Inside cost_ctrl()

Created by Cuiping Zhang, last modified on Nov 04, 2015

Overview

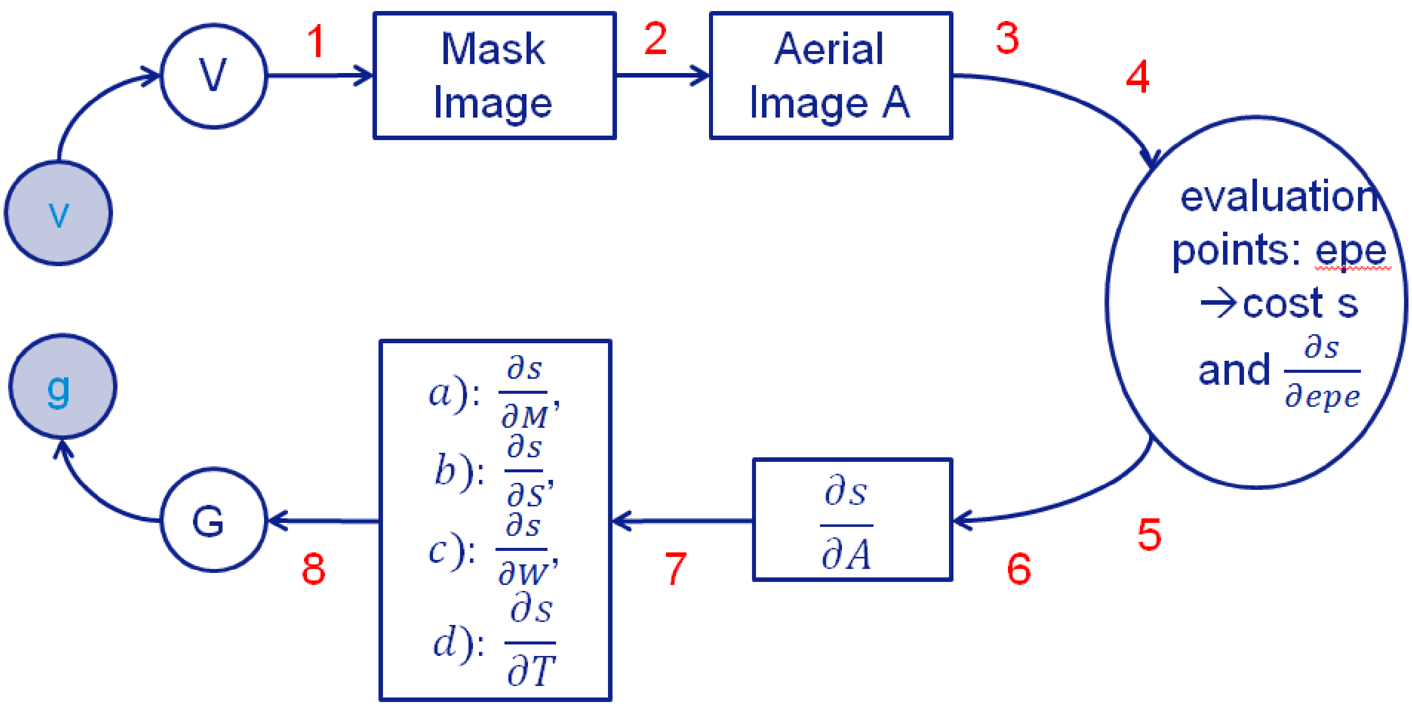
This page is to help understand cost_ctrl() function in ffoptleaf.cc.

Below is grad() flow again. For optical model, cost_ctrl() covers step #3, #4, #5, #6. For resist model, it covers step #4 and #5. Resist image can be obtained by simply thresholding aerial image for an optical model, so no resist image buffer is provided.

In another word,

- Optical model: input image ("imgs") to cost_ctrl() is aerial image. output includes cost s and ds/dA
- Resist model: input image ("imgs") to cost_ctrl() is resist image. output includes cost s and ds/dR

Let's assume an optical model is used. Details of cost_ctrl() for optical model will be given below. For resist model, only slight change is needed.



Estimate epe (edge place error) at a given evaluation point

Pixel size for aerial image is 28 nm. At an arbitrary location, its aerial image value "a" can be estimated through interpolation with an analytical 2D filter "comb" derived from sinc function. With interpolation, subpixel resolution is 1/400 of original pixel size.

$$a = \sum comb(jx, ii) comb(jj, jy) * a(k)$$

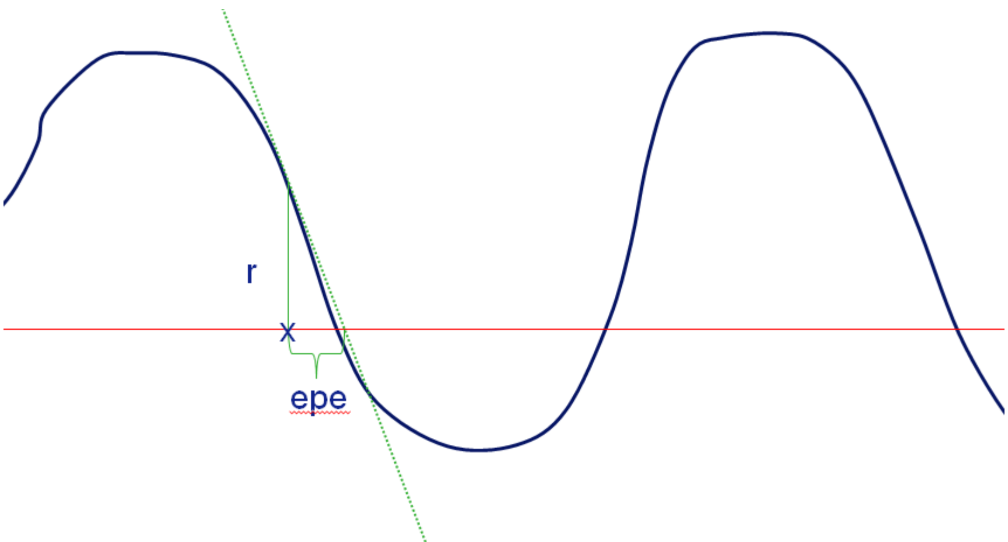
Resist image can be obtained by simply thresholding aerial image.

$$\mathrm{r} = \mathrm{a} - \frac{cts}{1+\Delta_{dose}}$$

"cts" is optical threshold value. Resist contour is formed by tracking points with r = 0. For an evaluation point, its epe is its distance to nearest resist contour. If epe is small enough, it can be estimated from resist value r and image slope:

$$e = \frac{r}{\sqrt{sn2}}$$

This is visualized below from 1D resist profile for an evaluation point located at x.



Formula for epe cost and its gradient

Epe cost

Epe-based cost is defined as

$$S_e = \sum w_0 * e^n = \sum w_0 * \left(\frac{r}{\sqrt{sn2}}\right)^n = \sum \frac{w_0}{sn2} * \left(\frac{r}{\sqrt{sn2}}\right)^{n-2} * r * r = \sum w * r * r$$

where n = 2, 4 are commonly used.

$$w = \frac{w_0}{sn2} * \left(\frac{r}{\sqrt{sn2}}\right)^{n-2}$$

S_e is summed over all evaluation points. w_0 is total weight at a given location. It is the multiplication of clip weight, PW weight, cutline weight, tag weight and other types of weight that contribute to this location. sqrt(sn2) is dose -adjusted local slope.

Cost gradient

The above cost function's partial derivative w.r.t. a variable v_k is

$$\frac{\partial S_e}{\partial v_k} = \sum w_0 * n * \left(\frac{r}{\sqrt{sn2}}\right)^{n-1} * \frac{\partial \frac{r}{\sqrt{sn2}}}{\partial v_k} = \sum w_0 * n * \left(\frac{r}{\sqrt{sn2}}\right)^{n-2} * \left(\frac{r}{\sqrt{sn2}}\right) * \frac{\partial \frac{r}{\sqrt{sn2}}}{\partial v_k}$$

$$= n * \sum \frac{w_0}{sn2} * \left(\frac{r}{\sqrt{sn2}}\right)^{n-2} * r * \sqrt{sn2} * \frac{\partial \frac{r}{\sqrt{sn2}}}{\partial v_k} = n * \sum w * r * \sqrt{sn2} * \frac{\partial \frac{r}{\sqrt{sn2}}}{\partial v_k}$$

"n" is later considered outside cost_ctrl() function by scaling cost (s = s/ (2*d_power) at the end of cost_optics().

For contribution from a single evaluation (or cutline end) point,

$$\frac{\partial S_e}{\partial v_k} = w * r * \sqrt{sn2} * \frac{\partial \frac{r}{\sqrt{sn2}}}{\partial v_k}$$

$$\frac{\partial S_e}{\partial v_k} = w * r * \sqrt{sn2} * \left(\frac{1}{\sqrt{sn2}} \frac{\partial r}{\partial v_k} - \frac{r}{2 * sn2 * \sqrt{sn2}} \frac{\partial sn2}{\partial v_k}\right) = w * r * \left(\frac{\partial r}{\partial v_k} - \frac{r}{2 * sn2} \frac{\partial sn2}{\partial v_k}\right)$$

Dose variable:

For the first item in above equation,

$$\mathrm{r} = \mathrm{a} - \frac{cts}{1+\Delta_{dose}}, \quad cts = \frac{d_{ct} * d_{aisf}}{ds}$$

$$\frac{\partial r}{\partial ds} = \frac{-1}{1+\Delta_{dose}} \frac{\partial cts}{\partial ds} = \frac{-1}{1+\Delta_{dose}} \frac{\partial \left(\frac{d_{ct} * d_{aisf}}{ds}\right)}{\partial ds} = \frac{cts}{1+\Delta_{dose}} * \frac{1}{ds}.$$

For the second item in above equation, dose-adjusted slope square is

$$\mathrm{sn2} = \mathrm{sn} * \mathrm{sn} + \mathrm{sne} * \mathrm{sne}$$ and $$\mathrm{sne} = \frac{d_{invsl}}{pixel} * ct = \frac{d_{invsl}}{pixel} * \frac{d_{ct} * d_{aisf}}{ds}$$

$$\frac{\partial sn2}{\partial ds} = 2 * sne * \frac{\partial sne}{\partial ds} = 2 * sne * \frac{d_{invsl} * d_{ct} * d_{aisf}}{pixel} * \frac{-1}{ds^2} = -2 * sne^2 * \frac{1}{ds}.$$

Therefore,

$$\frac{\partial S_e}{\partial dose} = w*r*\left(\frac{\partial r}{\partial v_k} - \frac{r}{2*sn2}\frac{\partial sn2}{\partial v_k}\right) = \frac{w*r*\frac{cts}{1+\Delta_{dose}} + \frac{w*r*r*sne^2}{sn2}}{ds}$$

$$= (w1*\frac{cts}{1+\Delta_{dose}} + w2*sn2e)/ds$$

Where w1 = w*r and w2 = w1*r/sn2. In cost_ctrl() function, it is "-" before the second item. Probably we don't need a bug revival, as the second item is negligible (flipping the sign shows no difference in the convergence curve and final cost).

Variables other than dose:

Again here is the general formula for gradient,

$$\frac{\partial S_e}{\partial v_k} = w*r*(\frac{\partial r}{\partial v_k} - \frac{r}{2*sn2}\frac{\partial sn2}{\partial v_k})$$

$$\frac{\partial S_e}{\partial v_k} = w*r*\left(\frac{\partial r}{\partial A_{ij}} - \frac{r}{2*sn2}\frac{\partial sn2}{\partial A_{ij}}\right)*\frac{\partial A_{ij}}{\partial v_k}$$

A_{ij} is aerial image. cost_ctrl() computes $\frac{\partial S_e}{\partial A_{ij}}$ only.

With fixed dose and

$$\mathrm{r} = \mathrm{a} - \frac{cts}{1+\Delta_{dose}}$$

$$\mathrm{sn2} = \mathrm{sn}*\mathrm{sn} + \mathrm{sne}*\mathrm{sne}$$

$$\frac{\partial r}{\partial A_{ij}} = \frac{\partial a}{\partial A_{ij}} \quad and \quad \frac{\partial sn2}{\partial A_{ij}} = 2*sn*\frac{\partial sn2}{\partial A_{ij}}$$

Cost gradient w.r.t. aerial image is

$$\frac{\partial S_e}{\partial A_{ij}} = w*r*\left(\frac{\partial a}{\partial A_{ij}} - \frac{r}{2*sn2}\frac{\partial sn2}{\partial A_{ij}}\right)$$

$$a = \sum comb(jx,ii)comb(jj,jy)*a(k) = \sum comb(jx)\,comb(jy)*a(k)$$

$$sx = \sum dcomb(jx,ii)comb(jj,jy)*a(k) = \sum dcomb(jx)comb(jy)*a(k)$$

$$sy = \sum comb(jx,ii)dcomb(jj,jy)*a(k) = \sum comb(jx)dcomb(jy)*a(k)$$

"dcomb" is first directive of the "comb" function.

Both sx and sy are further scaled with pixel size: sx = sx/pixel, sy = sy/pixel.

The first item in $\frac{\partial S_e}{\partial A_{ij}}$ is

$$\frac{\partial a}{\partial A_{ij}} = \sum comb(jx)comb(jy) * a(k)$$

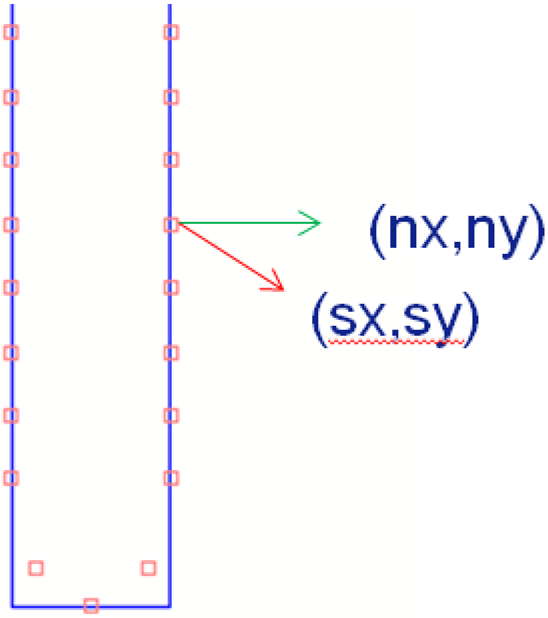
.

Given

$$\mathrm{sn2} = \mathrm{sn} * \mathrm{sn} + \mathrm{sne} * \mathrm{sne}$$

,

For job option "usenorm = 1" (default value), "sn" is normalized gradient (local gradient estimated from aerial image, then projected to normal direction).



$$sn = p.nx * sx + p.ny * sy \ for \ usenorm = 1$$

$$sn = sx * sx + sy * sy \ for \ usenorm = 0$$

Let $coefx = \begin{cases} p.nx & usenorm = 1 \\ sx & usenorm = 0 \end{cases}$, and $coefy = \begin{cases} p.ny & usenorm = 1 \\ sy & usenorm = 0 \end{cases}$

,

$$\frac{\partial sn}{\partial A_{ij}} = \left(coefx * \sum dcomb(jx)comb(jy) + coefy * \sum comb(jx)dcomb(jy)\right)/pixel$$

$$\frac{\partial S_e}{\partial A_{ij}} = w * r * \left(\frac{\partial a}{\partial A_{ij}} - \frac{r}{2 * sn2}\frac{\partial sn2}{\partial A_{ij}}\right)$$

$$= w * r * \left(\sum comb(jx)\, comb(jy) - \frac{r}{2 * sn2} * 2 * sn * \frac{\partial sn}{\partial A_{ij}}) \right)$$

$$= \ w * r * \sum comb(jx)\, comb(jy) - w * r * \frac{r}{2 * sn2} * 2 * sn * \left(coefx * \sum dcomb(jx)comb(jy) + coefy * \sum comb(jx)dcomb(jy)\right)/pixel$$

Above formula for $\frac{\partial S_e}{\partial A_{ij}}$ loops over 2D filter index (ii, jj) for the following

$$w * r * comb(jx)comb(jy) - w * r * \frac{r}{sn2} * \frac{sn}{pixel} * (coefx * dcomb(jx)comb(jy) + coefy * comb(jx) * dcomb(jy))$$

$$= w1 * comb(jx)comb(jy) - w3 * [coefx * dcomb(jx) * comb(jy) + coefy * comb(jx) * dcomb(jy)]$$

Where w1 = w * r, w2 = w1 * r/sn2, w3 = w * r * r / sn2 * sn / pixel = w2 *sn / pixel.

This is as simple as we could get for partial gradient of epe cost w.r.t. aerial image.

(end)

No labels