Data Consistency Conditions in Image Reconstruction from Projections

Rolf Clackdoyle

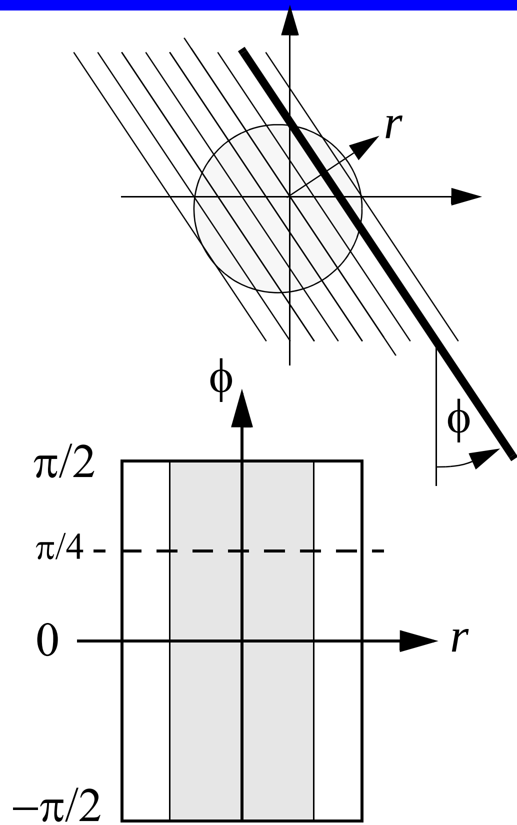
Hubert Curien Laboratory, CNRS UMR 5516, St. Etienne

Supported by ANR grant “DROITE” (ANR-12-BS01-0018)

Context

2D parallel projections:

2D parallel projections:



$p(\phi, \cdot)$ - parallel projection at angle ϕ

Context

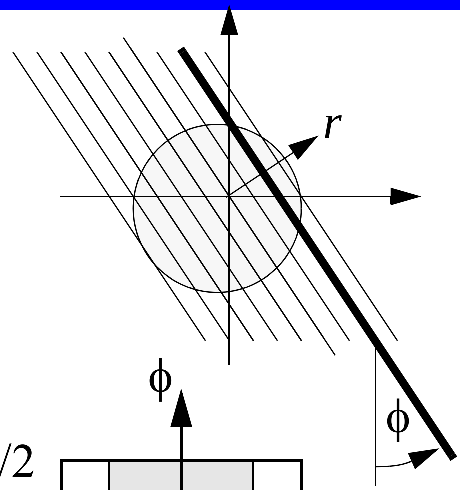
2D parallel projections:

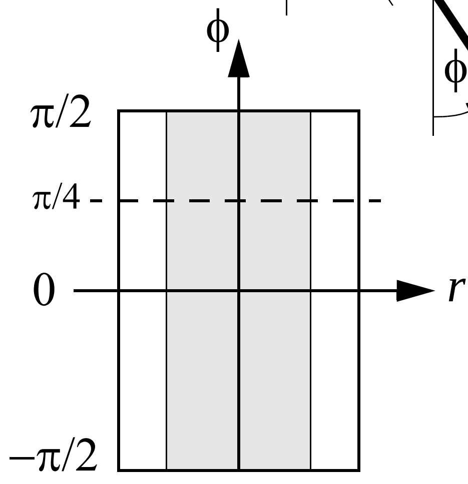
$$p(\phi, r) = \int f(r\alpha + s\beta)ds$$

$$\phi \in [-\pi/2, \pi/2]$$

$$\alpha = (\cos\phi, \sin\phi)$$

$$\beta = (-\sin\phi, \cos\phi)$$





$p(\phi, \cdot)$ - parallel projection at angle ϕ

For short.... $p = \boldsymbol{R}f$

Context

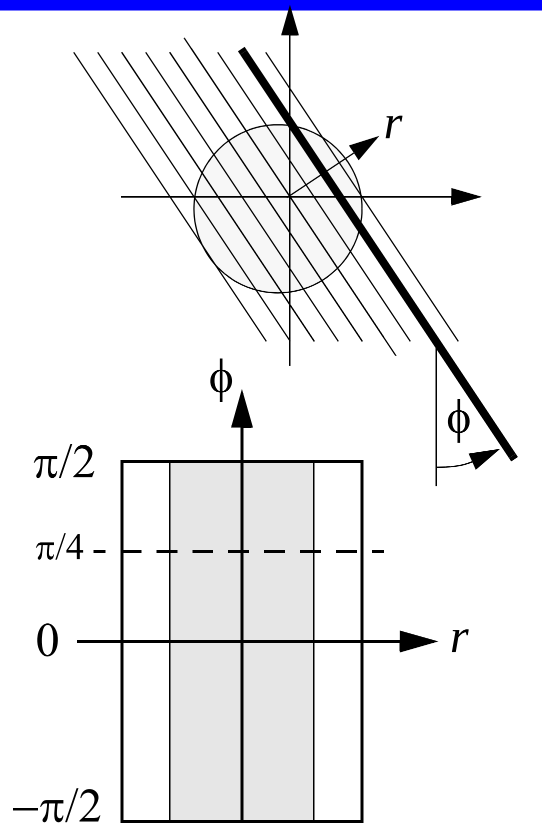
2D parallel projections:

$$p(\phi, r) = \int f(r\alpha + s\beta)ds$$

$$\phi \in [-\pi/2, \pi/2]$$

$$\alpha = (\cos\phi, \sin\phi)$$

$$\beta = (-\sin\phi, \cos\phi)$$



$p(\phi, \cdot)$ - parallel projection at angle ϕ

For short.... $p = \boldsymbol{R}f$

We are interested in **truncated** parallel projections.

Data Consistency Conditions (DCC)

Helgason-Ludwig DCC

$$\text{Let } M_0(\phi) = \int p(\phi, r) \quad dr.$$

Helgason-Ludwig DCC

$$\text{Let } M_0(\phi) = \int p(\phi, r) \quad dr.$$

If $p = \boldsymbol{R}f$ then $M_0(\phi) = \text{constant}$

Helgason-Ludwig DCC

$$\text{Let } M_0(\phi) = \int p(\phi, r) \quad dr.$$

If $p = \boldsymbol{R}f$ then $M_0(\phi) = \text{constant}$

$$\text{Let } M_n(\phi) = \int p(\phi, r)\; r^n \; dr.$$

Helgason-Ludwig DCC

$$\text{Let } M_0(\phi) = \int p(\phi, r) \quad dr.$$

If $p = \boldsymbol{R}f$ then $M_0(\phi) = \text{constant}$

$$\text{Let } M_n(\phi) = \int p(\phi, r)\, r^n\, dr.$$

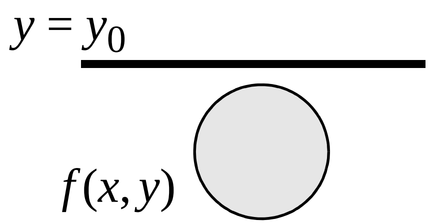
If $p = \boldsymbol{R}f$ then $M_n(\phi) = a_0 U^n + a_1 U^{n-1} V + \ldots a_n V^n$

where $U = \cos\phi$ and $V = \sin\phi$

New DCC

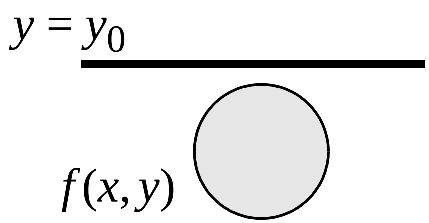
New DCC

Take some line outside the target object, w.l.o.g. the line $y = y_0$ with $y_0 > 0$.



New DCC

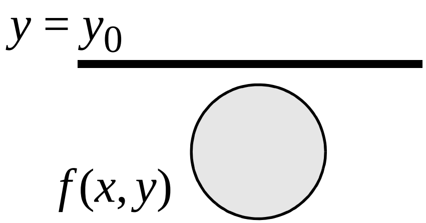
Take some line outside the target object, w.l.o.g. the line $y = y_0$ with $y_0 > 0$.



Let $B_0(x) = \int_0^{\pi} p(\phi, r^*) \dfrac{1}{\cos\phi}\, d\phi$, where $r^* = x\cos\phi + y_0 \sin\phi$.

New DCC

Take some line outside the target object, w.l.o.g. the line $y = y_0$ with $y_0 > 0$.

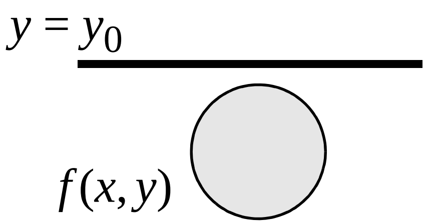


Let $B_0(x) = \int_0^{\pi} p(\phi, r^*) \dfrac{1}{\cos\phi}\, d\phi$, where $r^* = x\cos\phi + y_0 \sin\phi$.

If $p = \boldsymbol{R}f$ then $B_0(x) =$ constant

New DCC

Take some line outside the target object, w.l.o.g. the line $y = y_0$ with $y_0 > 0$.



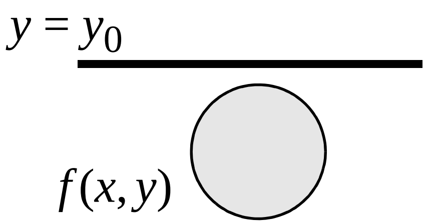
$$\text{Let } B_0(x) = \int_0^{\pi} p(\phi, r^*)\,\frac{1}{\cos\phi}\,d\phi, \text{ where } r^* = x\cos\phi + y_0\sin\phi.$$

If $p = \boldsymbol{R}f$ then $B_0(x) = \text{constant}$

$$\text{Let } B_n(x) = \int_0^{\pi} p(\phi, r^*)\,\frac{\tan^n\phi}{\cos\phi}\,d\phi, \text{ where } r^* = x\cos\phi + y_0\sin\phi.$$

New DCC

Take some line outside the target object, w.l.o.g. the line $y = y_0$ with $y_0 > 0$.



Let $B_0(x) = \int_0^{\pi} p(\phi, r^*) \dfrac{1}{\cos\phi}\, d\phi$, where $r^* = x\cos\phi + y_0 \sin\phi$.

If $p = \boldsymbol{R} f$ then $B_0(x) = \text{constant}$

Let $B_n(x) = \int_0^{\pi} p(\phi, r^*) \dfrac{\tan^n\phi}{\cos\phi}\, d\phi$, where $r^* = x\cos\phi + y_0 \sin\phi$.

If $p = \boldsymbol{R} f$ then $B_n(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n$

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\ r^n\ dr$$

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\ r^n\ dr$$

no truncation allowed

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\, r^n\, dr$$

no truncation allowed

(limited-angle OK)

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\, r^n\, dr \qquad B_n(x) = \int_0^{\pi} p(\phi, r^*)\, \frac{\tan^n \phi}{\cos \phi}\, d\phi$$

where $r^* = x\cos\phi + y_0 \sin\phi$

no truncation allowed

(limited-angle OK)

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\, r^n\, dr \qquad B_n(x) = \int_0^\pi p(\phi, r^*)\, \frac{\tan^n\phi}{\cos\phi}\, d\phi$$

where $r^* = x\cos\phi + y_0\sin\phi$

no truncation allowed

(limited-angle OK)

no limited-angle allowed

Comparing Helgason-Ludwig DCC and the new DCC

$$M_n(\phi) = \int p(\phi, r)\, r^n \, dr \qquad B_n(x) = \int_0^{\pi} p(\phi, r^*) \frac{\tan^n \phi}{\cos \phi}\, d\phi$$

where $r^* = x\cos\phi + y_0 \sin\phi$

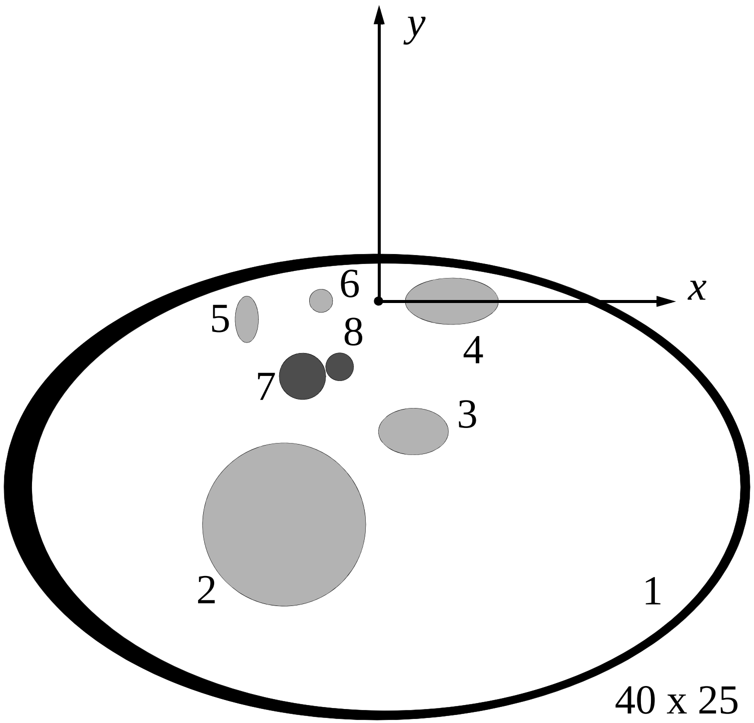
no truncation allowed

(limited-angle OK)

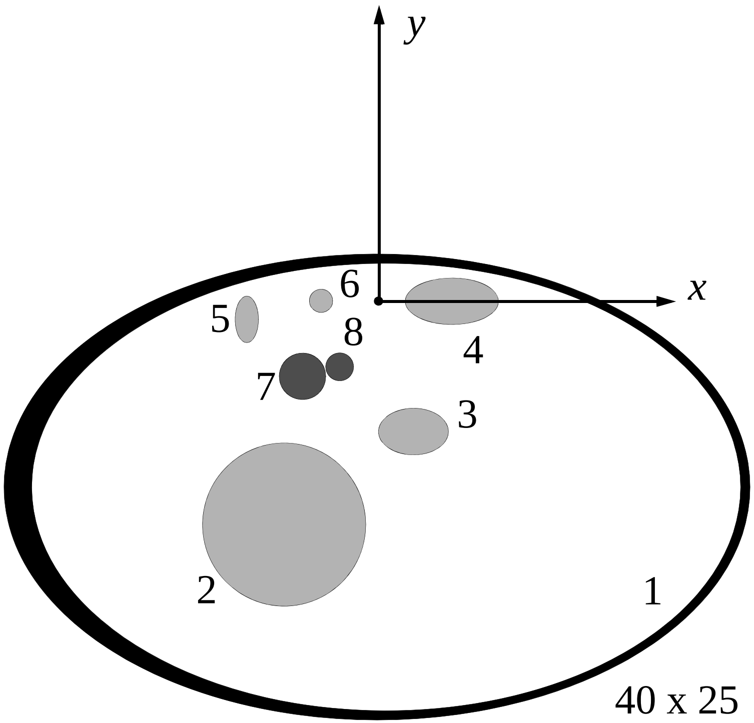
no limited-angle allowed

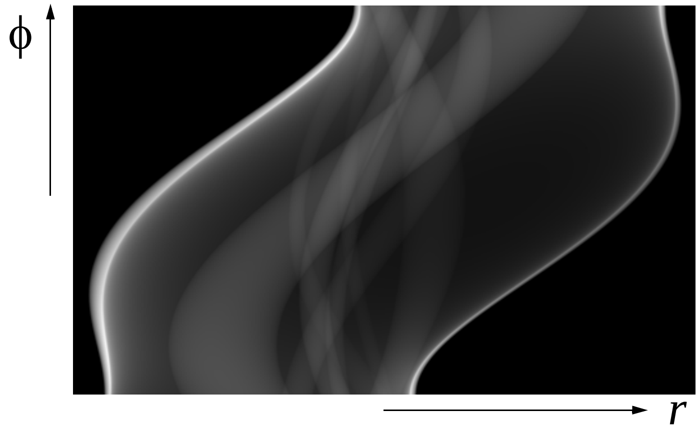
(<u>truncation OK</u>)

Simulations

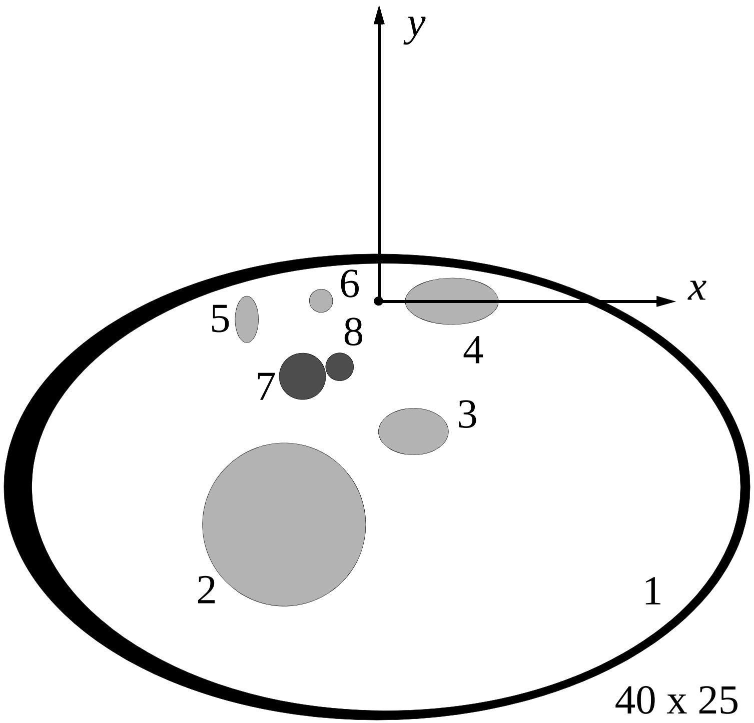


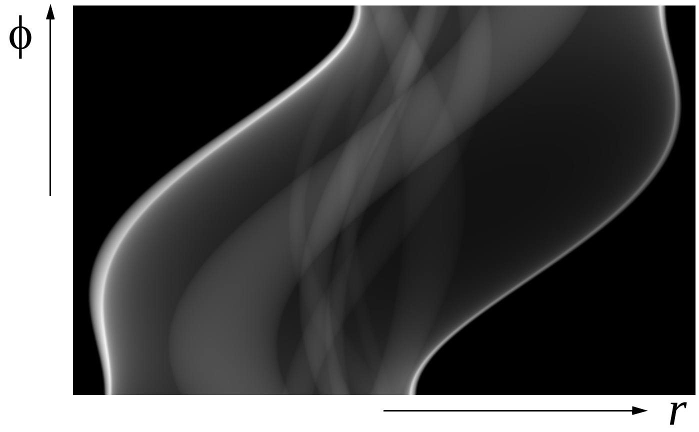
Simulations

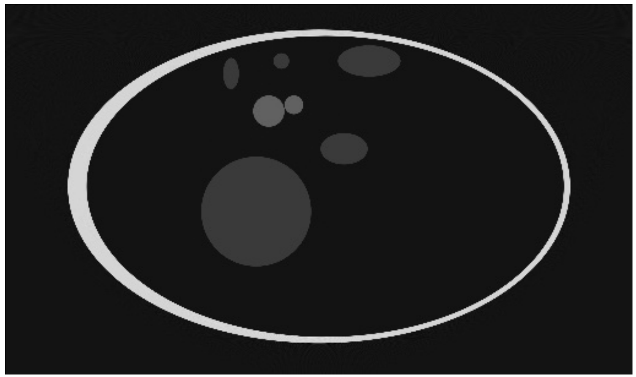




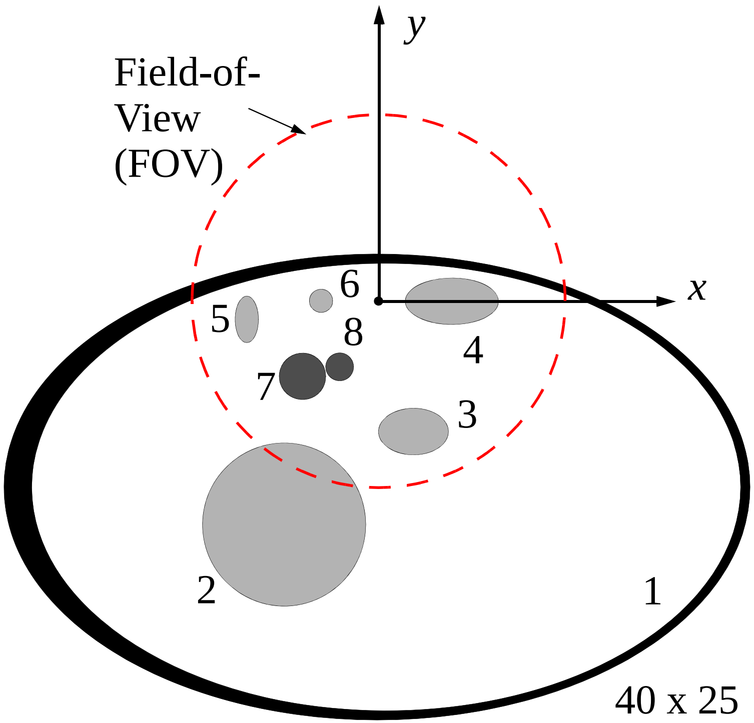
Simulations

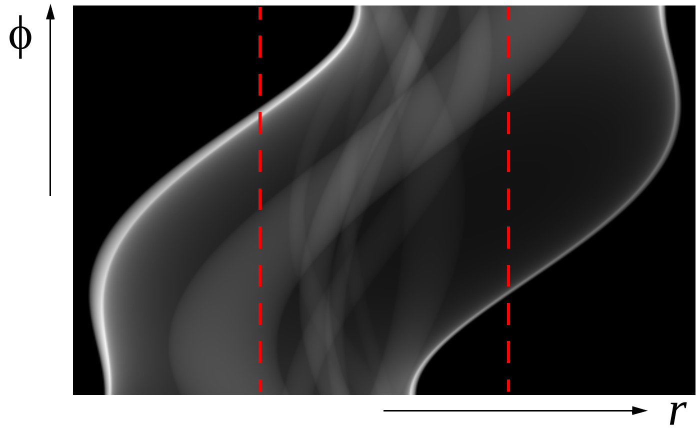




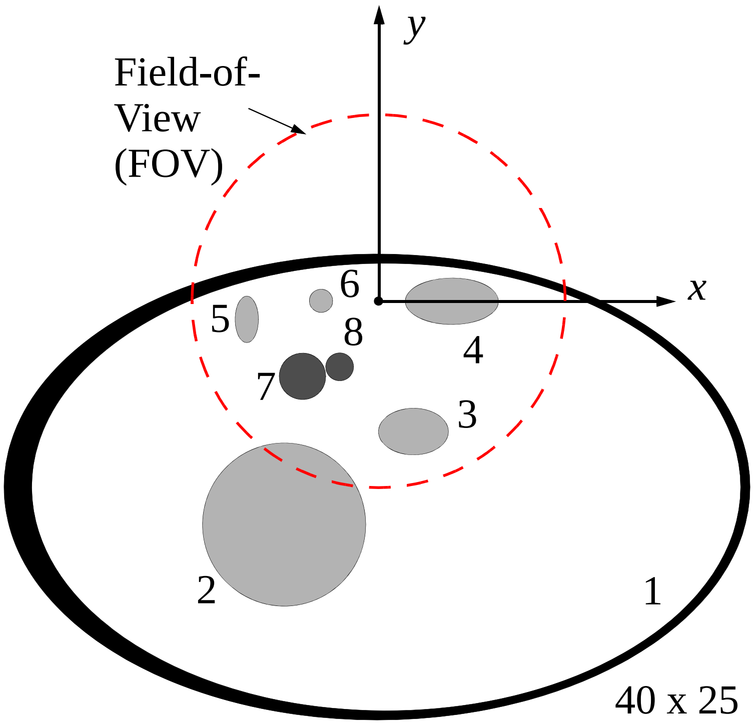


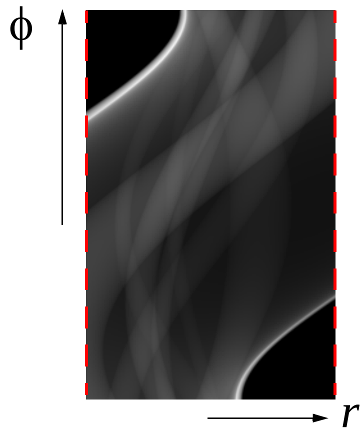
Simulations



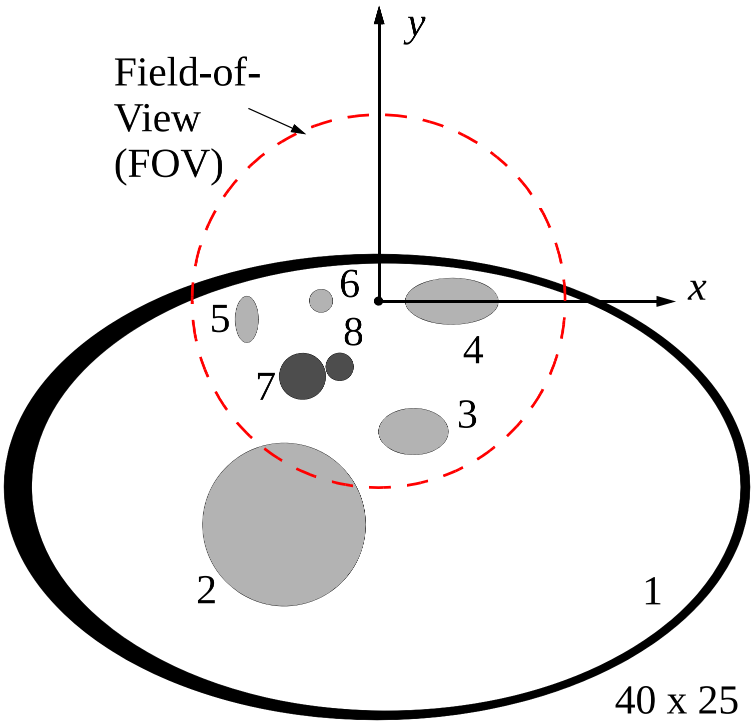


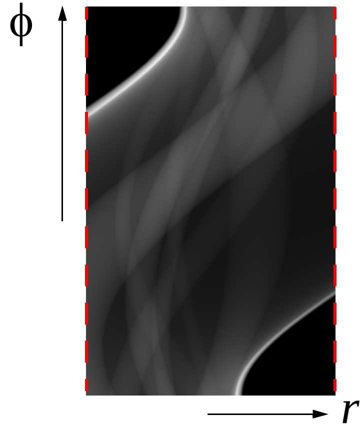
Simulations

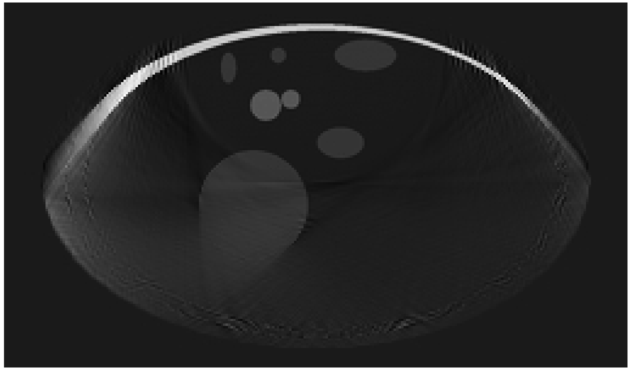




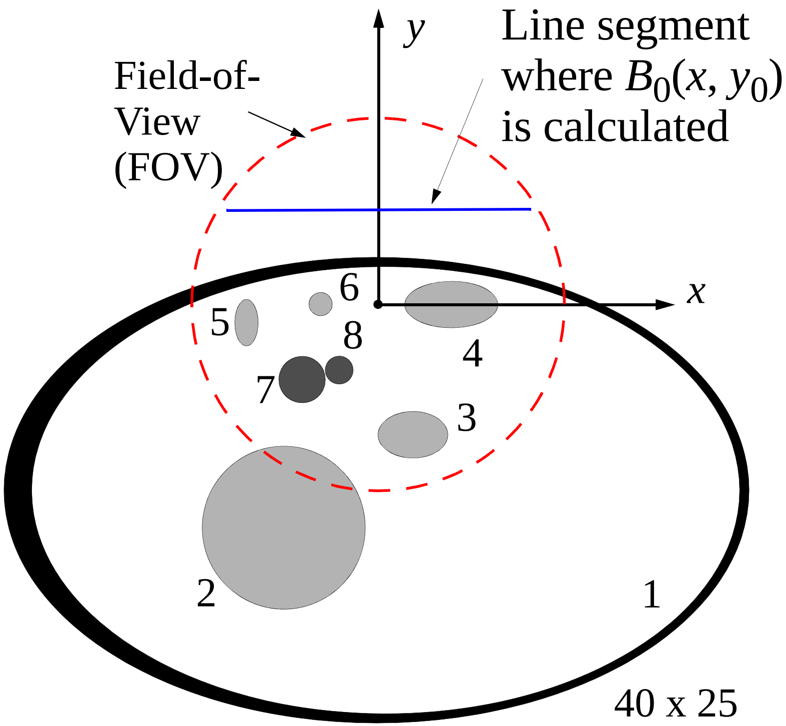
Simulations

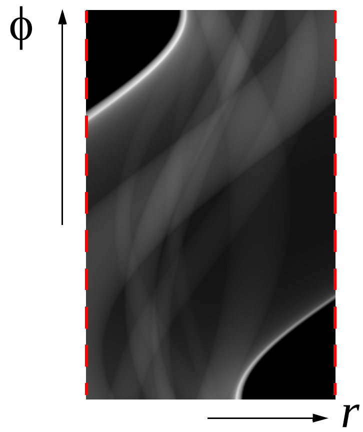




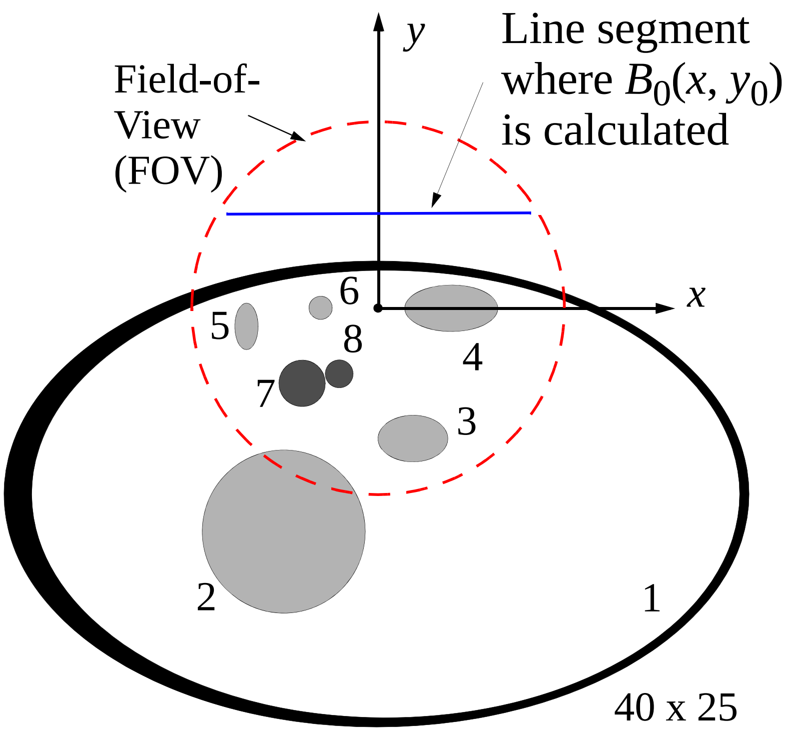


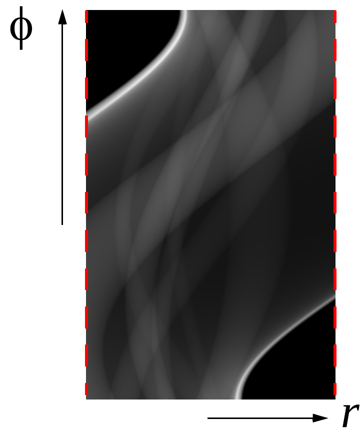
Simulations

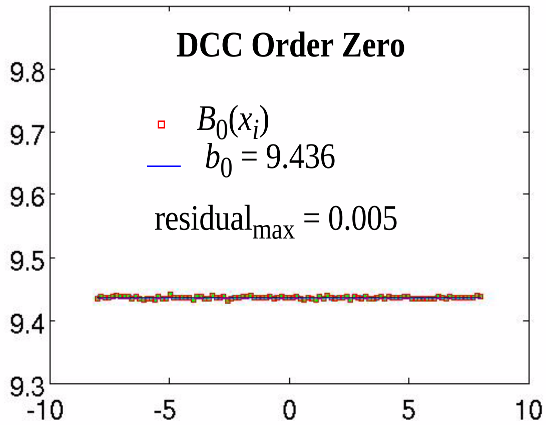




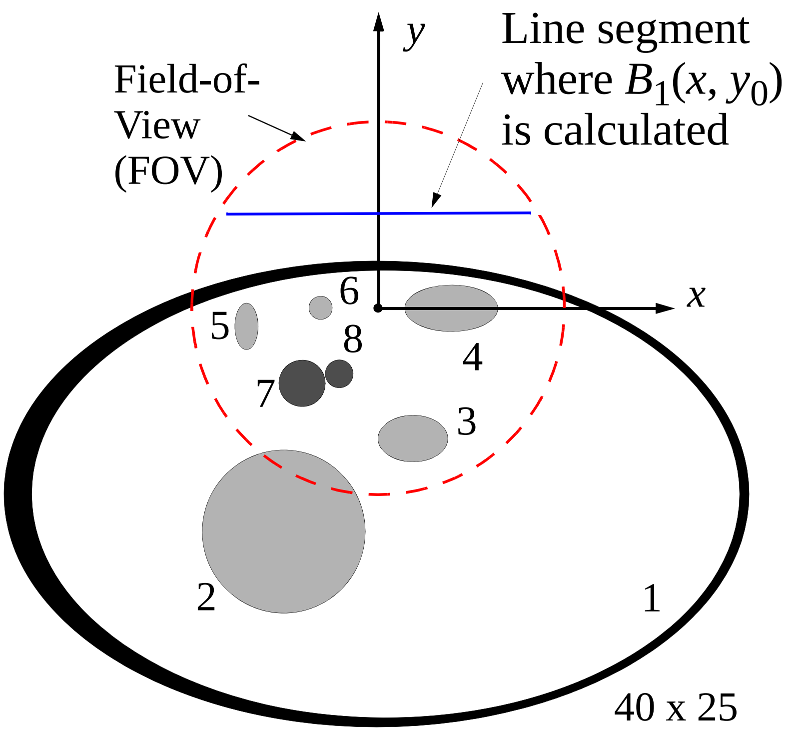
Simulations

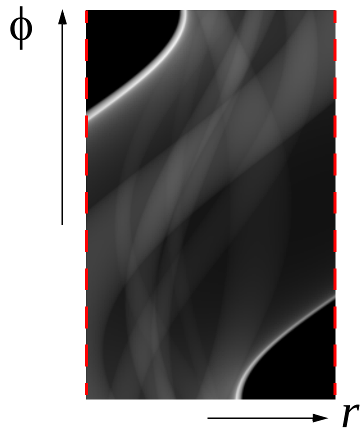




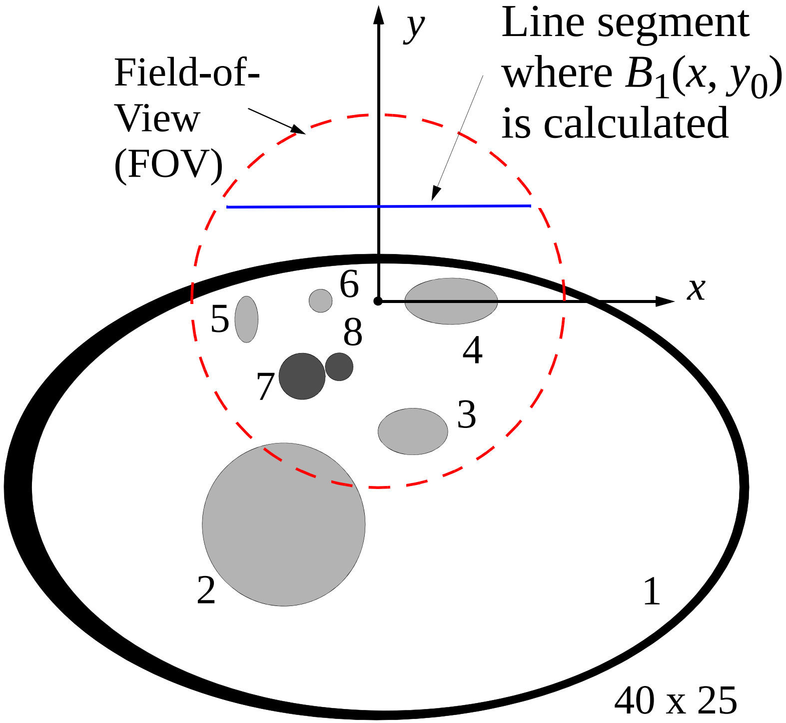


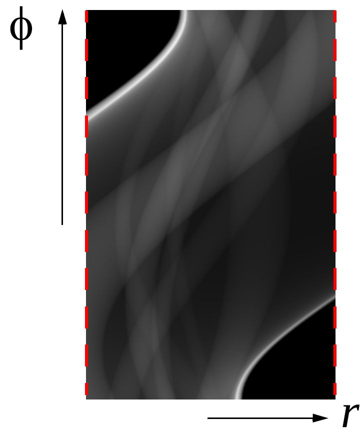
Simulations

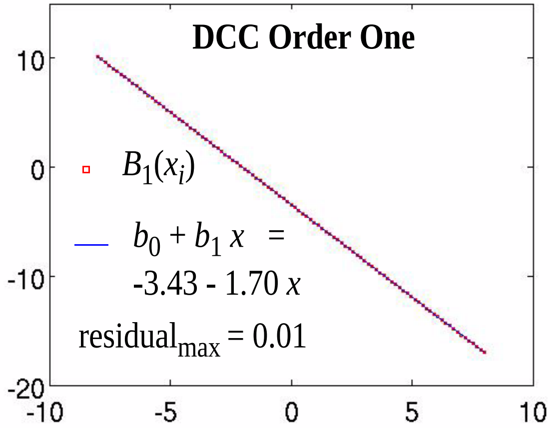




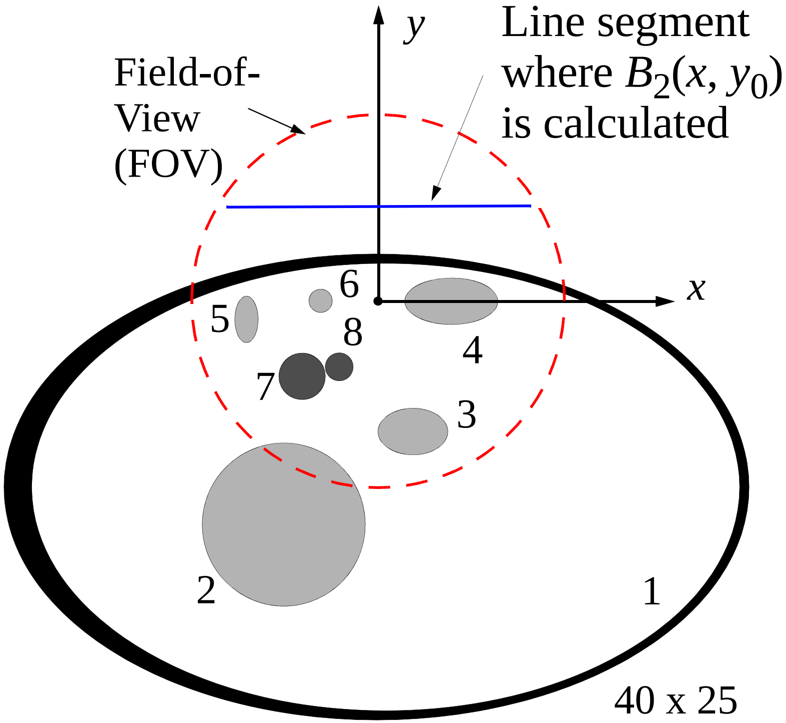
Simulations

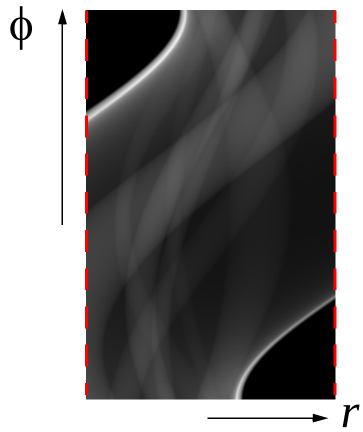




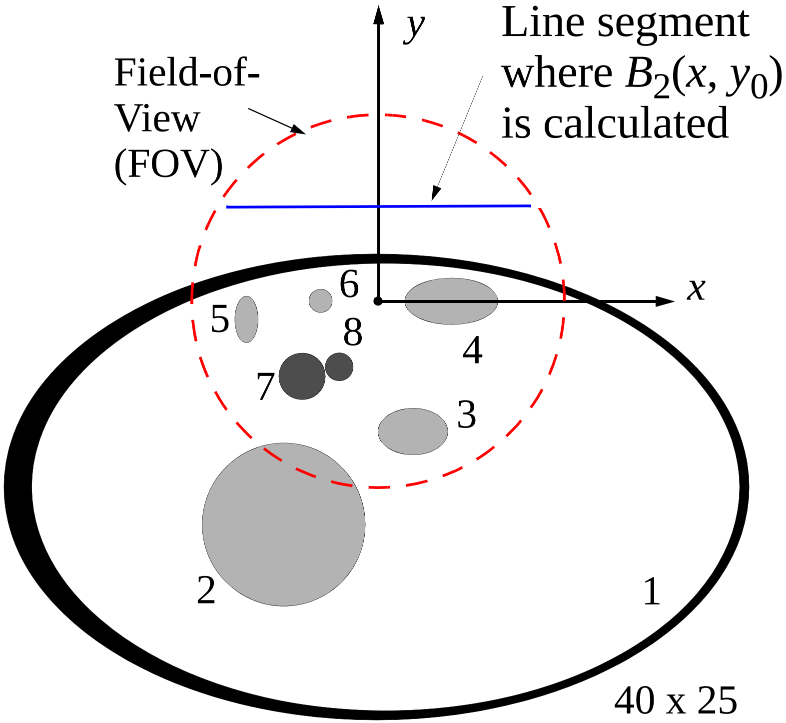


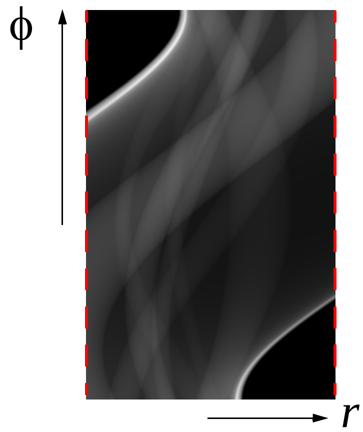
Simulations

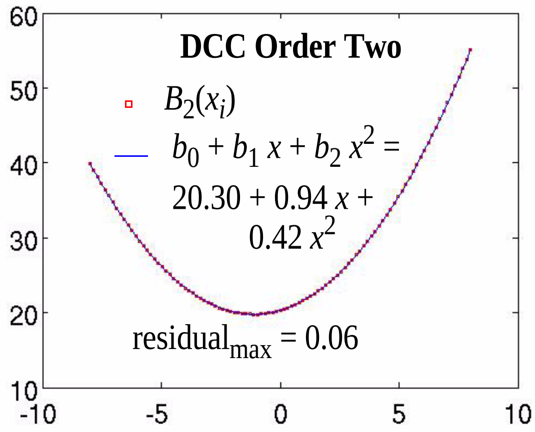




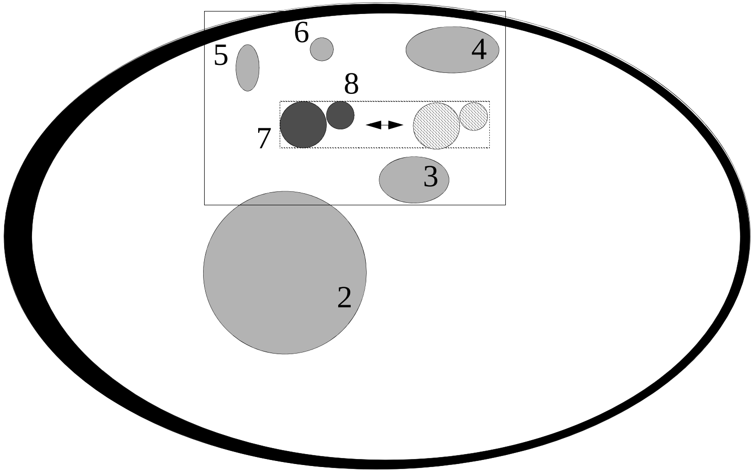
Simulations

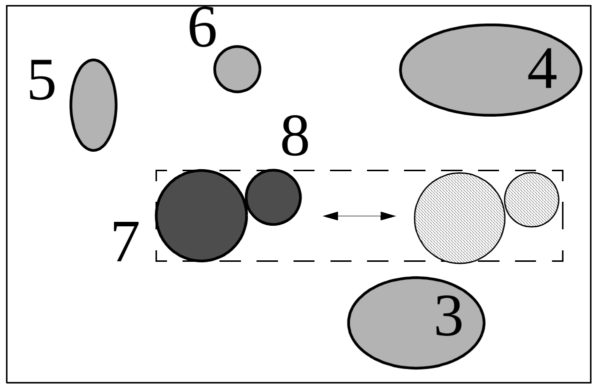


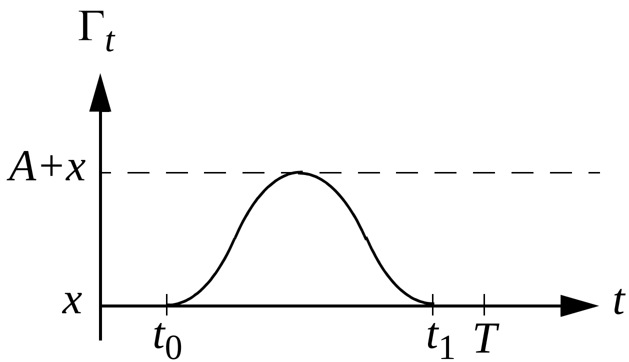




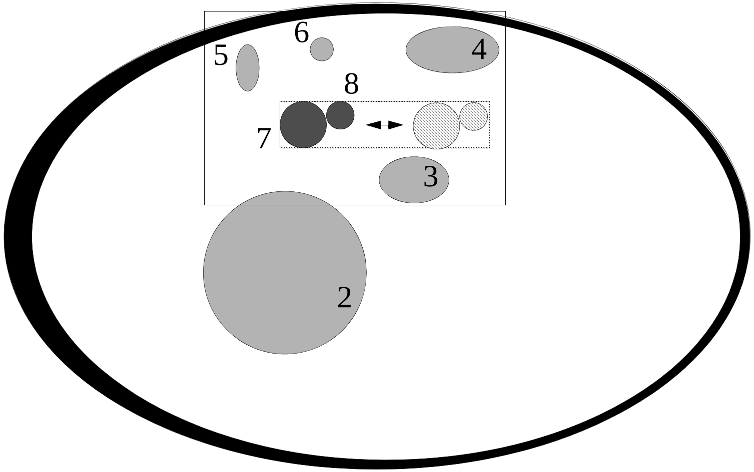
Simulations

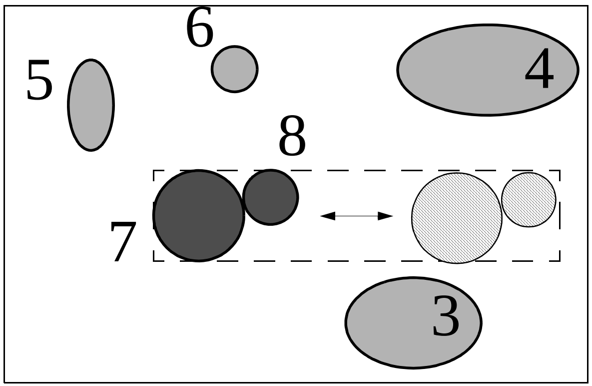


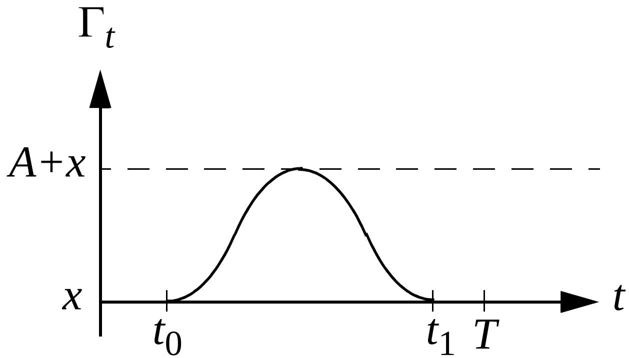




Simulations

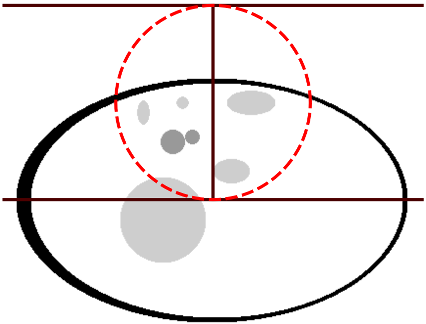


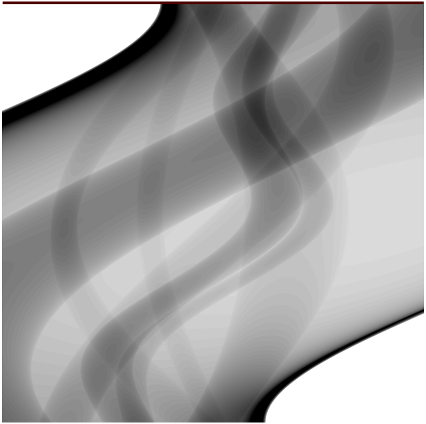




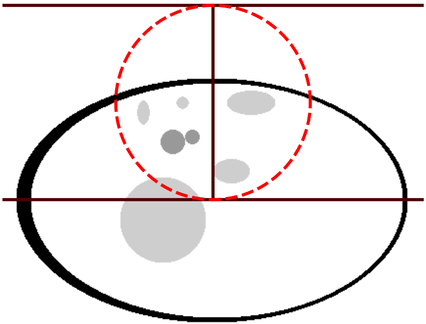
$$T = 18, \quad (t_0\,,\,t_1) = (2\,,\,17), \;\; A = 7$$

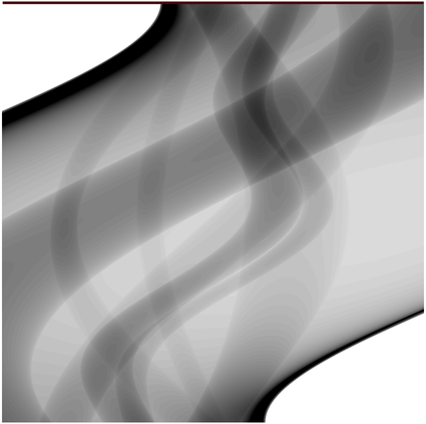
Dynamic phantom and parallel geometry



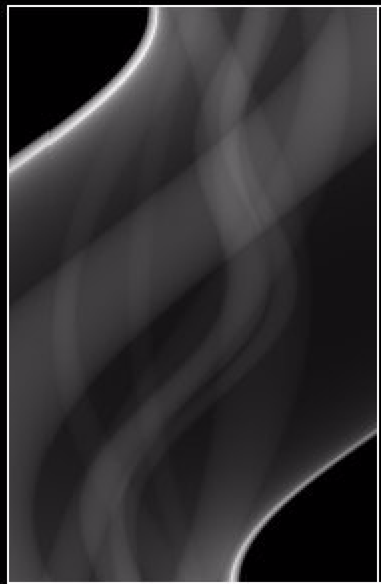


Dynamic phantom and parallel geometry

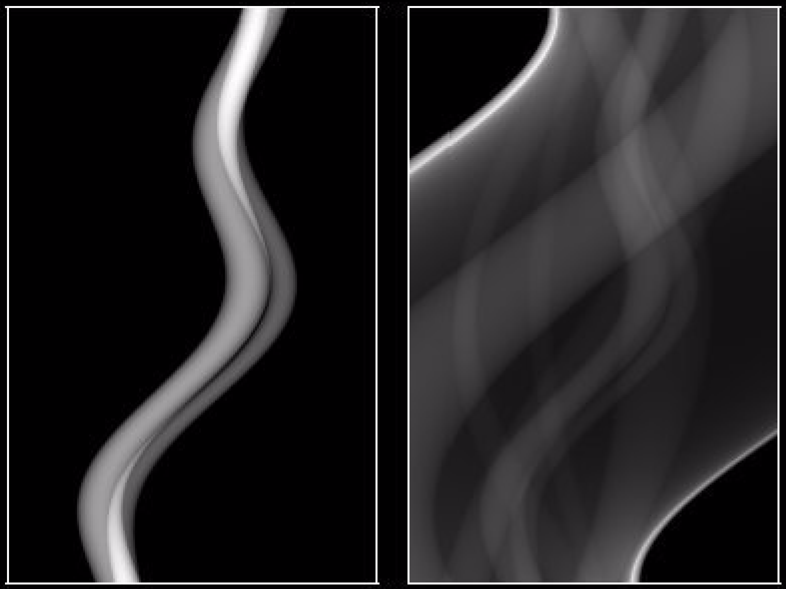




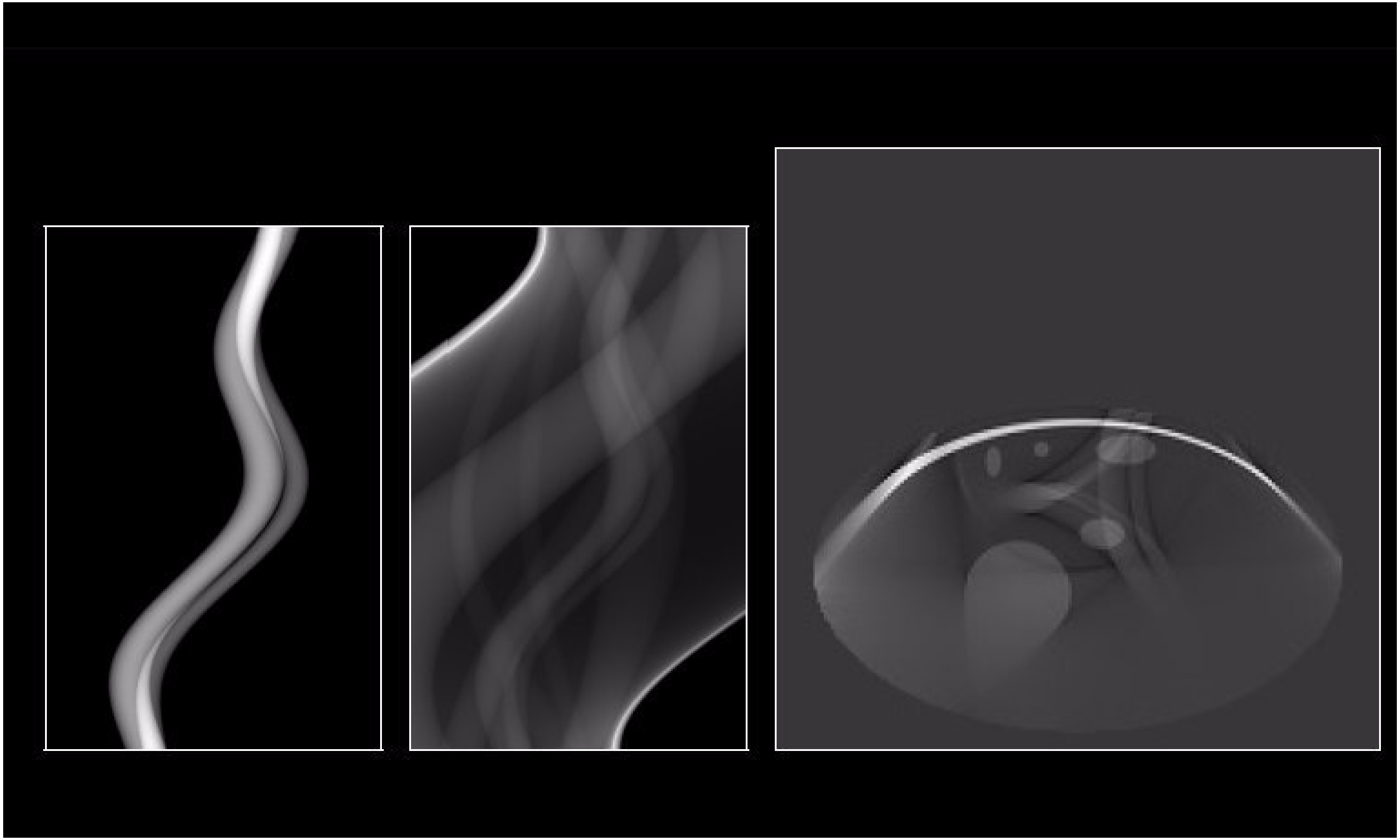
Simulations



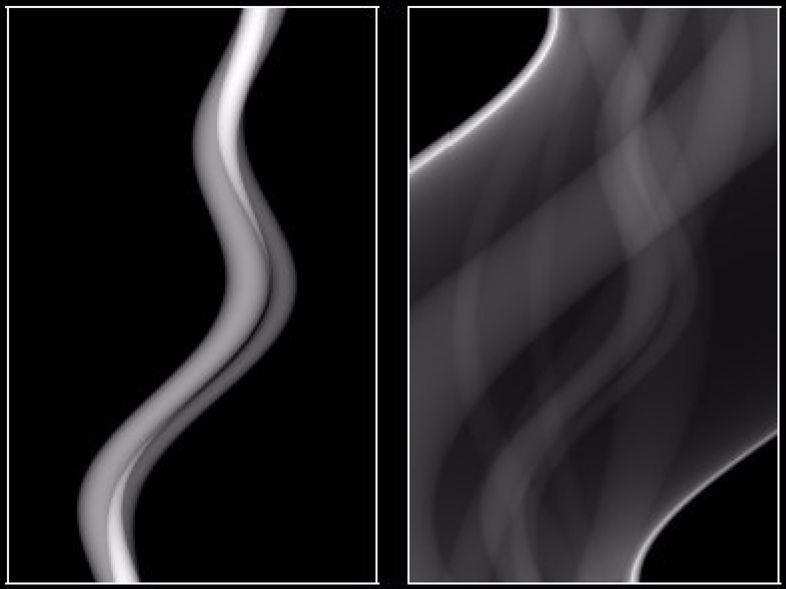
Simulations



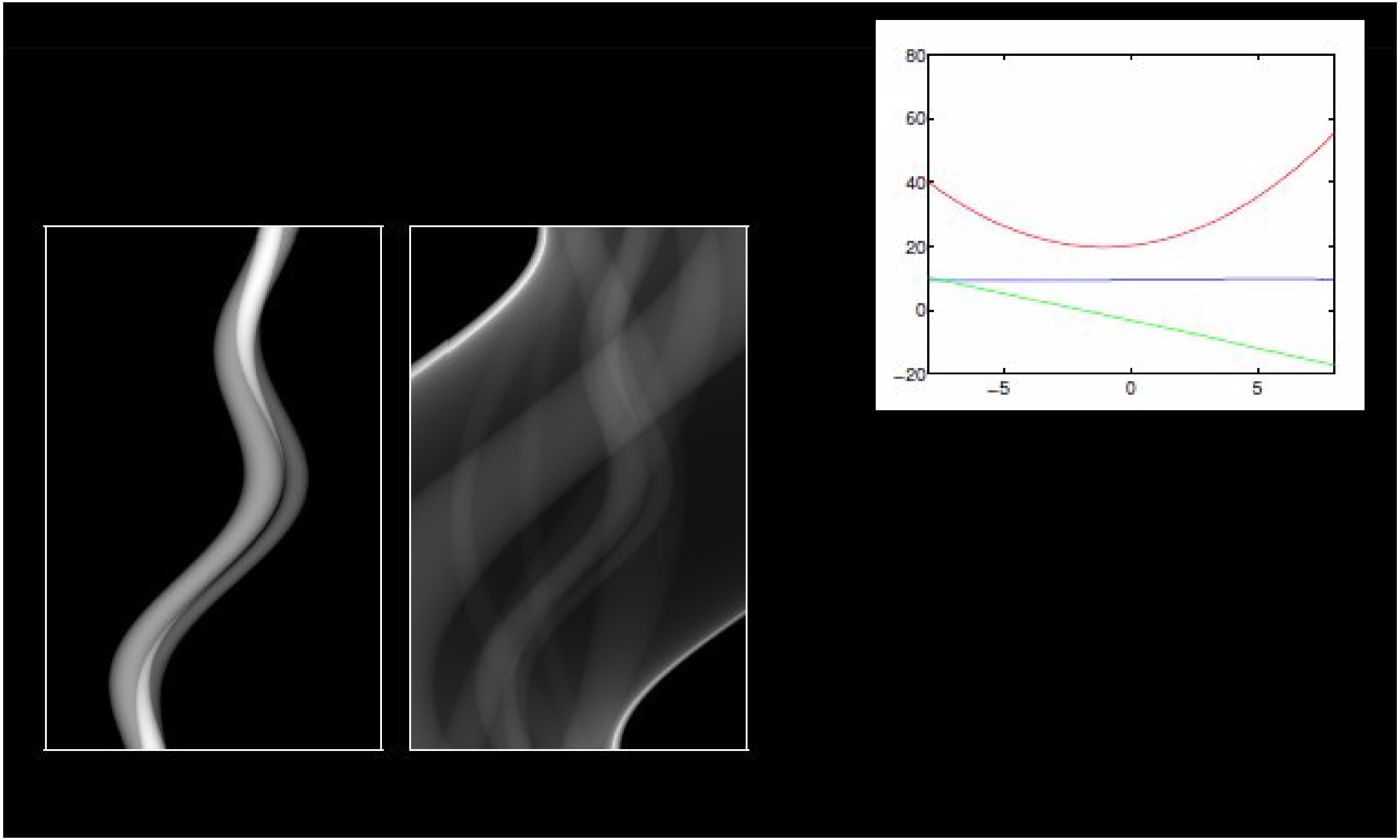
Simulations



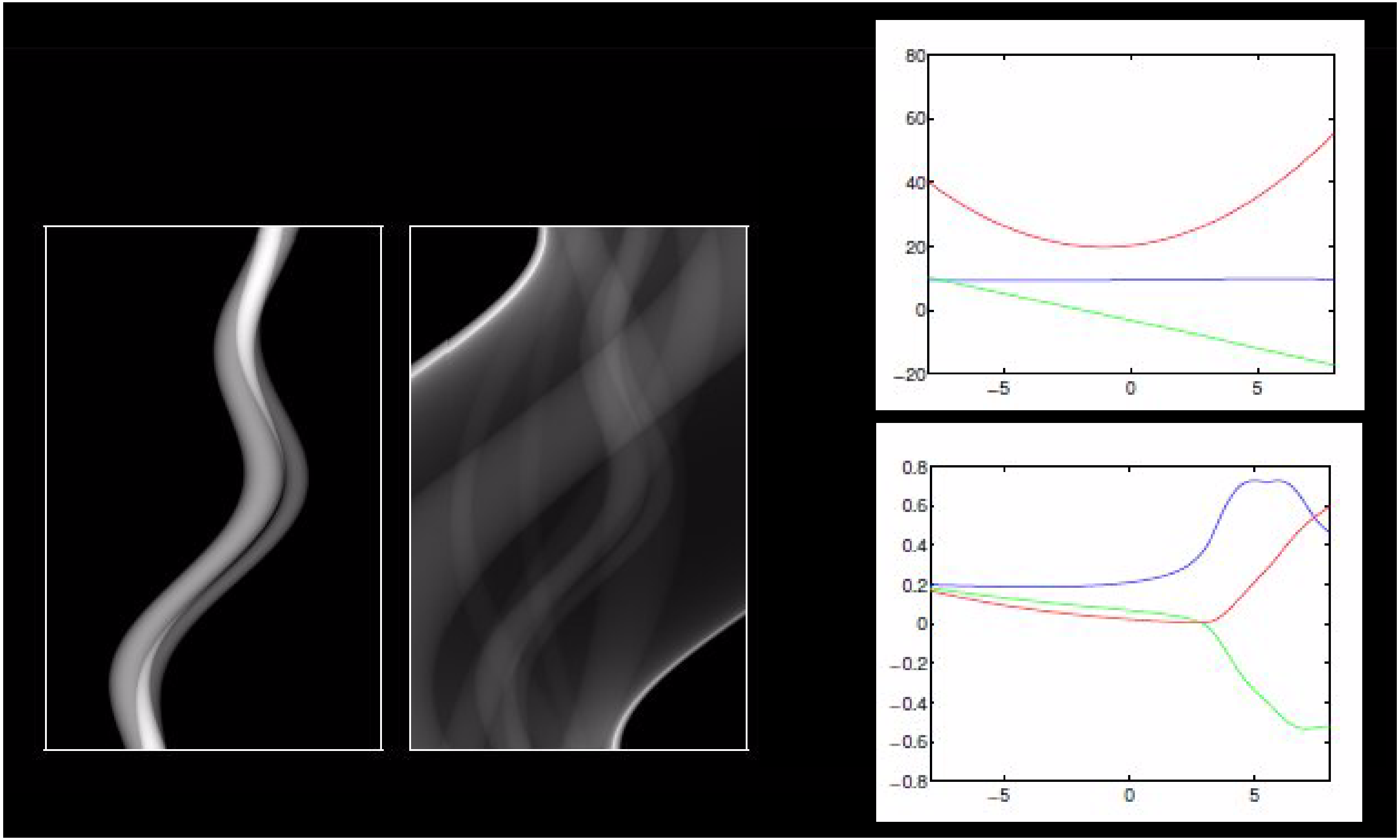
Simulations



Simulations



Simulations



Simulations

Search for the (non-linear) parameters (t_0, t_1) and A such that the DCC are "best satisfied" (are polynomials of the correct degree), when the moving component is subtracted from the data.

(The alternative, not using DCC, is to solve for (t_0, t_1), A, and $f(x, y)$ - which combines a small non-linear problem with a huge linear problem.)

Simulations

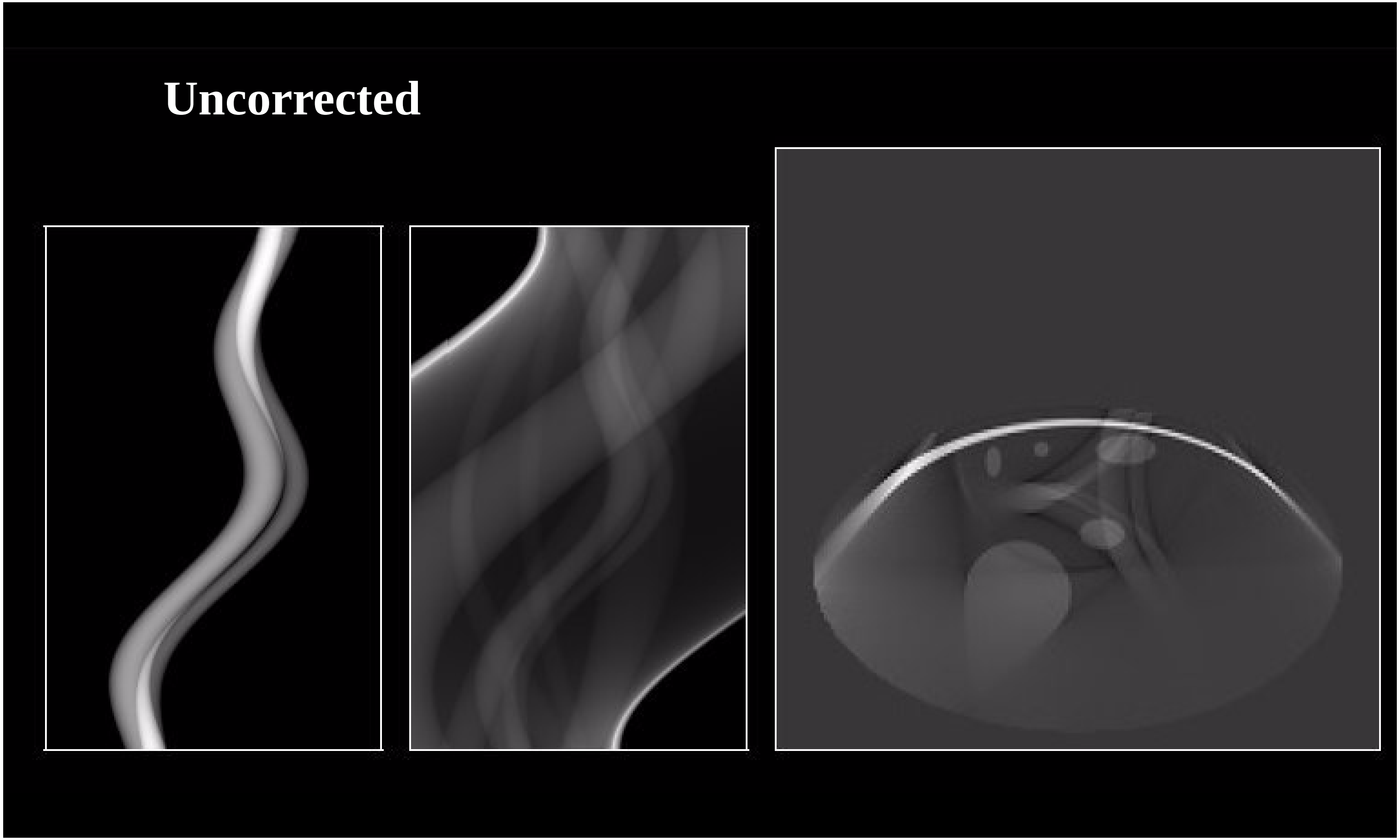
Search for the (non-linear) parameters (t_0, t_1) and A such that the DCC are "best satisfied" (are polynomials of the correct degree), when the moving component is subtracted from the data.

(The alternative, not using DCC, is to solve for (t_0, t_1), A, and $f(x, y)$ - which combines a small non-linear problem with a huge linear problem.)

The non-linear estimation found $(\tilde{t}_0, \tilde{t}_1) = (1.98, 16.99)$ and $\tilde{A} = 7.03$

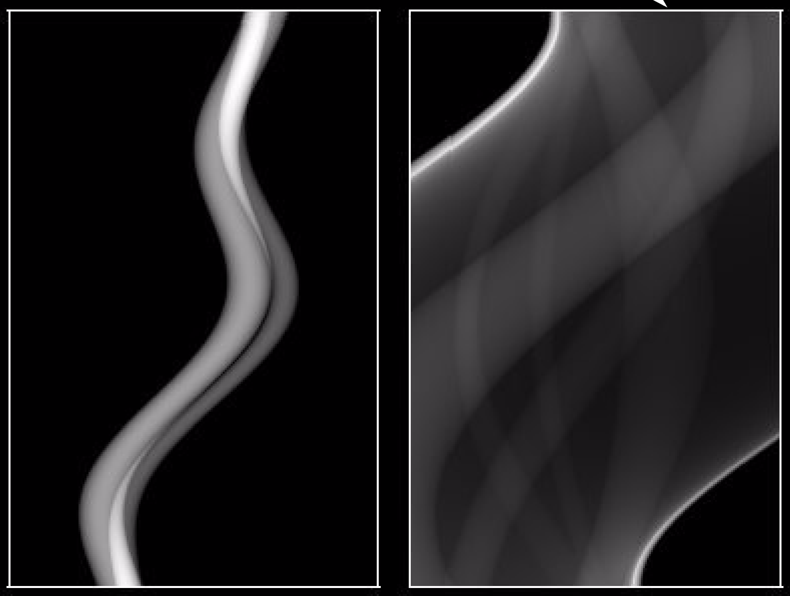
(The true values were $(t_0^*, t_1^*) = (2, 17)$ and $A^* = 7$)

Simulations

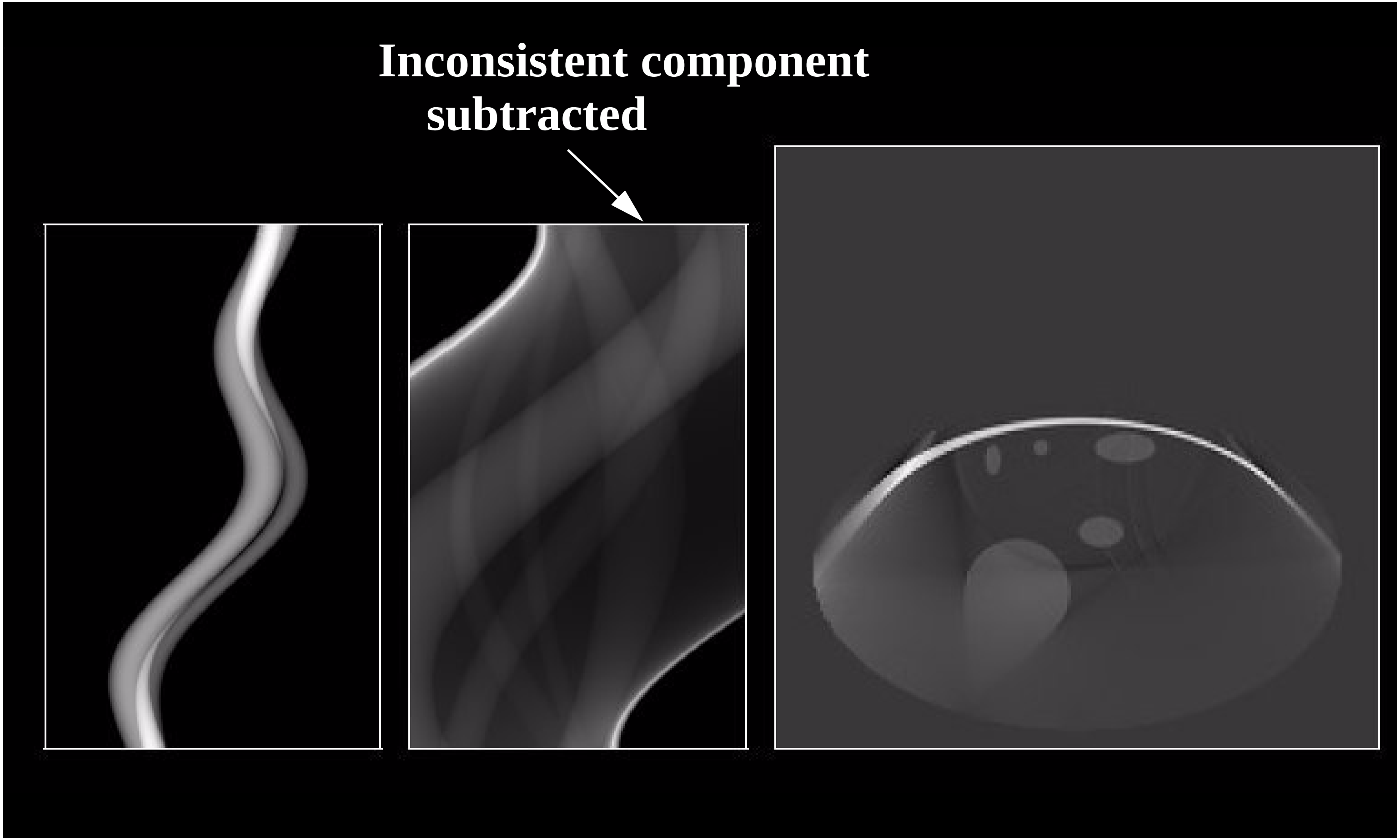


Simulations

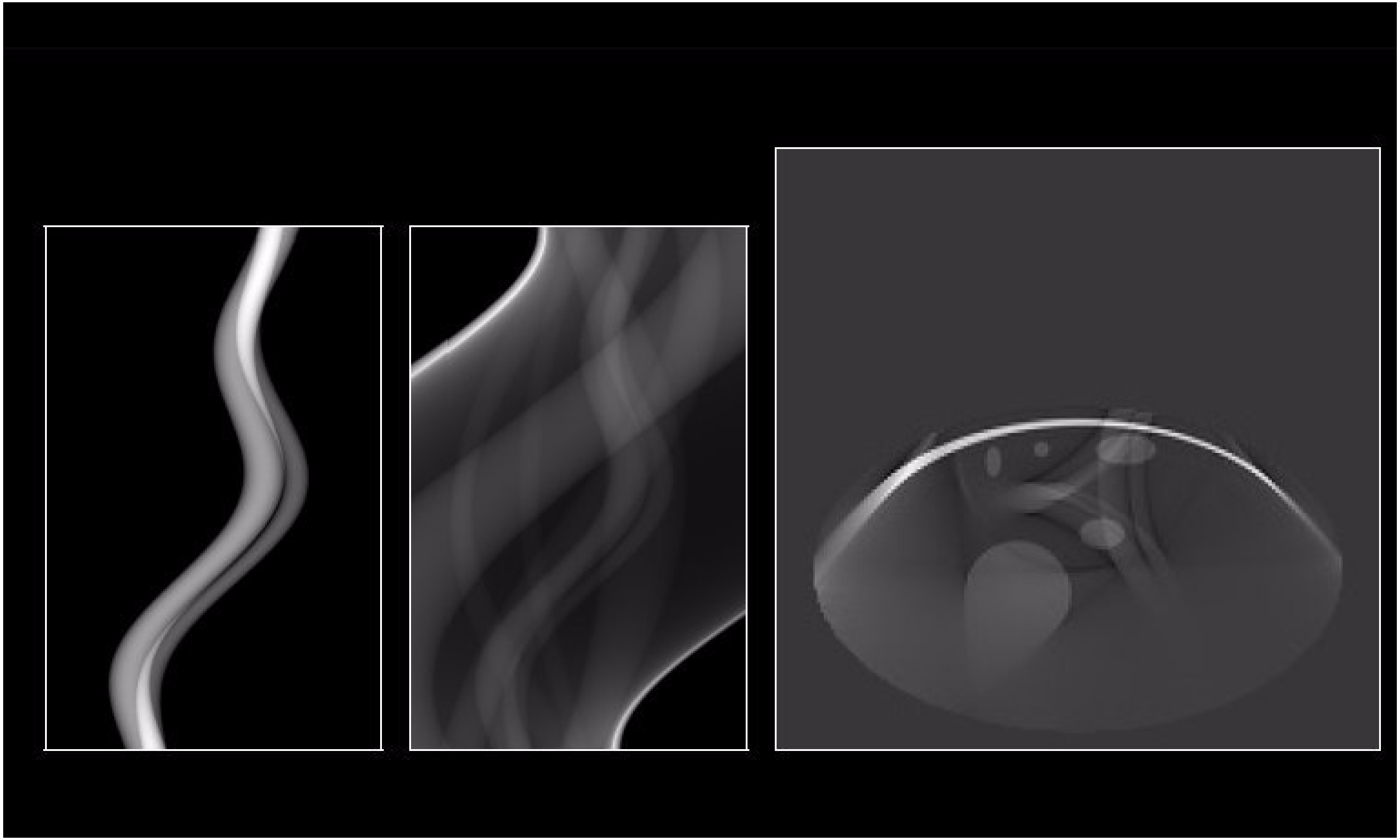
Inconsistent component subtracted



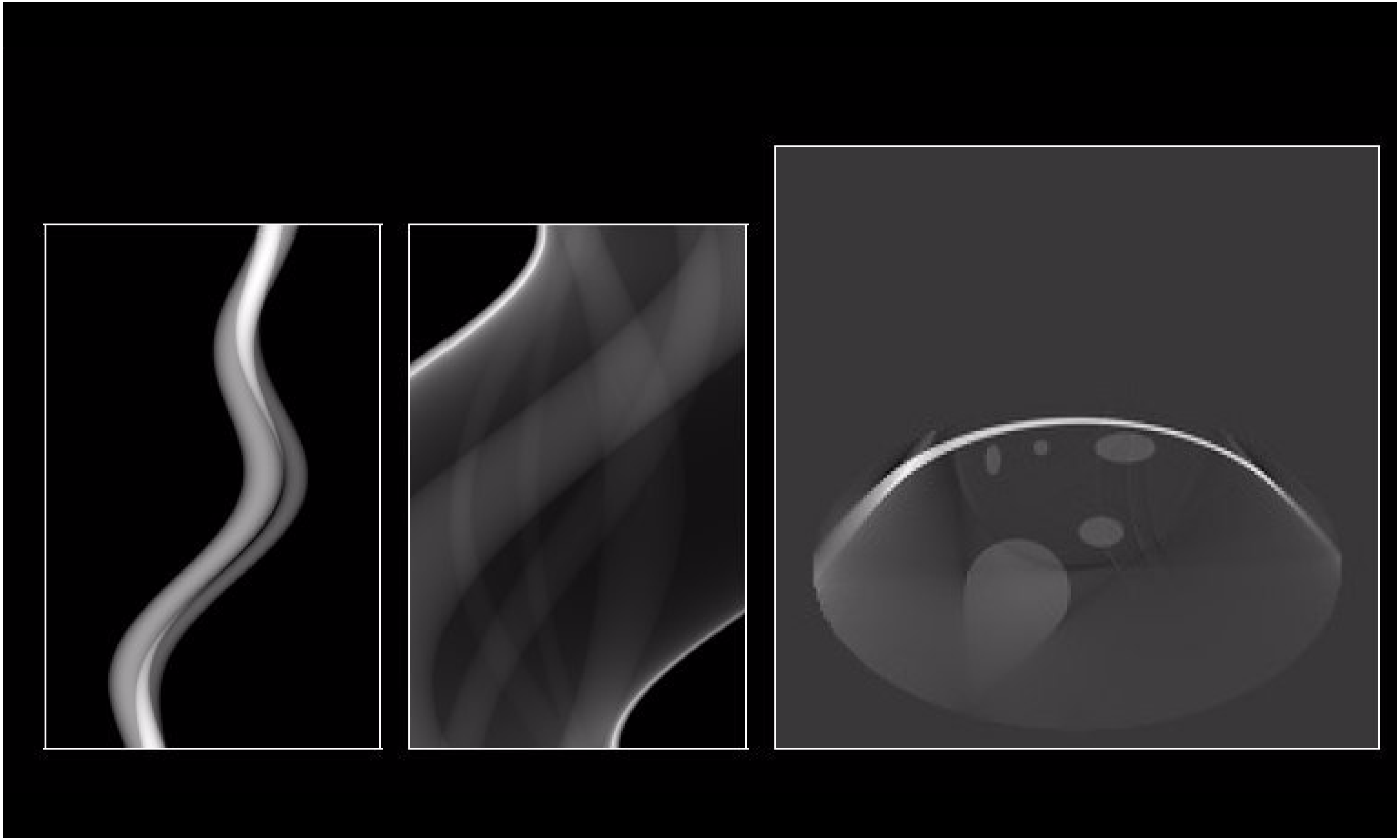
Simulations



Simulations



Simulations



Data Consistency Conditions - Summary

Data Consistency Conditions - Summary

- Data Consistency Conditions (DCC) express redundancies between projections

Data Consistency Conditions - Summary

- Data Consistency Conditions (DCC) express redundancies between projections
- Checking the DCC is a way to see if the data follows the forward model

Data Consistency Conditions - Summary

- Data Consistency Conditions (DCC) express redundancies between projections
- Checking the DCC is a way to see if the data follows the forward model
- The advantage of the DCC is that they can be checked without having to perform reconstruction (i.e. much quicker)

Data Consistency Conditions - Summary

- Data Consistency Conditions (DCC) express redundancies between projections
- Checking the DCC is a way to see if the data follows the forward model
- The advantage of the DCC is that they can be checked without having to perform reconstruction (i.e. much quicker)

- New DCC for truncated projections have been described (2014)

Data Consistency Conditions - Summary

- Data Consistency Conditions (DCC) express redundancies between projections
- Checking the DCC is a way to see if the data follows the forward model
- The advantage of the DCC is that they can be checked without having to perform reconstruction (i.e. much quicker)

- New DCC for truncated projections have been described (2014)
- DCC were successfully used to estimate non-linear motion parameters

Thanks for your attention!