

# CASToR v3.0

## Features



**Release: 1<sup>st</sup> half November 2019**



# Features v3.0



- **Dynamic reconstruction**

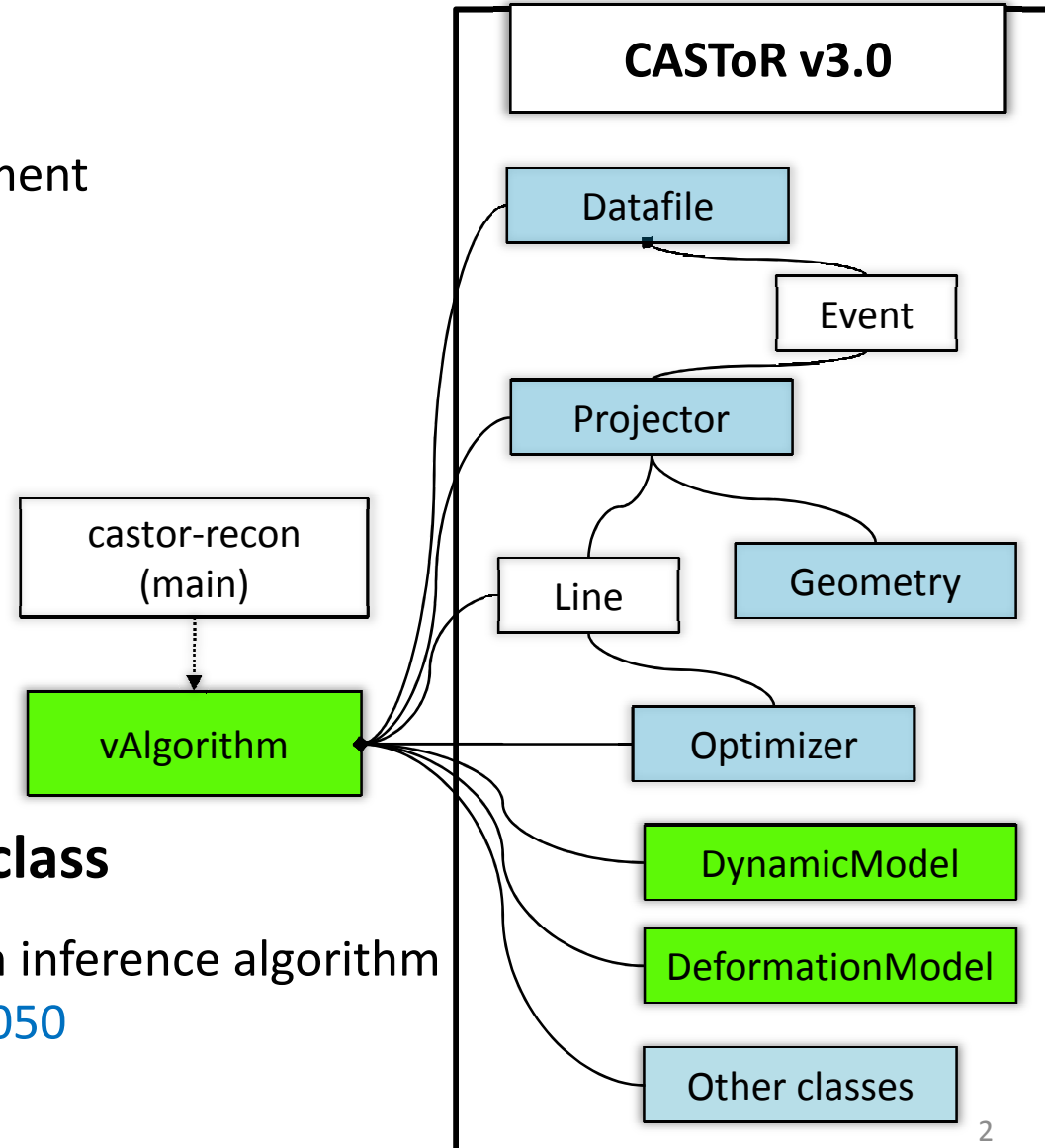
- Dynamic acquisition management
- Dynamic Model class
- Deformation Model class

- **TOF management update**

- Improved implementation

- **Generic iterative algorithm class**

- RCP-GS: probabilistic Bayesian inference algorithm  
DOI: [10.1109/TMI.2018.2886050](https://doi.org/10.1109/TMI.2018.2886050)





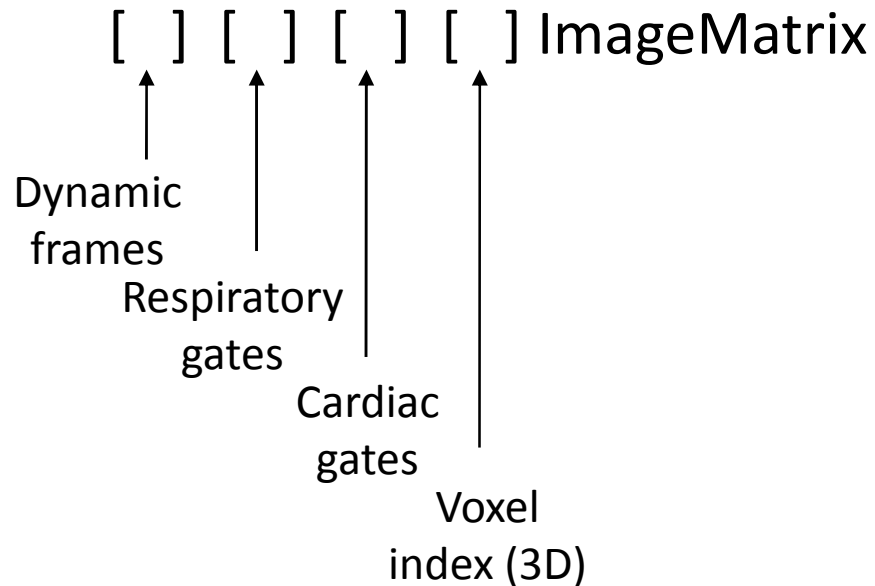
# Dynamic architecture

---

- **Dynamic dataset reconstruction**

- Dynamic acquisition: temporal sampling in **frames**
- Gated acquisition: datafile rebinning in **gates**

- **6D image dimensions**



Applications:

- Dynamic recon.
- Kinetic modelling
- Gated, dual-gated recon.
- Motion correction



# Dynamic data management: **Frames**

- **Command-line options:** `-frm 0,30,60,90:30`  
↑ ↑ ↑ ↑ }  
start times duration

- **CASToR datafile current formats**

	timestamp	other fields
Frame 1	20	_____
	32	_____
	35	_____
Frame 2	40	_____
	44	_____
	48	_____
Frame 3	56	_____
	60	_____
	65	_____
Frame 4	70	_____
	90	_____
	105	_____
	115	_____

List-mode

timestamp	other fields	
0	_____	Frame 1
0	_____	
0	_____	
30	_____	Frame 2
30	_____	
30	_____	
60	_____	Frame 3
60	_____	
60	_____	
90	_____	Frame 4
90	_____	
90	_____	

Histogram

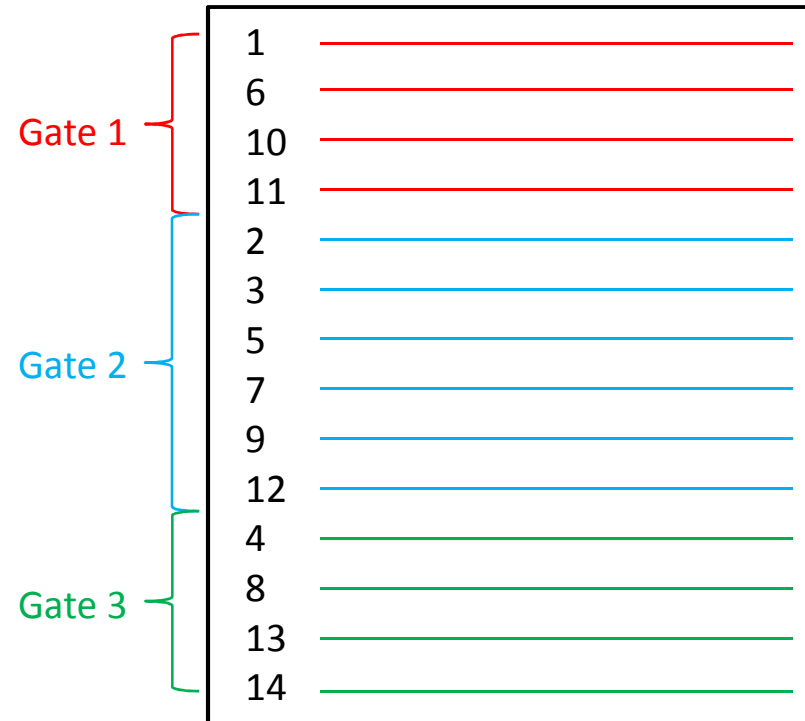
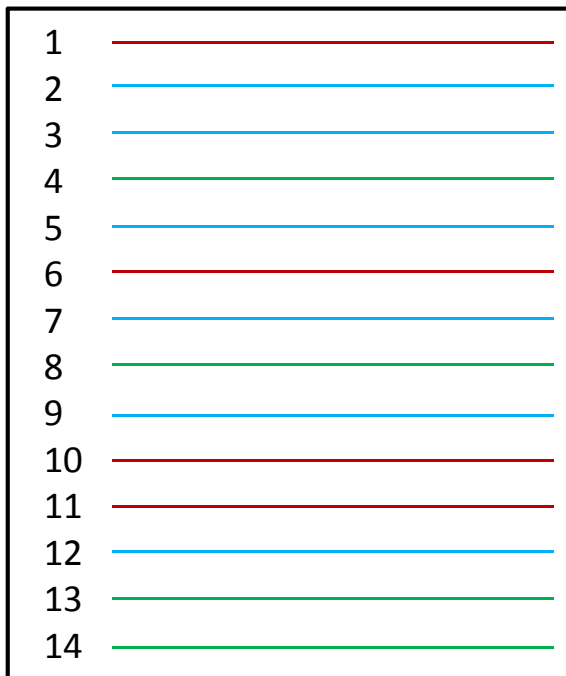


# Dynamic data management: Gating (list)

- Command-line options: **-g path/to/file.txt**

```
nb_respiratory_gates: 3 file.txt
nb_events_respiratory_gates:
4,6,4
```

- Gated datafile splitting (CASToR list-mode)



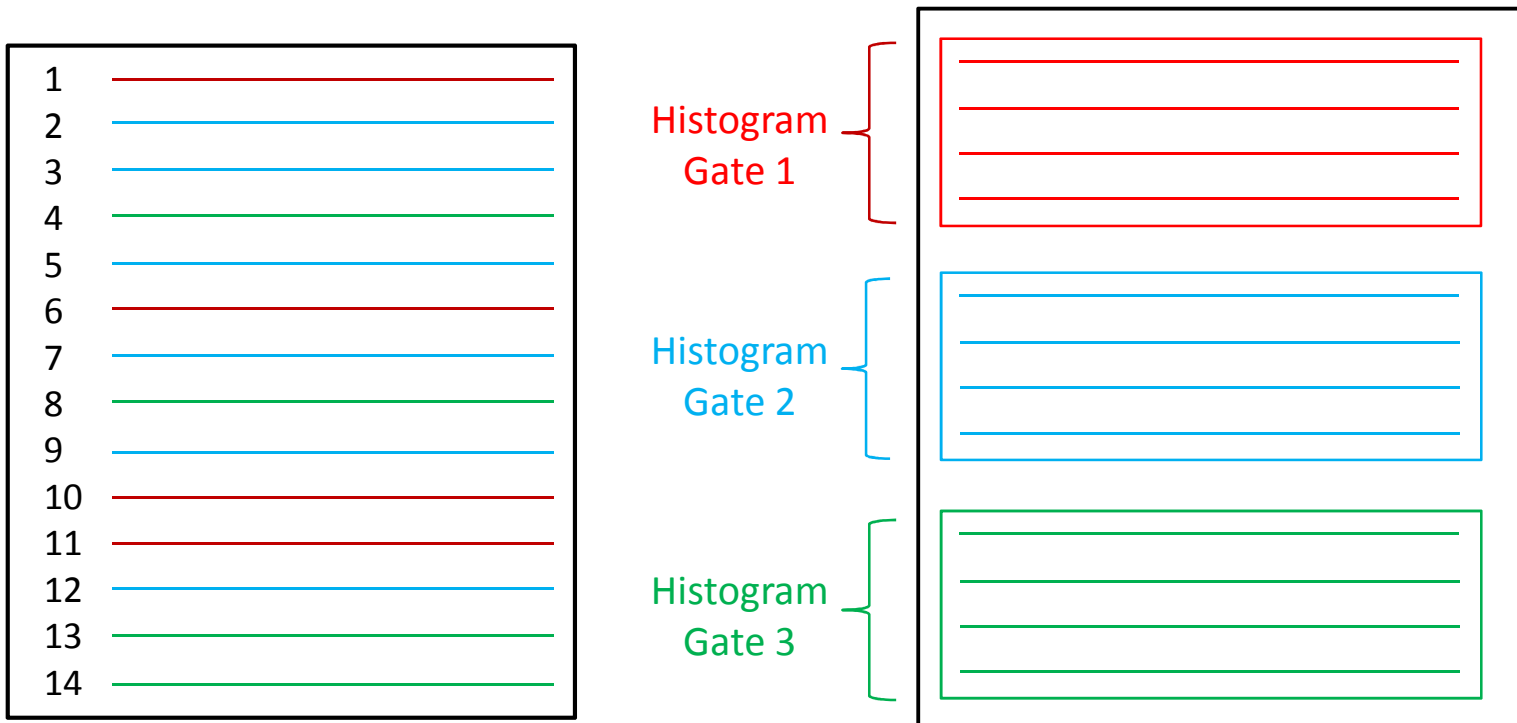


# Dynamic data management: Gating (histogram)

- Command-line options: **-g path/to/file.txt**

```
nb_respiratory_gates: 3 file.txt
nb_events_respiratory_gates :
nEventsHist, nEventsHist, nEventsHist
```

- Gated datafile splitting (CASToR histogram)





# 5D Dynamic data management

- 5D Datasets (frames + gates)

**-frm 0,15,30:15**

**-g path/to/file.txt**

file.txt

```
nb_respiratory_gates: 3
nb_events_respiratory_gates :
4,6,4
5,4,5
4,5,5
```

Frame1

Frame 2

Frame 3

Gate 1

Gate 2

Gate 3

Gate 1

Gate 2

Gate 3

Gate 1

Gate 2

Gate 3

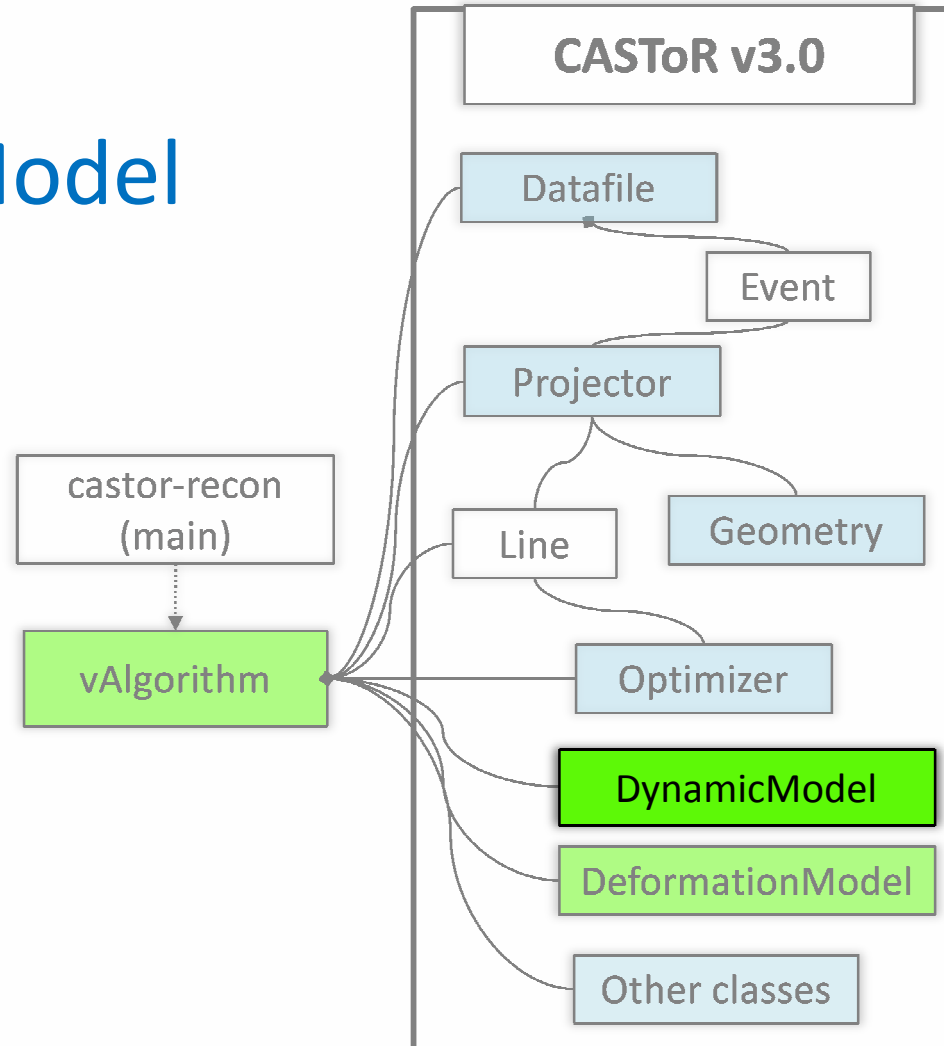




# CASToR v3.0 features



## DynamicModel class





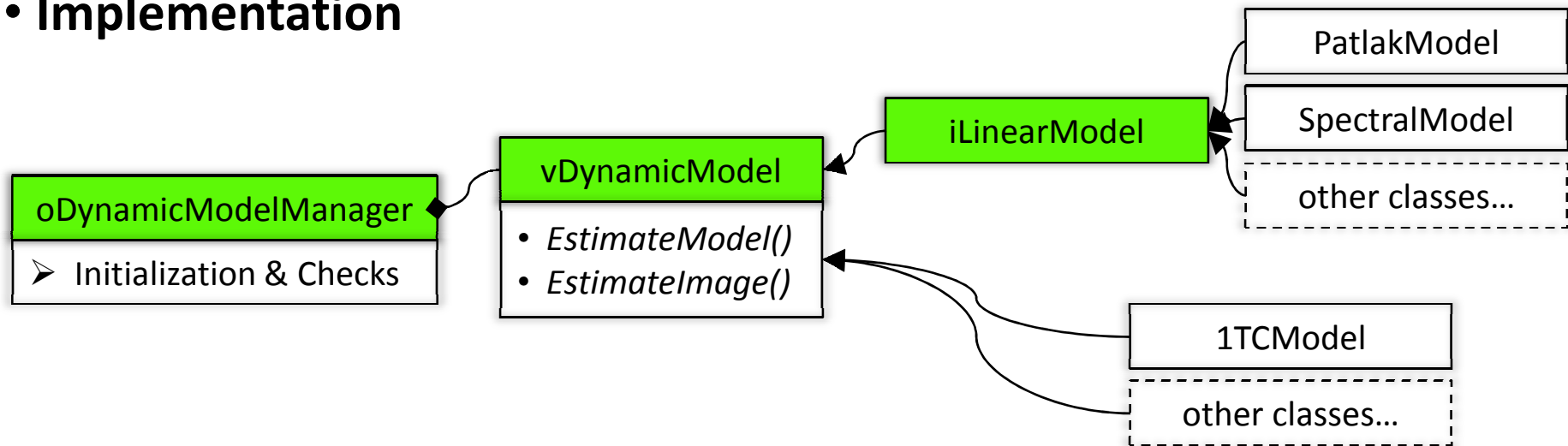


# DynamicModel class

## • Aims

- Parametric image reconstruction
- Temporal regularization
- Post-reconstruction kinetic fitting  
( *castor-imageDynamicTools* )

## • Implementation



-dynamic-model alias, list, of, parameters

-dynamic-model alias:path/to/file



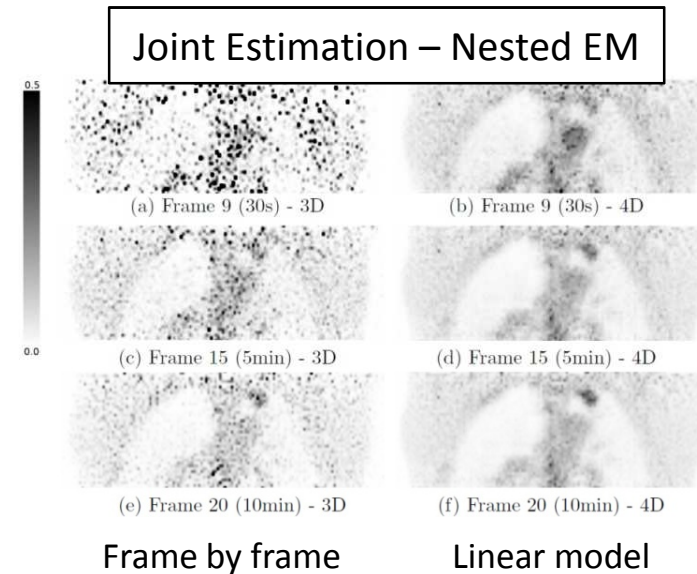
# Implemented models : LinearModel

## • Features

- Frames / Gates linear model
- Optimization methods:
  - Nested EM
  - Non negative Least-Square
  - Direct : Basis functions applied within system matrix

## • Initialization requirements

- dynamic-model `LinearModel:path/to/conf/file`
- Dynamic level (frames/gates)
- Linear basis functions
- Optimization method

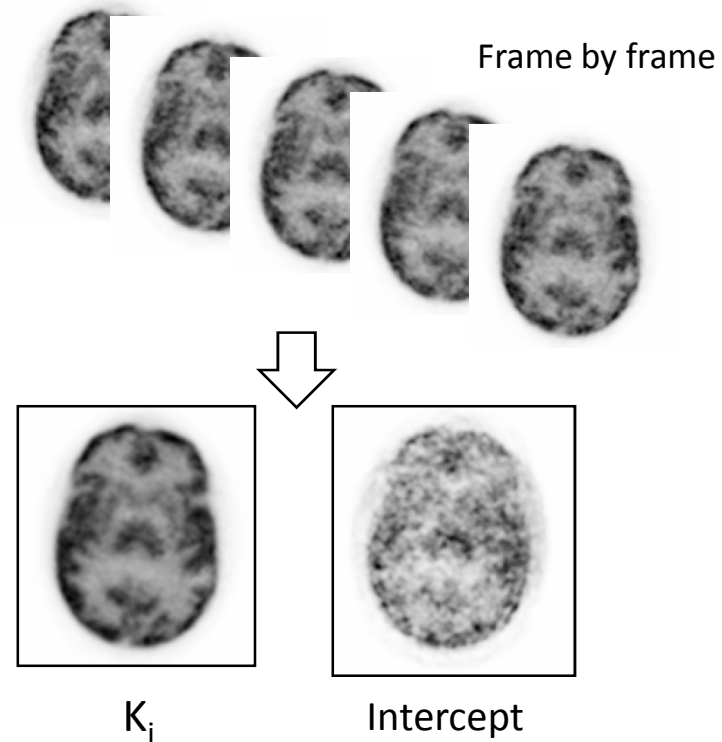




# Implemented models : PatlakModel

## • Features

- $K_i$  estimation
- Computation of Patlak basis functions from arterial input curve (AIC)
- Optimisation methods:
  - Nested EM
  - NNLS
  - Linear regression



## • Initialization requirements

- dynamic-model Patlak,  $Ic_{p1}, Ic_{p2}, \dots, C_{p1}, C_{p2}, \dots,$
- dynamic-model Patlak:path/to/conf/file
- Patlak basis functions
- Optimization method



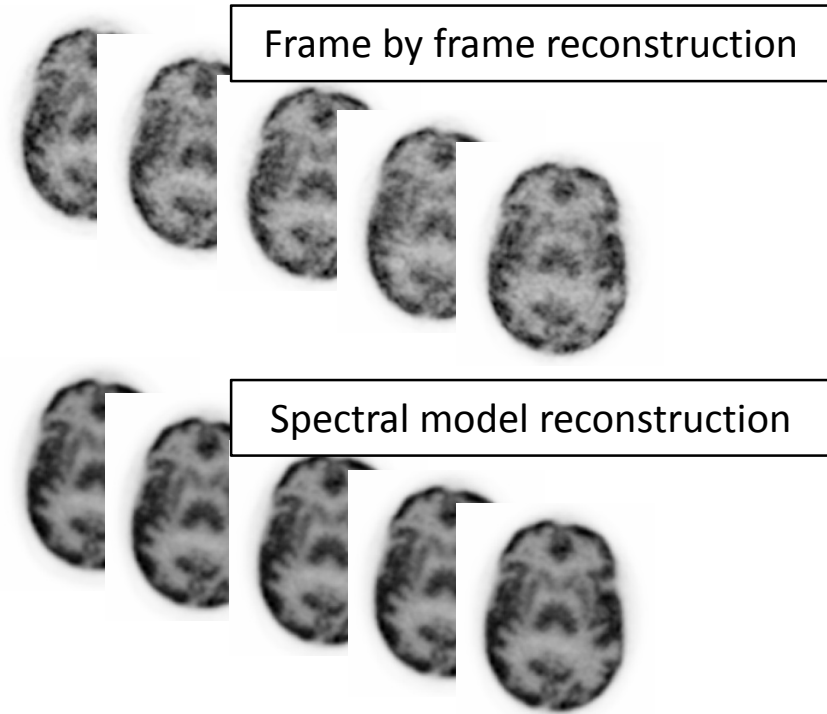
# Implemented models : SpectralModel

- **Features**

- Set of exponential basis functions  $\exp(-\beta_i t)$
- Fonctions convoluted with AIC
- Optimization methods:
  - Nested EM
  - NNLS
  - Direct

- **Initialization requirements**

- dynamic-model `Spectral:path/to/conf/file`
- Arterial input curve
- Spectral function parameters
- Optimization method

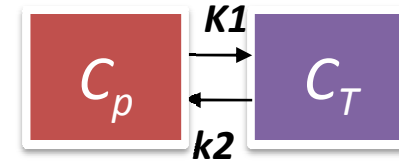




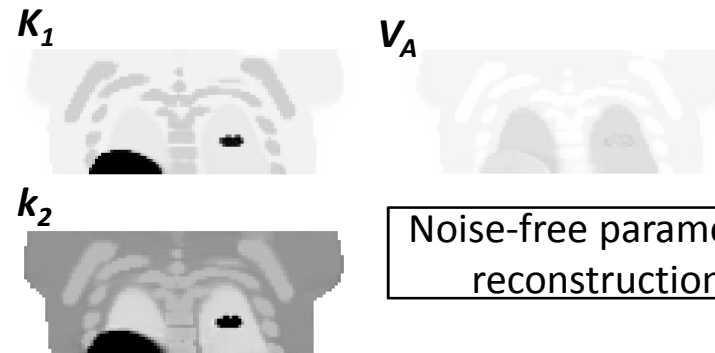
# 1 tissue compartment model

## • Features

- $K_1, k_2, V_A$  estimation
- Optimization methods:
  - NNLS
  - LS with Ridge-Regression



$$C_{PET}(t) = C_T(t) + V_A \cdot C_p(t)$$



## • Initialization requirements

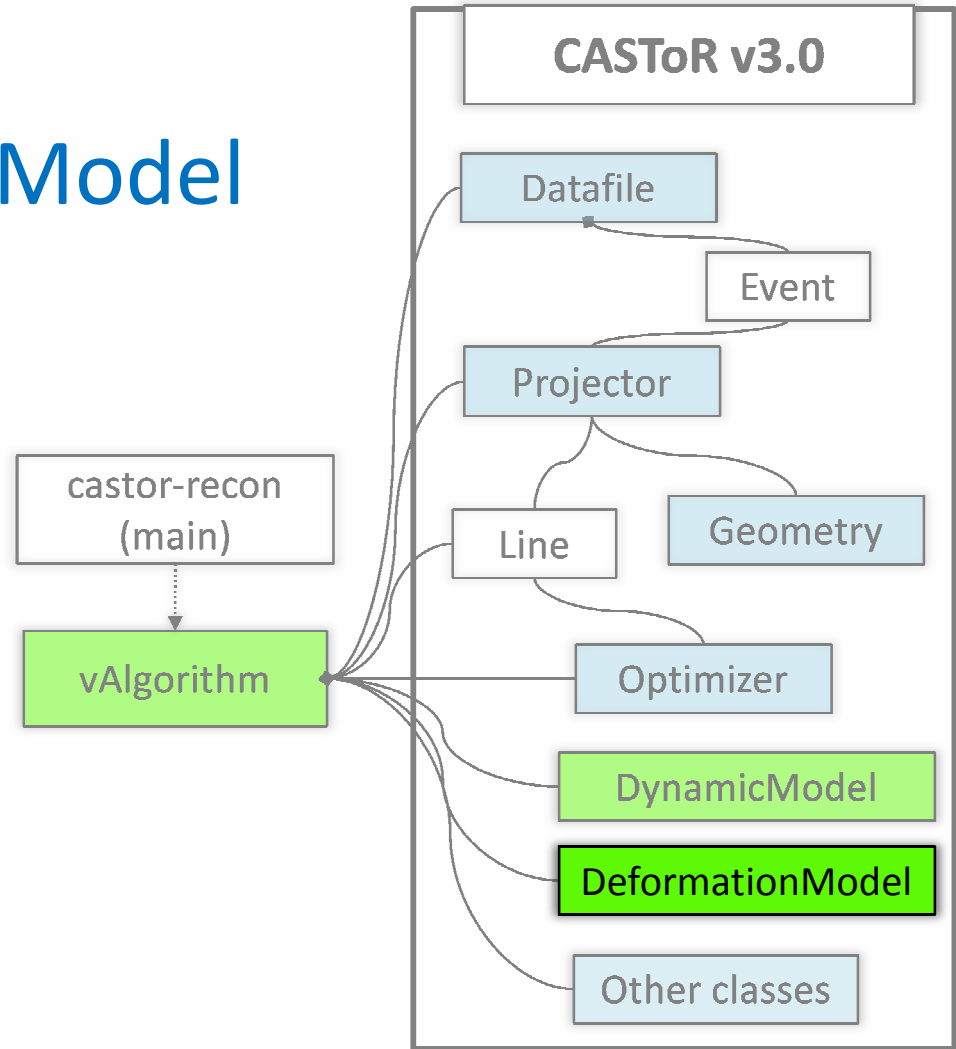
- dynamic-model \_1TCM,  $C_{p1}, C_{p2}, \dots$
- dynamic-model \_1TCM:path/to/conf/file
- Input function
- Optimization method
- Other parameters (integration method, ...)



# CASToR v3.0 features



## DeformationModel class

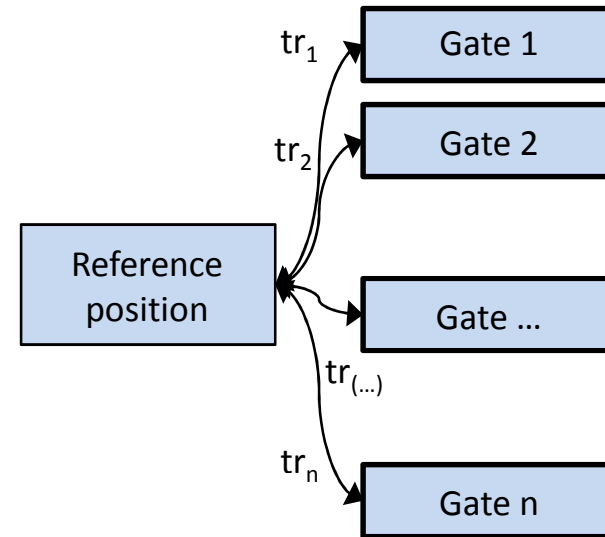




# Motion management

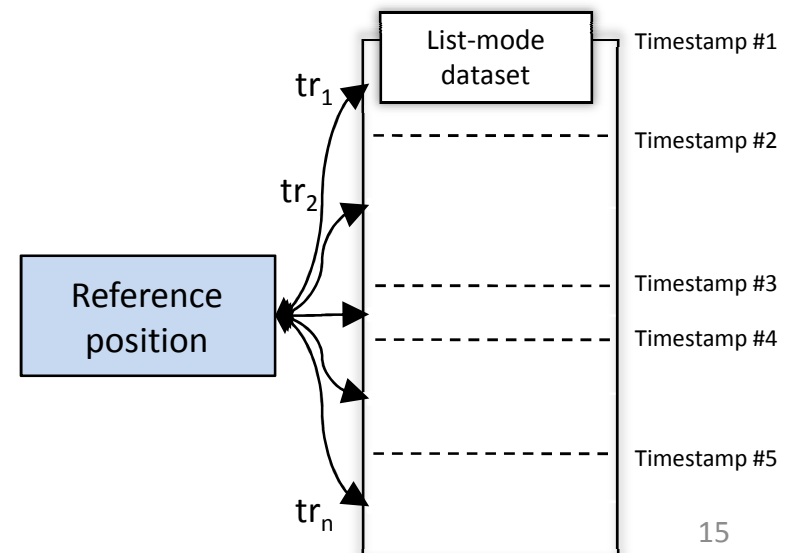
- **Gated dataset motion correction (MC)**

- Set motion correction of gated datasets :
  - rm**: respiratory motion
  - cm**: cardiac motion
- Transformation parameters:  
reference position ↔ each gate



- **Timestamp-based motion correction**

- Set involuntary patient MC  
**-im**
- Transformation parameters:  
reference position ↔ each data subsets
- Transformation timestamps:  
(-g path/to/conf/file)



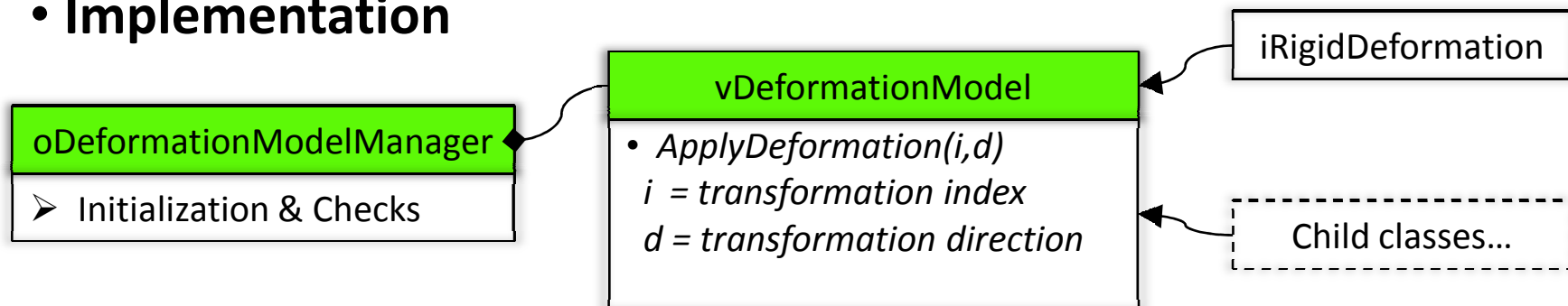


# DeformationModel class

## • Aims

- Image-based deformation
- Motion correction
- Main class manages triggers/image matrices
- New class only implements:
  - Deformation process
  - Transformation parameters

## • Implementation



```
-rm, cm, im  alias, list, of, parameters
```

```
-rm, cm, im  alias:path/to/file
```





# Implemented models : Rigid Deformation

- **Features**

- 6 parameters  
(3 translations / rotations)
- Tri-linear interpolation
- Rotation convention (XYZ, YXZ,...)

- **Initialization requirements**

-im deformationRigid,  $t_{x1}, t_{y1}, t_{z1}, r_{x1}, r_{y1}, r_{z1}, t_{x2}, t_{y2}, \dots$

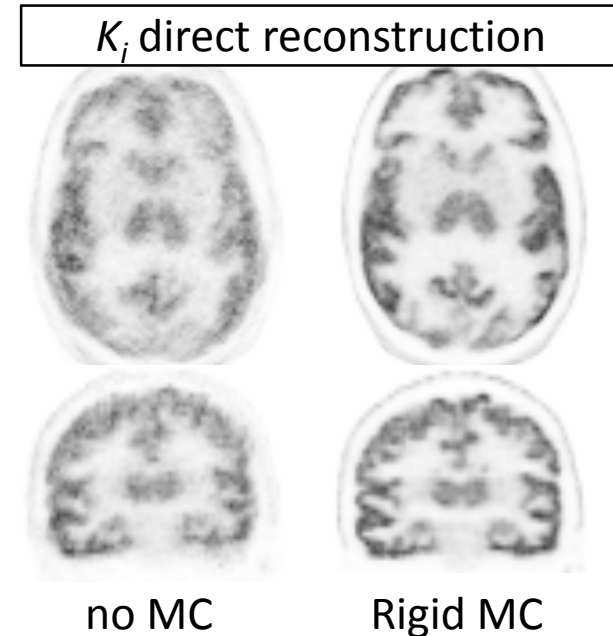
-im deformationRigid:path/to/conf/file

➤ Transformation parameters

➤ Motion trigger (-g option) :

- Motion timestamp

- Gating configuration





# Dynamic : Documentation

---

- **Documentation files** ([www.castor-project.org/documentation](http://www.castor-project.org/documentation))
  - General documentation
    - dynamic datafile
    - command-line options
  - Dynamic / Deformation models documentation
    - Models initialization
    - Development guidelines
- **In-code documentation**
  - -help-dynamic, -help-dynamic-model , -help-deformation-model
  - Dev: Template classes
- **Examples** ([www.castor-project.org/benchmarks](http://www.castor-project.org/benchmarks))
  - Dynamic recon. examples with simulated datasets
    - Different dynamic models
    - Random motion correction



# CASToR v3.0 features

---



Time of Flight  
update



# CASToR v2.0 Time-of-flight (TOF)

---

## Current implementation:

- **System matrix**

- TOF component conceptually independent of the projection component

- **Datafile**

- Histogram / List-mode

- **Corrections**

- TOF estimations for random/scattered count rate

- **Voxel sensitivity**

- Always the same, except with some approximation options



# CASToR v3.0 Time-of-flight

---

- **TOF histogram reconstruction**

- Computation options for TOF weights
  - Pre-computed / On-the-fly
  - Approximation / Exact formulas (integration/convolution)
- Computing speed Vs. Accuracy trade-off

- **TOF list-mode reconstruction**

- Continuous TOF measurements
- Quantized TOF measurements
  - Data file : additional fields
  - TOF weights computation : same options as for histogram data



# TOF Update : Documentation

---

- **Documentation files** ([www.castor-project.org/documentation](http://www.castor-project.org/documentation))
  - TOF documentation update
- **Paper**
  - “Time-of-flight (TOF) implementation for PET reconstruction in practice”  
Filipovic et al 2019, *Phys. Med. Biol*  
[doi.org/10.1088/1361-6560/ab4f0b](https://doi.org/10.1088/1361-6560/ab4f0b)
- **In-code documentation**
  - Helper command-line options :
    - proj-common
    - help-projm
- **Updated benchmarks** ([www.castor-project.org/benchmarks](http://www.castor-project.org/benchmarks))
  - PET Benchmarks



# Thanks for your attention!



[www.castor-project.org](http://www.castor-project.org)

Mailing-list: [castor-project.org/mailling-list](http://castor-project.org/mailling-list)