=1

# The EllipseBoundaryToImageFilter in ITK

*Release 0.00*

Zheen Zhao[1]

July 12, 2006

[1]Duke University, USA
kurtzhao@pmail.ntu.edu.sg

**Abstract**

The current conversion from an ellipse spatial object to an image using SpatialObjectToImageFilter filter could cost minutes because the filter traverses all pixels in the output image. This delay is quite noticeable during applications such as human-machine interface. This document describes a contribution to the Insight Toolkit ITK www.itk.org, which implements what we call an itk::EllipseBoundaryToImageFilter. This is a derived from SpatialObjectToImageFilter. It marks the output image by setting the pixels along the boundary of the input ellipse to the InsideValue and the OutsideValue elsewhere. This filter is fast because it traverses the boundary only. This paper is accompanied with the source code.

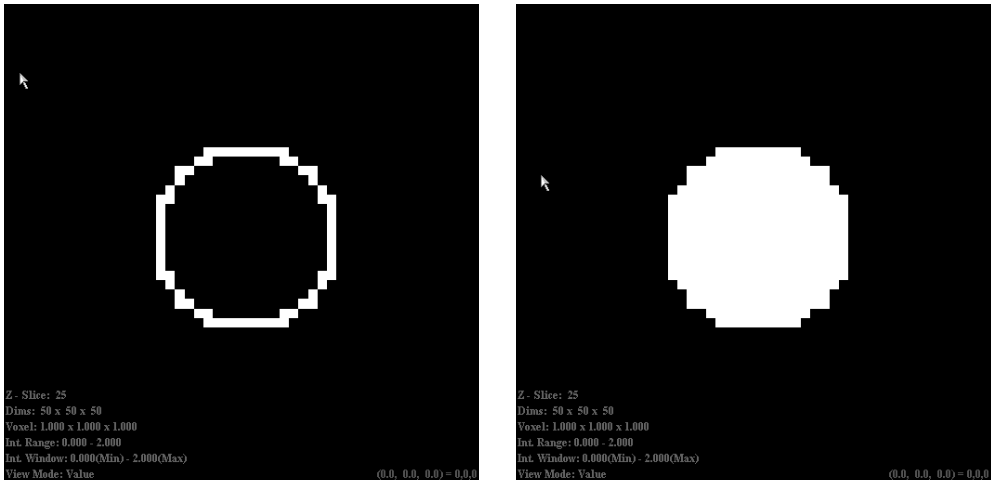
## Contents

## 1 Introduction

EllipseBoundaryToImageFilter has the same interface as SpatialObjectToImageFilter, except taking EllipseSpatialObject instead of general SpatialObject. It takes a EllipseSpatialObject and produce a image whose pixels are set to InsideValue along the ellipse's boundary and to OutsideValue elsewhere.

The motivation is that the current conversion from an ellipse spatial object to an image using SpatialObjectToImageFilter filter could cost minutes because the filter traverses all pixels in the output image. This delay is quite noticeable during applications such as human-machine interface. The proposed filter is efficient because it traverses the boundary of the input ellipse only. Figure 1 shows the cross-sections of the two images generated by the two filters. An identical ellipse was used as the input ellipse.

In this version, the proposed filter takes only one ellipse. However, if there is a demand, we can revise it to support GroupSpatialObject so that it can produce an image reflecting multiple ellipses.



Z - Slice: 25
Dims: 50 x 50 x 50
Voxel: 1.000 x 1.000 x 1.000
Int. Range: 0.000 - 2.000
Int. Window: 0.000(Min) - 2.000(Max)
View Mode: Value
(0.0, 0.0, 0.0) = 0,0,0

EllipseBoundaryToImageFilter

Z - Slice: 25
Dims: 50 x 50 x 50
Voxel: 1.000 x 1.000 x 1.000
Int. Range: 0.000 - 2.000
Int. Window: 0.000(Min) - 2.000(Max)
View Mode: Value
(0.0, 0.0, 0.0) = 0,0,0

SpatialObjectToImageFilter

Figure 1: Cross-section of the images generated from the two filters by a same ellipse. The left shows the boundary and the right shows the whole ellipse.

=1

# The EllipseBoundaryToImageFilter in ITK

*Release 0.00*

Zheen Zhao[1]

July 27, 2006

[1]Duke University, USA
kurtzhao@pmail.ntu.edu.sg

**Abstract**

The current conversion from an ellipse spatial object to an image using SpatialObjectToImageFilter filter could cost minutes because the filter traverses all pixels in the output image. This delay is quite noticeable during applications such as human-machine interface. This document describes a contribution to the Insight Toolkit ITK www.itk.org, which implements what we call an itk::EllipseBoundaryToImageFilter. This is a derived from SpatialObjectToImageFilter. It marks the output image by setting the pixels along the boundary of the input ellipse to the InsideValue and the OutsideValue elsewhere. This filter is fast because it traverses the boundary only. This paper is accompanied with the source code.

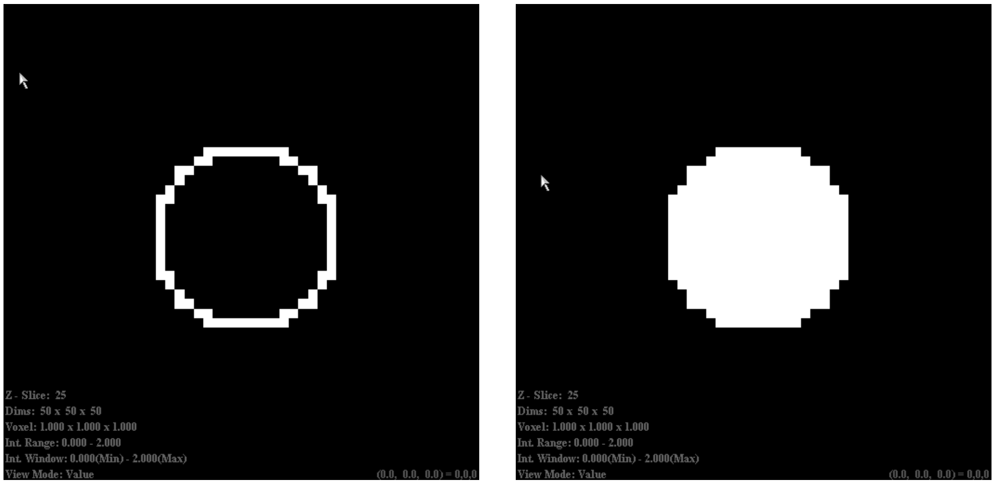
## Contents

## 1 Introduction

EllipseBoundaryToImageFilter has the same interface as SpatialObjectToImageFilter, except taking EllipseSpatialObject instead of general SpatialObject. It takes a EllipseSpatialObject and produce a image whose pixels are set to InsideValue along the ellipse's boundary and to OutsideValue elsewhere.

The motivation is that the current conversion from an ellipse spatial object to an image using SpatialObjectToImageFilter filter could cost minutes because the filter traverses all pixels in the output image. This delay is quite noticeable during applications such as human-machine interface. The proposed filter is efficient because it traverses the boundary of the input ellipse only. Figure **??** shows the cross-sections of the two images generated by the two filters. An identical ellipse was used as the input ellipse.

In this revision, the proposed filter is able to take multiple ellipse.



```
Z - Slice:  25
Dims:  50 x  50 x  50
Voxel: 1.000 x 1.000 x 1.000
Int. Range: 0.000 - 2.000
Int. Window: 0.000(Min) - 2.000(Max)
View Mode: Value                    (0.0,  0.0,  0.0) = 0,0,0
```

EllipseBoundaryToImageFilter

```
Z - Slice:  25
Dims:  50 x  50 x  50
Voxel: 1.000 x 1.000 x 1.000
Int. Range: 0.000 - 2.000
Int. Window: 0.000(Min) - 2.000(Max)
View Mode: Value                    (0.0,  0.0,  0.0) = 0,0,0
```

SpatialObjectToImageFilter

Figure 1: Cross-section of the images generated from the two filters by a same ellipse. The left shows the boundary and the right shows the whole ellipse.