

Thierry CANTENOT

REPORT

---

# Digital Image Processing

## « Assignments »

---



2014-2015

# Summary

<b>A Geometric transforms</b>	<b>1</b>
A.1 Problem statement . . . . .	1
A.2 Python implementation . . . . .	1
A.3 Translation . . . . .	3
A.4 Rotation . . . . .	6
A.5 Scaling . . . . .	9

# A. Geometric transforms

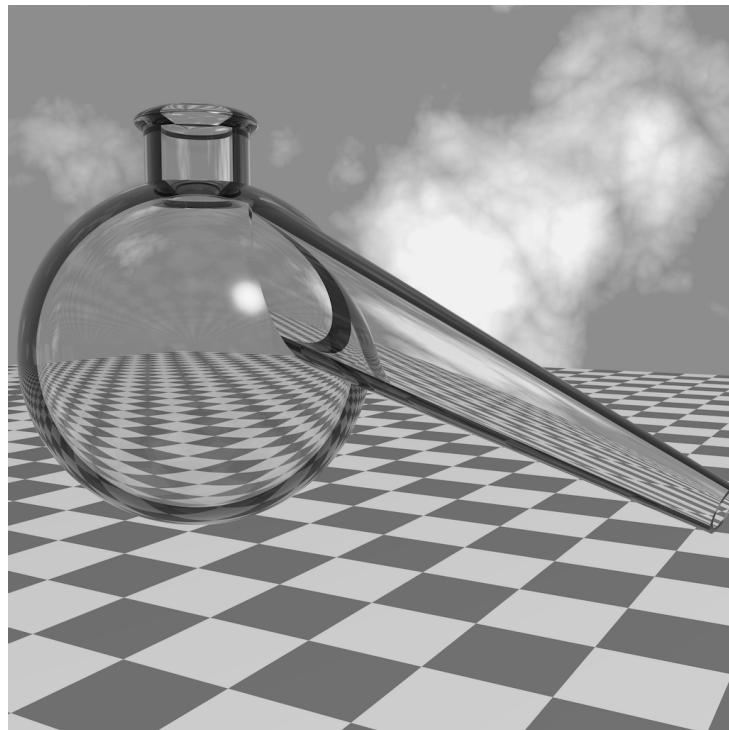
## A.1 Problem statement

Develop a geometric transform program that will rotate, translate, and scale an image by specified amounts, using the nearest neighbor and bilinear interpolation methods, respectively.

## A.2 Python implementation

```
Usage : problem6.py [-h] -i INPUT  
(-t TRANSLATE [TRANSLATE ...] | -r ROTATE | -s SCALE) (-nearest |  
-bilinear) [-ntruncate] [-debug]
```

Use `python problem6.py -h` to see the help.

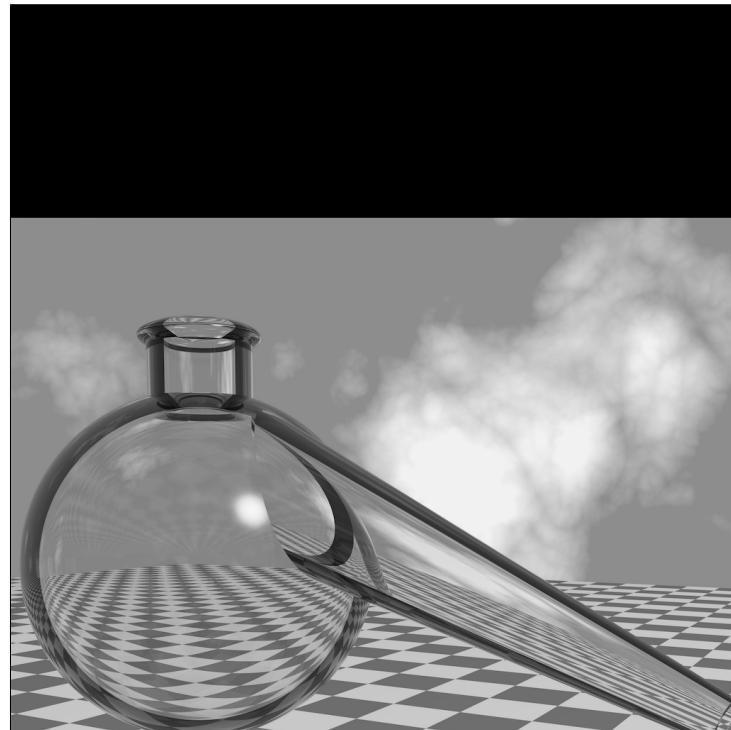


**FIGURE A.1** – Original image

### A.3 Translation

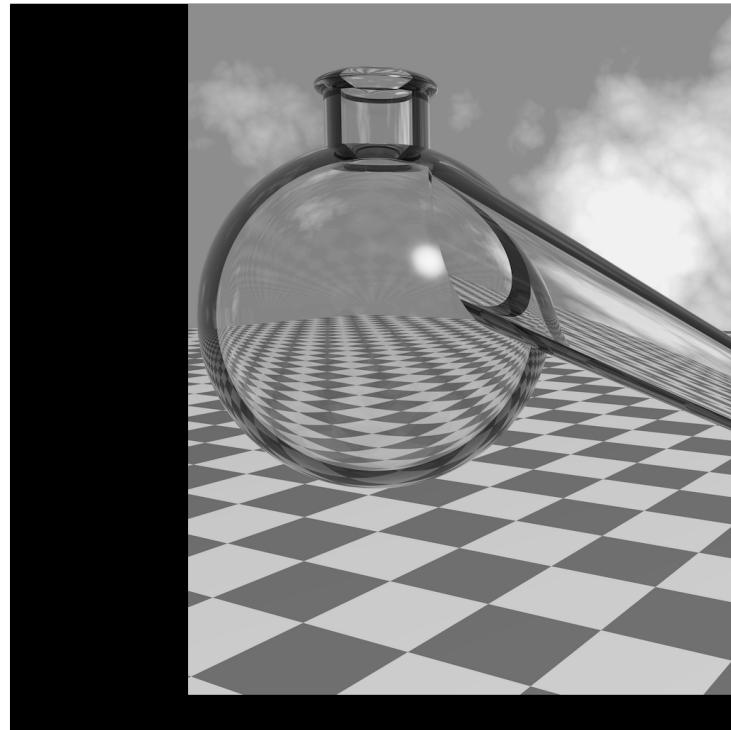
`python problem6.py -i ray_trace_bottle.tif -bilinear -t tx ty`, where  $(tx, ty)$  is the 2D translation vector.

**FIGURE A.2** – Translation  $(0, 300)$  nearest



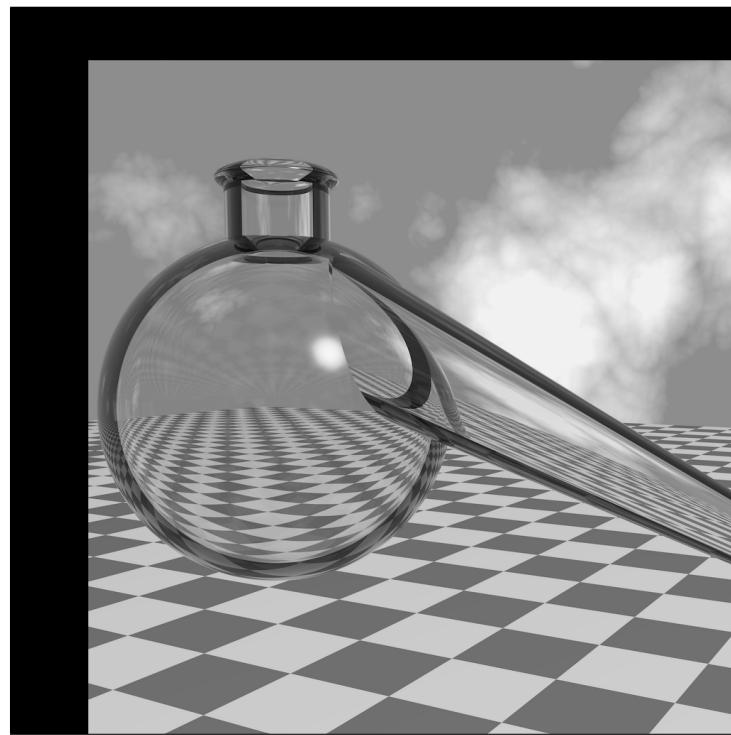
**FIGURE A.3** – Translate  $(0, 300)$  bilinear

**FIGURE A.4** – Translation (250, -100) nearest



**FIGURE A.5** – Translate (250, -100) bilinear

**FIGURE A.6** – Translation (109.8, 75.5) nearest



**FIGURE A.7** – Translation (109.8, 75.5) bilinear

## A.4 Rotation

`python problem6.py -i ray_trace_bottle.tif -nearest -r theta`, where theta is the angle of rotation.

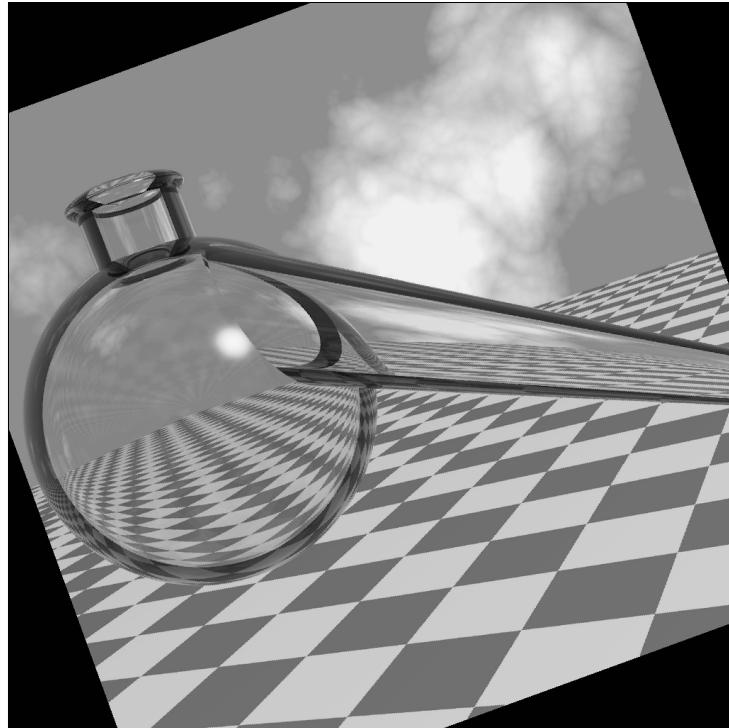


FIGURE A.8 – Rotate 20° nearest

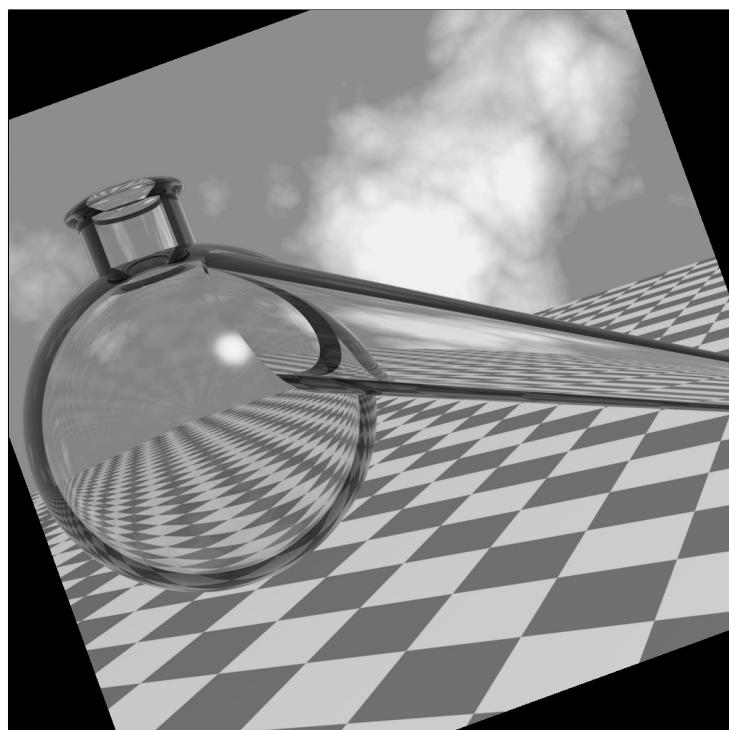
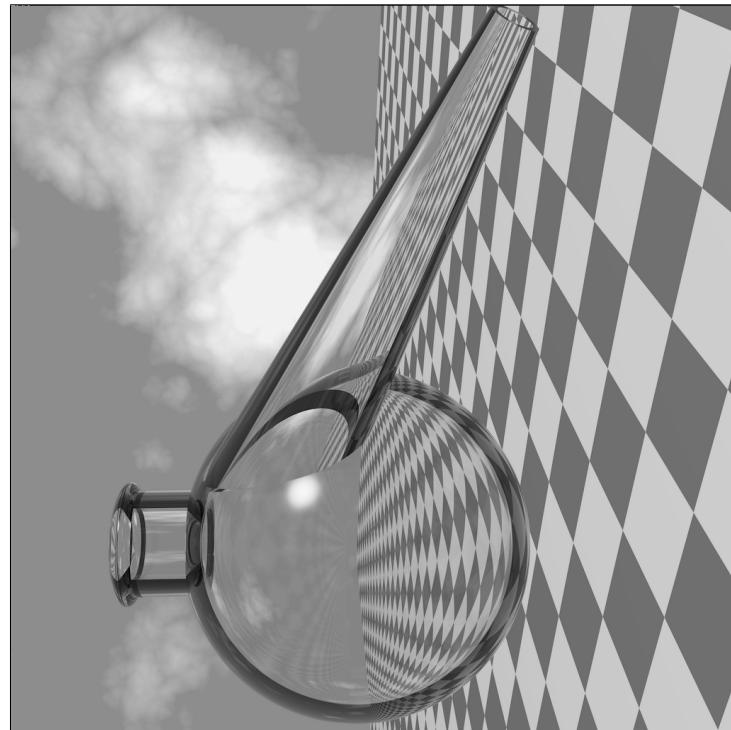
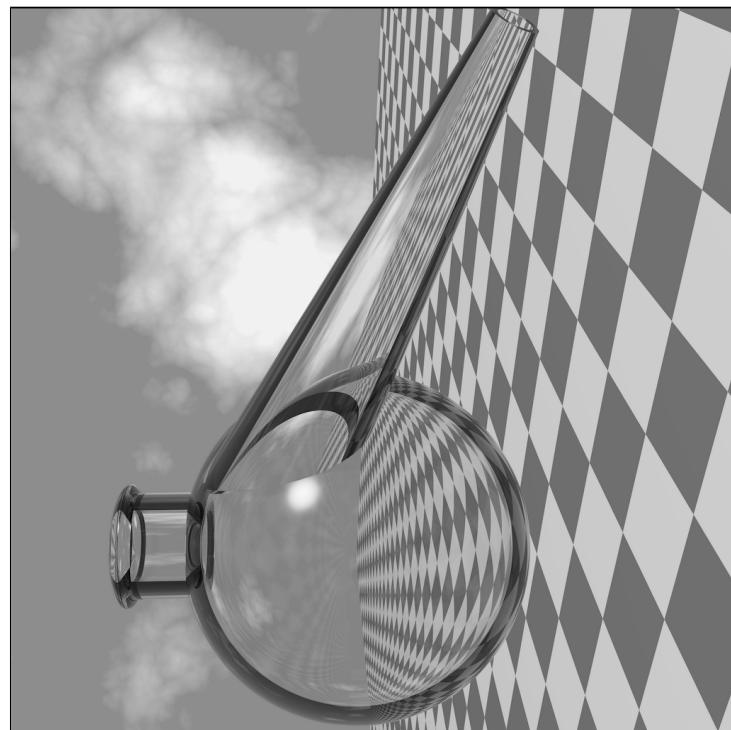


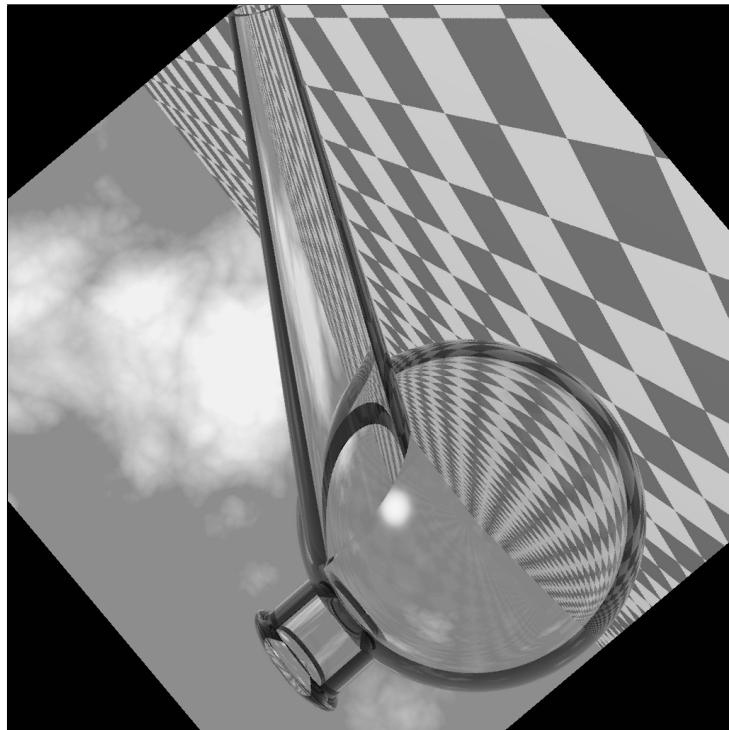
FIGURE A.9 – Rotate 20° bilinear



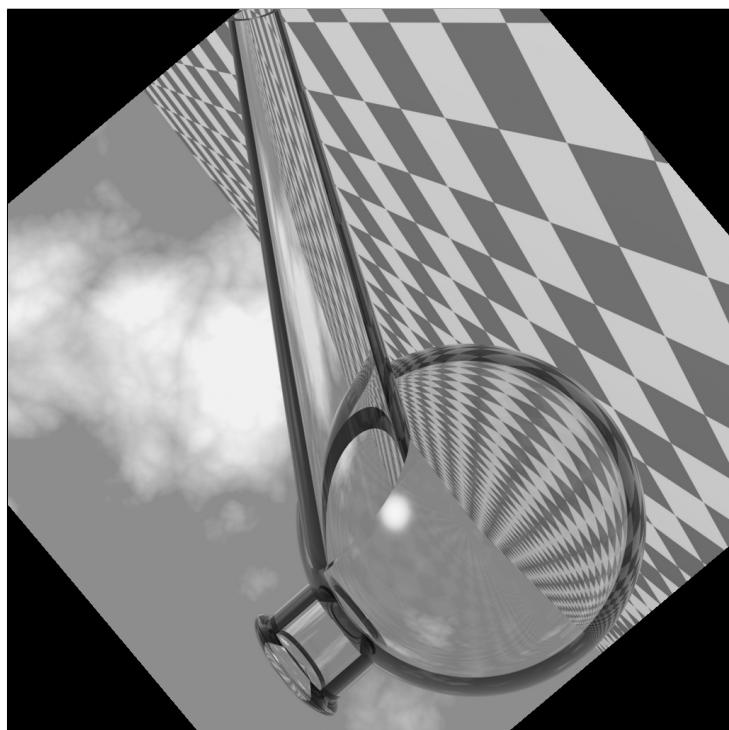
**FIGURE A.10** – Rotate 90° nearest



**FIGURE A.11** – Rotate 90° bilinear

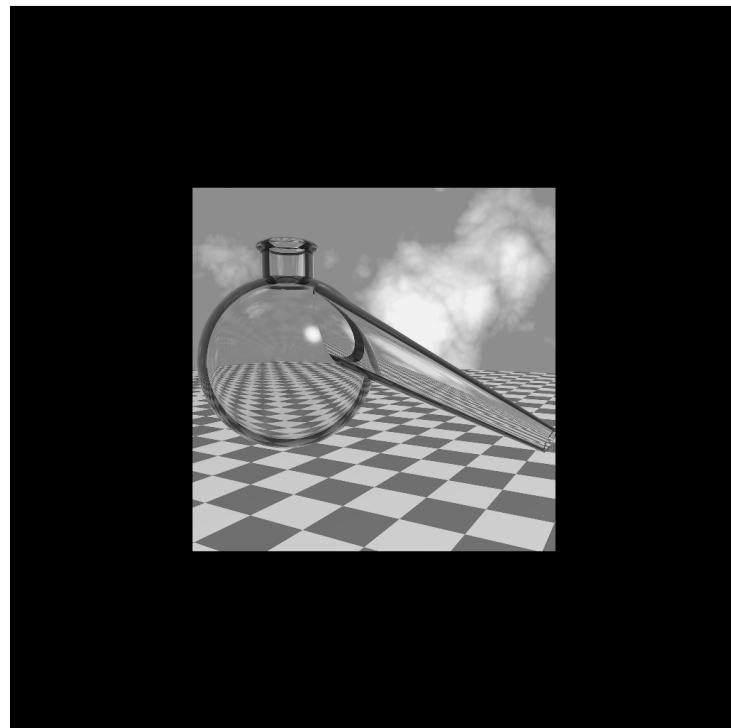


**FIGURE A.12** – Rotate 130° nearest

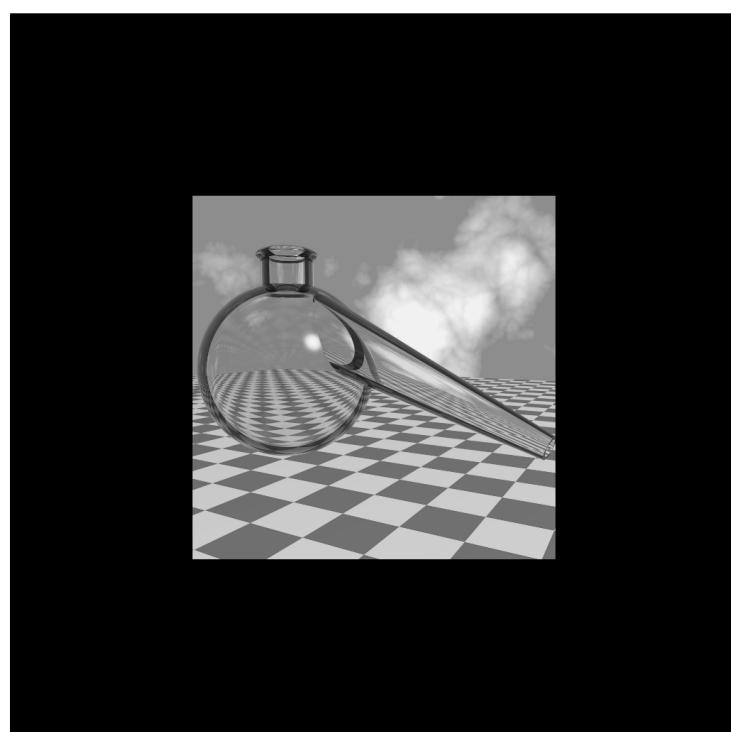


**FIGURE A.13** – Rotate 130° bilinear

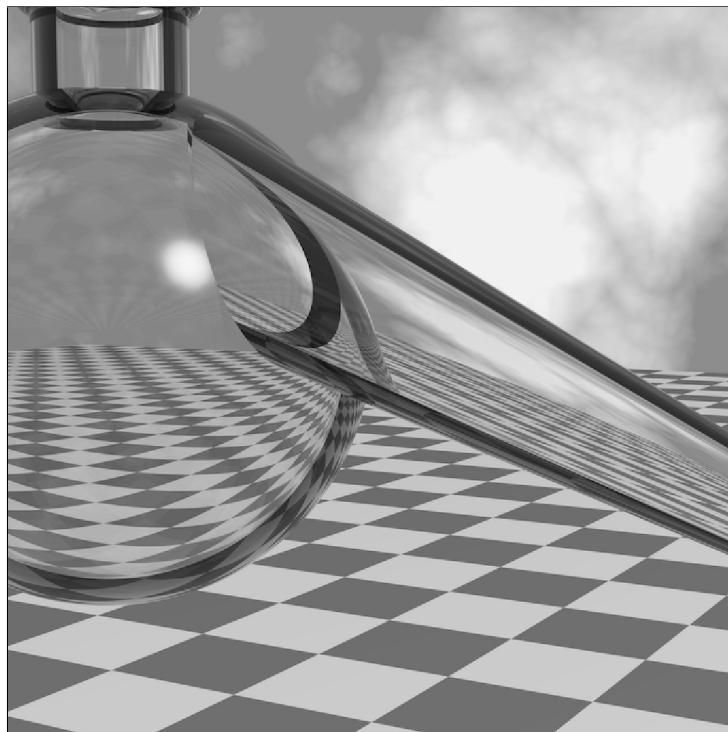
## A.5 Scaling



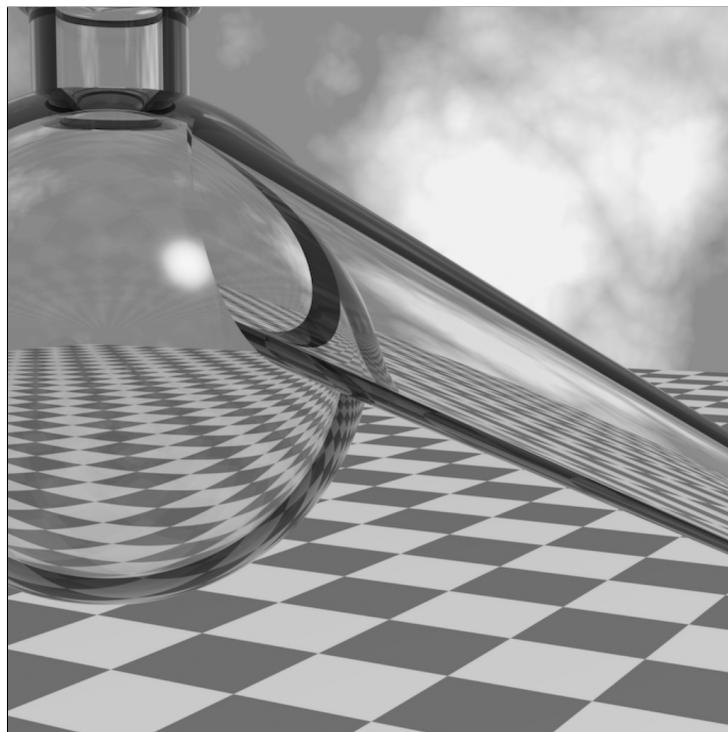
**FIGURE A.14** – Scaling 0.5 nearest



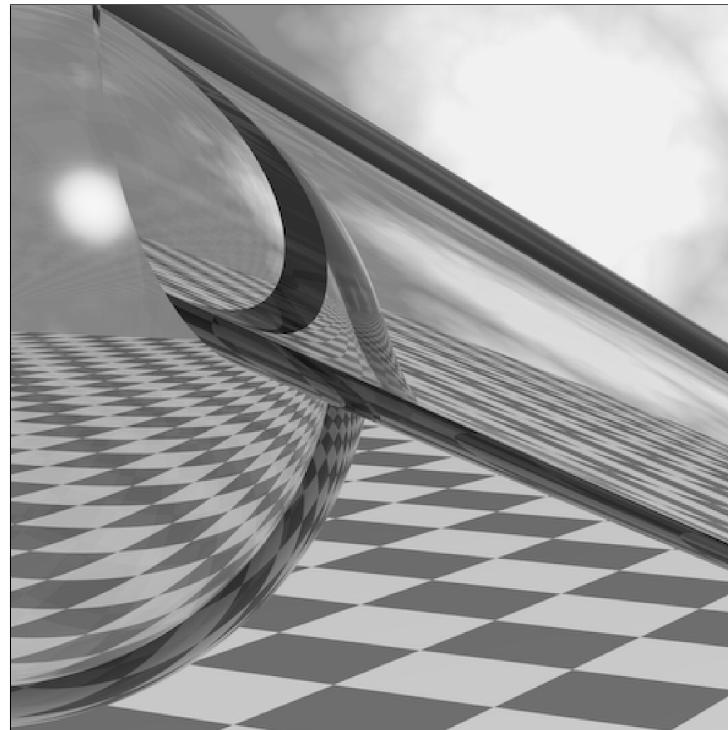
**FIGURE A.15** – Scaling 0.5 bilinear



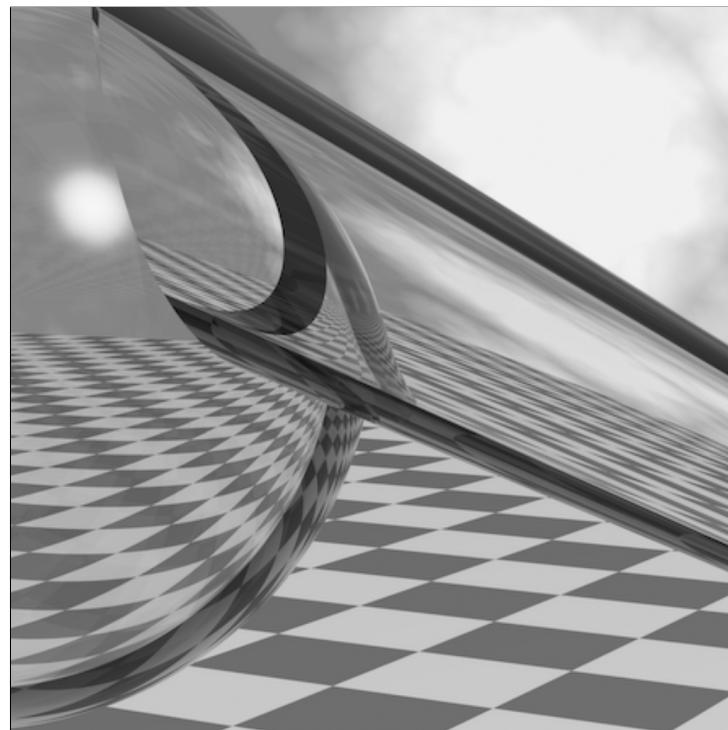
**FIGURE A.16** – Scaling 1.5 nearest



**FIGURE A.17** – Scaling 1.5 bilinear



**FIGURE A.18** – Scaling 2.3 nearest



**FIGURE A.19** – Scaling 2.3 bilinear