Points and Lines in a Plane

3D Computer Vision – Lab Session Task

(CTU FEE subjects B4M33TDV, BE4M33TDV, XP33VID)

Martin Matoušek and Jaroslav Moravec

September 2024

CENTER FOR MACHINE
PERCEPTION

Image Boundary and Coordinate System

Let the image area has an extent [1, 1] to [800, 600]. Draw its boundary.

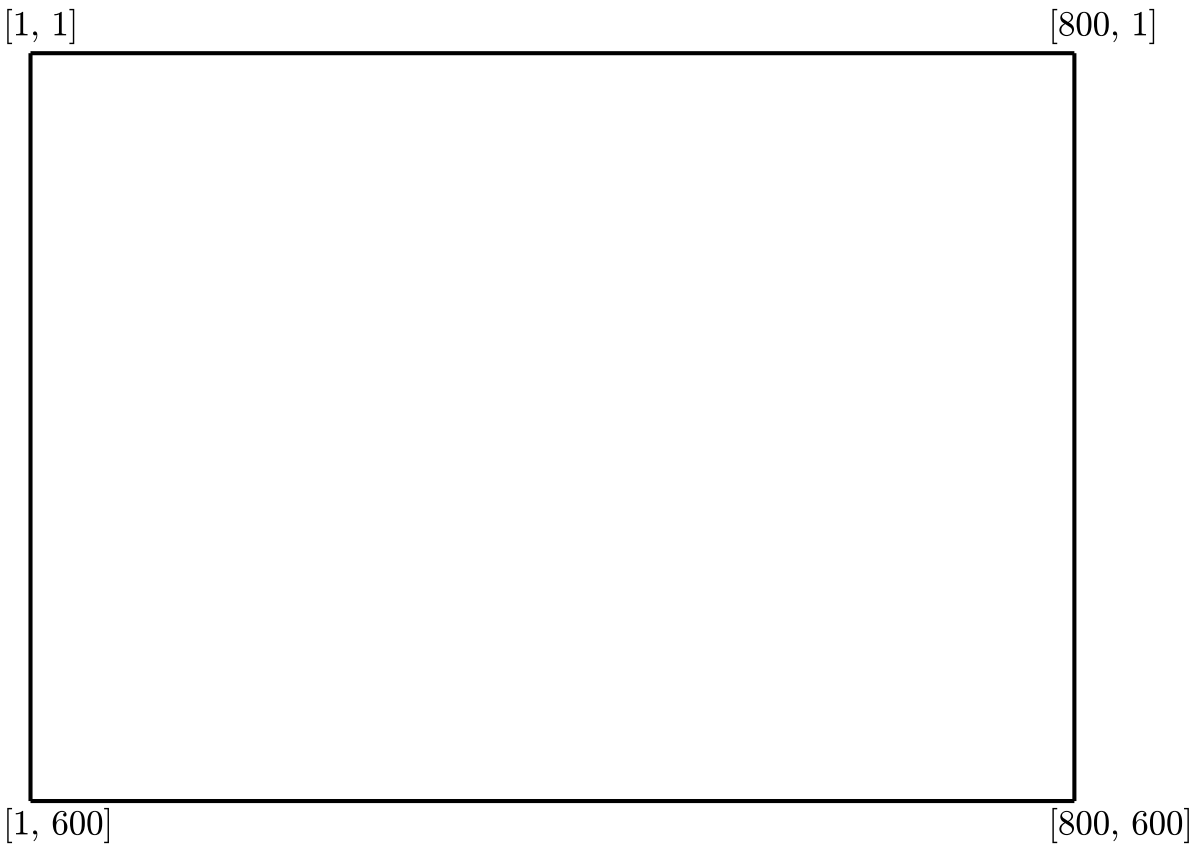
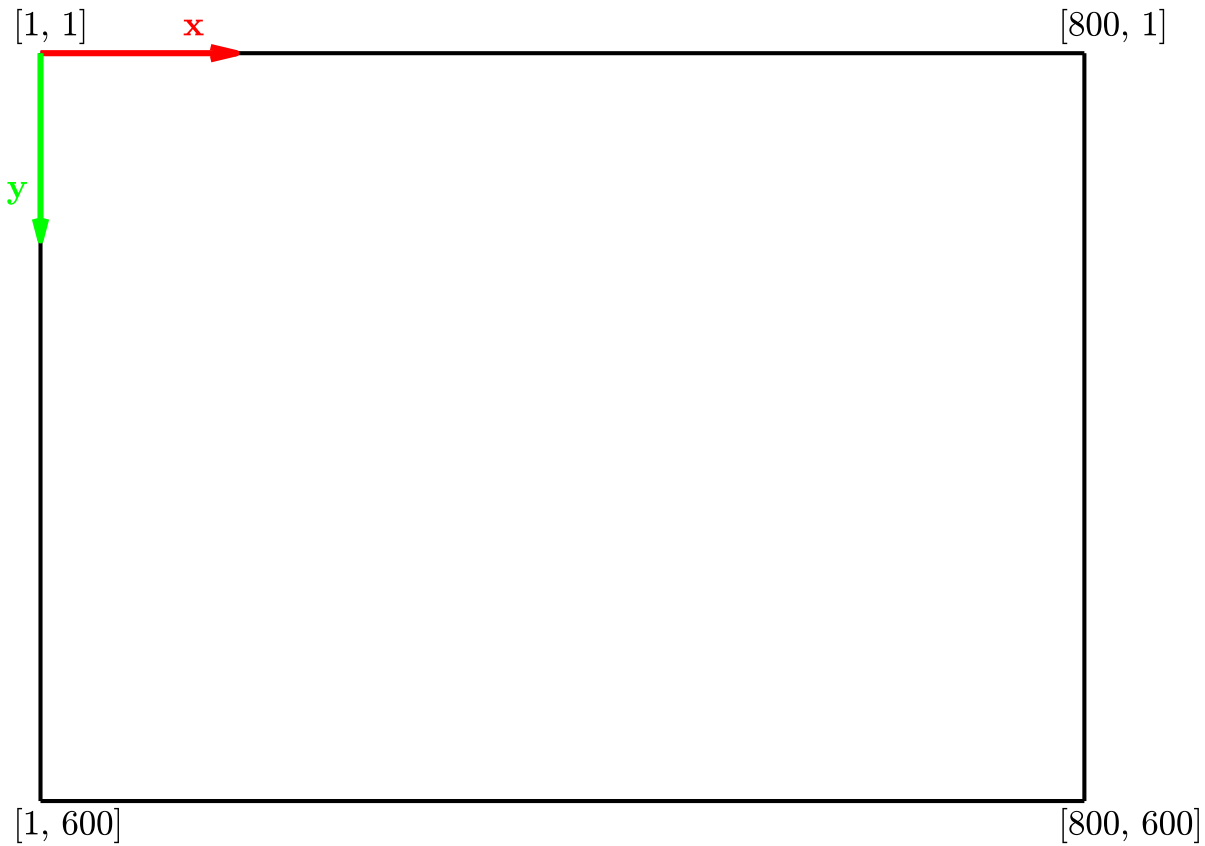


Image Boundary and Coordinate System

Let the image area has an extent [1, 1] to [800, 600]. Draw its boundary.
Use the image coordinate system, i.e., x-axis pointing **right** and y-axis pointing **down**.

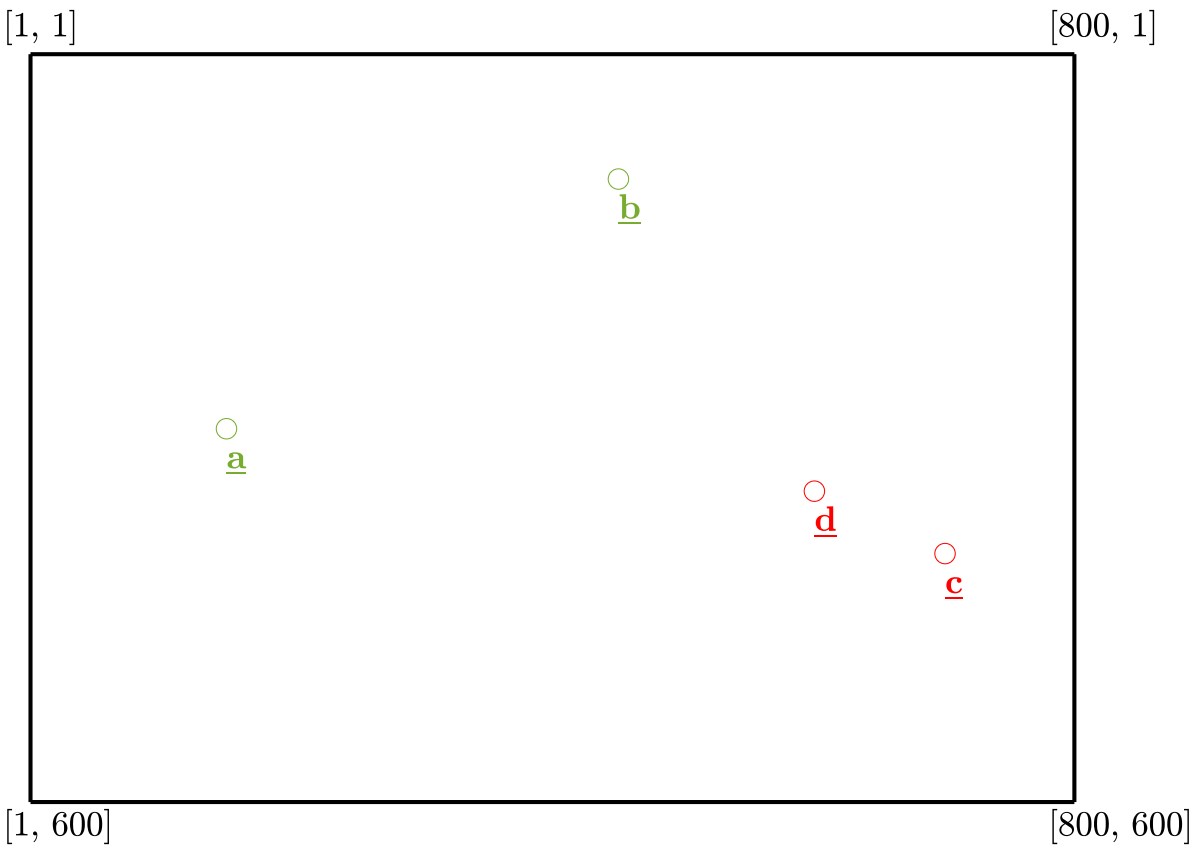


Hint: Python

```
plt.gca().invert_yaxis()
```

Drawing Points and Lines: Points

Allow entering two pairs of points within this area and display them.

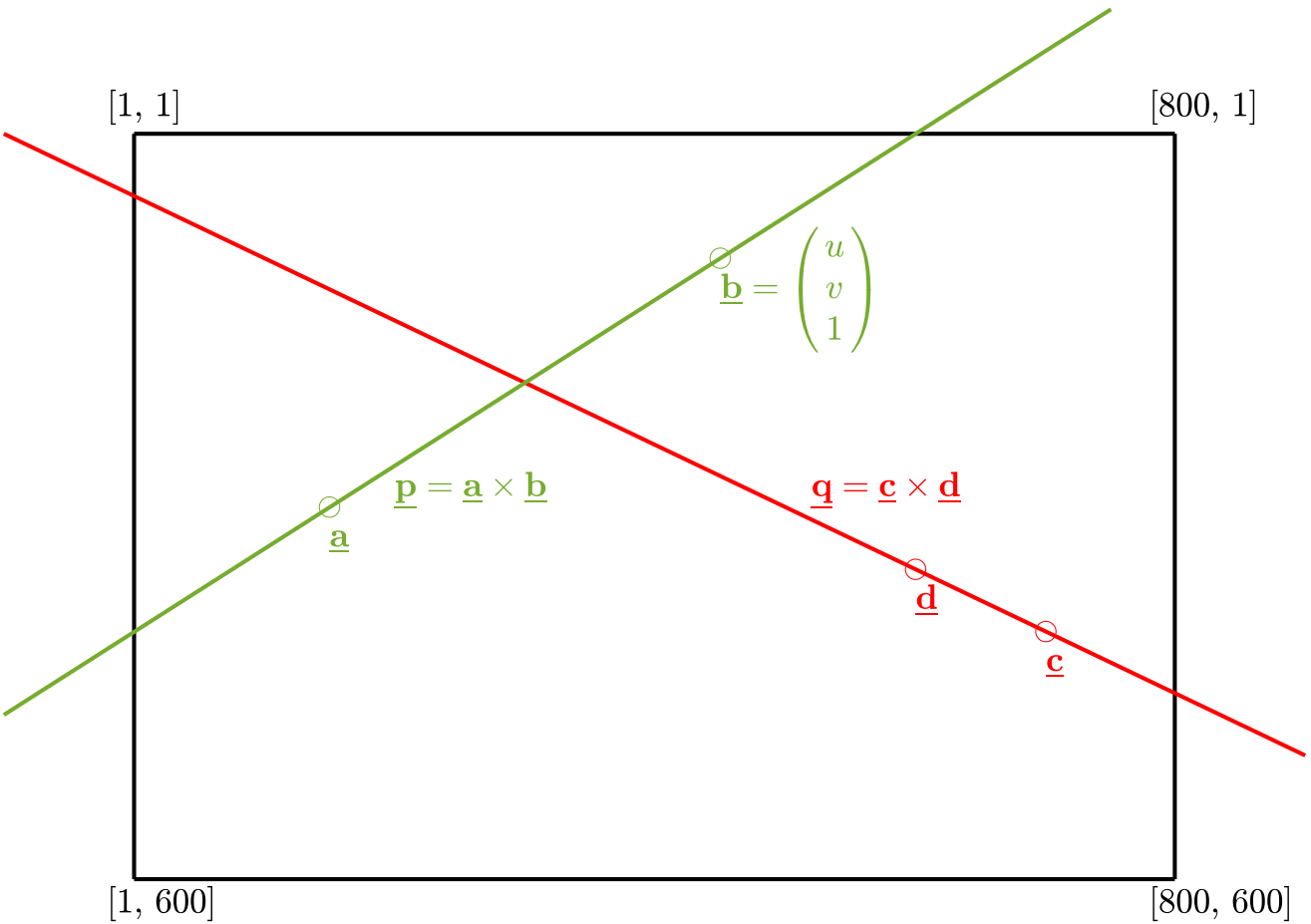


Hint: Python

```
x = plt.ginput(4)
```

Drawing Points and Lines: Join of Points

Allow entering two pairs of points within this area and display them.
Calculate the straight line passing through the first pair and the straight line passing through the second pair. **Use homogeneous representation.**



Toolbox (your code)

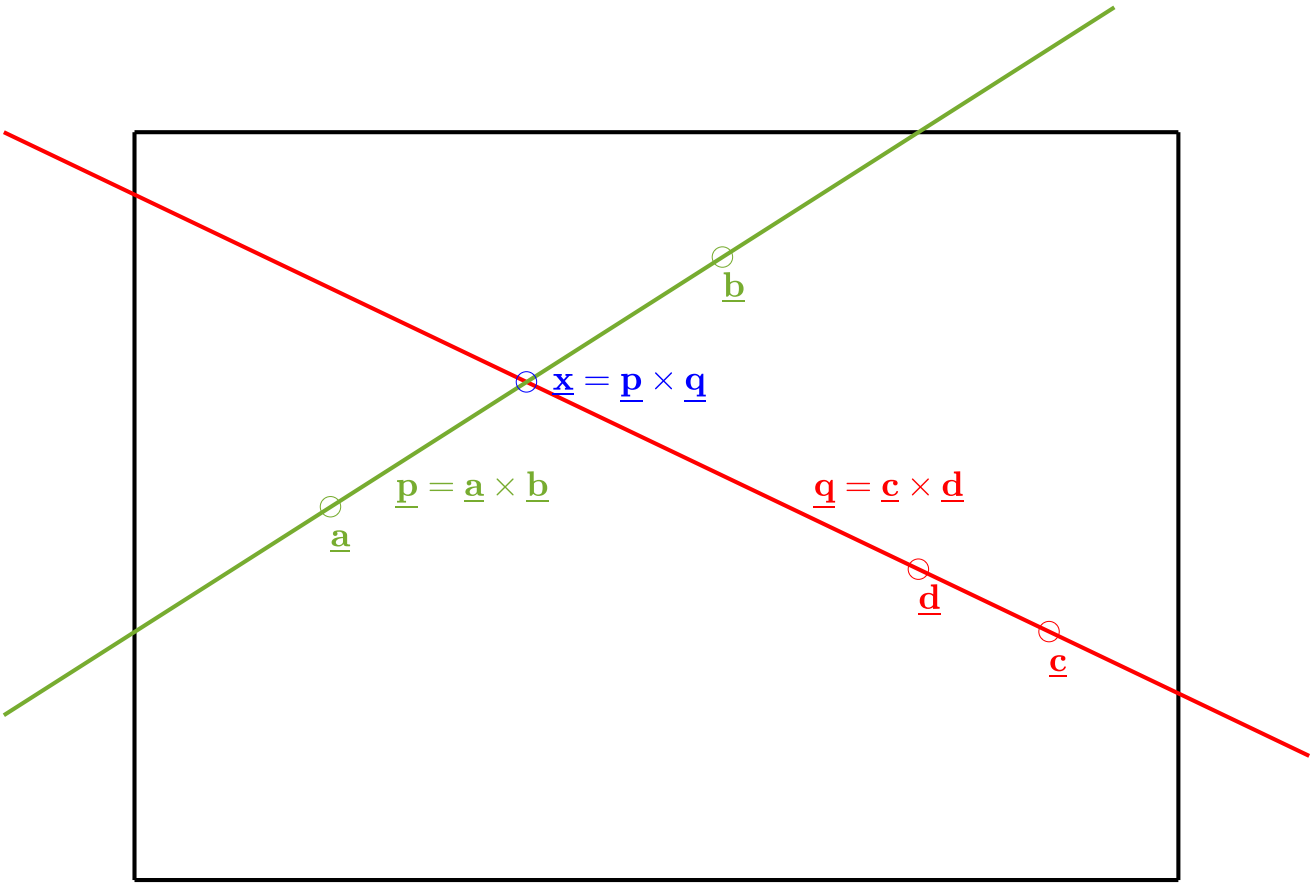
`u_p=e2p(u_e)`
transforms vectors of coordinates from euclidean to projective (homogeneous) coordinates

`u_e=p2e(u_p)`
transforms vectors of coordinates from projective (homogeneous) to euclidean coordinates

Hint: Python

```
p = np.cross(a,b)
```

Calculate the intersection of both lines and draw it, **if it is inside the image area**.

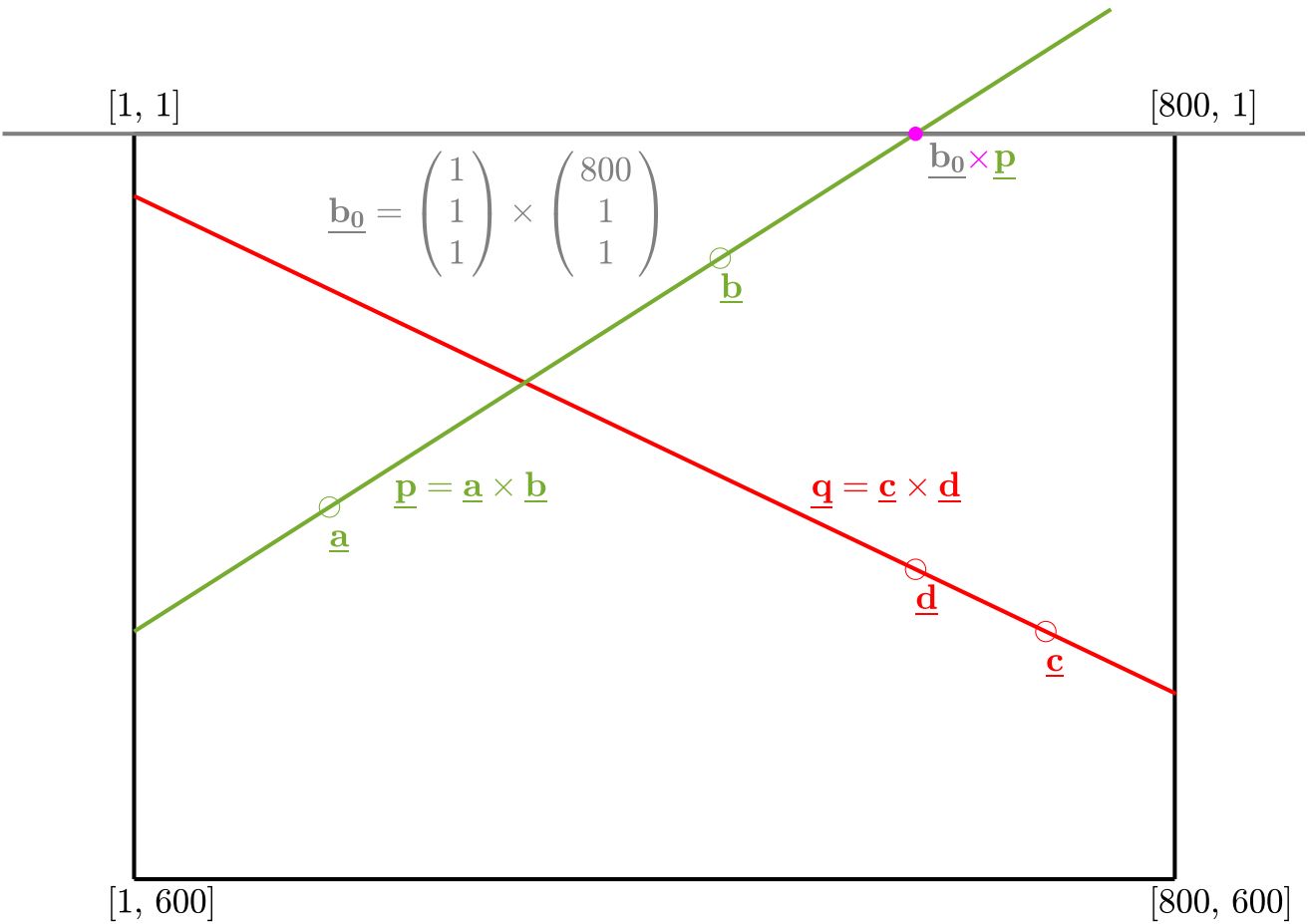


Hint: Python

```
x = np.cross(p,q)
```

Drawing Points and Lines: Crop Lines

Calculate intersections of each line with the image area and draw line segments inside.



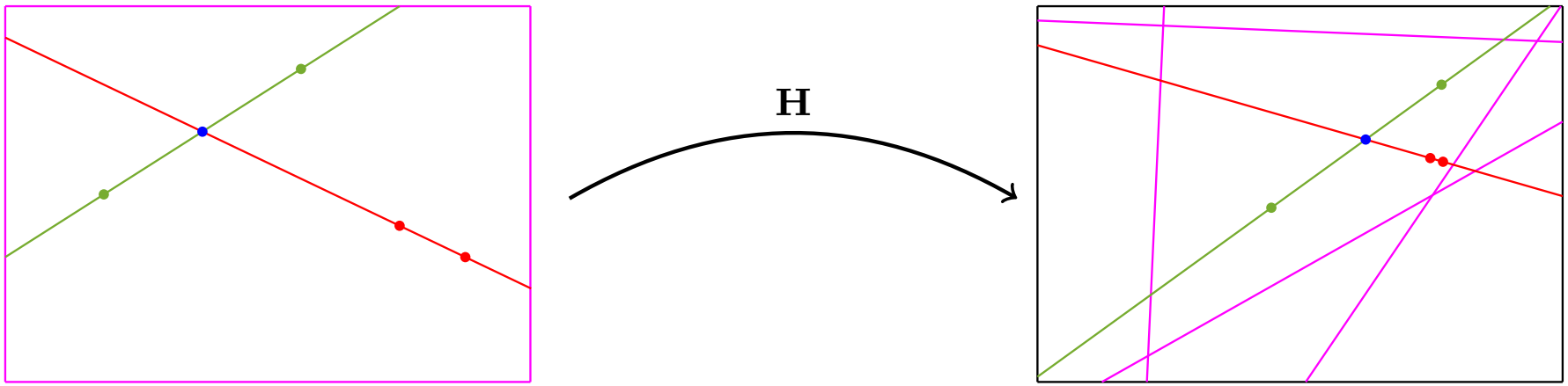
Toolbox

`u1, u2 = cropline(l, bnd)`
returns intersections of a line $\underline{\mathbf{l}}$ with a given image boundary

Apply the following homography

$$\mathbf{H} = \begin{pmatrix} 3.0 & 0.3 & 190.0 \\ 0.2 & 2.0 & 30.0 \\ 0.003 & 0.002 & 1.0 \end{pmatrix}$$

to points: $\underline{\mathbf{x}}' \simeq \mathbf{H}\underline{\mathbf{x}}$ and lines (and image boundary): $\underline{\mathbf{l}}' \simeq \mathbf{H}^{-\top}\underline{\mathbf{l}}$.
Plot everything into another figure using the original image boundaries (as visualised below on the right in black).



Transform the Line at Infinity

Use the same homography

$$\mathbf{H} = \begin{pmatrix} 3.0 & 0.3 & 190.0 \\ 0.2 & 2.0 & 30.0 \\ 0.003 & 0.002 & 1.0 \end{pmatrix}$$

to transform the line at infinity (ideal line) $\underline{\mathbf{n}}_\infty = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ and plot it inside the image boundary (see below in blue). Please, show us the two figures: