A patch-based framework for new ITK functionality: Joint fusion, denoising, and non-local super-resolution

Release 0.00

Nick Tustison[1] Brian Avants[2], Hongzhi Wang[3], Long Xie[5], Pierrick Coupé[4], Paul Yushkevich[5], and Jose Vicente Manjón[5]

March 15, 2017

[1]University of Virginia, Charlottesville, VA, USA
[2]Biogen, Cambridge, MA, USA
[3]IBM Research, San Jose, CA, USA
[4] Laboratoire Bordelais de Recherche en Informatique, Talence, France
[5]University of Pennsylviania, Philadelphia, USA
[6]ITACA institute, Universidad Politécnica de Valencia, Valencia, Spain

Abstract

In an earlier Insight Journal article, we introduced an ITK implementation of the adaptive patch-based image denoising algorithm described in [3]. We follow-up up that offering with a generalized non-local, patch-based ITK class framework and a refactored denoising class. In addition, we provide two ITK implementations of related, well-known algorithms. The first is a non-local super resolution method described in [1, 2]. The second is the multivariate joint label fusion algorithm of [4, 5] with additional extensions, denoted as "joint intensity fusion", which will be described in a forthcoming manuscript. Accompanying these ITK classes are documented programming interfaces which use our previously introduced unique command line interface routines. Several 2-D examples on brain imaging data are provided to qualitatively demonstrate performance.

Latest version available at the Insight Journal [http://hdl.handle.net/10380/3570]

1 Introduction

Since the non-local (NL) means algorithm introduced in several early papers, the resulting family of research techniques has grown to significant prominence within the medical imaging community. In addition to continued research output, patch-based techniques have been the subject of workshops (Patch-MI 2015 and PMI'16, both MICCAI-sponsored workshops). Given the interest and potential applications, this submission is a potential "conversation starter" for the development of an ITK software framework for accommodating such techniques.

2 ITK Implementation

2.1 Parent and child classes

Based on the implementations of different algorithms from the various authors, the common routines were grouped to the following parent class:

- itkNonLocalPatchBasedImageFilter.h
- itkNonLocalPatchBasedImageFilter.hxx

The common routines contained within this class include defining both the patch and search neighborhoods. We also define two patch similarity metrics—sum of squared differences (SSD) and Pearson's correlation. We also provide methods for "vectorizing" image patches and multiple image patches (for multi-modal applications) for facilitating similarity metric calculation.

We provide the following classes for the adaptive denoising algorithm:

- itkAdaptiveNonLocalMeansDenoisingImageFilter.h
- itkAdaptiveNonLocalMeansDenoisingImageFilter.hxx

The following helper class

- itkVarianceImageFilter.h
- itkVarianceImageFilter.hxx

These have been refactored since our previous contribution such that the adaptive denoising class is derived from the parent class. Again, the second class is the variance analog to the `itk::MeanImageFilter` which "Computes an image where a given pixel is the [variance] value of the the pixels in a neighborhood about the corresponding input pixel."

For the super-resolution algorithm, we provide the following class:

- itkNonLocalSuperresolutionImageFilter.h
- itkNonLocalSuperresolutionImageFilter.hxx

which subsumes functionalities described in both [1, 2].

Finally, for the joint fusion algorithm [4, 5] we provide the following class:

- itkWeightedVotingFusionImageFilter.h
- itkWeightedVotingFusionImageFilter.hxx

Note that all classes are multi-threaded.

2.2 Programs

We provide the following routines which use the above-listed classes:

- DenoiseImage.cxx
- NonLocalSuperResolution.cxx
- jointFusion.cxx

All these routines use an ANTs-based command line interface facilitated by the following helper classes:

- antsCommandLineOption.h
- antsCommandLineOption.cxx
- antsCommandLineParser.h
- antsCommandLineParser.cxx

These latter files are not essential for other applications or for inclusion into ITK. They are included simply to facilitate the generation of a stand-alone application that people can use.

One can invoke the help menu for all these routines via the `--help` option invoked at the command line. To see the progress of the filter, one also needs to set the verbose flag, i.e., `-v 1`. Below we give the help menu for each program.

2.3 Denoise image

```
$ DenoiseImage --help 1

COMMAND:
     DenoiseImage
          Denoise an image using a spatially adaptive filter originally described in J. V.
          Manjon, P. Coupe, Luis Marti-Bonmati, D. L. Collins, and M. Robles. Adaptive
          Non-Local Means Denoising of MR Images With Spatially Varying Noise Levels,
          Journal of Magnetic Resonance Imaging, 31:192-203, June 2010.

OPTIONS:
     -d, --image-dimensionality 2/3/4
          This option forces the image to be treated as a specified-dimensional image. If
          not specified, the program tries to infer the dimensionality from the input
          image.

     -i, --input-image inputImageFilename
          A scalar image is expected as input for noise correction.

     -n, --noise-model Rician/(Gaussian)
          Employ a Rician or Gaussian noise model.

     -x, --mask-image maskImageFilename
          If a mask image is specified, denoising is only performed in the mask region.
```

```
      -s, --shrink-factor (1)/2/3/...
            Running noise correction on large images can be time consuming. To lessen
            computation time, the input image can be resampled. The shrink factor, specified
            as a single integer, describes this resampling. Shrink factor = 1 is the
            default.

      -p, --patch-radius 1
                           1x1x1
            Patch radius. Default = 1x1x1

      -r, --search-radius 3
                            3x3x3
            Search radius. Default = 3x3x3.

      -o, --output correctedImage
                     [correctedImage,<noiseImage>]
            The output consists of the noise corrected version of the input image.
            Optionally, one can also output the estimated noise image.

      -v, --verbose (0)/1
            Verbose output.

      -h
            Print the help menu (short version).

      --help
            Print the help menu.
```

2.4 Super-resolution

```
$ NonLocalSuperResolution --help 1

COMMAND:
      NonLocalSuperResolution
            Non-local super resolution described in the following papers: 1) JV Manjon, P
            Coupe, A Buades, V Fonov, DL Collins, and Montserrat Robles. Non-local MRI
            Upsampling.Medical Image Analysis, 14:784-792, 2010 and2) JV Manjon, P Coupe, A
            Buades, DL Collins, and Montserrat Robles. MRI Superresolution Using
            Self-Similarity and Image Priors.International Journal of Biomedical Imaging,
            2010.

OPTIONS:
      -d, --image-dimensionality 2/3/4
            This option forces the image to be treated as a specified-dimensional image. If
            not specified, the program tries to infer the dimensionality from the input
            image.

      -i, --input-image inputImageFilename
            A low-resolution image input image to be superresoluted.

      -j, --interpolated-image inputImageFilename
            An interpolated version of the low-resolution image (such as B-spline). One
```

```
      should specify either this option as a secondary input or a high-resolution
      multi-modal counterpart (cf the -k option).

-k, --reference-image inputImageFilename
      A high resolution reference multi-modal image. Assumed to be in the same space
      as the low-resolution input image (i.e., registered).One should specify either
      this option as a secondary input or an interpolated version (cf the -j option).

-p, --patch-radius 1
                         1x1x1
      Patch radius. Default = 1x1x1

-r, --search-radius 3
                          3x3x3
      Search radius. Default = 3x3x3.

-g, --intensity-difference-sigma 1.0
      Intensity difference sigma. Default = 1.0

-t, --patch-similarity-sigma 1.0
      Patch similarity sigma. Default = 1.0

-s, --scale-levels 32x16x8x2x1
      Scale levels. Default = 32x16x8x2x1

-n, --interpolation Linear
                          NearestNeighbor
                          Gaussian[<sigma=imageSpacing>,<alpha=1.0>]
                          BSpline[<order=3>]
                          CosineWindowedSinc
                          WelchWindowedSinc
                          HammingWindowedSinc
                          LanczosWindowedSinc
      Several interpolation options are available in ITK. These have all been made
      available.

-o, --output outputImage
      The output consists of the noise corrected version of the input image.
      Optionally, one can also output the estimated noise image.

--version
      Get Version Information.

-v, --verbose (0)/1
      Verbose output.

-h
      Print the help menu (short version).

--help
      Print the help menu.
      <VALUES>: 1
```

2.5 Joint fusion

```
$ jointFusion --help 1

COMMAND:
     antsJointFusion
          antsJointFusion is an image fusion algorithm developed by Hongzhi Wang and Paul
          Yushkevich which won segmentation challenges at MICCAI 2012 and MICCAI 2013. The
          original label fusion framework was extended to accommodate intensities by Brian
          Avants. This implementation is based on Paul's original ITK-style implementation
          and Brian's ANTsR implementation. References include 1) H. Wang, J. W. Suh, S.
          Das, J. Pluta, C. Craige, P. Yushkevich, Multi-atlas segmentation with joint
          label fusion IEEE Trans. on Pattern Analysis and Machine Intelligence, 35(3),
          611-623, 2013. and 2) H. Wang and P. A. Yushkevich, Multi-atlas segmentation
          with joint label fusion and corrective learning--an open source implementation,
          Front. Neuroinform., 2013.

OPTIONS:
     -d, --image-dimensionality 2/3/4
          This option forces the image to be treated as a specified-dimensional image. If
          not specified, the program tries to infer the dimensionality from the input
          image.

     -t, --target-image targetImage
                              [targetImageModality0,targetImageModality1,...,targetImageModalityN]
          The target image (or multimodal target images) assumed to be aligned to a common
          image domain.

     -g, --atlas-image atlasImage
                            [atlasImageModality0,atlasImageModality1,...,atlasImageModalityN]
          The atlas image (or multimodal atlas images) assumed to be aligned to a common
          image domain.

     -l, --atlas-segmentation atlasSegmentation
          The atlas segmentation images. For performing label fusion the number of
          specified segmentations should be identical to the number of atlas image sets.

     -a, --alpha 0.1
          Regularization term added to matrix Mx for calculating the inverse. Default =
          0.1

     -b, --beta 2.0
          Exponent for mapping intensity difference to the joint error. Default = 2.0

     -c, --constrain-nonnegative (0)/1
          Constrain solution to non-negative weights.

     -p, --patch-radius 2
                           2x2x2
          Patch radius for similarity measures. Default = 2x2x2

     -m, --patch-metric (PC)/MSQ
          Metric to be used in determining the most similar neighborhood patch. Options
          include Pearson's correlation (PC) and mean squares (MSQ). Default = PC (Pearson
```

```
      correlation).

-s, --search-radius 3
                        3x3x3
                        searchRadiusMap.nii.gz
      Search radius for similarity measures. Default = 3x3x3. One can also specify an
      image where the value at the voxel specifies the isotropic search radius at that
      voxel.

-e, --exclusion-image label[exclusionImage]
      Specify an exclusion region for the given label.

-x, --mask-image maskImageFilename
      If a mask image is specified, fusion is only performed in the mask region.

-o, --output labelFusionImage
                intensityFusionImageFileNameFormat
                [labelFusionImage,intensityFusionImageFileNameFormat,<labelPosteriorProbabilityImageFileNameFormat>]
      The output is the intensity and/or label fusion image. Additional optional
      outputs include the label posterior probability images and the atlas voting
      weight images.

--version
      Get version information.

-v, --verbose (0)/1
      Verbose output.

-h
      Print the help menu (short version).

--help
      Print the help menu.
      <VALUES>: 1
```

3 Usage examples

Instead of providing examples contained within the submission, we point to self-contained github repos which showcase usage for each of the three programs.

3.1 Denoise image

- https://github.com/ntustison/DenoiseImageExample

3.2 Super-resolution

- https://github.com/ntustison/NonLocalSuperResolutionExample

3.3 Joint fusion

Note that for the following example, the user will need to replace “antsJointFusion” with “jointFusion” due to some legacy issues in preparing this submission.

- https://github.com/ntustison/MalfLabelingExample

References

[1] José V Manjón, Pierrick Coupé, Antonio Buades, D Louis Collins, and Montserrat Robles. Mri superresolution using self-similarity and image priors. *Int J Biomed Imaging*, 2010:425891, 2010. (document), 2.1

[2] José V Manjón, Pierrick Coupé, Antonio Buades, Vladimir Fonov, D Louis Collins, and Montserrat Robles. Non-local mri upsampling. *Med Image Anal*, 14(6):784–92, Dec 2010. (document), 2.1

[3] José V Manjón, Pierrick Coupé, Luis Martí-Bonmatí, D Louis Collins, and Montserrat Robles. Adaptive non-local means denoising of mr images with spatially varying noise levels. *J Magn Reson Imaging*, 31(1):192–203, Jan 2010. (document)

[4] Hongzhi Wang, Jung W Suh, Sandhitsu R Das, John B Pluta, Caryne Craige, and Paul A Yushkevich. Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell*, 35(3):611–23, Mar 2013. (document), 2.1

[5] Hongzhi Wang and Paul A Yushkevich. Multi-atlas segmentation with joint label fusion and corrective learning-an open source implementation. *Front Neuroinform*, 7:27, 2013. (document), 2.1