if supervised learning is used) with metrics that measure the similarity between model parameters.

∘ large field



9.2.2. **Deep Learning registration (DLIR)** [13]




Network 1: Try to solve Registration problem

→ Example 2: GAN

Network 2: Fooling Network 1

Cost and Rewards



Other possibility is to use Generative Adversarial Networks, for instance as shown in the figure: the generator tries to solve the registration problem by learning the deformation field from the reference and sensed images, and generates a solution (warped image). The Discriminator network learns to effectively distinguish between registered images and original images, so that the solutions that are not good enough will get “caught” in the discriminator and then discarded. The two networks compete with each other to obtain the best global result.

UNIVERSIDAD DE GRANADA

# Activity 6

COSI Computational Colour and Spectral Imaging

## Wrap-up exercise (II): the registration DISSECTION OF A FROG.

**3 class points**

**What you need to do:**

Search for a paper describing a registration method published in the last two years.

**Dissect it according to the taxonomy we have studied. Classify the steps that are present in the method, and the general strategy of the algorithm.**

**Be critical with your frog. Identify some shortbacks or potential limitations of the method.**

∘ Be critical with paper
→ 1000 words maximum

**Send to the instructor:**
**a) the reference**
**b) the main steps of your taxonomy (dissection results)**
**c) the most representative advantage and shortback of the method.**
**DEADLINE: May 16th 14 h.**



This final activity is better done as Homework. You will have access to a very recent paper describing a registration algorithm, and you will try to perfom a taxonomy of the method described according to the classification studied in this section and the general framework for image registration algorithms. Then, you will point out the most relevant advantages and dissadvantages of the method, showing what you have learnt during our lectures on image registration.

Finished with Theory!

UNIVERSIDAD DE GRANADA

## References and Links

COSI Computational Colour and Spectral Imaging


[1] A. Goshtasby. Advances in Computer Vision and Pattern Recognition. Image Registration, 2012. Principles, Tools and Methods. Springer. Latest Edition: 2017.

[2] J. Le Moigne et al. Image Registration for Remote Sensing. Cambridge University Press, 2012.

[3] http://es.mathworks.com/help/images/registering-multimodal-mri-images.html

[4] http://www.jacmp.org/index.php/jacmp/article/viewFile/3450/2219/30253

[5] http://parklab.johnshopkins.edu/Research.html

[6] L.G. Brown. A survey of image registration techniques. ACM Computing Surveys,Vol 24, 325-376 (1992).

[7] B. Zitova, J. Flusser. Image registration methods: a survey. Image and Vision Computing 21, 977-1000 (2003)

[8] L. Kitchen and A. Rosenfeld. Gray level corner detection. Technical Report #887, Computer Science Center, University of Maryland (1980)




Next lecture: Coding and activity.

<HW: Do beforehand>

# References and Links


[9] http://www.vlfeat.org/

[10] Eckhard T, Eckhard J, Valero EM, Nieves JL. Nonrigid registration with free-form deformation model of multilevel uniform cubic B-splines: application to image registration and distortion correction of spectral image cubes.
Appl Opt. 2014 Jun 10;53(17):3764-72. doi: 10.1364/AO.53.003764.

[11] Jia Eckhard, Timo Eckhard, Eva M. Valero, Juan Luis Nieves, and Estibaliz Garrote Contreras.
Outdoor scene reflectance measurements using a Bragg-grating-based hyperspectral imager.
Appl. Opt. Vol. 54, Issue 13, pp. D15-D24 (2015) •https://doi.org/10.1364/AO.54.000D15

[12] Eckhard T., Eckhard J., Valero E.M., Hernández-Andrés J.
Subpixel Accurate Calibration of Line-Scan Multi-Spectral Images Using a Polynomial or B-Spline based Registration Model.
Journal of Imaging Science and Technology, Volume 61, Number 3, May 2017, pp. 30503-1-30503-11(11)

[13] Chen, X., Diaz-Pinto, A., Ravikumar, N., & Frangi, A. F. (2021). Deep learning in medical image registration.
Progress in Biomedical Engineering, 3(1), 012003.




Image registration: sensed image - better to go with the higher resolution to compensate the loss in the processing step.