Rekonstruksi Citra

Image Reconstruction

- Image reconstruction is an imaging technique that produces cross-sectional image of an object through the processing of the signal trans-axial projection of the object.
- The term is usually used in image reconstruction tomographic imaging scope.

Tomographic Image Reconstruction

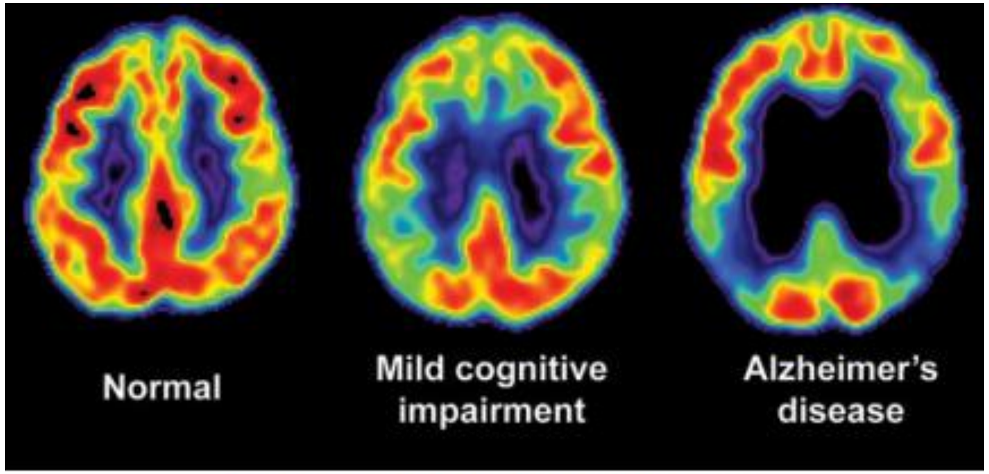


Figure: PET brain images

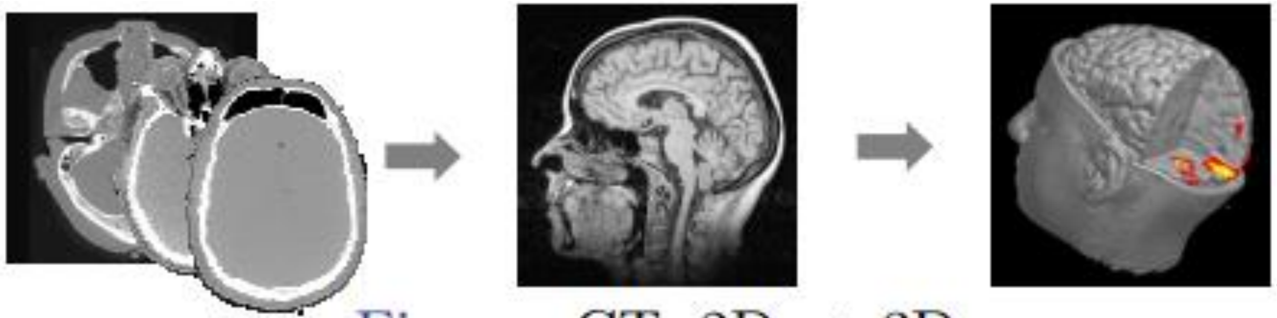
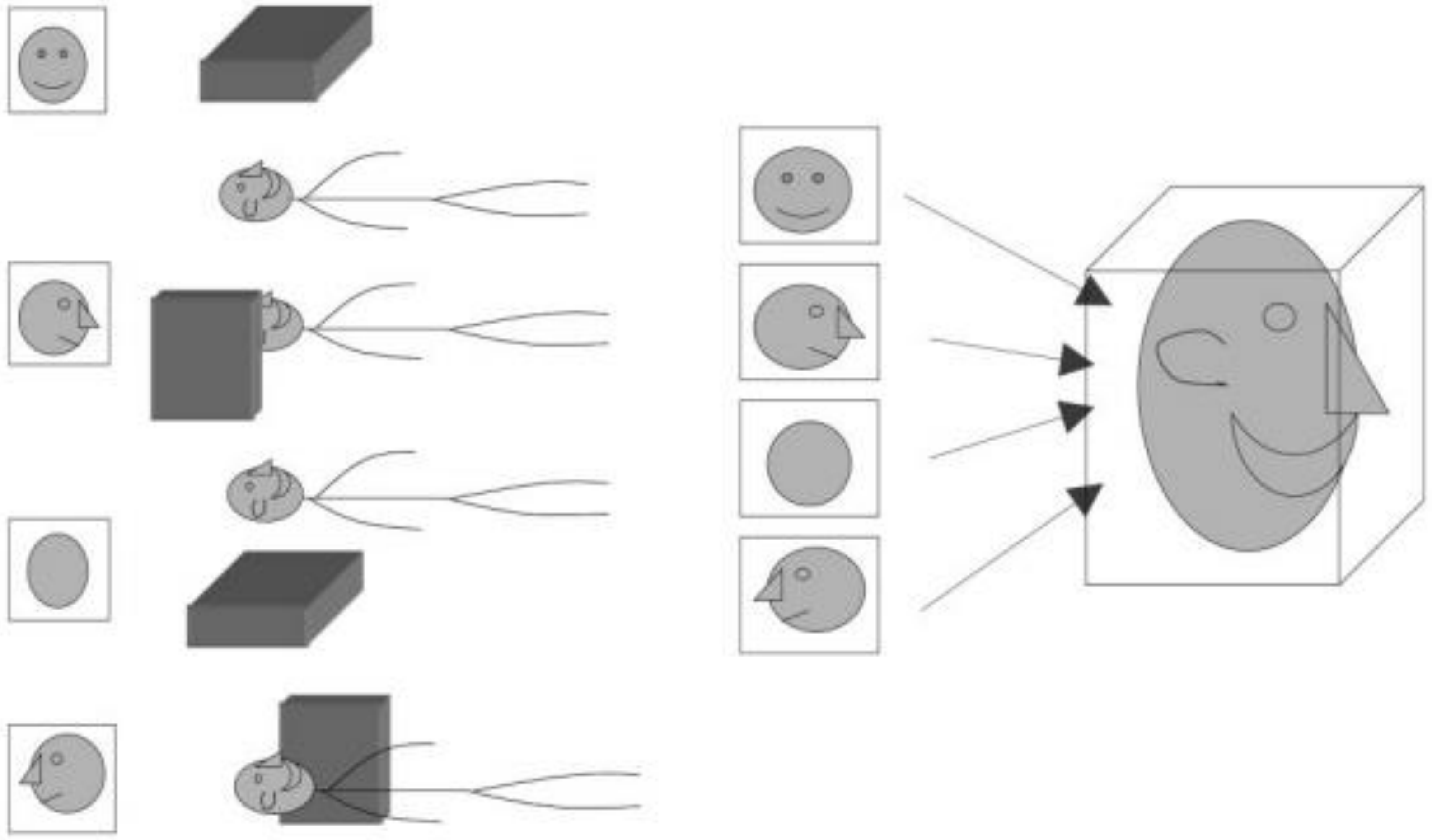
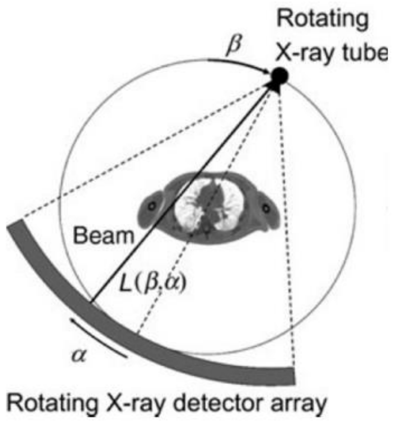


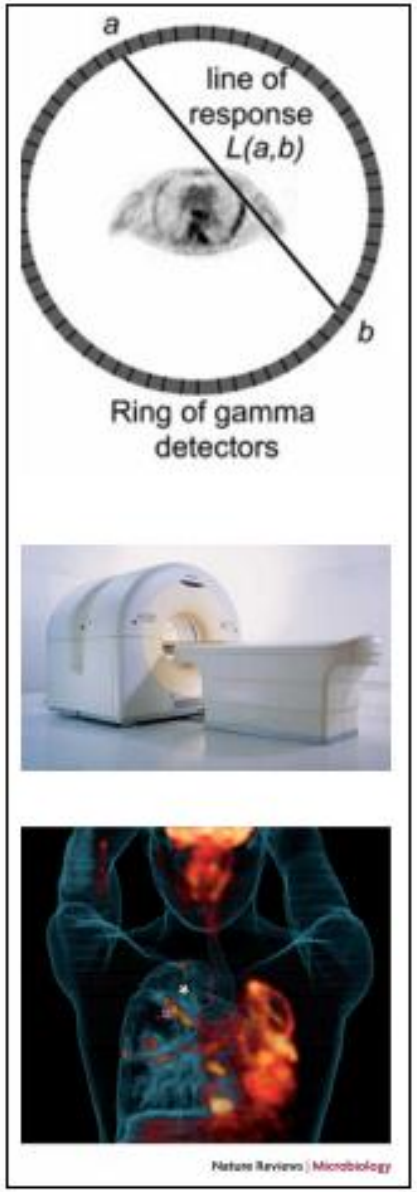
Figure: CT: 2D → 3D

Projection data acquired from different views are used to reconstruct the image.



Computer Tomografi (CT)





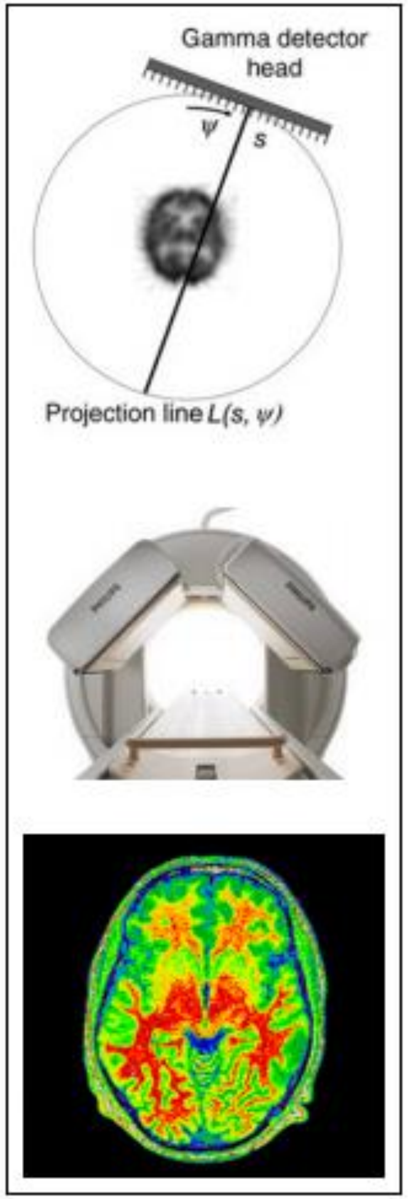
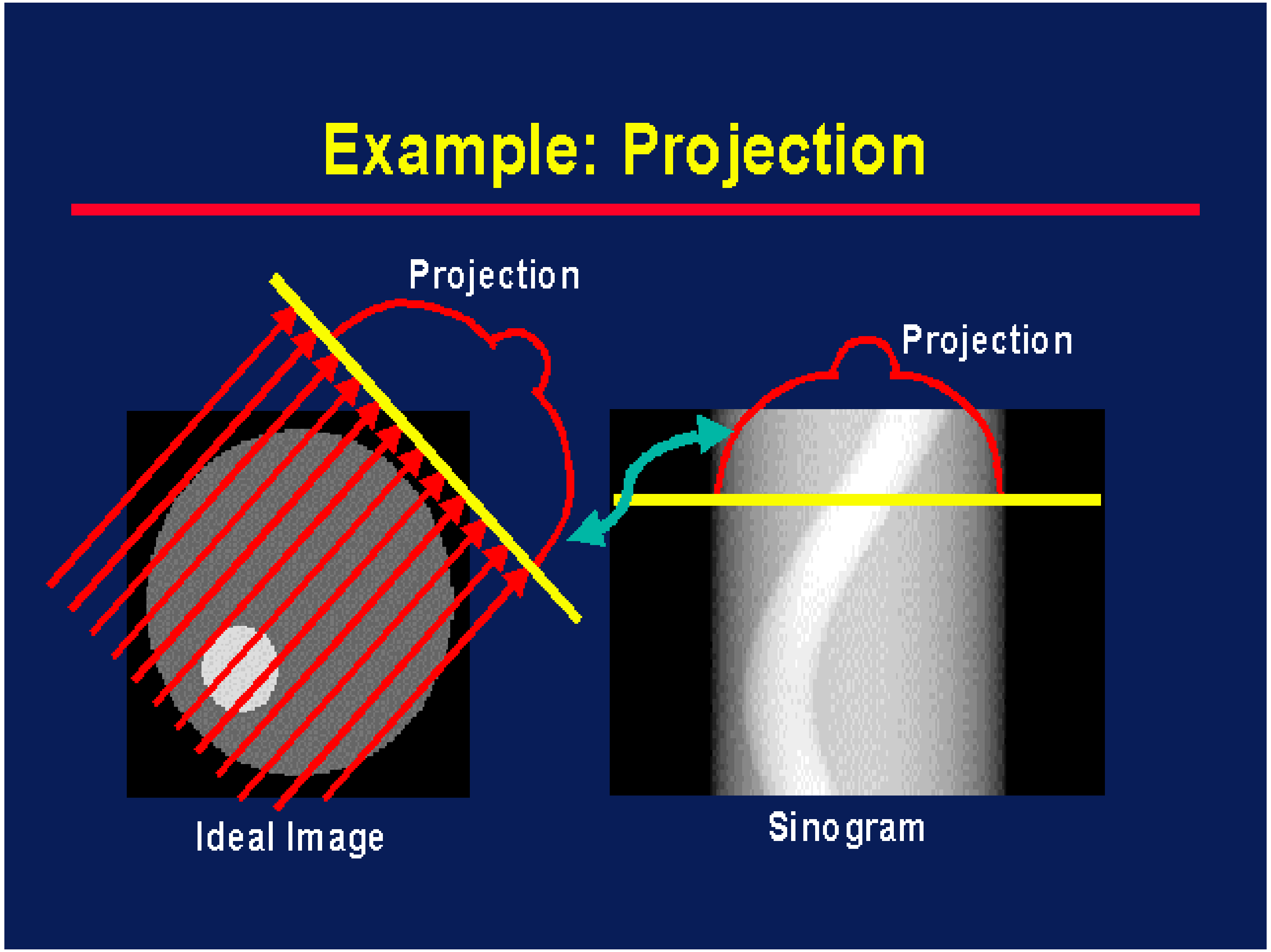


Image Reconstruction

- Analytical tomographic Image Reconstruction
- Iterative Image Reconstruction

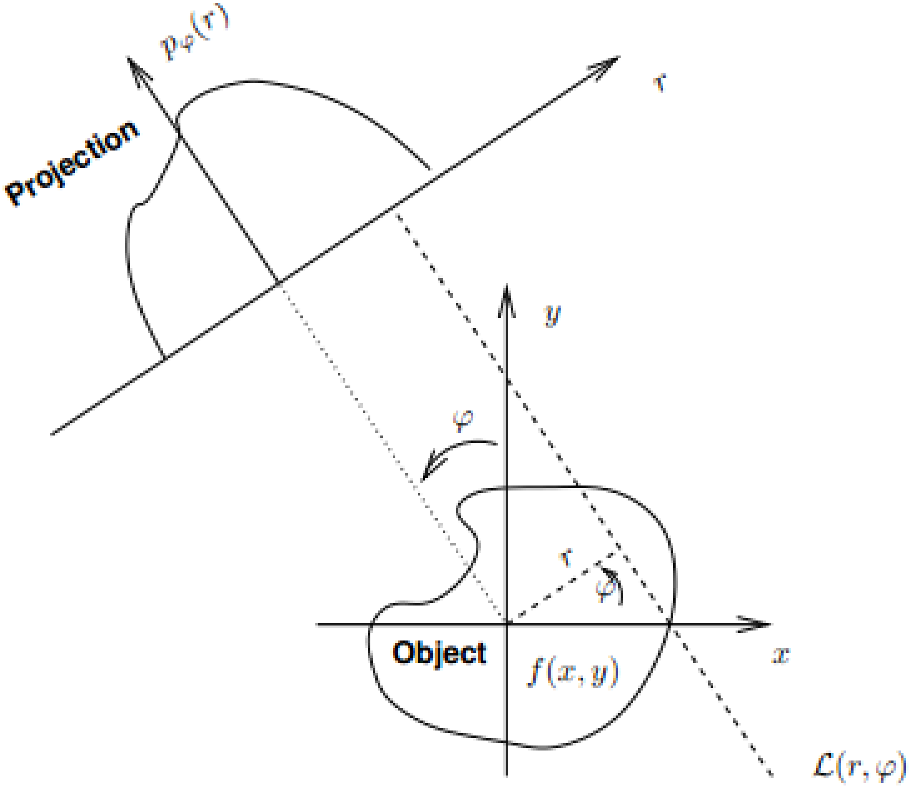
Analytical tomographic Image Reconstruction

- Radon transform in 2D
- Back projection


Example: Projection
Projection
Projection
Ideal Image
Sinogram

Radon transform in 2D

- The foundation of analytical reconstruction methods is the Radon transform, which relates a 2D function *f*(*x*,*y*) to the collection of line integrals of the function.
- Let *L*(r,φ) denote the line in the Euclidean plane at angle *ϕ* counter-clock wise from the y axis and at a signed distance *r* from the origin:



Radon transform in 2D

- Let $p_\varphi(r)$ denote the line integral through f(x, y) a long the line L(r, φ).
- A straight line in Cartesian coordinates can be described either by its slope-intercept form, y = ax+b
- As can be seen by using trigonometry, the inclination is :

$$a = -\frac{\cos\varphi}{\sin\varphi} \qquad b = \frac{r}{\sin\varphi}$$

Radon transform in 2D

$$y = ax + b$$

$$y = x\left(-\frac{\cos\varphi}{\sin\varphi}\right) + \frac{r}{\sin\varphi}$$

$$y + x\frac{\cos\varphi}{\sin\varphi} = \frac{r}{\sin\varphi}$$

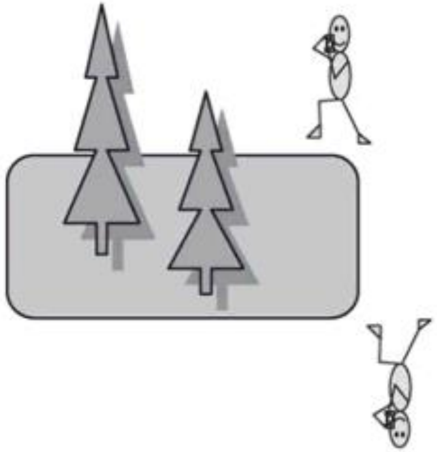
$$y\sin\varphi + x\cos\varphi = r$$

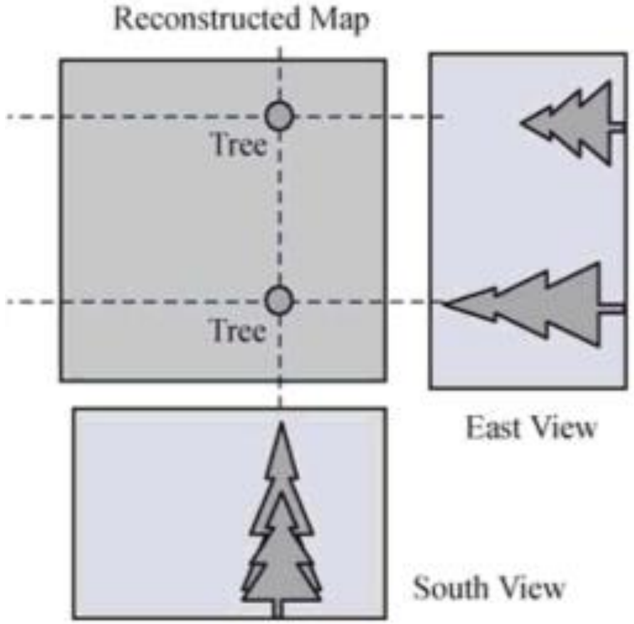
$$y\sin\varphi + x\cos\varphi - r = 0$$

$$p_\varphi(r) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x\cos\varphi + y\sin\varphi - r)\,\mathrm{d}x\,\mathrm{d}y$$

BASIC IDEA OF BACK PROJECTION

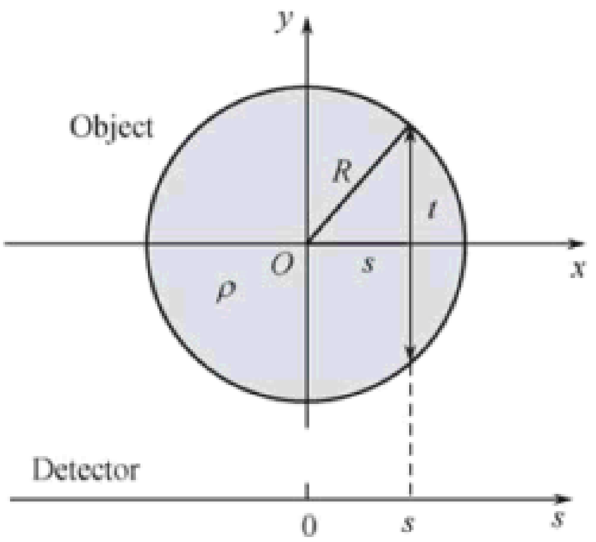
- PHOTO PROJECTION
  - Two trees in a park, make 2 pictures from east and south, try to create a map of the park.

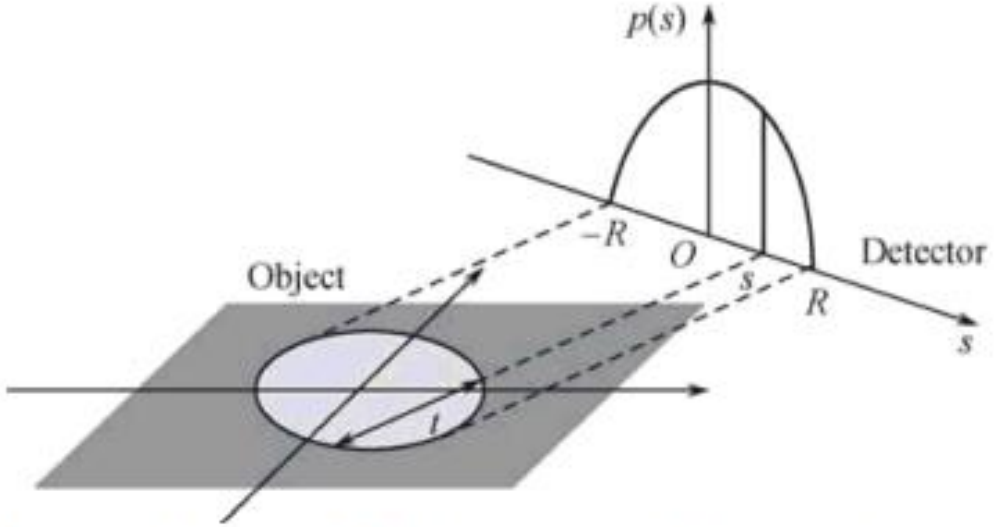




PROJECTION

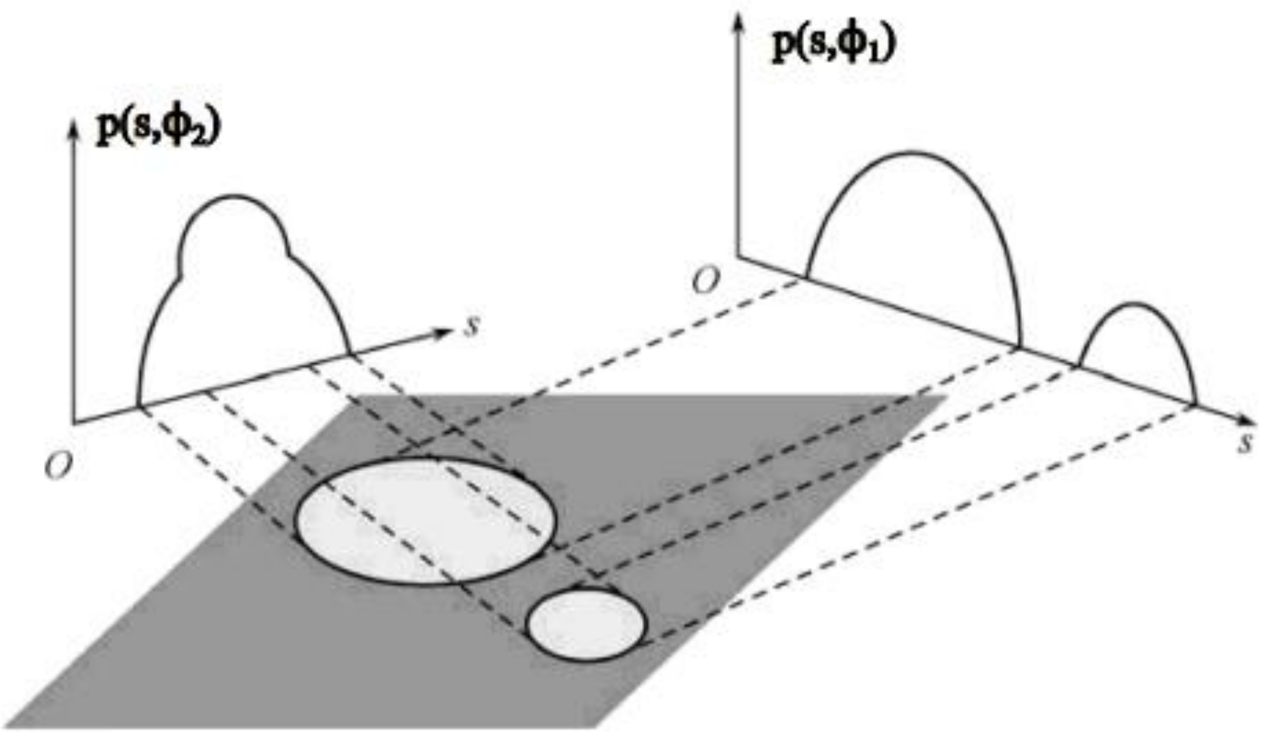
- LINE INTEGRAL PROJECTION
  - Projection $p(s;\phi)$ at angle ϕ , s is coordinate on detector





Projection $p(s)$ the same for any ϕ

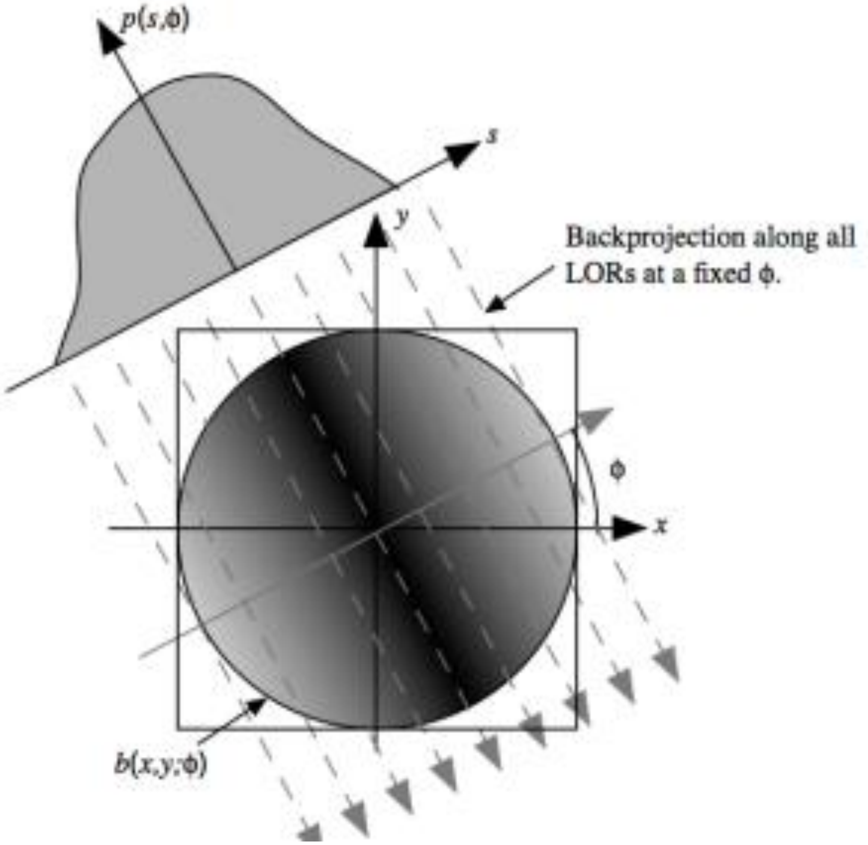
- LINE INTEGRAL PROJECTION



Projection $p(s, \phi)$ depends on orientation

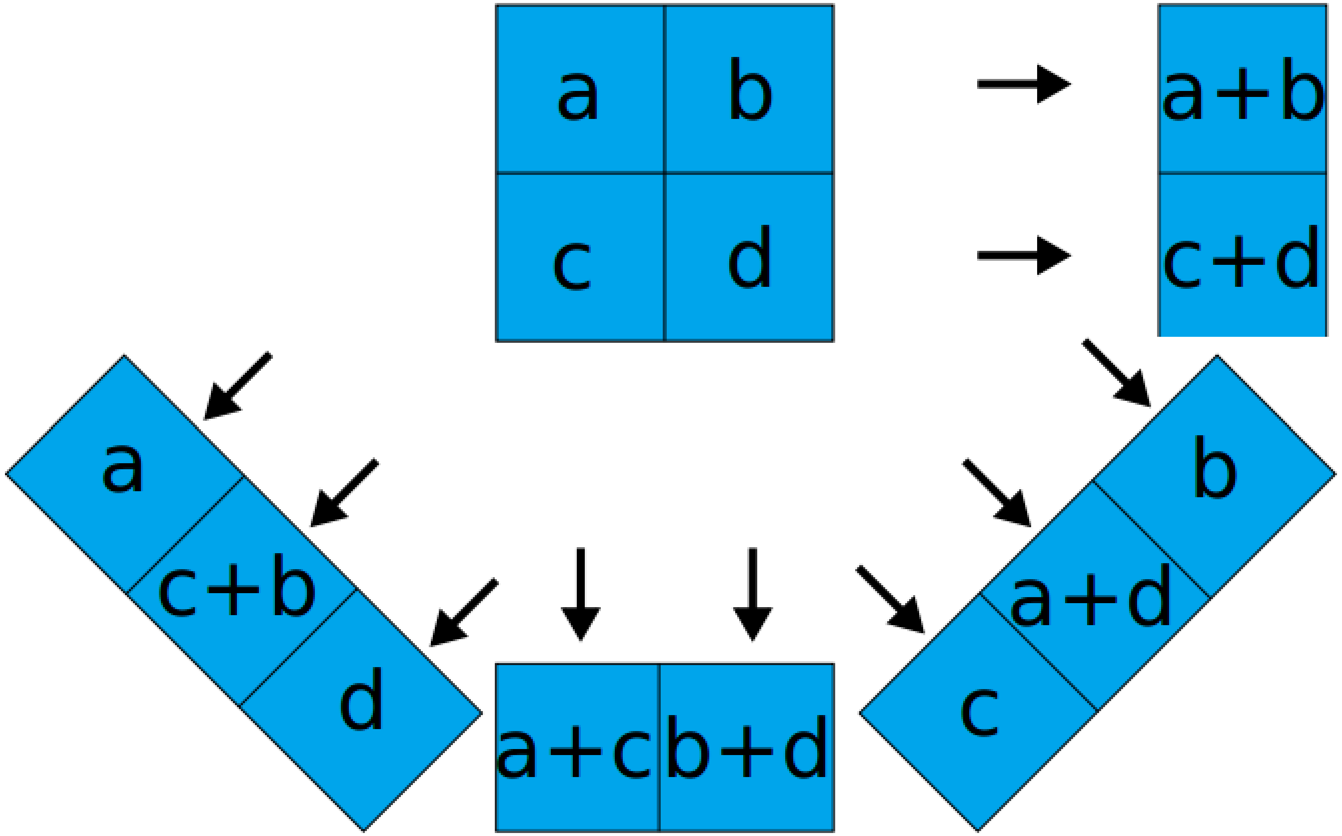
Back projection

- Placing a value of $p(s;\phi)$ back into the position of the appropriate LOR (Lines of Response)



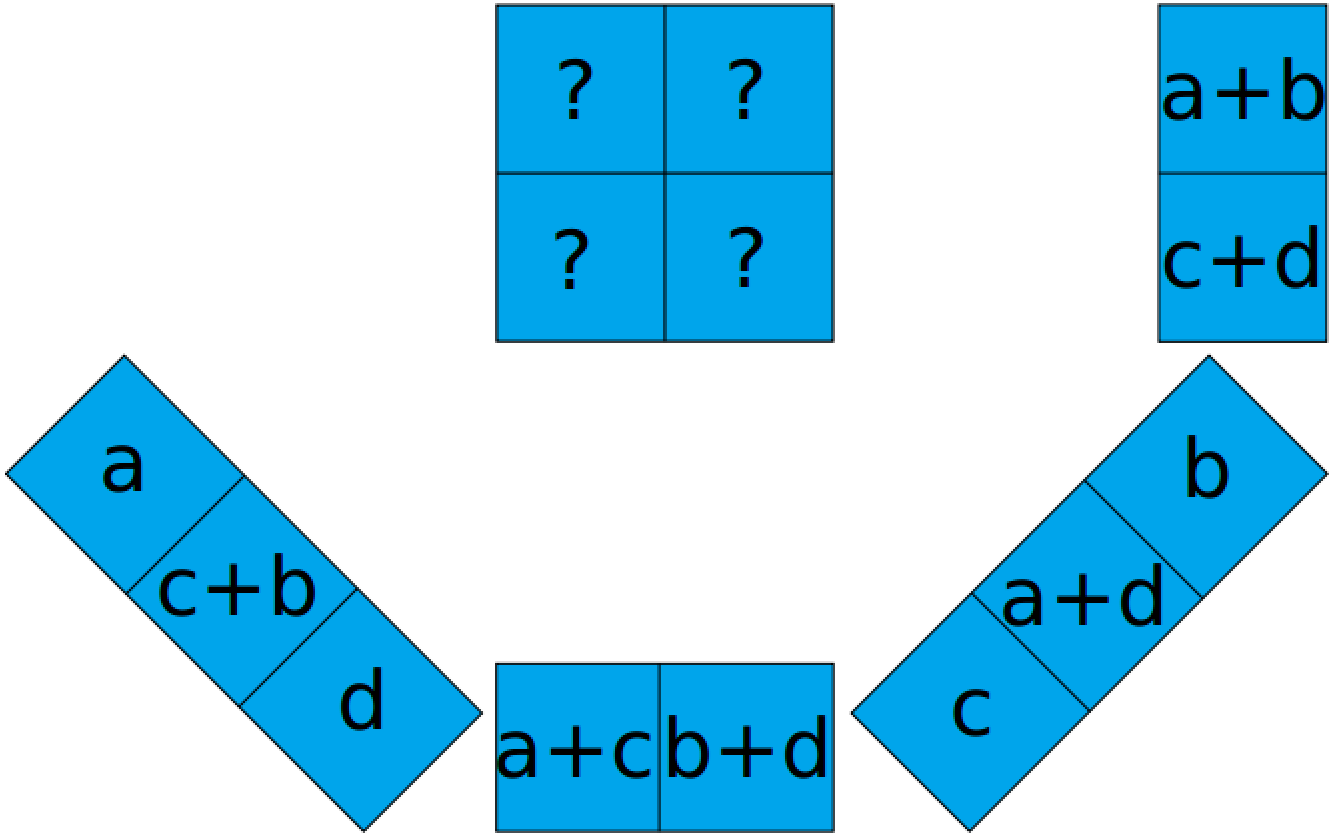
Back projection Example

- Projection

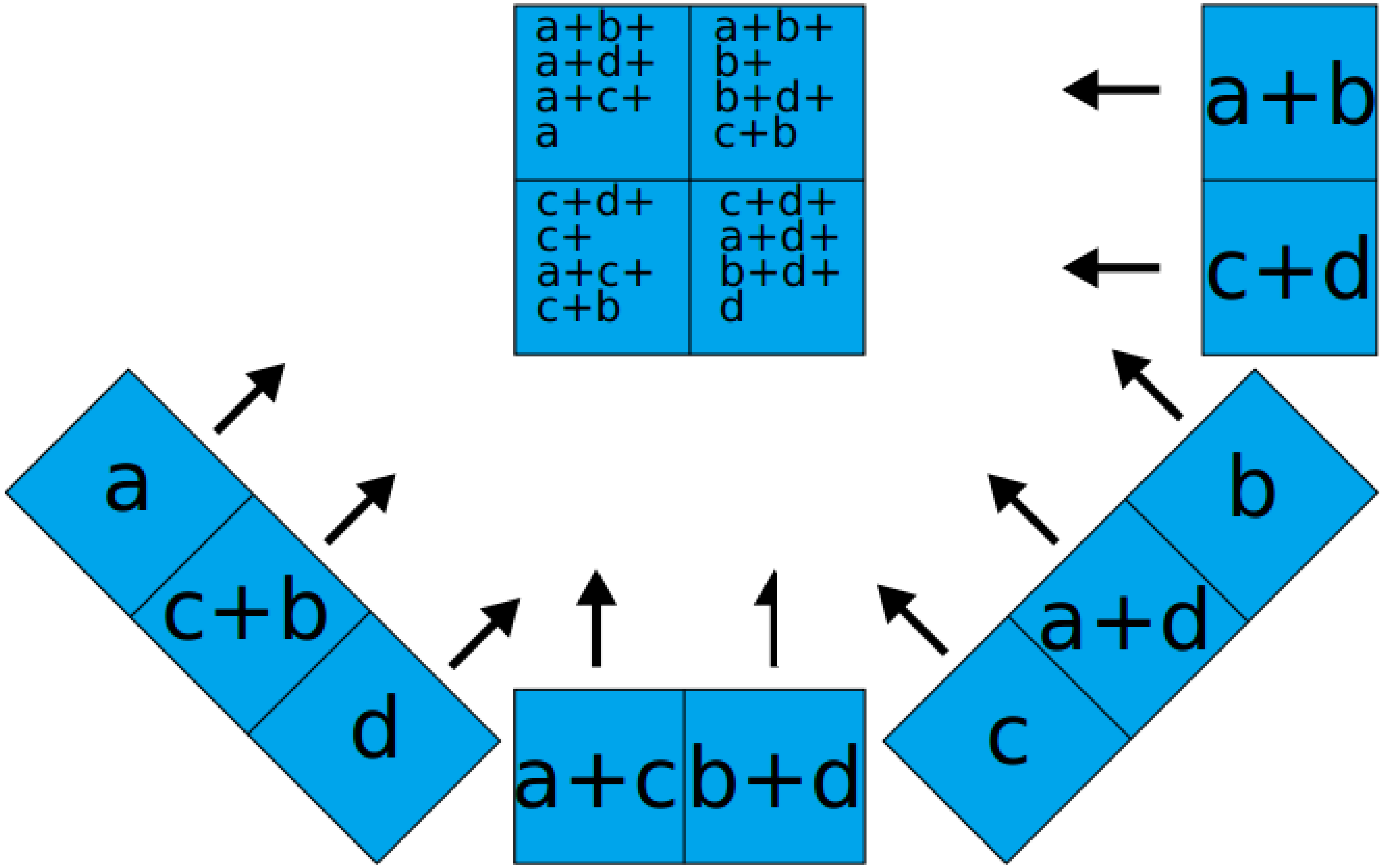


Back projection Example

- Back projection

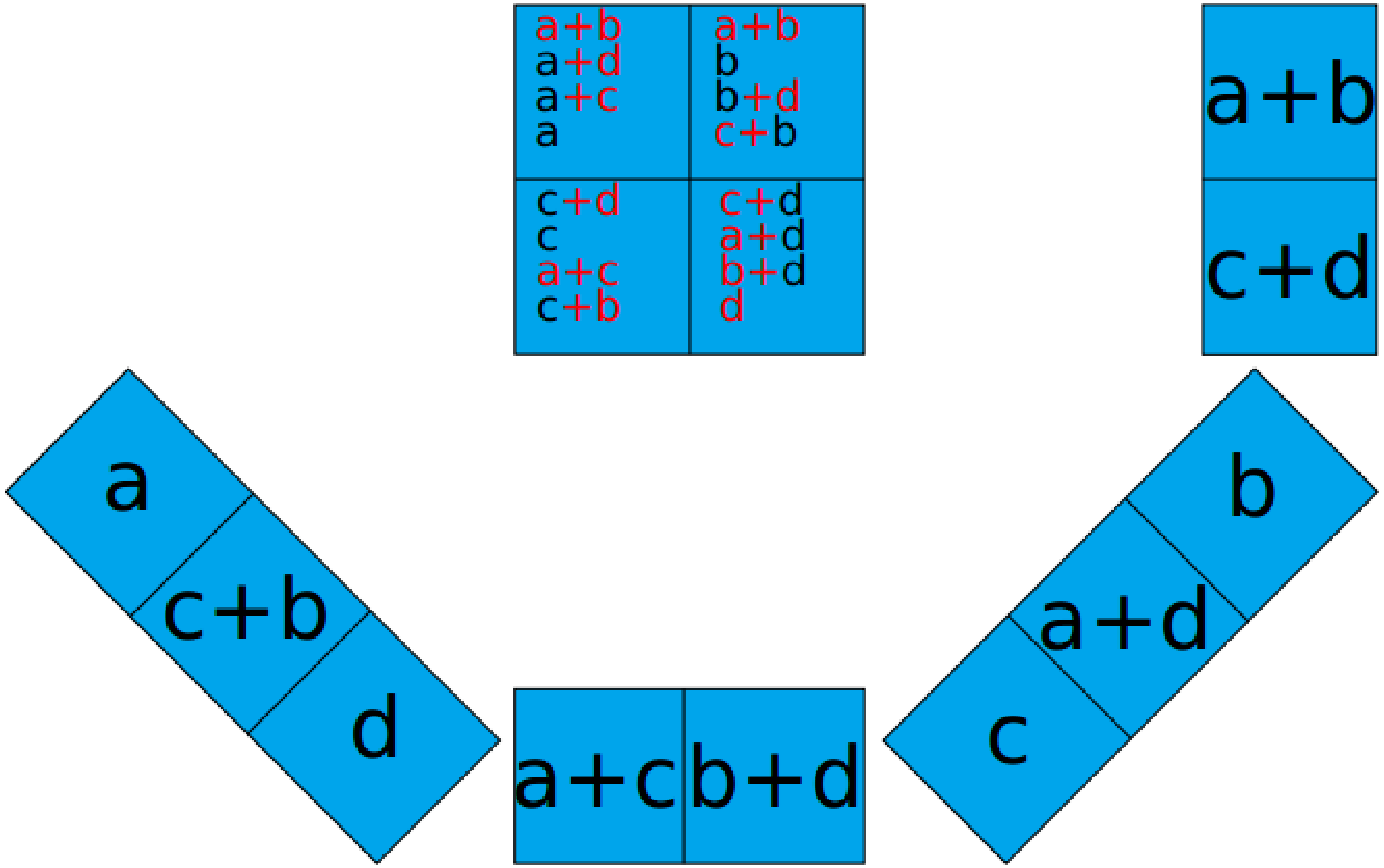


Back projection



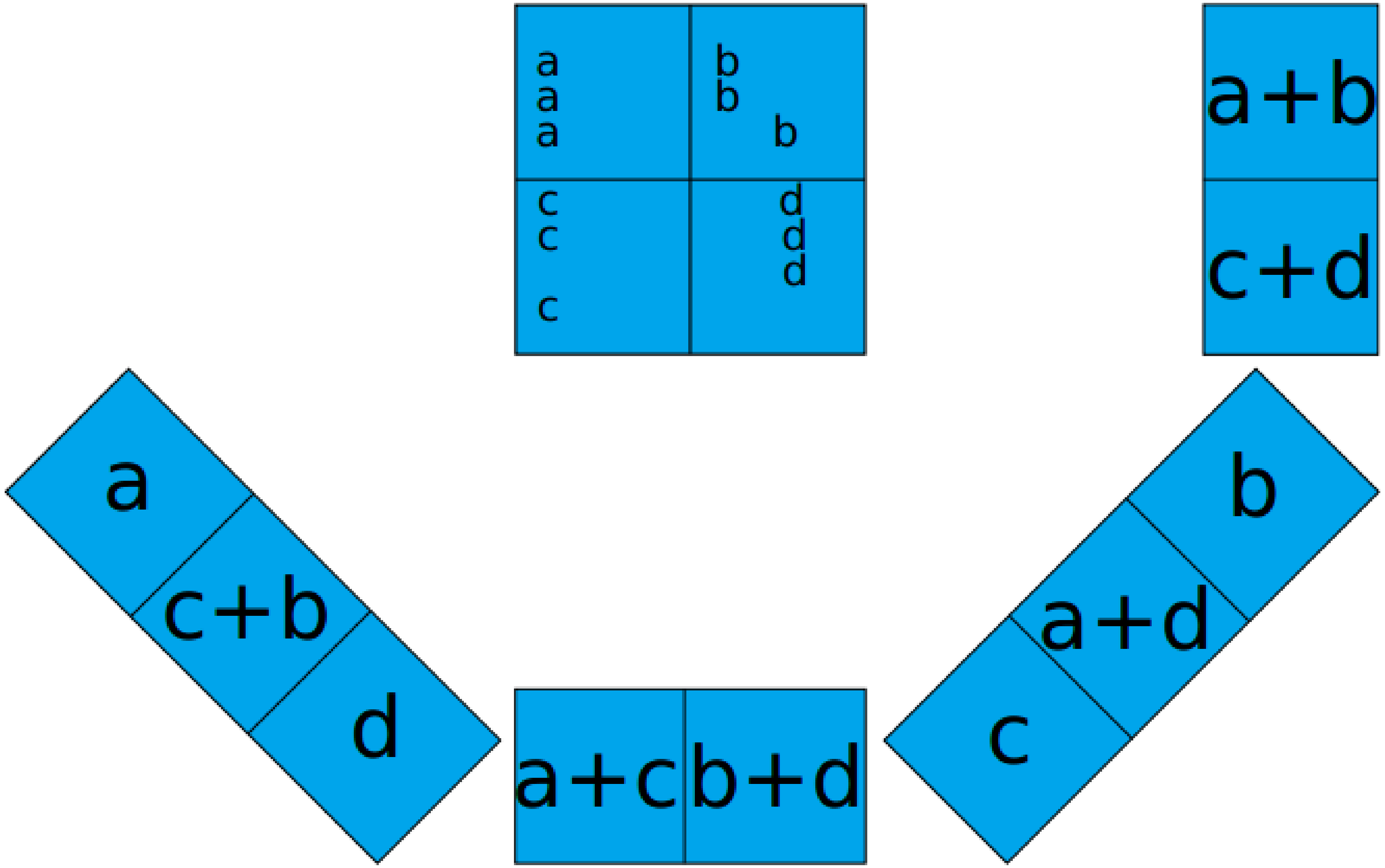
Back projection

Subtract projection sum from each entry



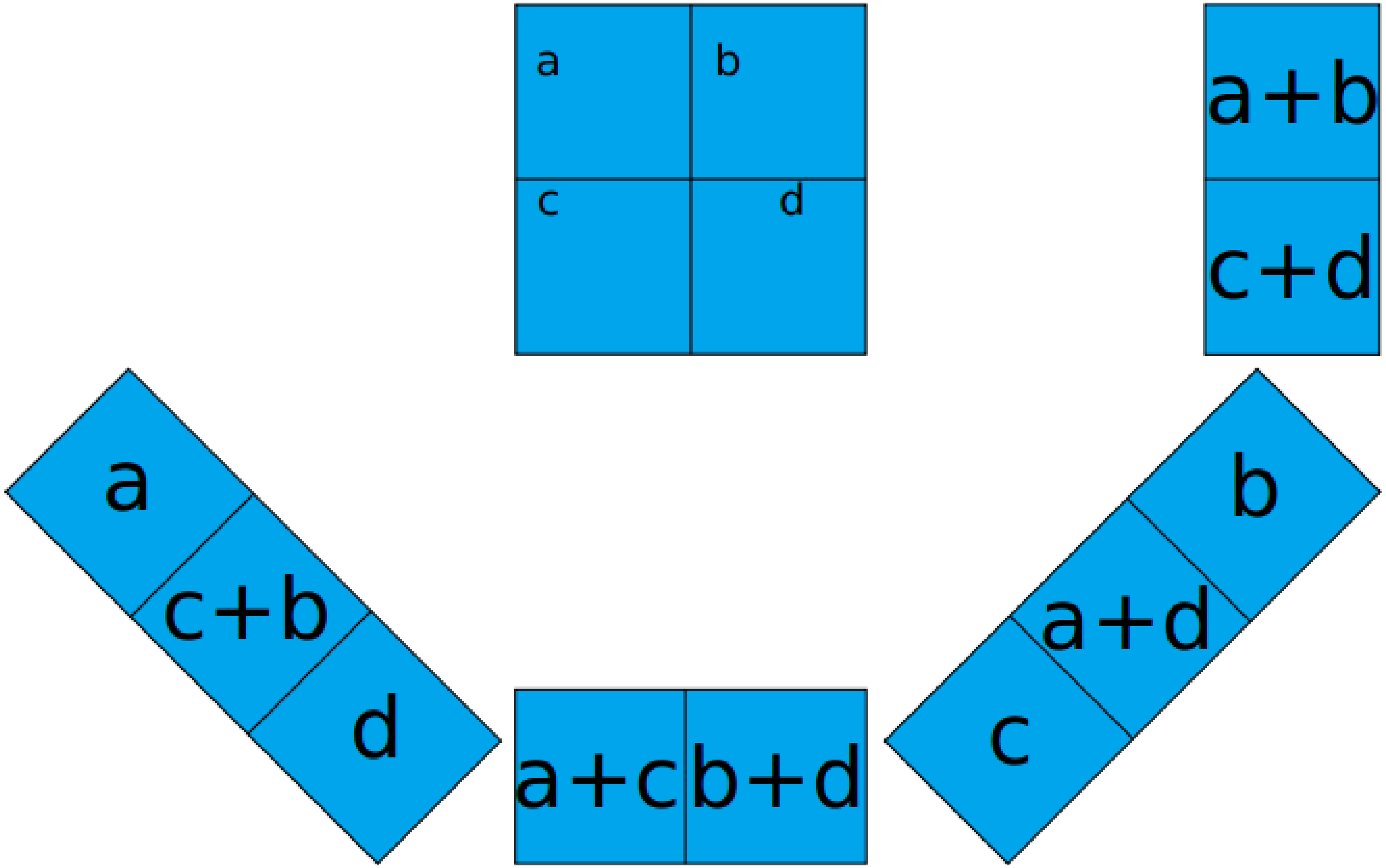
Back projection

Subtract projection sum from each entry

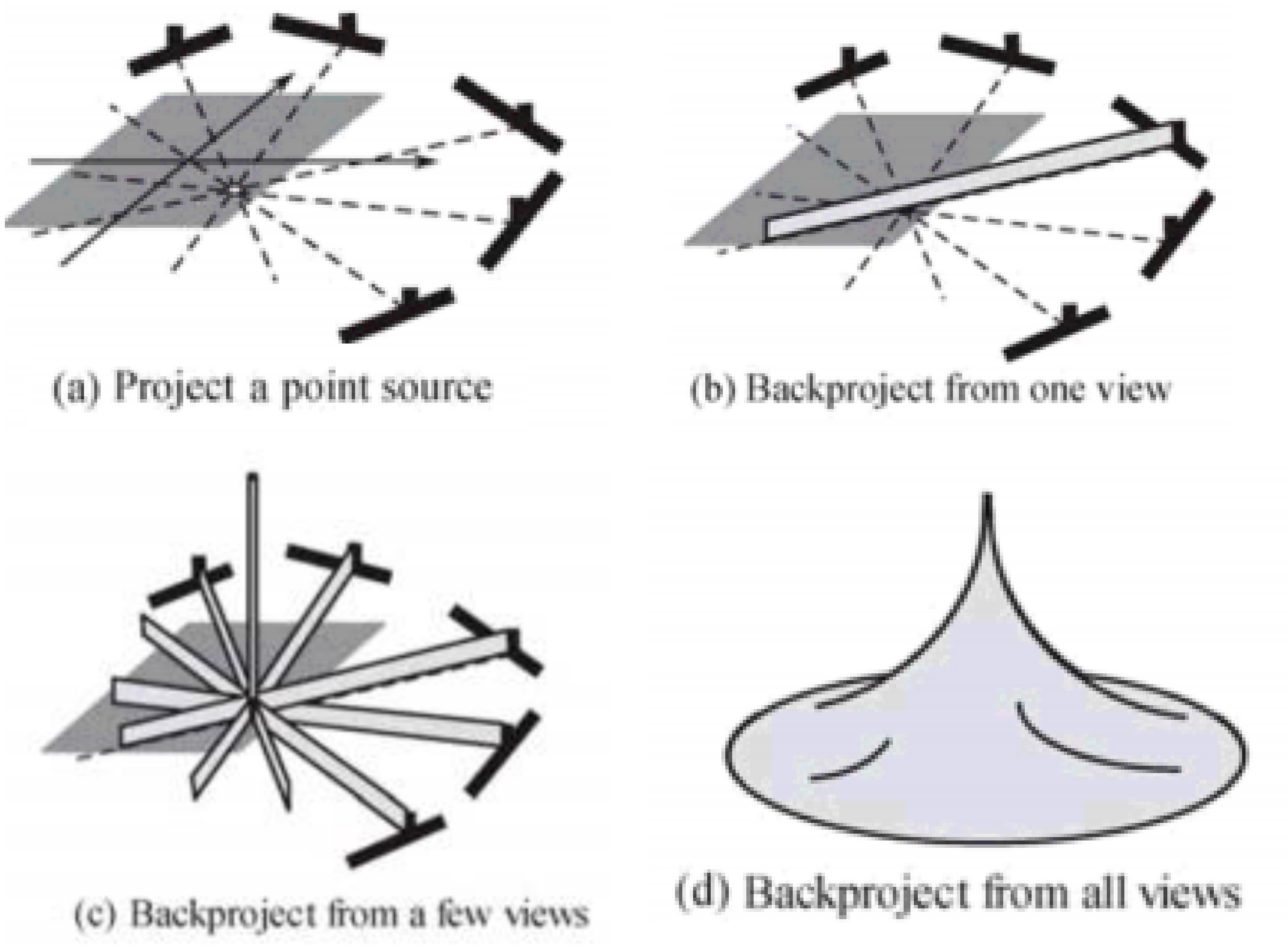


Back projection

Divide by number of projections $-1 = 3$

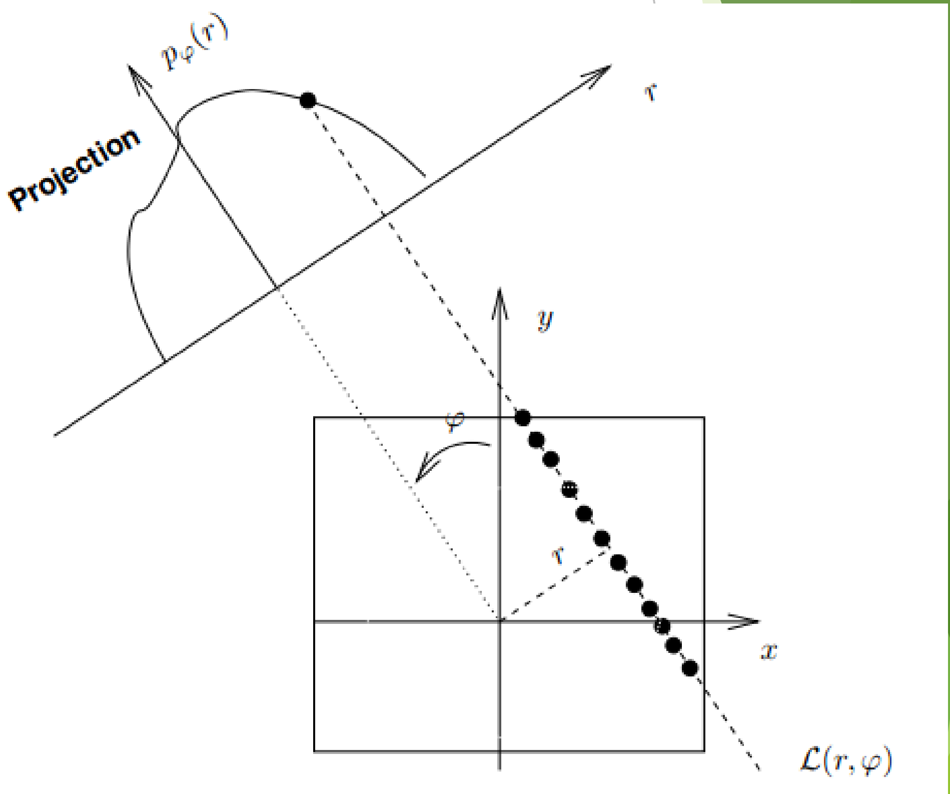


Back projection



BACK PROJECTION

- The Radon transform maps a 2D object $f(x,y)$ into a sinogram $p\phi(r)$ consisting of line integrals through the object.
- One approach to try to recover the object from $p\varphi(r)$ would be to take each sinogram value and "smear" it back in to object space along the corresponding ray, as illustrated in Figure.
- This type of operation is called back projection and is fundamental to tomographic image reconstruction.
- Unfortunately in its simplest form this procedure does not recoverth object $f(x,y)$, but in stead yields a blurred version of the object $f_b(x,y)$.
- This blurred version $f_b(x, y)$ is called a laminogram or layergram



ITERATIVE METHODS

Maximum Likelihood - Expectation Maximisation (ML-EM)

- Objective function to maximise: log-Likelihood (ML)
- Maximisation algorithm: Expectation Maximisation (EM)

ML-EM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

ML-EM Algorithm

- Initial guess for the image (uniform) $x_i^{(0)}$

ML-EM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

- Simulate measurements from estimate (forward proj.)

$$y_j^{\text{simu}} = \sum_k A_{kj} x_k^{(0)}$$

ML-EM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

ML-EM Algorithm

- Compare this with actual measurements

$$\text{Ratio } R_j = \frac{y_j}{y_j^{\text{simu}}}$$

ML-EM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

- Improve image estimate (backward projection)

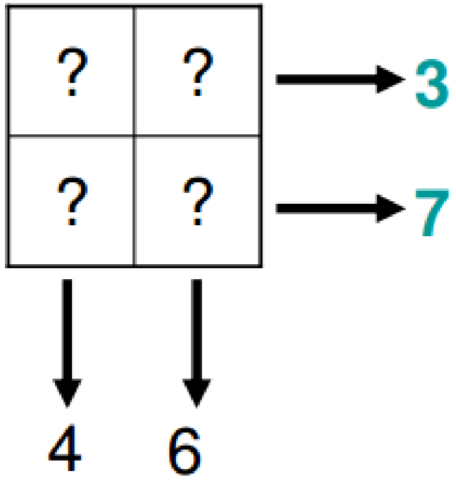
$$x_i^{(1)} = x_i^{(0)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} R_j$$

ML-EM

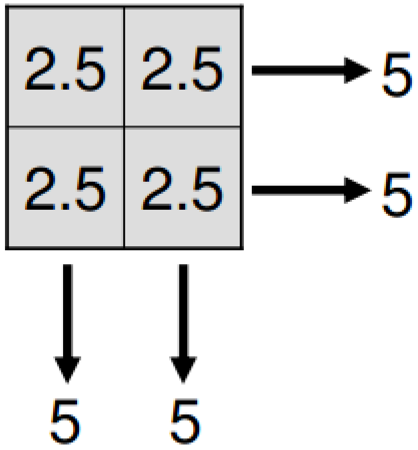
$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

- Repeat until convergence

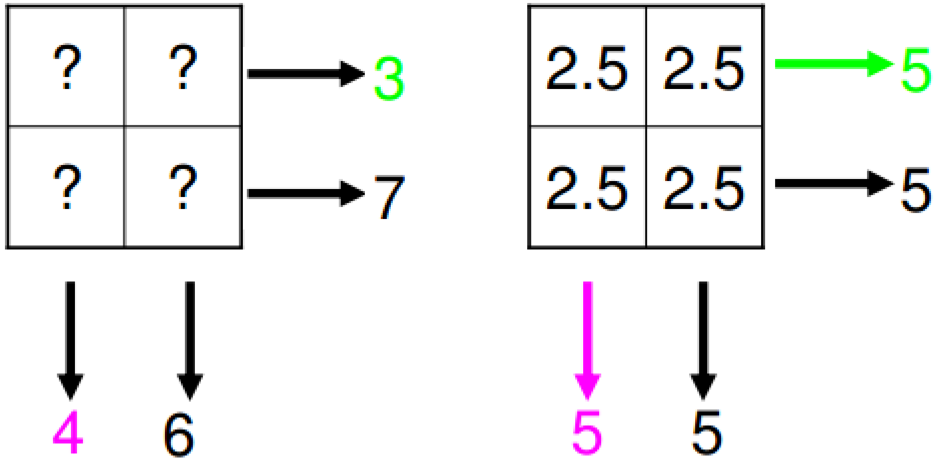
ML-EM Algorithm



x=(3+7)/4=10/4=**2.5**



ML-EM Algorithm

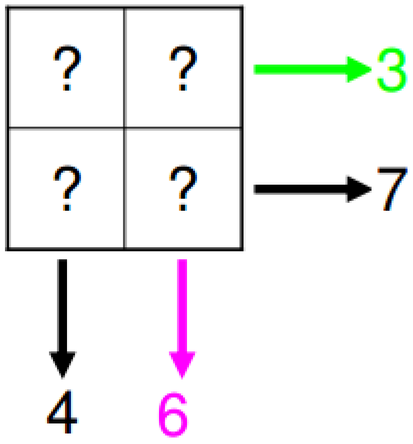


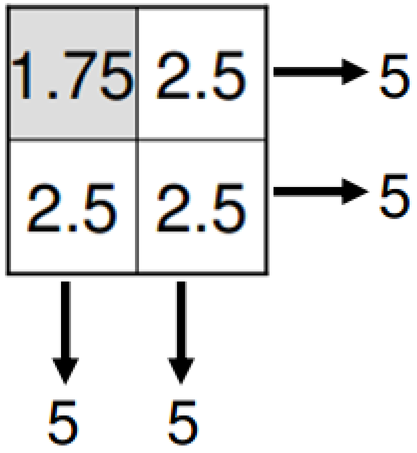
c11 = (3/5 + 4/5)/2 = 0.7

x11 = 0.7 x 2.5 = 1.75

1.75	2.5
2.5	2.5

ML-EM Algorithm



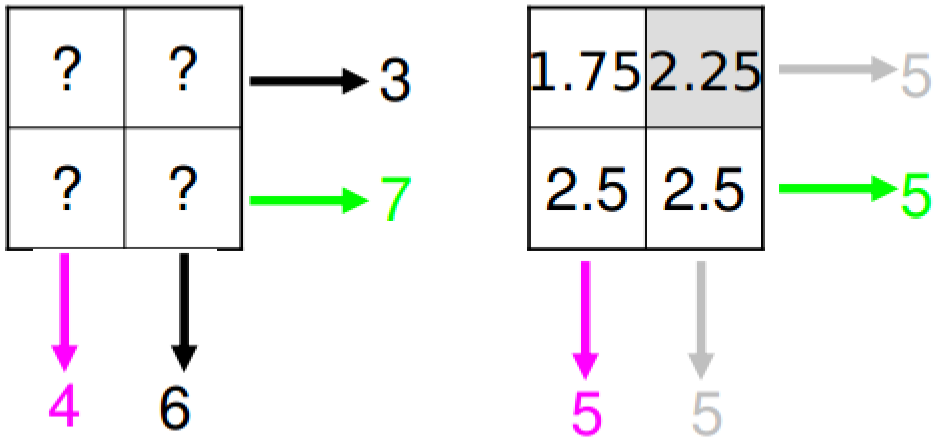


c12 = (3/5 + 6/5)/2 = 0.9

x12 = 2.25

1.75	2.25
2.5	2.5

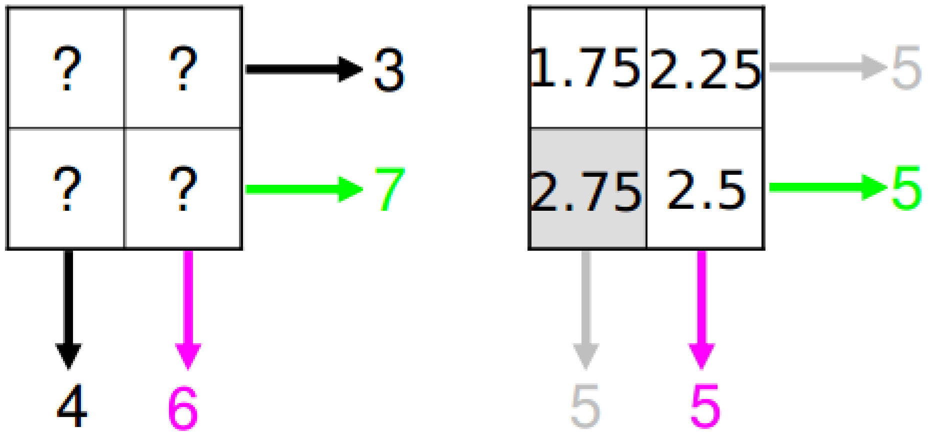
ML-EM Algorithm



c13 = (7/5 + 4/5)/2 = 1.1
x13 = 2.75

1.75	2.25
2.75	2.5

ML-EM Algorithm



c14 = (7/5 + 6/5)/2 = 1.3

x14 = 3.25

1.75	2.25
2.75	3.25