­­

Cone-Beam CT Forward Projector: Notation and Calculations

­­­

# Definition of variables:

## Coordinates:

Coordinates in scanner space. The scanner space is defined as follows: The axis of rotation is located at and parallel to the -axis; The x-axis is the line through the source that is perpendicular to the detector plane.

Coordinates in object space. The -axis ***is*** the axis of rotation in the object space.

## Index parameters:

Voxel indices corresponding to

Detector element indices corresponding to -axis and the -axis

View angle index

## Scanner parameters:

Location of the source. The source is assumed to be on the -axis with .

Location of the center of rotation. Axis of rotation is parallel to -axis.

The location of the center of the first detector corresponding to . As the detector array is defined to be perpendicular to the -axis, all points on the detector have the -coordinate .

Angle associated with first view.

Detector width and spacing in -plane.

Detector width and spacing in -plane.

Spacing between view angles.

Number of detectors in array in -plane.

Number of detectors in array in -plane.

Number of discrete view angles.

Sinogram

## Image parameters:

Location of the center of first voxel corresponding to .

The width/height/depth of the voxel.

Number of voxels in -direction

Number of voxels in -direction

Number of voxels in -direction

3D image volume

## Intermediate parameters

### Angles

The view rotation angle of the object.

Angle between -axis and line through source and voxel center in -plane.

Angle between -axis and line through source and voxel center in -plane.

Angle of the line through the voxel center relative to the voxel side in the -plane.

Angle mapped to the range .

### Cooridinates

The center of the voxel location (scanner coordinates).

The center of the voxel location (object coordinates).

Location of the center of the detector of interest.

### Indices and index related variables

Smallest index such that .

Forms array: .

Largest index such that .

Forms array: .

Smallest index such that .

Forms array: .

Largest index such that .

Forms array: .

Maximum number of detector pixels affected by any voxel:

Maximum number of detector pixels affected by any voxel:

Index value corresponding to a -location of a voxel given

Forms array: .

### Matrix Parameters

Sample density of -vales. .

Number of discrete -values for computing .

Distance between discrete -values.

Minimum -value when computing -matrix.

Maximum -value when computing -matrix.

### Lengths/ distances

Absolute distance between projection center and detector center in -plane.

Absolute distance between projection center and detector center in -plane.

Width of the projection profile in the -plane.

Width of the projection profile in the -plane.

Length of the overlap of the projection profile and the detector (-plane).

Length of the overlap of the projection profile and the detector (-plane).

The magnification factor.

Array of these for different view angles.

### Matrix entries

The -matrix .

The -matrix

The matrix entry of the final forward projection matrix .

# Relationships:

Matrix entry definition:

Transformation from object to scanner coordinates:

Calculation of detector center:

Calculation of view angle:

Calculation of voxel center:

Rounding: denotes the closest integer to .

# Comute\_B(…)

## Overview

Inputs:

Scanner Parameters:

Image Parameters:

Auxiliary Parameters:

Outputs:

Auxiliary Outputs:

## Computation

For to

{

For to

{

For to

{

For to

{

}

}

}

}

# Compute\_C

## Overview

Inputs:

Scanner Parameters:

Image Parameters:

Matrix Parameters:

Output:

## Calculations

For to

{

For to

{

For to

{

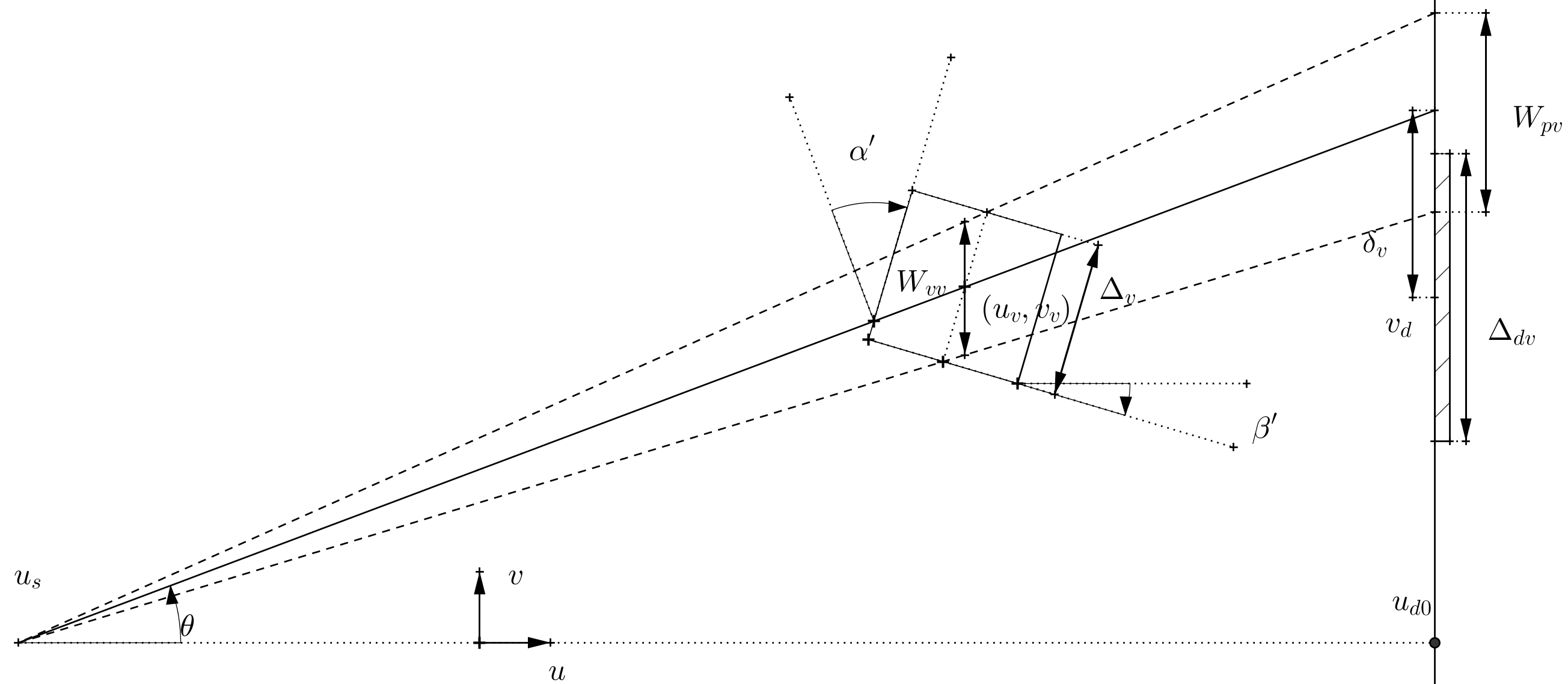
}

}

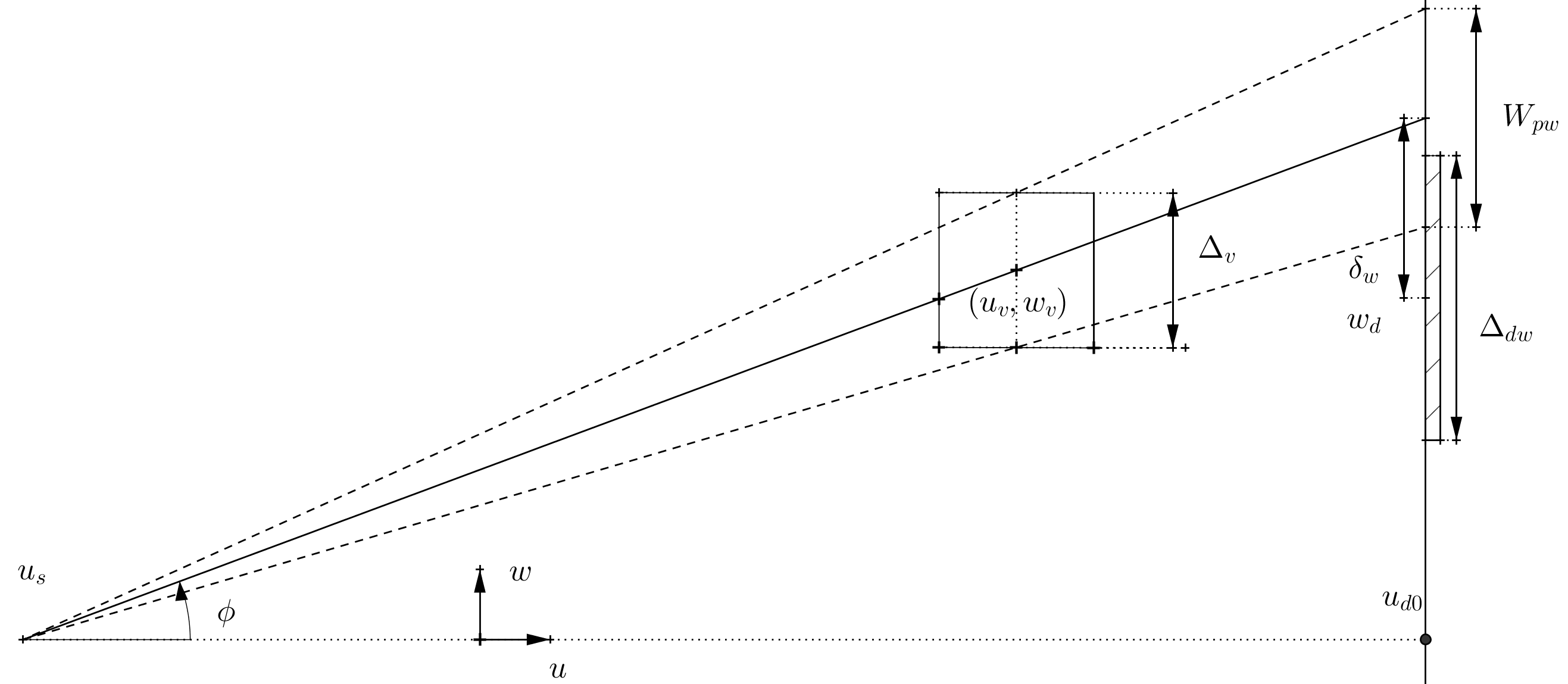
}

# Figures

## -plane



## -plane



Functions Archive

(Obsolete because some functions have been merged or reorganized)

(variable names or I/O might be outdated)

## Computation of upper bound on the number of elements per column:

Before we write the coefficients of one column into we need pre-allocate the array.

The array has size

where is (simplified) computed as the other ’s using the ‘worst’ possible voxel location.

This gives us an upper bound for the size of .

## How to access from ?

as a 1D array. Then we get the matrix entry

Define [ sic ] to be the cumulative sum of :

so that is the index of .

With this the access of gets simplified to

We for we can use this identity:

So for this we need the arrays

[ sic ] and

.

We do not need and .

# Function VoxProjectVSingle()

## Overview

Inputs:

Parameters:

The voxel indices:

The measurement index:

Output:

Projection coefficients for a given voxel, detector and view:

## Calculations

### Calculation of the voxel and detector location:

# Function VoxProjectVDetectors()

## Overview

Inputs:

Parameters:

The voxel indices:

The view index:

Output:

Starting detector index:

Stopping detector index:

Number of elements in output array, :

Array of projection coefficients for a given pixel and view:

## Calculations

**Note**: It can actually be the case that

or

For correct execution it is necessary to check .

For to

{

}

# VoxProjectVDetectorsViews() (Slow version, nested functions)

## Overview

Inputs:

Parameters:

The voxel indices:

The number of discrete view angles:

Output:

Array of starting detector indices :

Array of stopping detector indices :

The cumulative sum of : [ sic ]

Array of Projection coefficients for a given pixel:

## Computation

For to

{

the concatenation of and

}

# Function VoxProjectWSingle ()

## Overview

Inputs:

Detector Parameters:

Image Parameters:

Index Parameters:

Auxiliary Inputs:

implicitly dependent on

Output:

The matrix entry:

## Calculations

// consider precomputing the green terms “outside the loop”

## Example execution:

For to

{

…

…

}

## More detail on the -term:

But

Therefore, more efficient:

compute , then

Can find 4-term polynomial with

Coefficients: (Canonical basis)

1.000068257087092

0.000000000000002

0.497357448138487

-0.000000000000007

-0.108505313733435

0.000000000000004

0.025483280999067

Using weighted LS-approach yields polynomial   
Coefficients: (Canonical basis)

1.000043510922152

0.000000000000006

0.497945181087399

-0.000000000000069

-0.110405098124138

0.000000000000065

0.026945672884054

See error plot below:

