# WMA Identifier/Scorer Application (WMAS)

#### Work Notes

## Some images with and without artifacts for evaluation

Case 314 was reconstructed nominally and with interpolation every 2,3,4 and 5 lines.

Saved at: “E:\Windmill\DATA for Scorer”

## Bug with display of directed data – 14/2/23

There is different data for each of 4 directions.

Better to display only a single shared buffer.

Make the data for directory local (not shared) –

And prepare a single shared buffer for display…

## Save and restore (case related) hints – 14/2/23

Hints are saved to a special directory under the current case.

Save and restore seems to work.

## Understand how the Client App can follow graphics on ImageR – 14/2/23

There are directories like “D:\MyLog\Graphics\WMA\_Roi1\_im162”

Apparently, there are (up to) 32 saved versions of the ROI in each.

mnSaveFiles = 32;

#### The design seems to be insufficient

There is only one message for graphic save (or active) – and no clear way to see which graphic was saved.

Upgrade: Add ID to graphic elements – and send them with the save message!

## Collecting Hints – 13-14/2/23

Hint Ellipses are either WMA or No-WMA –

Hints are

1. Initiated by the scorer app
2. Manipulated interactively by ImageR
3. Saved and restored by WMAS

Technical comments

1. Start with manipulating, saving and restoring a single WMA hint
2. WMA should be RED – No-WMA should be Green
3. Initialize save directories to be empty when new hint is started
4. Save hints in different directories for pros and cons

## Revive the Demo App – 12-13/2/23

3>Target GetReferencedVCProjectsInfo:

3> C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\MSBuild\Microsoft\VC\v160\Microsoft.CppBuild.targets(1008,11): error MSB4006: There is a circular dependency in the target dependency graph involving target "GetReferencedVCProjectsInfo".

3>

3> Done building project "DemoApp.vcxproj" -- FAILED.

#### Create new project and new solution

There is a problem to create Dialog Based MFC application –

But, after some attempts the wizard opened with the proper options.

All old files were brought back and a working Demo App was committed and saved to GitHub.

## Next – Collect Hints: WMA and no-WMA ROIS for a single case – 12/2/23

Current goal –

When I have a collection of hints, try to train a small CNN to distinguish between them!

Every pixel inside a “hint area” can be considered a training goal.

As the CNN is small – hope it can’t just remember the specific pixels.

Try to prove that by training on some hints you get good results on others.

It should help that, if the identification is not working fine, I add hints where it is wrong.

#### First – accumulate hints

Define hints and save them for next runs.

How does the client app receive the saved graphics?

## Failure to build ImageR – 12/2/23

Fails to find “MyUtils.lib” – why? It shouldn’t even try! – there is ”yUtils.lib”!

ImageR went back to “win32” configuration – and there it was still “MyUtils”.

Project file edited manually…

## Allow GE to be freezed – in ImageR (and XML) – 7-8/2/23

It works

## Design better tools to improve WMAS algorithm – 7/2/23

Manually define a set of ROIS on different selected images. Give them manual scores.

For any “trial” of a variant of the algorithm –

Compare the relative scores to the manual scores –

Check whether there is improvement or regression.

May add graphs

## Improve basic detection algorithms – 5/2/23

Maybe use lower threshold to real diff – to distance noisy non-WMA areas.

## The scorer should run in a different thread than the GUI – 4/2/23

And update only to the latest image position…

It is working.

Added base class to all score computers.

Added support to 4 directions.

## Use 4 directions – 4/2/23

New class doing all computations for a single direction.

Hold array of this class.

Look for maximum score over 4 directions.

New class CDirScore – start with Prep Diff and holds the positive and negative dir dif.

## Try Some real progress – 4/2/23

1. Scores are displayed as shared memory - Done
   1. Next: set columns to 3 and show 2 additional intermediary images.
   2. Focus window of shared memory around point with artifact
2. Get position broadcast from ImageR - Done
   1. Try to compute score on leafing
   2. Show location of score on all images by ROI

## Resume work – 3/2/23

First work on WMAS – make it usable.

Goals:

1. Add shared memory to display scoring data - done
2. Find directed diff on all 4 directions
   1. Create new class for directed diff
3. Make WMAS follow leafing in ImageR – and update scores accordingly
   1. Can it be real time?
4. Show scores as graphics on all current images
   1. Identify area of artifact
5. Make WMAS check whole volume and show highest scores

## Give score to area with highest level of Smoothed Directed Amplitude – 24/1/23

Try to use just “max” after smooth…

It seems to give reasonable score for “image 127”.

## Prefer areas with consistent directed diff – 24/1/23

Define “consistency range” – try multiplication factor of [1/2 – 2]

Check how many values within range in 5\*5 bounds pixels – and multiply by it.

What is the input for this “consistency”?

Try using Dir-Amp before smooth…

Seems to work well!

## Mask strong edges – 24/1/23

Compute edge-score for each pixel. Smooth it.

Mask areas with strong edge in original Diff.

Masking strong edges do great service to distinguish relevant edges!

## Find local direction with amplitude – 23/1/23

Try:

Dir Amplitude =

if (current dir) diff amplitude \* 2

if (neighbor dir) diff amplitude

if (vert dir) - diff amplitude

Combine Dir Amp image for same direction for pos and neg

Smooth it for 3\*3, 5\*5, etc.

#### Conclusion –

It is pretty good –

But still now strong diffs, even after zoom, take precedent.

Better to identify relatively large areas with “constant” level of directed diffs –

More computations can be applied to peak areas in the smoothed directed noise.

Also, may mask or reduce effect of area with strong edges in the original image…

## Find local direction of diff around any point – 23/1/23

Working (initially) only for points that are local high…

Work in 5\*5 square around the point.

In the outer pixels there are 18 pixels – 9 pairs of opposite pixels.

Choose the pair with highest lower pixel.

Give index in the range [1-9] to the central pixel.

Do it for positive and negative separately.

#### Correction

Start with 3\*3 square, there are 4 possible directions, give them scores in the range [1-4].

## Separate negative and positive diff – 23/1/23

Separation is working…

## Compute WMA specific score – 16/1/23

Some ideas:

1. Look for consistent diff – clip every diff down to the average of its 2 closest neighbors
2. Look for area where both positive and negative diff is high

## Let scorer give initial scores – 15/1/23

It may be average abs diff 🡪 It works.

## Start working on Windmill Scorer – 14/1/23

New class to load files – CmultiDataF in ImageRLib.

Now take the two current images and give some score and region definition.

## Demo App problem – 14/1/23

1. C:\Program Files (x86)\Microsoft Visual Studio\2019\
2. Professional\MSBuild\Microsoft\VC\v160\Microsoft.CppBuild.targets(1008,11): error MSB4006: There is a circular dependency in the target dependency graph involving target "GetReferencedVCProjectsInfo".

## Very initial scorer functionality – 14/1/23

1. Load 2 volumes
2. Remember loaded volumes for next run
3. Display volumes with ImageR
4. For current image (as selected in ImageR)
   1. Compute Diff
   2. Look for areas with big difference
   3. Show these areas on ImageR
   4. Compute and Display score

## New application started – 13/1/23

The new application is in “D:\SW\WindmillScorer”

It is also in GITHUB at [git@github.com:YoavHaifa/WindmillScorer.git](mailto:git@github.com:YoavHaifa/WindmillScorer.git)

Prepare special design doc…

## Basic Wanted Functionality – 8/1/23

1. Load 2 images volumes
   1. High Resolution
   2. Low resolution
2. Identify WMA
3. Score by severity
4. Work both with GUI and as command-line