2D Alignment
Houston, December 2002

Pawel A. Penczek
The University of Texas – Houston Medical School,
Department of Biochemistry and Molecular Biology,
6431 Fannin, MSB6.218, Houston, TX 77030, USA.
phone: (713) 500-5416
fax: (713) 500-0652
Pawel.A.Penczek@uth.tmc.edu

The University of Texas
Health Science Center at Houston
Medical School

2D alignment methods

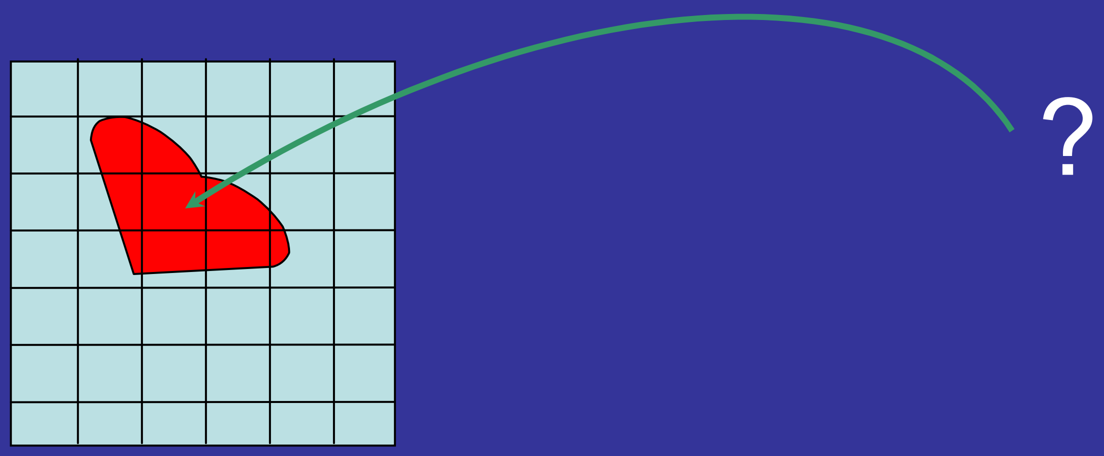
- 2D alignment is at the core of many data processing steps in single particle analysis. Therefore, it is essential that this step is done possibly accurately and rapidly.

- There is no efficient method that would allow simultaneous estimation of translation and rotation parameters between two 2D images. Therefore, various strategies are used depending on the required accuracy, noise level in the data, and the time constraints.

- Two major problems:
  (a) in general, a change of image orientation requires interpolation of the data, which adversely affects the quality of the image (resolution);
  (b) we have to make assumptions about the surroundings of the image outside of the image frame.

Interpolation of images

We tend to think about images as continuous objects, but in computer reality images are sampled on rectangular grids and represented as sets of numbers at discrete locations.



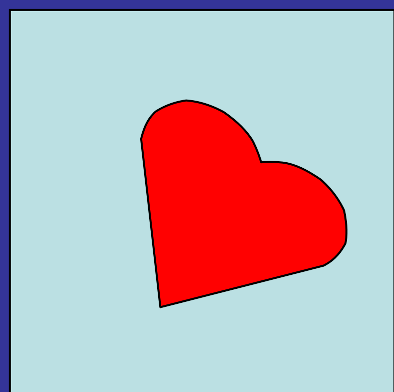
Interpolation errors

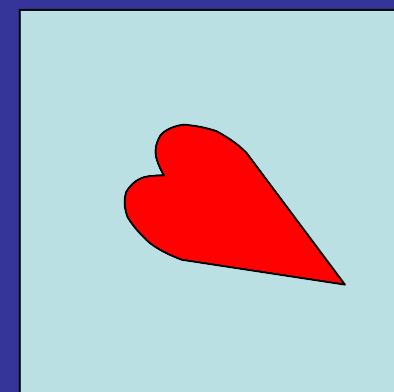
bilinear

quadratic

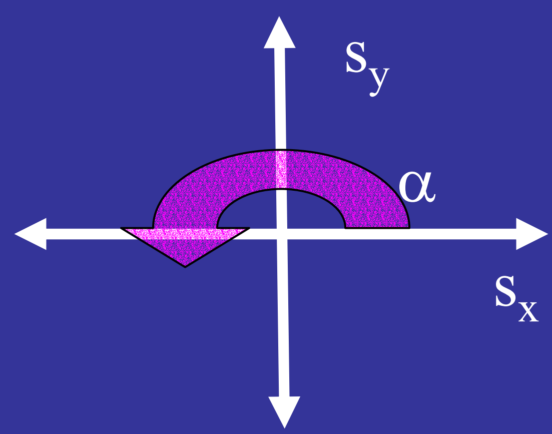
The alignment problem

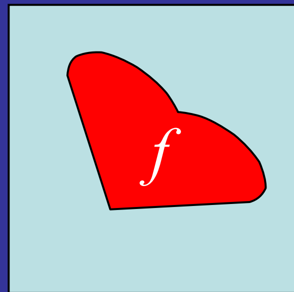
Two 2D images:

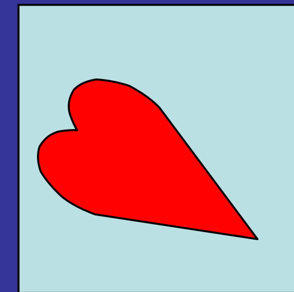




Three degrees of freedom:







Two images are aligned if the least square discrepancy between them is minimized:

$$\int \left| f\left(\mathbf{x}; s_x, s_y, \alpha \right) - g(\mathbf{x}) \right|^2 d\mathbf{x} \rightarrow min$$

$$\int |f|^2 d\mathbf{x} + \int |g|^2 d\mathbf{x} - 2\int f\left(\mathbf{x}; s_x, s_y, \alpha \right) g(\mathbf{x}) dx \rightarrow min$$

$$const + const - c\left(s_x, s_y, \alpha \right) \rightarrow min$$

$$c\left(s_x, s_y, \alpha \right) \rightarrow max$$

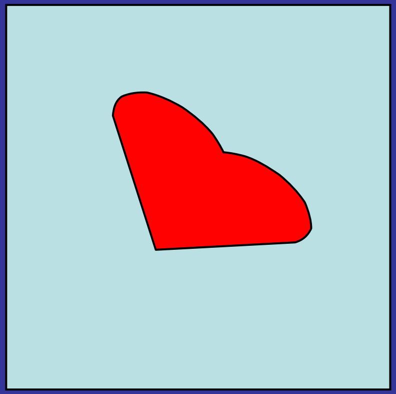
Two images are aligned if the least square discrepancy between them is minimized:

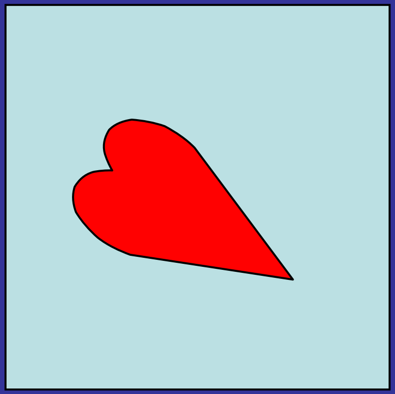
$$c\left(s_x, s_y, \alpha \right) \rightarrow max$$

Maximum of the cross-correlation function

Alignment as a discrete problem

Two 2D images:





Translation *or* rotation can be found effectively using the Fast Fourier Transform algorithm.

To find translation *and* rotation we can use one of the two possible strategies:

- use gradient-based methods to move towards best orientation

- use exhaustive search (explore all possible orientations)

Images are represented on a two-dimensional rectangular grid.

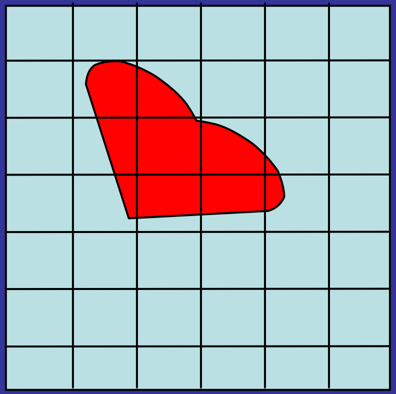
The accuracy of the search is *finite*.

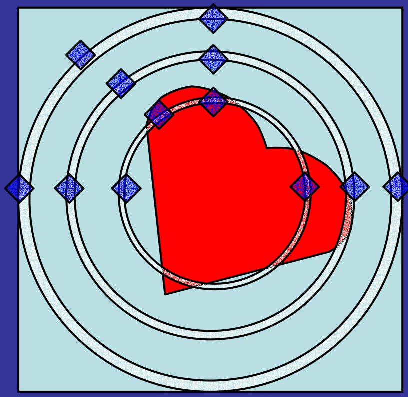
Gradient-based methods are not suitable.

The number of orientations that have to be explored is *finite*.

Possible orientations of images are discrete and finite in respect to both translation and rotation.

Possible orientations of images are discrete and finite in respect to both translation and rotation.





Number of possibilities:

$(2k+1)^2 2\pi\ r$

For search range k=10 and radius of the object r=60, the number of possible orientations is 2770.

For n images the number of possibilities is

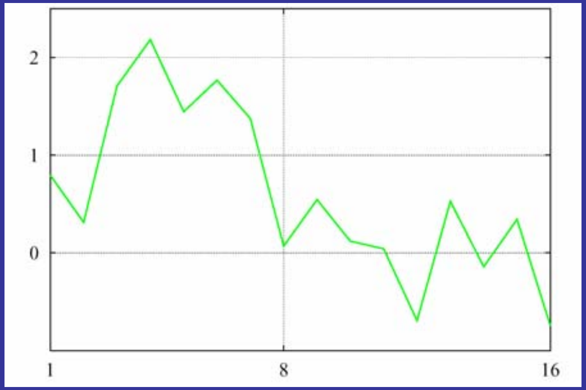
$\{(2k+1)^2\ (2\pi\ r)\}^n$

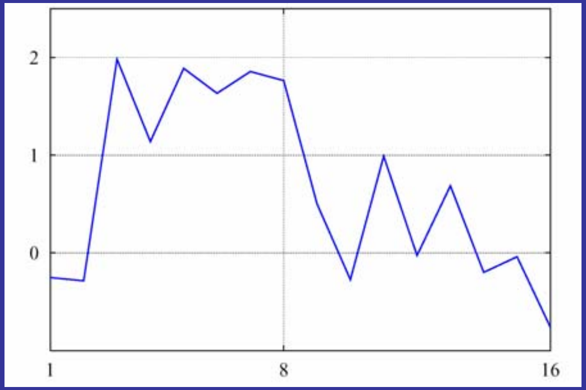
For 100 images $2770^{100}=10^{344}$!

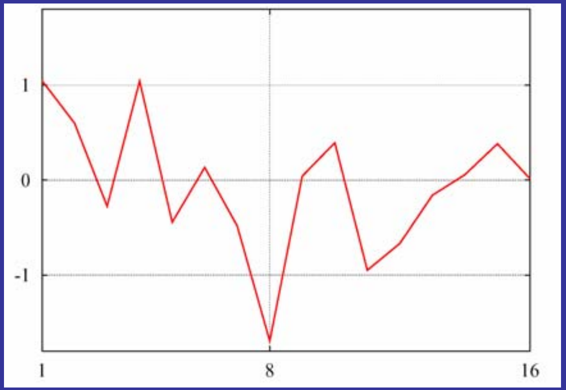
Similarity measures between images

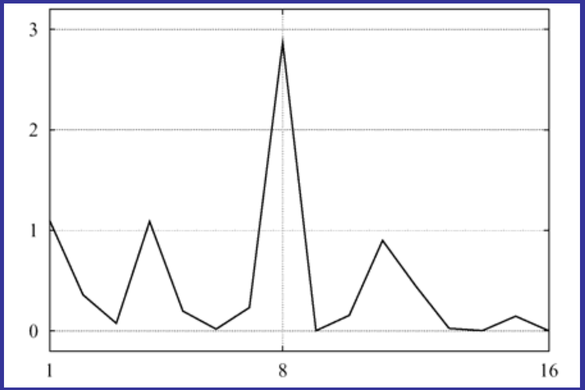
- Euclidean measure
- Correlation coefficient

Euclidean distance



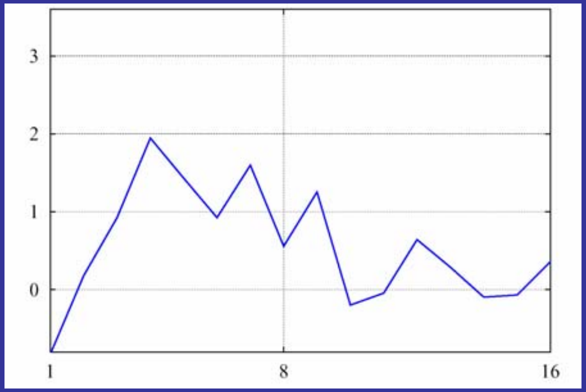


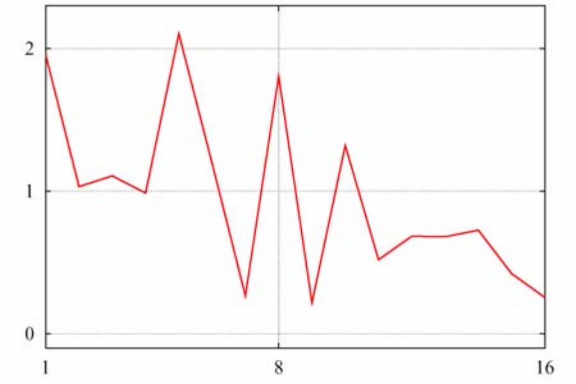


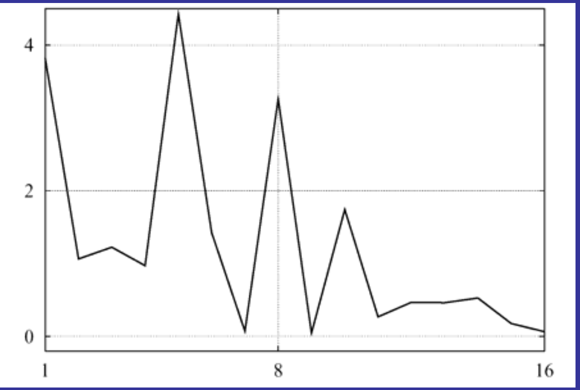


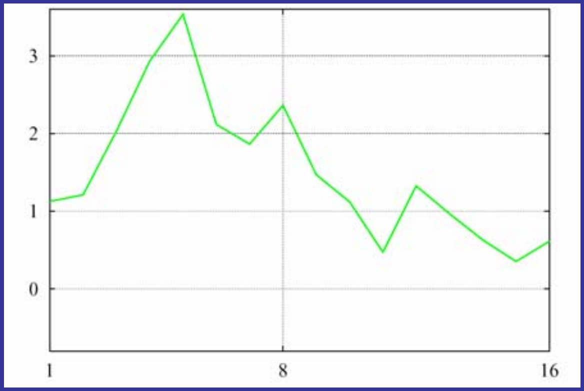
7.6

Euclidean distance depends on the scaling of images (both additive and mulitplicative).



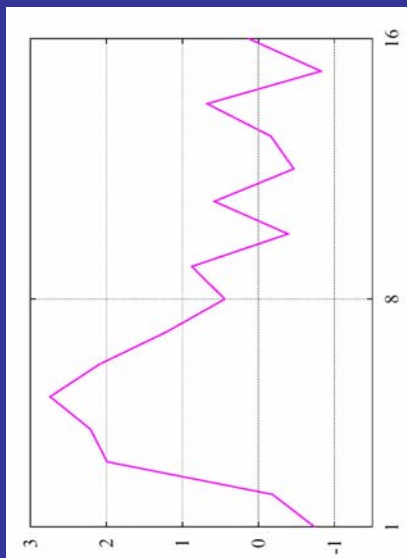


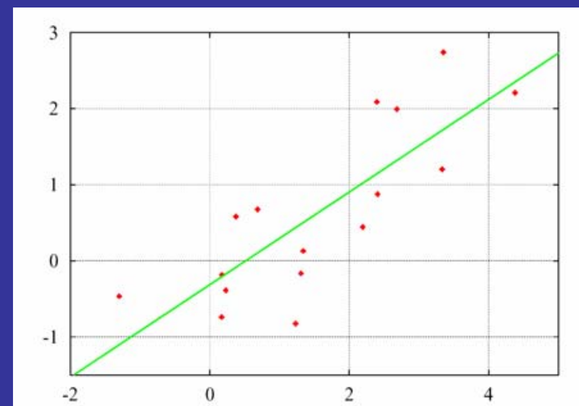




20.0

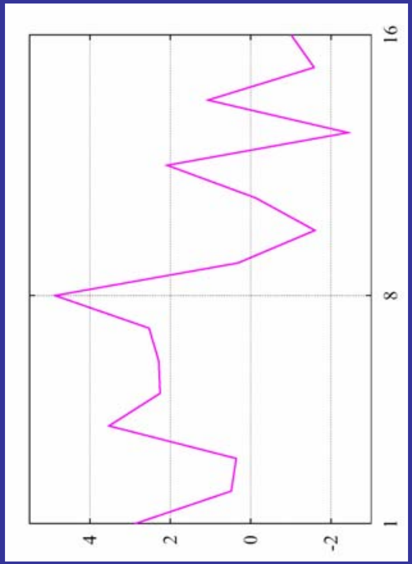
Correlation coefficient

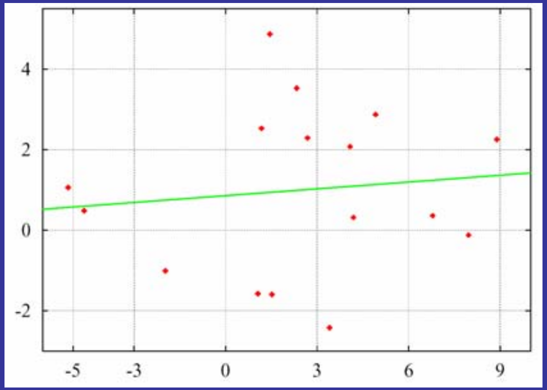




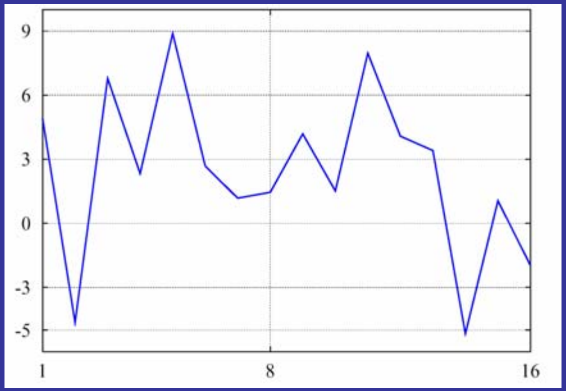
0.81

Correlation coefficient





0.11



Cross-correlation

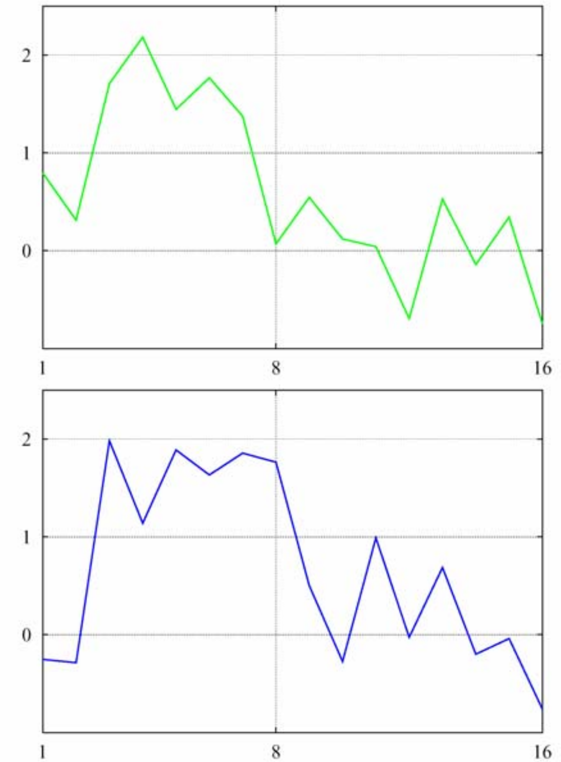
is used as a tool to align (bring into register) images that are in different orientations.

One of the images is shifted with respect to the other, reference image, and for each shift position the similarity between two images is calculated.

A set of similarity values as a function of image position is called ***cross-correlation function*** *(CCF).*

The maximum of CCF indicates the best mutual orientation between a pair of images.

Calculation of crosscorrelation - *methods*



$$Corr(f,g) = c(l) = \sum_{k=1}^{n} f(k)g(k+l) \quad ,l = 0,\ldots,n-1$$

requires n^2 operations.

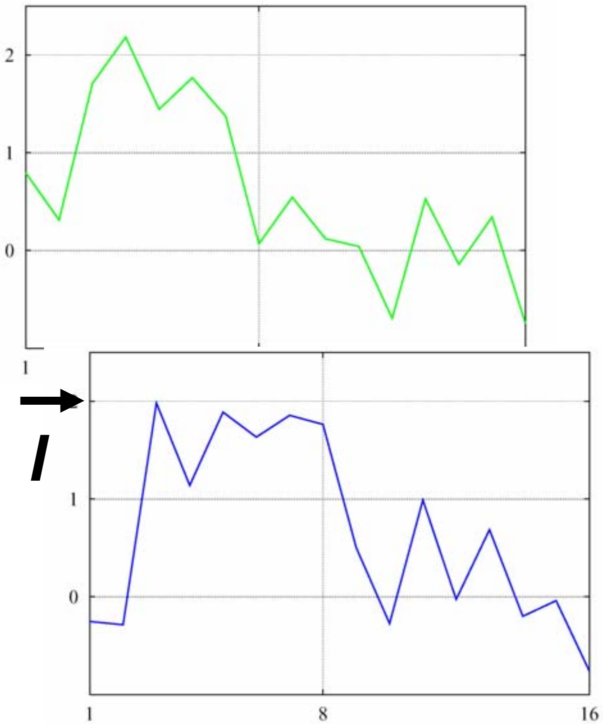
Substantial reduction of the number of operations can be achieved using the FFT.

$Corr(f,g) \Leftrightarrow FG^*$ "Correlation Theorem"

1D FFT	$n \log_2 n$
1D FFT	$n \log_2 n$
multiplication	n
1D FFT^{-1}	$n \log_2 n$
Total $\cong$	$3n \log_2 n$

FFT-based method is more efficient for *n*>9!

Calculation of crosscorrelation - *methods*



$$c(l)=\sum_{k=1}^{n} f(k)g(k-l) \qquad ,l=1,\dots,n$$

both f and g are defined for $k=1,\dots,n$

WHAT IS OUTSIDE OF THE KNOWN RANGE??

Two solutions:

1. assume that outside of the known image there is "nothing" (zeroes)

$$c(l)=\frac{1}{\left|k-l+1\right|}\sum_{k=1}^{n} f(k)g(k+l) \qquad ,l=0,\dots,n-1$$

2. assume that the image is "periodic" (circularly closed)

$$c(l)=\sum_{k=1}^{n} f(k)g(\mathrm{mod}(k+l,n)) \qquad ,l=0,\dots,n-1$$

Both methods have problems:

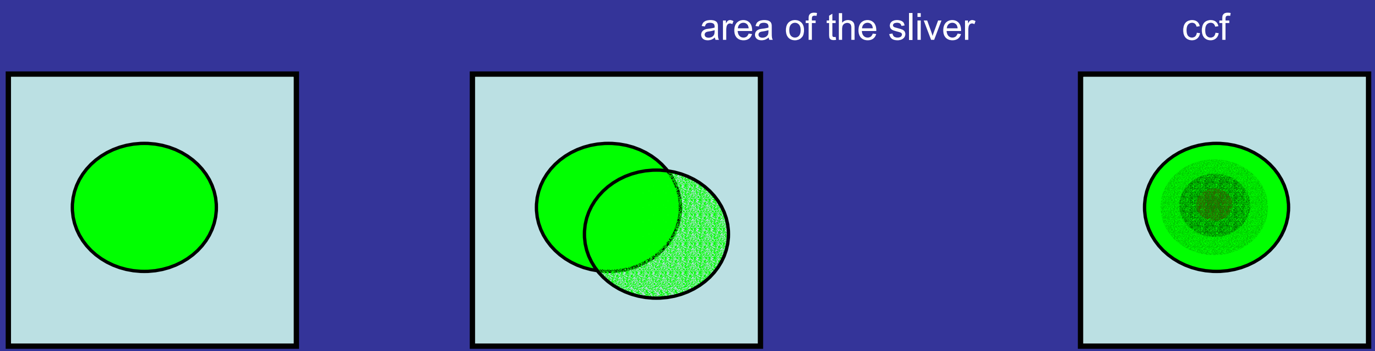
1. If the normalization coefficient is omitted, it favors position *l*=0 of the image; different ccfs have different “statistical reliability” (*l*=0 has lower fluctuation than |*l*|>>0)
2. If the object is smaller than the window size there is no way reduce the influence of noise

Calculation of crosscorrelation - *methods*

Masking of the image is bad – it falls into the first category of the ccf calculation, i.e., it is based on the assumption that outside of the known image there is “nothing”.

Thus, if the normalization coefficient is omitted (as it usually is the case), it favors position l=0 of the image,

or, different ccfs have different “statistical reliability” (l=0 has lower fluctuation than $|l|$>>0).



Calculation of crosscorrelation - *methods*

A possible solution for the problem of masking is adopted in the method based on the resampling into polar coordinates around the systematically selected centers of the image.

L. Joyeux, P.A. Penczek / Ultramicroscopy 92 (2002) 33–46 35

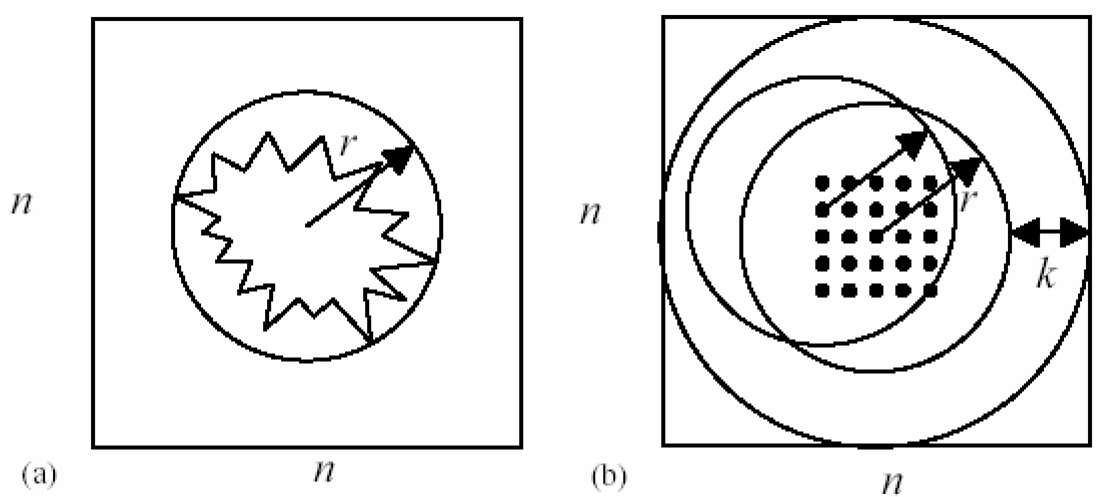


Fig. 1. The geometrical constraints of the 2D alignment problem. (a) The reference particle view is placed within a square image frame $n \times n$ pixels and its size is such that it can be bounded by a circle with a radius no larger than $r = n/2$. (b) The particle view, which size is bounded by the same radius as the reference view, can be located within a circle centered on discrete locations within the image frame, such that the maximum translation is $k = n/2 - r$. The number of possible translations is $(2k+1)^2 = l$.

The size of the object and the window size determine the number of possible object locations.

L. Joyeux, P.A. Penczek / Ultramicroscopy 92 (2002) 33–46 35

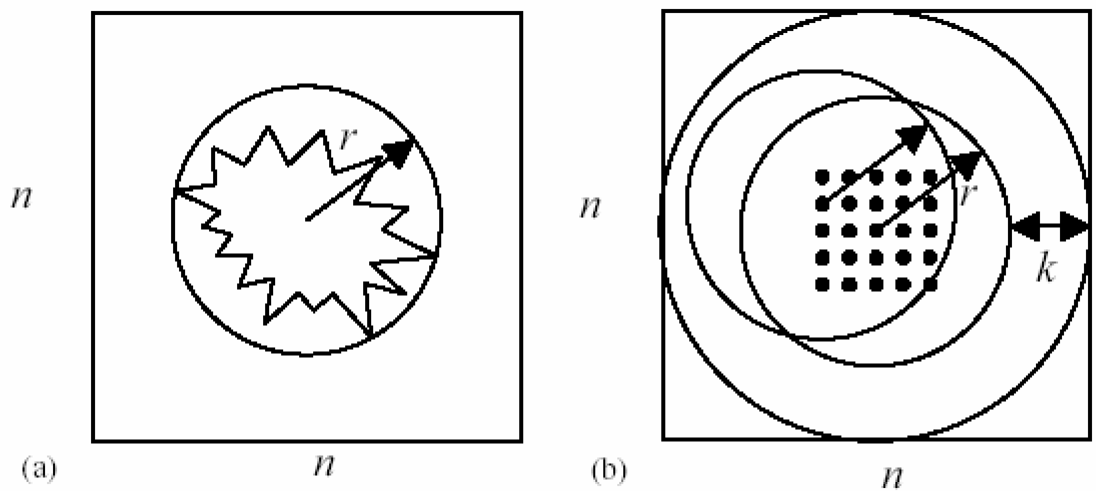
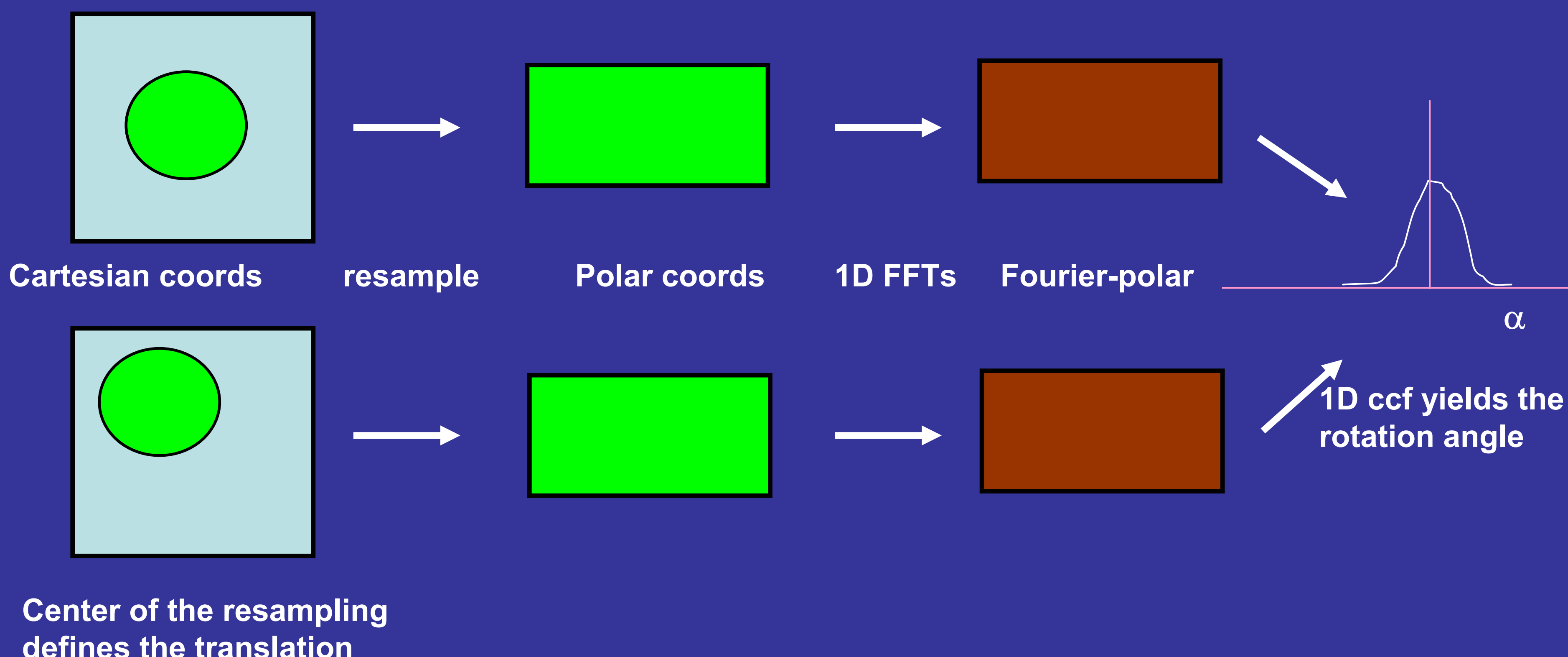


Fig. 1. The geometrical constraints of the 2D alignment problem. (a) The reference particle view is placed within a square image frame $n \times n$ pixels and its size is such that it can be bounded by a circle with a radius no larger than $r = n/2$. (b) The particle view, which size is bounded by the same radius as the reference view, can be located within a circle centered on discrete locations within the image frame, such that the maximum translation is $k = n/2 - r$. The number of possible translations is $(2k + 1)^2 = l$.

Calculation of crosscorrelation - *methods*

A possible solution for the problem of masking is adopted in the method based on the resampling into polar coordinates around the systematically selected centers of the image.



Methods of 2D alignment

- *Direct alignment in real space*
- *Direct alignment using 2D FFT*
- *Sinograms*
- *Indirect alignment using autocorrelation function*
- *Alignment using resampling to polar coordinates*

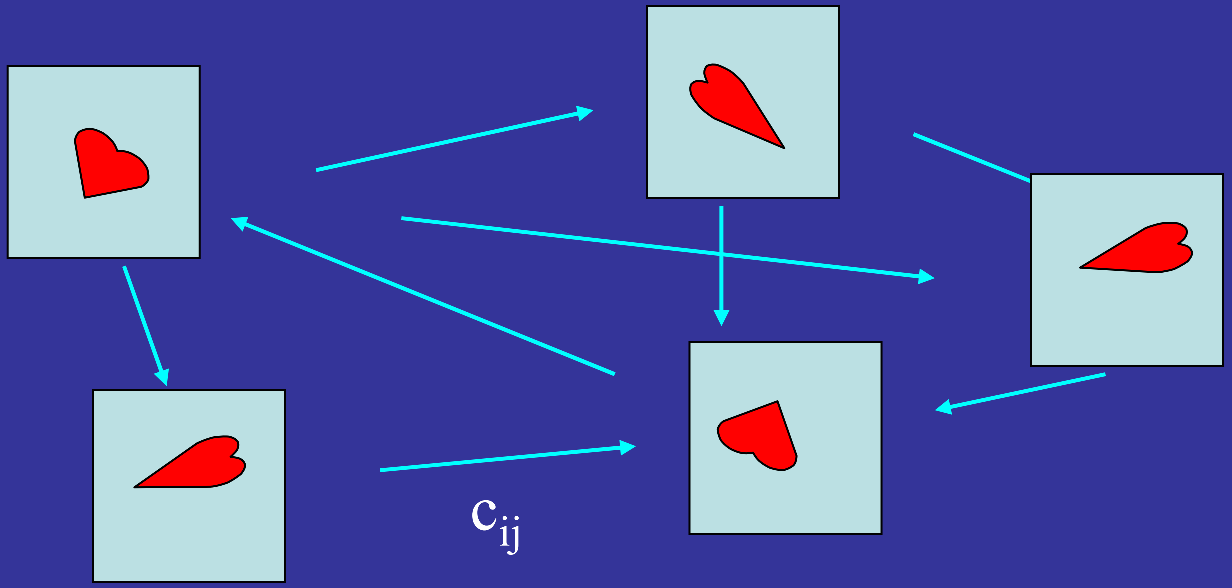
Two images are aligned if the least square discrepancy between them is minimized:

$$c\left(s_x, s_y, \alpha\right) \rightarrow max$$

Maximum of the cross-correlation function

How to define the best alignment for n *objects?*

Alignment of n objects



The distances between all pairs of images have to be minimized simultaneously.

$$\sum_{k=1}^{n-1} \sum_{l=k+1}^{n} \int \left| f_k\left(\mathbf{x}; s_x^k, s_y^k, \alpha^k\right) - f_l\left(\mathbf{x}; s_x^l, s_y^l, \alpha^l\right)\right|^2 d\mathbf{x} \rightarrow min$$

Sum of distances between all pairs of images.

$$\sum_{k=1}^{n-1} \sum_{l=k+1}^{n} \int \left| f_k\left(\mathbf{x}; s_x^k, s_y^k, \alpha^k\right) - f_l\left(\mathbf{x}; s_x^l, s_y^l, \alpha^l\right) \right|^2 d\mathbf{x} =$$

$$\sum_{k=1}^{n} \int \left| f_k\left(\mathbf{x}; s_x^k, s_y^k, \alpha^k\right) - \left\langle f \right\rangle_k \right|^2 d\mathbf{x} \rightarrow min \quad ,$$

where

$$\left\langle f \right\rangle_k = \frac{1}{n-1} \sum_{\substack{l=1 \\ l \neq k}}^{n} f_l\left(\mathbf{x}; s_x^l, s_y^l, \alpha^l\right)$$

Sum of distances between each image and sums (average) of all remaining images.

Sum of distances between all pairs of images.

$$\sum_{k=1}^{n} \int \left| f_k\left(\mathbf{x}; s_x^k, s_y^k, \alpha^k\right) - \langle f \rangle_k \right|^2 d\mathbf{x} \rightarrow min \quad ,$$

where

$$\langle f \rangle_k = \frac{1}{n-1} \sum_{\substack{l=1 \\ l \neq k}}^{n} f_l\left(\mathbf{x}; s_x^l, s_y^l, \alpha^l\right)$$

Sum of distances between each image and sums (average) of all remaining images.

Suggests an alignment algorithm:
orientation of each image is refined against the current average of remaining images.

There is no algorithm that would guarantee location of the global minimum (best possible alignment of a set of n images).

Methods of 2D alignment of *n* images

No matter what the claim of the author might be and whether the author realizes it or not,

all methods of 2D alignment of *n* images try to circumvent the problem

that there is *no algorithm that would guarantee the optimum alignment of n images.*

Methodology of 2D alignment

http://www.wadsworth.org/spider_doc/spider/docs/align.html

The approaches to 2D particle alignment can be subdivided into several categories. The main division is created by the availability of a reference image, and the secondary division by the degree of variability within the data set, i.e., in how many orientations the particle is observed to lie in a micrograph.

Types of alignment problems:

- A reference image is known or can be easily approximated, and there is only one particle orientation (with possible small variations). This case will be called *Reference-based alignment*.
- A small number of reference images are known or can be easily approximated, and their number is known, and particle orientations are well defined (with possible small variations). This case will also be called *Reference-based alignment*.
- An approximation of a reference image is known and there is only one particle orientation (with possible small variations). This case will be called *Alignment with the reference refinement*.
- Reference images are not known, but the data set can in principle be divided into a number (unknown) of homogeneous classes. This case will be called *Multireference alignment*.
- Reference images are not known, but the data set can in principle be divided into a number (unknown) of homogeneous classes. The particles can be centered. This case will be called *Rotationally invariant K-means algorithm*.
- Reference images are not known, and there is no clear groupings in the data set. This case will be called *Reference-free alignment*.