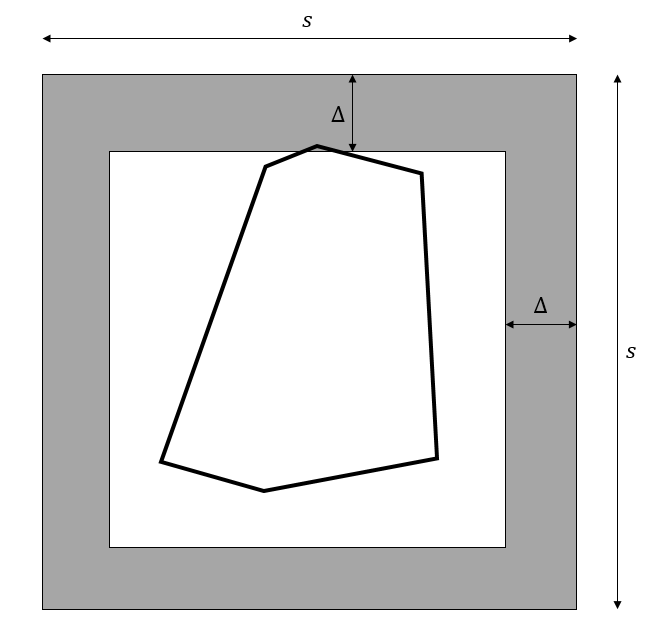
# 周りのマージンも含めてレンダリング

ビルを描画した際に、ビルがキャンバスからはみ出しているかどうかをチェックする方法として、一番外側の1ピクセルをチェックし、線があればはみ出しているとみなすという方法があるが、これでは、ぎりぎりはみ出さないケースもはみ出していると判定される。また、ほんの少しはみ出すケースもはみ出していると判定されて除外されてしまうため、トレーニング画像に偏り（大きいビルが少ない）が発生する。

この偏りを少し軽減するための方法が、キャンバスよりも大きめに描画し、外側ピクセルに線があるかどうかチェックするという方法だ。ただし、。これにより、少しだけはみ出した画像は除外されない。

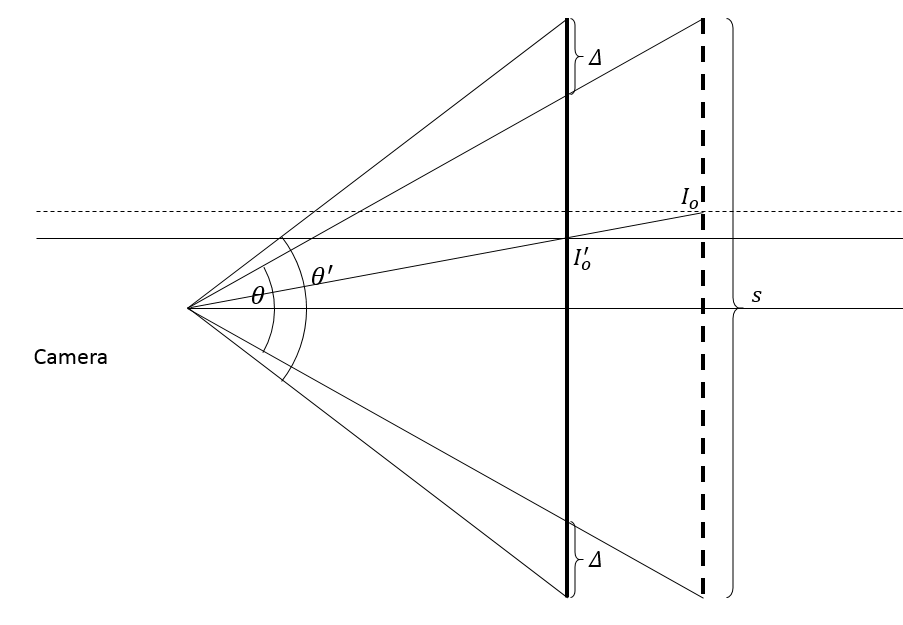


**Figure 1: Rendering with margins.**

カメラパラメータは、次のように変更する必要がある。まず、field of viewは、

次に、image planeの中心座標は、

なお、field of viewを変更しても、カメラのZ座標などは変更しない。



こうして得られるレンダリング画像は、サイズはのままだが、外側は、キャンバスの外側が描画されている。よって、外側をチェックすることで、描画されたビルがはみ出しているかどうか正確に判定できる。

以下、コード。

camera.fovy = 2 \* utils::rad2deg(atan2(height() \* 0.5, (height() \* 0.5 - margin) / tan(utils::deg2rad(camera.fovy \* 0.5))));

camera.center.x = camera.center.x \* (height() - margin \* 2) / height();

camera.center.y = camera.center.y \* (height() - margin \* 2) / height();

camera.updatePMatrix(width(), height());

...(snip)...

// add noise

if (edgeNoise) {

for (int ci = 0; ci < contour.size(); ++ci) {

contour[ci].first.x = (contour[ci].first.x - margin) / (width() - margin \* 2) \* image\_size + round(utils::genRand(-image\_size \* edgeNoiseMax \* 0.01f, image\_size \* edgeNoiseMax \* 0.01f));

contour[ci].first.y = (contour[ci].first.y - margin) / (height() - margin \* 2) \* image\_size + round(utils::genRand(-image\_size \* edgeNoiseMax \* 0.01f, image\_size \* edgeNoiseMax \* 0.01f));

contour[ci].second.x = (contour[ci].second.x - margin) / (width() - margin \* 2) \* image\_size + round(utils::genRand(-image\_size \* edgeNoiseMax \* 0.01f, image\_size \* edgeNoiseMax \* 0.01f));

contour[ci].second.y = (contour[ci].second.y - margin) / (height() - margin \* 2) \* image\_size + round(utils::genRand(-image\_size \* edgeNoiseMax \* 0.01f, image\_size \* edgeNoiseMax \* 0.01f));

}

}