=1

Contributions to the Normalized Correlation and the Mean Squares Metric

Release 0.00

Marius Staring[1]

April 10, 2006

[1]Image Sciences Institute, University Medical Center Utrecht,
Q0S.459, P.O. Box 85500, 3508 GA Utrecht, The Netherlands

Abstract

This document describes contributions on the NormalizedCorrelationImageToImageMetric and the MeanSquaresImageToImageMetric of the Insight Toolkit ITK www.itk.org. For the first metric a two time speed-up can be achieved by rewriting the code to loop only once over the fixed image. This is instead of the two times that is used in the current ITK code. The reduction in computation time comes at the cost of an additional storage of a parameters array. For both metric we have implemented the option to use only a random subset of the fixed image voxels for calculating the metric value and its derivatives. This reduces the computation time (substantially), while convergence properties are maintained. This paper is accompanied with the source code.

Contents

1 Contribution 1: loop over the fixed image only once

The `itk::NormalizedCorrelationImageToImageMetric` in the current ITK release 2.6 implements the `GetDerivative()` and the `GetValueAndDerivative()` functions by using two loops over the fixed image. The code can be rewritten to loop over this image only once. In order to see this we derive formulae for the metric value and its derivative to the transform parameters. We do this for the slightly more complex case where the mean of the fixed and moving image is subtracted from the sample values.

The Normalized Correlation NC with subtracted means is given by:

$$\mathrm{NC}(p) = \frac{\sum_x \left(f(x) - \overline{f} \right) \left(m(x+u(x,p)) - \overline{m} \right)}{\sqrt{\sum_x \left(f(x) - \overline{f} \right)^2 \sum_x \left(m(x+u(x,p)) - \overline{m} \right)^2}}, \tag{1}$$

where f and m are the fixed and the moving image, respectively, $\overline{f}$ and $\overline{m}$ their means, and $u(x,p)$ the deformation as a function of the spatial position x and the transform parameters p. Rewriting (1) and substituting $\mathrm{sff} = \sum_x f(x)^2$, $\mathrm{smm} = \sum_x m(x)^2$, $\mathrm{sfm} = \sum_x f(x) * m(x+u(x,p))$, $\mathrm{sf} = \sum_x f(x)$, and $\mathrm{sm} = \sum_x m(x+u(x,p))$, as is done in the source code, we arrive at:

$$\mathrm{NC}(p) = \frac{\mathrm{sfm} - \mathrm{sf}\cdot\mathrm{sm}/N}{\sqrt{(\mathrm{sff}-\mathrm{sf}\cdot\mathrm{sf}/N)\,(\mathrm{smm}-\mathrm{sm}\cdot\mathrm{sm}/N)}}, \tag{2}$$

where N is the number of samples the metric is calculated over.

We define $\mathrm{NC} = A/B$, with A the enumerator of (1) and B the denominator. The derivative of A to the transform parameters is:

$$\frac{\partial A}{\partial p} = \frac{\partial A}{\partial m}\frac{\partial m}{\partial x}\bigg|_{x+u(x,p)} \frac{\partial(x+u(x,p))}{\partial p}, \tag{3}$$

where $\partial m/\partial x|_{(x+u(x,p))}$ is the gradient of the moving image at the mapped point $x+u(x,p)$ (which is `gradient` in the source code), and where $\partial(x+u(x,p))/\partial p = \partial T/\partial p|_{(x+u(x,p))}$ is the Jacobian of the transform at x (`jacobian` in the source code). So, we have

$$\frac{\partial A}{\partial p} = \sum_x f(x)\cdot\mathrm{gradient}\cdot\mathrm{jacobian} - \frac{1}{N}\cdot\mathrm{sf}\cdot\sum_x \mathrm{gradient}\cdot\mathrm{jacobian} \tag{4}$$

and

$$\frac{\partial B}{\partial p} = [(\mathrm{sff}-\mathrm{sf}\cdot\mathrm{sf}/N)(\mathrm{smm}-\mathrm{sm}\cdot\mathrm{sm}/N)]^{-\frac{1}{2}}\cdot(\mathrm{sff}-\mathrm{sf}\cdot\mathrm{sf}/N) \tag{5}$$

$$\cdot\left[\sum_x m(x+u(x,p))\cdot\mathrm{gradient}\cdot\mathrm{jacobian} - \frac{1}{N}\cdot\mathrm{sm}\cdot\sum_x \mathrm{gradient}\cdot\mathrm{jacobian}\right]. \tag{6}$$

Substituting $\mathrm{sumF} = \sum_x f(x)\cdot\mathrm{gradient}\cdot\mathrm{jacobian}$ (see `sumF` in the source code) and $\mathrm{sumM} = \sum_x m(x+u(x,p))\cdot\mathrm{gradient}\cdot\mathrm{jacobian}$ (see `sumM` in the source code), and $\mathrm{differential} = \sum_x \mathrm{gradient}\cdot\mathrm{jacobian}$ (see `differential` in the source code), yields

$$\frac{\partial \mathrm{NC}}{\partial p} = \frac{B\frac{\partial A}{\partial p} - A\frac{\partial B}{\partial p}}{B^2} \tag{7}$$

$$= \frac{(\mathrm{sumF}-\frac{1}{N}\cdot\mathrm{sf}\cdot\mathrm{differential}) - \frac{\mathrm{sfm}-\mathrm{sf}\cdot\mathrm{sm}/N}{\mathrm{smm}-\mathrm{sm}\cdot\mathrm{sm}/N}\cdot(\mathrm{sumM}-\frac{1}{N}\cdot\mathrm{sm}\cdot\mathrm{differential})}{\sqrt{(\mathrm{sff}-\mathrm{sf}\cdot\mathrm{sf}/N)\,(\mathrm{smm}-\mathrm{sm}\cdot\mathrm{sm}/N)}}. \tag{8}$$

All the elements from (8) can be calculated in one loop over the fixed image voxels. After the loop the value and derivative can be calculated from (2) and (8), respectively. The two times reduction in computation time comes at the cost of an additional storage of a parameters array. See the source accompanying this paper for more details.

Other differences compared to the current ITK version of `itk::NormalizedCorrelationImageToImageMetric` are

- The use of `Value()` instead of `Get()` when using the iterator: it is faster.
- When checking for a nonzero RealType denominator, substitute `denom != 0.0` with `denom < 0.00000001`.
- Getting rid of an unused `DerivativeType derivativeM1` in the `GetValueAndDerivative()`.

2 Contribution 2: use only a random subset of the fixed image voxels

Instead of calculating the value and derivative over all fixed image voxels, it is possible to calculate it over a random subset of those voxels. If this is done an error is made in the estimation of the value and the derivative. Because a new subset is randomly chosen every iteration of the optimizer, a random unbiased error is made. With this approach, Klein et al. [1] have shown for mutual information that the computation time per iteration can be extremely decreased, without affecting the rate of convergence, final precision, or robustness, by using a very small subset of the image to compute the derivative of the cost function. In his experiments he showed that the size of this subset can go down to about 2500 voxels for 32 histogram bins for 2D and 3D images.

We have implemented this using the `itk::ImageRandomConstIteratorWithIndex`. Two options are added to the `itk::NormalizedCorrelationImageToImageMetric`: a boolean defining the use of this random subset option (`UseAllPixels`, which is by default true), and a number defining how many fixed image voxels are used (`NumberOfSpatialSamples`) to calculate the value and the derivative in case `UseAllPixels` is false. Tests show that if `UseAllPixels` is set to true, the output of the new algorithm is exactly exactly equal to the current ITK implementation.

We have also implemented this option for the `itk::MeanSquaresImageToImageMetric`.

References

[1] Stefan Klein, Marius Staring, and Josien P. W. Pluim. Comparison of gradient approximation techniques for optimisation of mutual information in nonrigid registration. In *SPIE Medical Imaging: Image Processing*, volume 5747 of *Proceedings of SPIE*, pages 192 – 203. SPIE Press, 2005. 2