MITRE Product: MP060181

Cropping WSQ

Options and Impacts

Margaret Lepley, The MITRE Corporation

September 2006

Sponsor: DOJ/FBI
Dept. No.: G036
Contract No.: W15P7T-05-C-F600
Project No.: 0706E02X

MITRE

Abstract: WSQ[1] is a wavelet-based compression algorithm used by the FBI with a large and growing archive of WSQ fingerprint images, often collected in a digital 10-print file[2] that contains a 4-finger flat impression as well as single finger rolled images. Sometimes a sequence error occurs and a single finger image is obviously incompatible with the 4-finger view of the same finger. When this occurs, past practice with hardcopy has been to crop a section of the 4-finger image to replace the erroneous single finger image. At other times, when a 4-finger flat scanner is used, only the 4-finger image may be available. This report studies various methods of cropping a 4-finger electronic WSQ file, and determines which method and operational instructions best maintain image quality. Other aspects of fingerprint cropping, such as automatically finding the crop location, and removal of other fingerprints or writing outside the print of interest are not addressed in this report.

Executive Summary

- 2 Crop Window placements compared
  - No restrictions (UL off grid)
  - UL corner restricted to 32x32 grid
  - (Finger location already known)
- 3 Cropping Algorithms compared
  - Expand/Crop/Recompress with current tools
  - Crop WSQ wavelet coefficients – custom code
  - Smart Expand/Crop/Recompress – custom code
- Best Results from
  - UL corner of crop window on 32x32 grid
  - Smart Exp/Crop/Recompress – maintaining Quantization tables

MITRE

Where the crop window is placed has a large impact on the final cropped image quality. We assumed that some outside source identified the region of interest, and then tried cropping at exactly that location (no cropping restriction), or at nearby restricted grid locations (ensuring the upper left corner of the crop area was a multiple of 32). Of these two the restricted cropping had much better performance for all the cropping algorithms attempted.

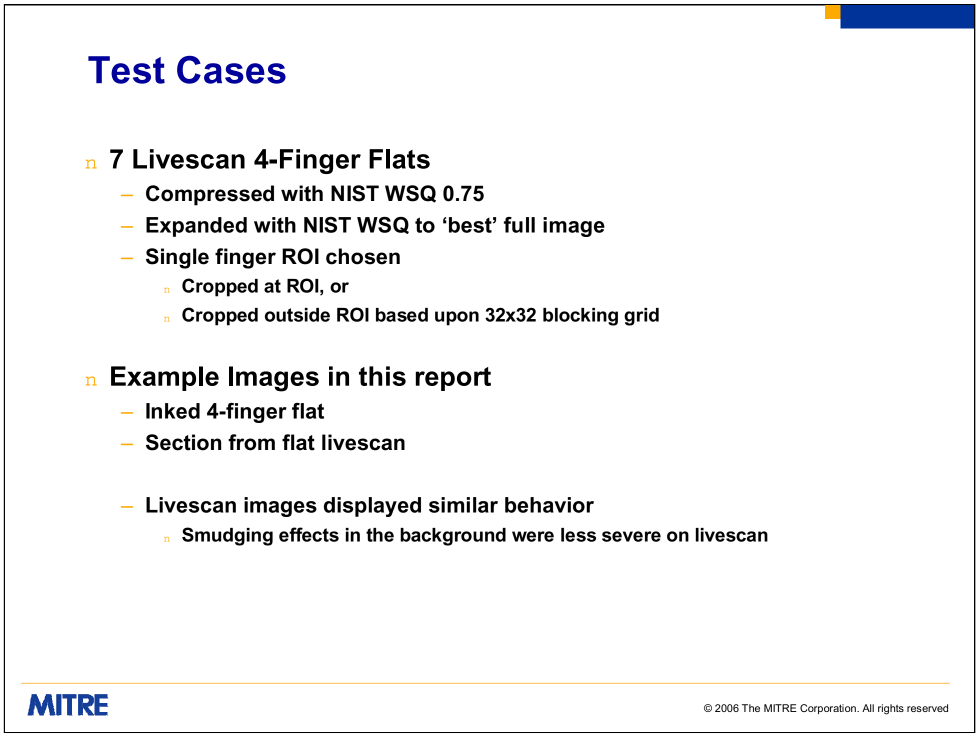
Three methods of cropping were compared. The first approach is brute-force, easily available with current tools: expand the WSQ to an uncompressed format, crop using your favorite image tool, and then recompress using standard WSQ. The second approach performs the cropping entirely in wavelet space, maintaining only the quantized wavelet coefficients needed to reconstruct an image for the cropped area. The third approach is similar to the first, but varies in two important ways: the intermediate image is floating point and the quantization parameters used in the original WSQ file are maintained in the new cropped WSQ file. Of these three techniques, the third, smart exp/crop/recompress, had the best performance.

Details of the testing and relative performance of each algorithm are given in the remainder this report.

Roadmap

MITRE

This report is comprised of 5 sections. The first section lays out the basic testing setup including test data, evaluation software and cropping options. The next three sections take a detailed look at each of the 3 cropping algorithms being analyzed. Finally the most important comparison images and conclusions are summarized.



A set of 7 livescan 4-finger flat images, with a slight range of qualities (none that would be rated poor) was used for testing. Each original image was compressed with the NIST version of WSQ[3] using the standard 0.75 bitrate. These WSQ files were the starting point for all experiments, and the original uncompressed images were not referred to again.

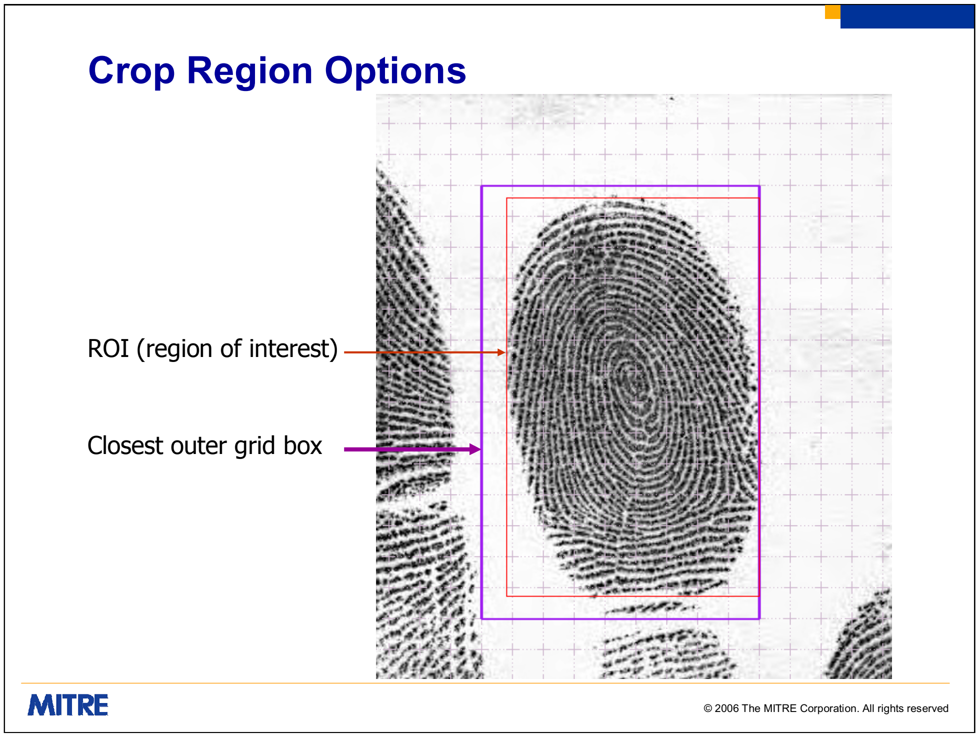
For each image, a single finger region of interest (ROI) was manually located and recorded as upper left (UL) position and width/height. These ROI parameters were recorded and used consistently with all methods throughout the testing.

After testing a publically releasable image was obtained for illustration purposes. This is an reasonably clean inked 4-finger image and demonstrates many of the cropping effects seen on the livescan images. Since the inked print has more background noise, some of the effects were more pronounced on the inked print. A few small zoomed subsections from a livescan are shown in a couple of the examples to demonstrate some quality alterations not seen on the inked example.

Overview of Full Image: after WSQ expansion

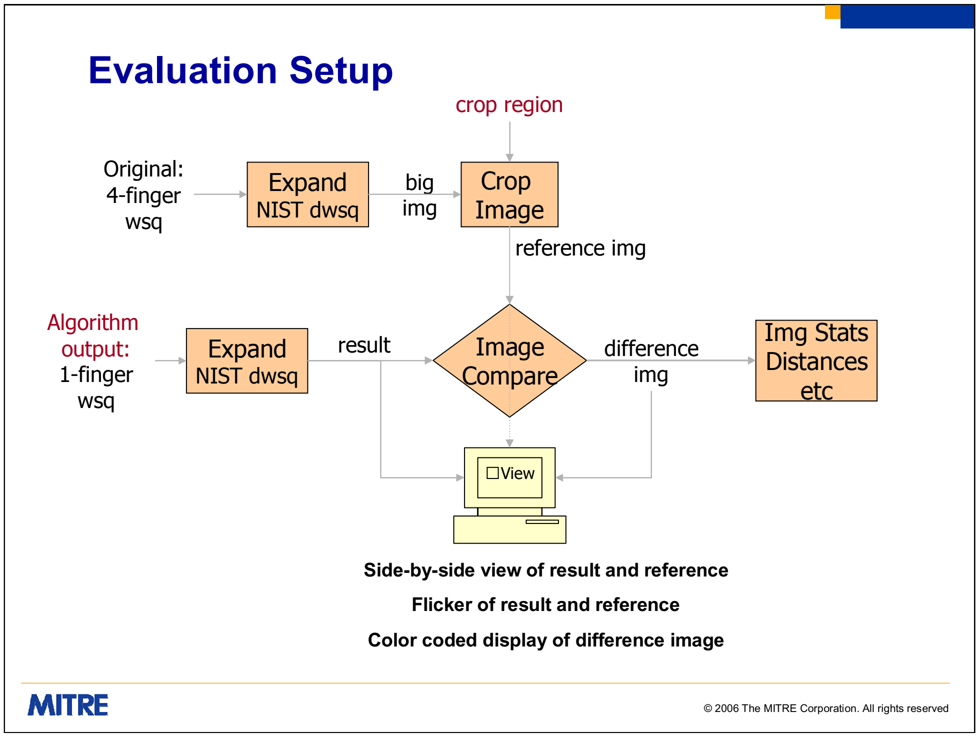
MITRE

This is one of the full 4-finger images (scaled down to fit) viewed after decompression from the WSQ file. The ring finger (3rd from left) was used for the region of interest (ROI) and will be shown throughout this report.



Here’s a full scale view in the vicinity of the ROI. A 32x32 grid (beginning at UL of the full image) has been overlaid on the image, as well as several possible cropping locations.

The ROI itself has an UL corner that isn’t on a 32x32 grid corner and is one cropping option. The alternative is a larger crop box that is aligned to the grid. This distinction between the original ROI, with an UL corner off grid, vs. a crop box that has UL corner on grid is very important and will play a key role the results.



The different cropping algorithms were evaluated using a reference crop image created by expanding the 4-finger WSQ and cropping it directly to the same area that the algorithm uses. The single finger WSQ produced by the algorithm being tested was then expanded and compared directly with the reference image. Both side-by-side static viewing as well as flicker were used to locate and categorize any changes. In addition the reference and result image were differenced (absolute difference) and this difference image was viewed with a color table designed to illustrate different scales of change. The difference image was also evaluated with image statistics such as histogram, mean, max, standard deviation, etc.

Specialized code was used to see how far into the difference image various DN values occurred. This made it possible to see which algorithms were better at protecting some interior portion of the image from quality changes, even if the boundary area had more severe alterations.

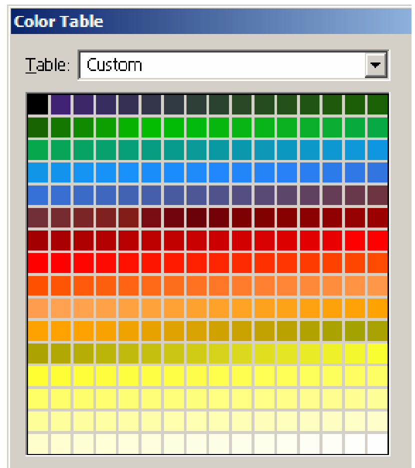
The basic parameters that could alter in this setup are highlighted in red: the location of the crop region and the algorithm used to produce the 1-finger WSQ.

Difference Color Table

|Cropped – Reference|

- **Black = identical**
- **Purple = not visible**
- **Dark green = zoom flicker only, no structure change**
- **Bright grin = visible w/o flicker**
- **Bright blue/red/etc = visible, may be structural change**

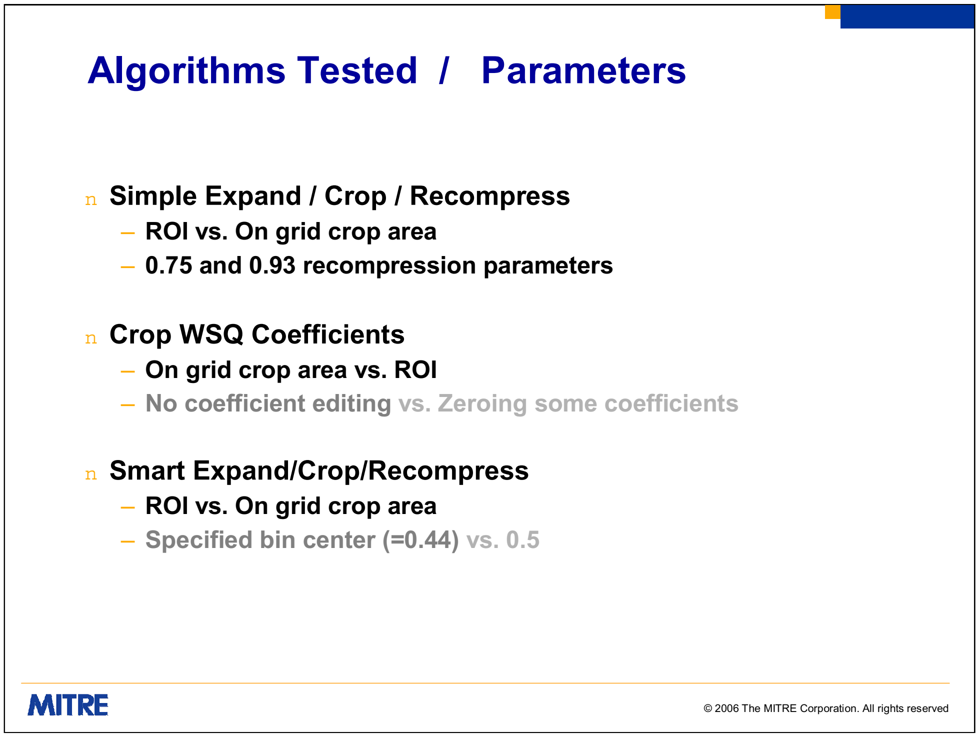
- **DN value loosely tied to structural change**
  - **Broad area constant difference**
    - **low structural impact**
  - **Streaks, small dot changes**
    - **higher structural impact**



MITRE

Since many of the differences were small, and hard or impossible to see when displayed as grayscale, a color table was used. Goals for the color table included: a) make a clear distinction between 0 and other values, b) make the very low differences non-intrusive, c) make color changes in ranges possibly related to visual characteristics. This is by no means the ideal color table, and any hypotheses formulated based upon image viewing with the color table was also checked with other methods. It does however, help to quickly draw attention to factors that may be of interest for further investigation.

Make sure to view any pages showing difference images in color!



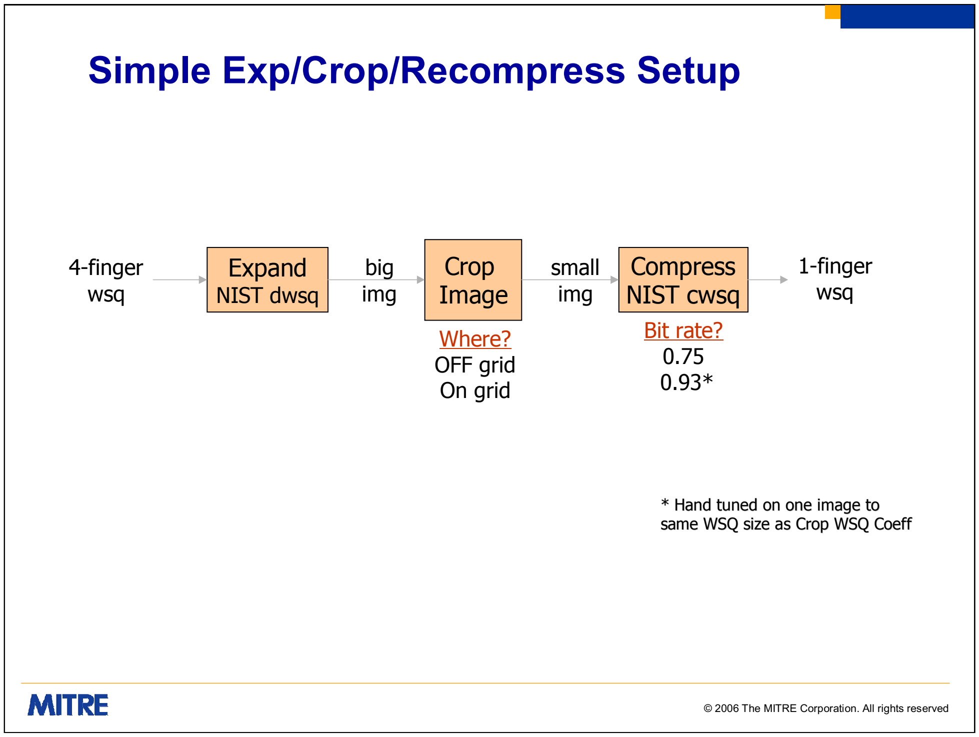
All three algorithms tested performance on the original ROI vs. the slightly larger window aligned with the 32x32 grid.

Since standard WSQ requires a compression bitrate specification, the simple expand/crop/recompress algorithm was tested using both the original 4-finger bitrate parameter (0.75) and using a larger value tuned to match the crop WSQ coefficients filesize.

Cropping the WSQ coefficients was originally performed without any coefficient editing. Later tests tried zeroing some coefficients but determined that this gave even worse performance. The discussion shown here will only apply to the cropping without coefficient editing.

The smart expand/crop/recompress testing was originally performed using the 0.44 bin center for expansion. A later test tried 0.5 for the expansion bin center but did not find any appreciable change in the results, so all the results shown here use the 0.44 bin center expansion.

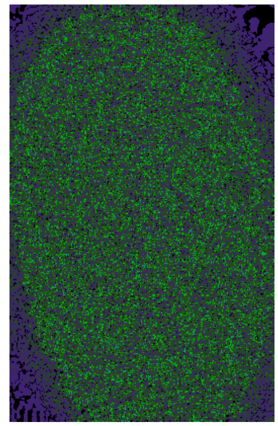
Each algorithm and results are shown in turn with specifics on processing appearing in each section.



Expand/Crop/Recompress sounds very straightforward but it turns out there are some choices along the way. The impacts of the exact placement of the crop box and the recompression bitrate parameter were studied. The WSQ Compressor and Decompressor code came from NIST Fingerprint Image Software (NFIS) v2, available on CD.

Results with ROI (off grid) crop

Simple 0.75 Recompression

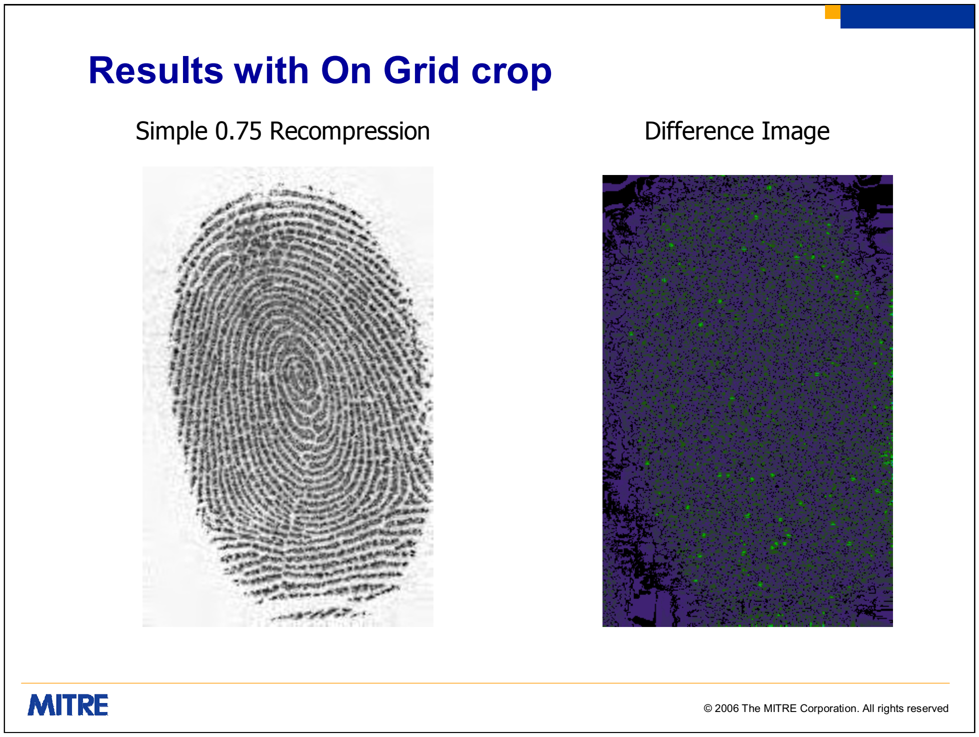


Difference Image

MITRE

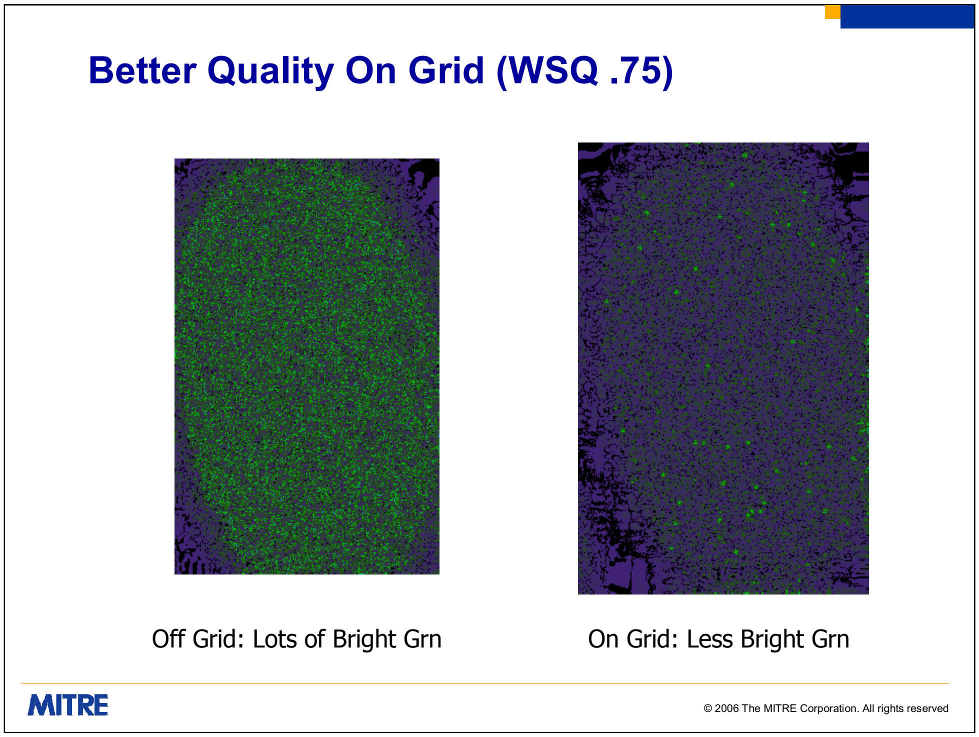
Here only the ROI was cropped from the expanded image and then recompressed using 0.75. Notice there's a lot of change that has a big enough difference it could be structural. Closer inspection of the images shows that while there is flickering and differences can be seen (pores getting brighter), there isn't major impact on structural elements.

(A view of reference 4-finger expansion in the vicinity of this print can be found on slide 33.)



Here the expanded image was cropped to the nearest outer grid boundary, and then recompressed. There are still some light blue locations, but the overall amount of bright green and blue has decreased.

Flicker of the recompression with the reference image is available in slideshow mode. (Powerpoint version of the document which enables flicker is available from author on request. See last page for contact information.)

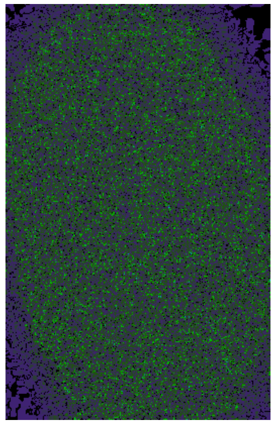


Here the off/on grid differences for bitrate 0.75 are shown side by side and it is apparent that the on grid results are better (fewer bright colors).

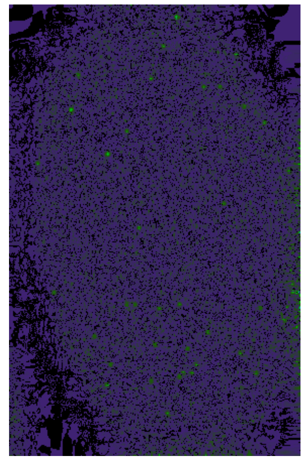
There is a good reason this occurs. The wavelet transform used in WSQ (and many other wavelet compression algorithms) is not translation invariant.[4] For each level of transform there an even position and an odd position, i.e. it is invariant only under shifts of size 2. When D levels of transform are combined shifts of 2^D are needed for invariance. WSQ uses 5 levels of wavelet transform so is only invariant when shifts of 32 are applied in each direction (row and column). Another way to think about this is to realize that each lowpass coefficient corresponds to a block of 32x32 pixels in the original image. In WSQ this relationship is fixed so that the first LL coefficient always matches the 32x32 area in the upper left corner of the image. If a cropped image aligns with the original blocks, the computations will be working with nearly the same data (modulo quantization effects). However if cropping forces the blocking to shift from its original layout, the computations will be working with very different values. The combination of quantization effects with a new computation gives more error than recomputing with just slightly modified values.

Some wavelet compression specifications (such as JPEG2000[5]) allow modification of the blocking offset, so that high quality cropping is possible at any location without reference to the original transform location. Since WSQ is not one of those formats, users should be pay attention to the 32x32 grid for highest quality cropping results.

Better Quality On Grid (WSQ .93)



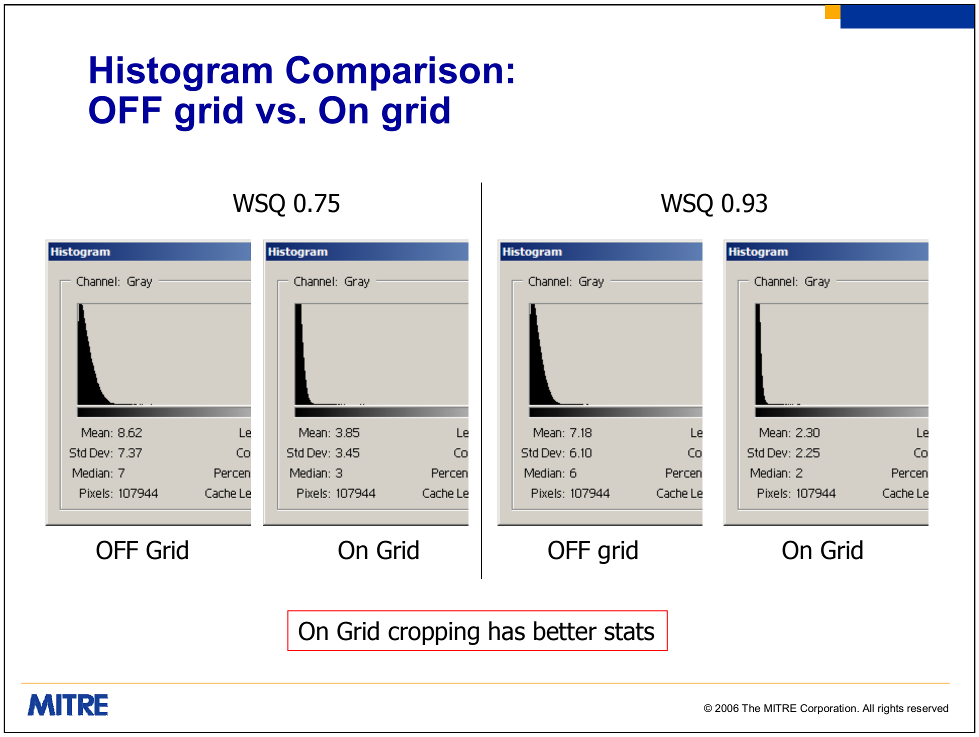
Off Grid: Lots of Bright Grn



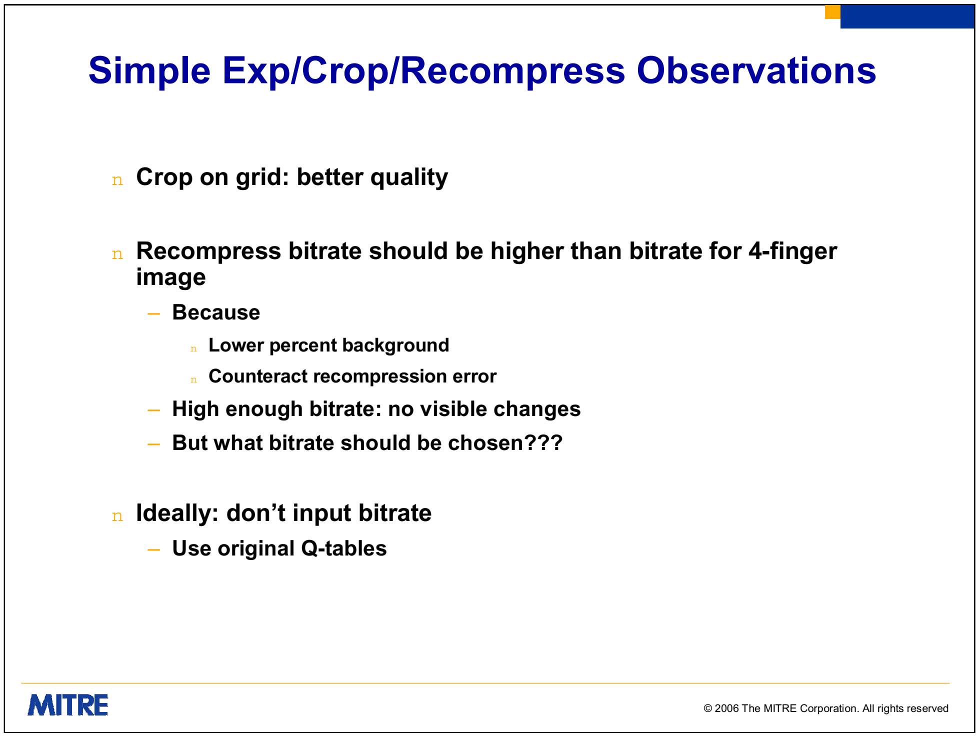
On Grid: Much less Green

MITRE

Here we show this wasn't just a fluke. Another compression parameter was used (0.93). This parameter made the on-grid expand/crop/recompress filesize match the Crop WSQ Coeff filesize on one of the livescan images. In both instances (compression parameter 0.75 and 0.93) the on-grid recompression looks dramatically better!

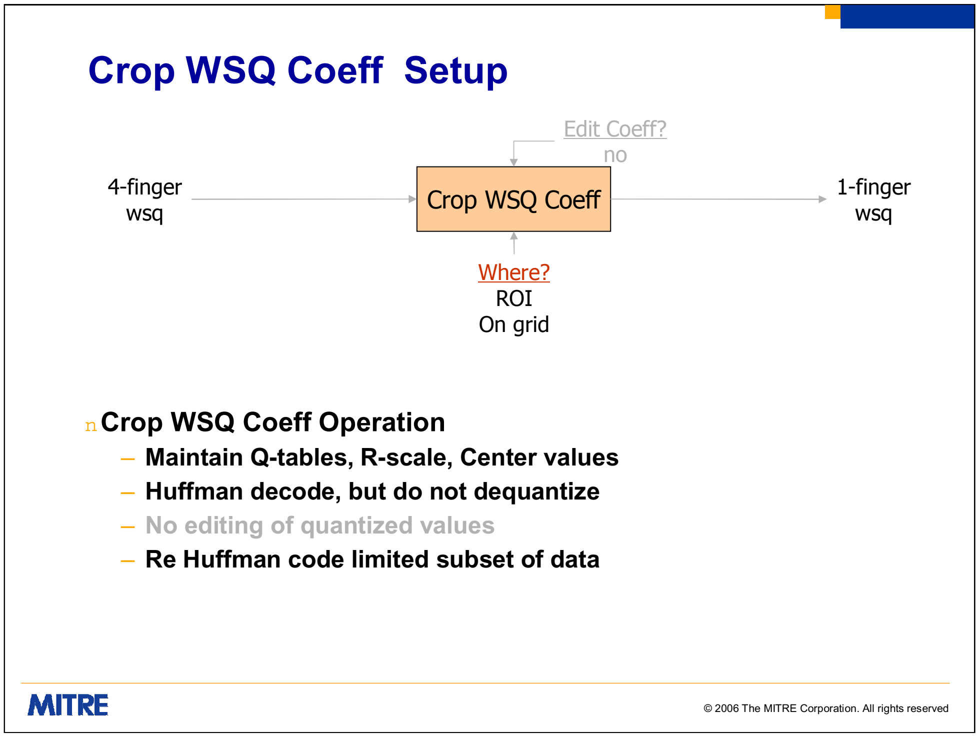


In addition to viewing the difference images, difference image statistics were computed. For the on grid data, the statistics were computed in the ROI area alone (shown here) for comparative purposes. However it should be noted that the statistics in the border area were not much different from those shown here. These stats confirmed that any simple exp/crop/recompress operation should occur at the 32-pixel block boundaries, no matter what compression bitrate is used.



In addition to seeing the sensitivity to location of the crop window, it also clear that the quality of the results are highly dependent on the bitrate that is chosen during recompression. Choosing the same bitrate as was used in the original compression definitely adds error to the image (both because the 4-finger image typically has a large percentage of background area while the single finger crop does not, and because recompression inherently tends to add error). But it is not obvious what bitrate should be chosen.

Ideally it would be easier if a bitrate didn't need to be entered at all and the quantization parameters in the 4-finger WSQ could be used to direct the crop recompression. This in observation is the basis of the smart expand/crop/recompress algorithm described in detail later.



In this case the cropping occurred entirely within one piece of code that did not entirely expand the image.

During Crop WSQ most of the file header is untouched (only width/height fields changed). Only the 3 blocks of Huffman coding are redone.

(This is cropping from the wavelet data in its most simplistic form. Other more complex operations that modify some wavelet coeff values are also possible and reviewed at the end of this section.)

NIST NFISv2 WSQ source code was used with slight alterations for this implementation

Expectations of Crop WSQ Coeff

- Upper Left corner on 32-pixel grid
- Central area will be lossless
- Lossy border size?
  - Up to 80? pixels on all sides
  - Lossy ≠ Visible change
    - See images later in brief for visuals

MITRE

Based upon the wavelet transform theory, there were certain expectations about how coefficient cropping would behave.

Due to the 32x32 shifts required for translation invariance (see explanation on page 13), it was anticipated that results would be very poor for Off grid cropping.

Due to the localization but slight spread of the filters, it was expected that some central area of the image would be identical to the reference, while some fairly large border area would be lossy. The exact size of the lossy border both numerically and visually for WSQ was in question.

Off Grid: Cropping WSQ Coefficients

- Cropped directly at ROI (not on grid)
- Resulting expansion is very poor due to misalignment errors
- ***Do NOT do this!!!***

MITRE

This is what happens when cropping WSQ coefficients using an UL corner off the 32x32 grid. By default the code snaps to closest (outer) grid location to avoid this behavior! The code was forced to do this only to show what a bad idea it is.

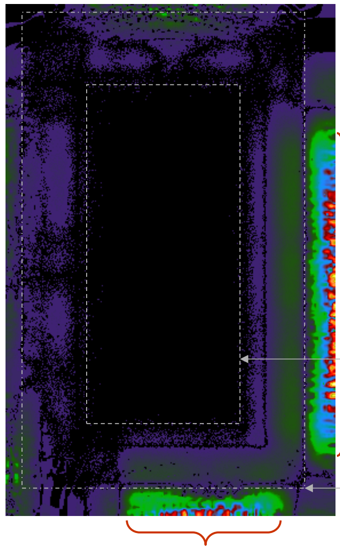
Crop WSQ Coefficients On Grid

- **Nearest Grid box**
- **Grey smudge**
  - **at right edge**
  - **at bottom**
- **What else?**
  - **flicker**

MITRE

This time a crop box aligned with the grid was used, with much better results. However near the border there are some obvious smudges that are not present in the reference image. Flickering shows other differences, all near the border of the image.

Crop WSQ Coeff Differences



<u>Crop WSQ – Full Expand</u>

- Black = identical
- Purple = not visible
- Dark grn = zoom flicker only, no structure change
- Bright grn = visible w/o flicker
- Bright blue/red/etc = visible, may be structural change

Watch out for structure loss

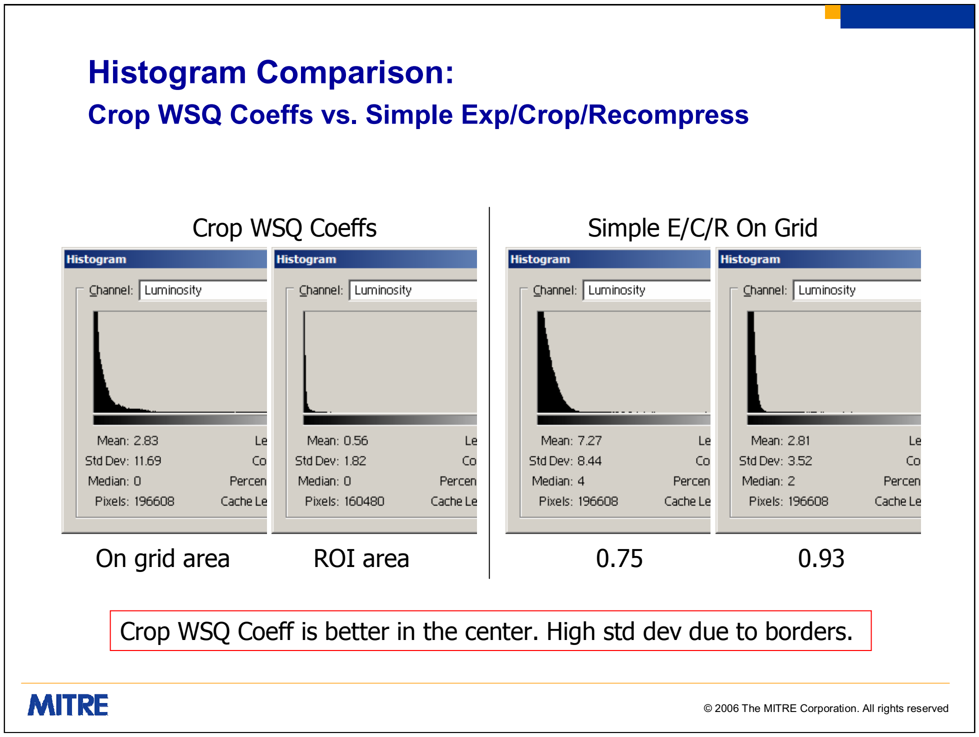
Perfectly lossless inside 93 pixel border

Effective visible loss border

MITRE

The dotted rectangles are not part of the difference image. They were added for illustrative purposes.

As expected the errors are restricted to a boundary region. Turns out 1DN differences can creep as much as 93 pixels in from the edge (on 7 livescan test images). But the largest differences stay much closer to the edge. Flickering showed that the brighter colors highlight visible differences (if you know where to look). And streaky / dotted higher end colors often indicate structural changes.   The actual location of structural change (determined by visual examination) lies about 16-20 pixels in from the border in livescan images.



Here the difference image histograms when cropping on grid are shown. These histograms come from the livescan processing rather than using the inked print. Two of them for crop WSQ coeffs: the first across the entire image including borders, the second only in the original ROI. This highlights that the stats at the border are significantly different than the center. These histograms are then compared to the histograms from the simple exp/crop/recompression (at two different bitrates). (Only one histogram across the entire image shown, since the ROI and border performance were quite similar.) These histograms show that recompressing at the same bitrate is not as good as cropping the WSQ coefficients directly, but recompressing at a higher bitrate can be quite effective [if you know what bitrate to choose].

Crop WSQ Coeff Observations

- Based upon 7 livescan trials
  - Max intrusion distance from image edge:

DiffDN	1	2	3	20	21	35	56+
#Pix	93	73	60	39	26	22	20

  - Ridges very near edge change structure
    - Alter direction
    - Fade out / Darken
    - Structure change seen at max 20 from edge
      - Typically more like 16 or less from edge
      - (≤ 0.04 inches from border for 500 dpi)

MITRE

This info was gleaned from the aggregate across the 7 livescan test images. First is a measure of how far into the image various grayscale differences extend. A DN difference of 1 can be up to 93 pixels toward the interior, but a DN difference of 2 will not extend that far in (furthest seen was 73 in). Etc. The changes causing structural differences weren't directly linked to a specific difference value, but we can see that the larger value differences are restricted to an area that is ~20 pixels from the border. For the inked print shown in this report the DN 1 differences extend further towards the center of the image, but the structural differences still remain limited to a small border region.

Crop WSQ Coeff Observations (cntd)

- **Areas not impacted**
  - **Crop through wide white area**
    - **'Enough' white on both sides of crop line**
- **Visible changes when:**
  - **Small white area between print and crop line**
  - **Crop line immediately between print and background**
  - **Ridges which cross crop line**
- **Often visible changes are not structural**
  - **Smudging background**

The changes in border areas occur because the wavelet transform no longer has all the original data used in its computation (due to spread) and is forced to use the symmetric extension rules to fill in for the missing values. When the nearby area that is cropped away is similar to the mirrored data then the cropping impact will be low. But often it is not (background vs. fingerprint), which can cause significant change (smudges, shifting angles on ridges, etc).

Zoomed in Comparison

- **Near Boundary**

Reference | Crop WSQ Coeff | Simple Expand/Crop/Recompress 0.75 | 0.93

MITRE

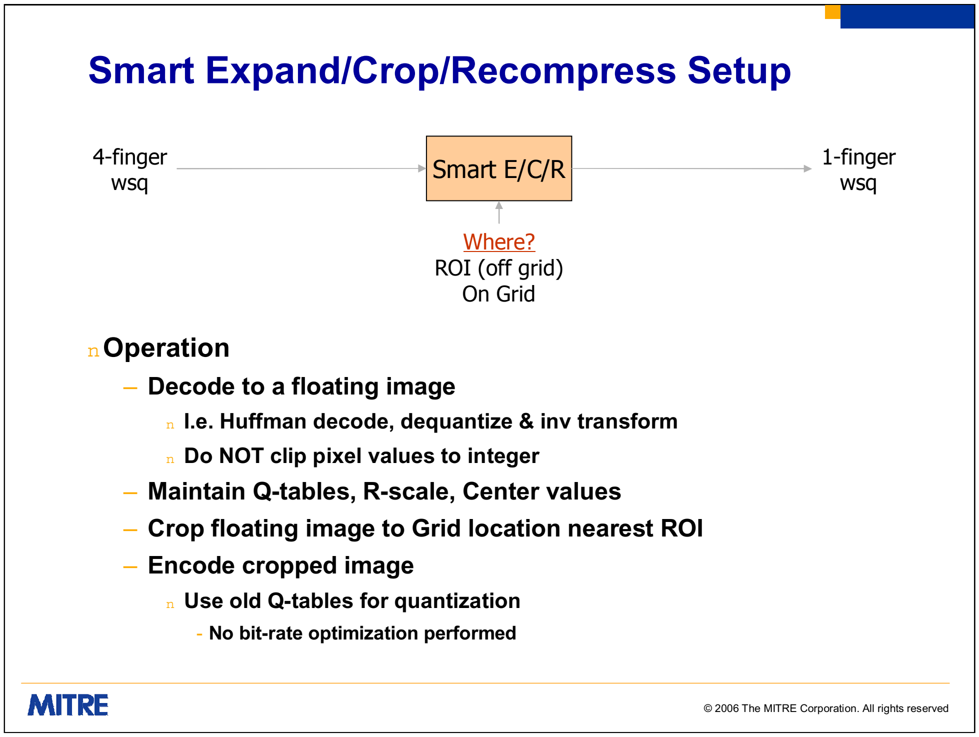
Here's a high res view of a border area containing ridges from one of the livescan test images. This gives a detailed view of the structural changes that Crop WSQ can cause near the border. When viewed in slideshow mode, the various result images can flicker with the reference. Backspace goes back to the reference and space or mouse click moves forward to the result images.

Other Thoughts

- **Zero Out Coefficients not used to reconstruct ROI**
  - **Attempting to blur the border area**
  - **Doesn't work!!!**
    - **Areas near ridges don't blur, but instead get a ridges on top of ridges pattern**
    - **These falsely created patterns could be confused with ridges**
- **Suspected Avenue for Improvements**
  - **Recompute border coefficients using mirrored extension**
    - **Same end effect as Smart E/C/R**

Only one other trial was attempted with crop WSQ coefficients. In this case all coefficients that did not contribute to computation of the ROI image pixels were reset to zero rather than maintaining their original values. The goal being a blurring effect. Unfortunately that was not the result, and other odd and disturbing effects appeared in the border area of the image. Therefore the idea of simple coefficient editing was abandoned.

More complex editing, where the coefficients in the entire border area are recomputed using the reflection are likely to work, but are essentially the same as the smart expand/crop/recompress method described in the next section.

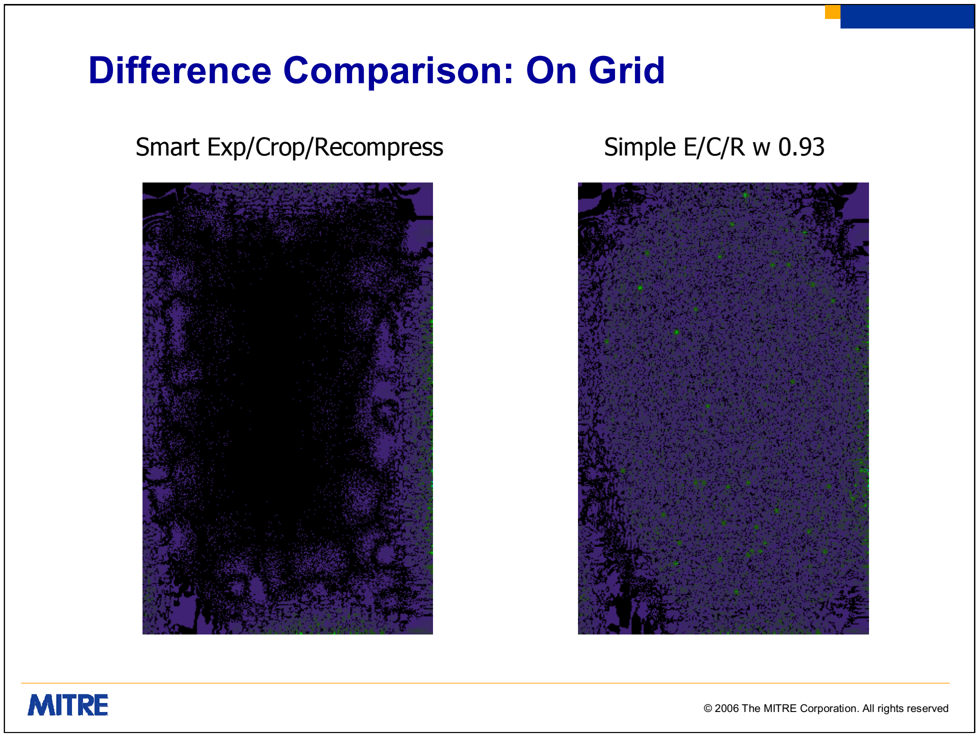


Although the image will be expanded, cropped, and recompressed, it is done within a single piece of code which can read and retain the original Q-tables. Since the expansion/crop/recompression is done internally, the intermediate 'reconstructed' image is stored with floating precision and never clipped to unsigned char. This removes one potential source of loss that was present in all previous expand/recompress tests.

During smart expand/crop/recompress most of the file header is untouched (only width/height fields changed) to maintain the integrity of the quantization parameters. Otherwise, only the 3 blocks of Huffman coding are redone in the file itself. In this sense it is very similar to cropping WSQ coefficients. However, internally smart exp/crop/recompress is doing more processing than cropping WSQ coefficients, and will run slower. Some special techniques can be used to reduce the amount of processing, but it will always be somewhat related to at least the cropped image size.

This smarter method removes any need to choose the compression bitrate, since it is set automatically.

Once again the NIST NFISv2 WSQ source code was used with slight alterations for this implementation.

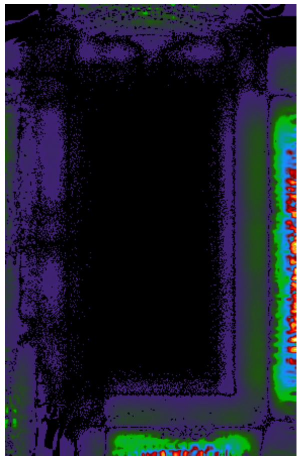


Here's a comparison of the differences when using smart expand/crop/recompress vs. the best simple expand/crop/recompress using an on-grid crop location.   Smart expand/crop/recompress is obviously better, particularly in the image interior.

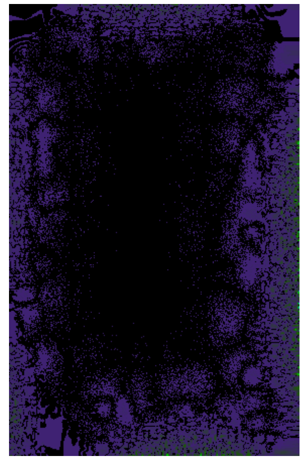
Although the results are not shown here, smart expand/crop/recompress was tried using the original off grid ROI for the crop area. This gave results similar to the simple expand/crop/recompress when used on the ROI … ie not so good, but not terrible.

Difference Comparison: On Grid

Crop WSQ Coeff

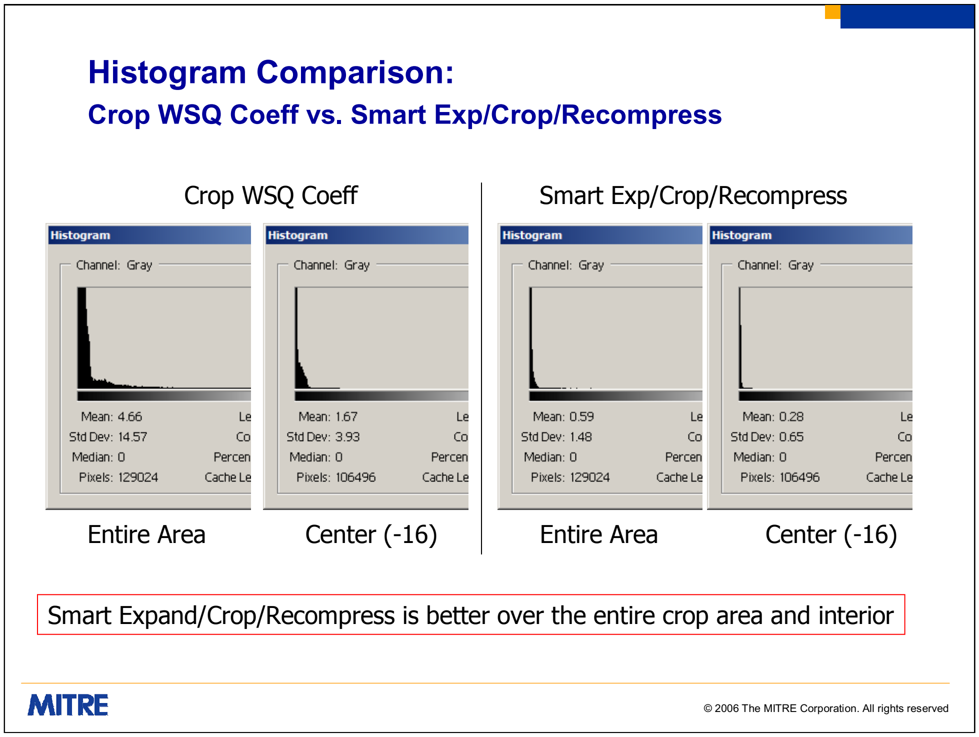


Smart Exp/Crop/Recompress



MITRE

Here the crop WSQ coefficient differences are compared to smart expand/crop/recompress. Although there are a few non-zero differences ranging a little further into the image interior, the border impacts are much reduced with smart expand/crop/recompress.



Here are histograms for the difference images on the previous slide. They confirm the visual perception from the color coded difference images. Smart expand/crop/recompress is closer to the reference expansion (cropped area of the expanded 4-finger image): both over the entire crop area (snapped to grid boundaries) and also 16 pixels into the center.

Observations

- Based upon 7 livescan trials
  - Max intrusion distance from image edge:

Difference DN	1	2	3	4	20	21	35	45	56+
Pix distance into interior									
Crop WSQ Coeff	93	73	60	59	39	26	22	21	20
Smart E/C/R	147	80	70	51	21	20	10	4	3

  - Max Difference values
    - Crop WSQ Coeff frequently 255
    - Smart E/C/R often about 70
  - Conclusion re Smart Exp/Crop/Recompress
    - Very small differences (1-3) appear a little further into image than with Crop WSQ Coeff
    - Larger differences stay closer to the edge
      - or don't appear at all

MITRE

Here we take a closer look at how far the non-zero differences intrude into the interior of the image. The data was collected from 7 livescan crops, not from the sample image shown in this report. A difference of 1 gray level was seen as far as 93 pixels in for crop WSQ coeff and 147 pixels in for smart expand/crop/recompress. However, once the gray level difference was 4 or more, the smart expand/crop/recompress version didn't intrude as far as when cropping WSQ coefficients. Larger differences were restricted to very close to the edge (4 pixels) in smart expand/crop/recompress vs. ~20 for cropping WSQ coefficients. And the 'larger' differences near the border were considerably smaller in magnitude in smart expand/crop/recompress.

The placement of the wavelet coefficients that were changed compared to the original values (used when cropping WSQ coefficients) was tracked as well. Typically changes occurred more often in the lower frequency subbands, and all changes were limited to the borders of the individual subbands. On the 7 images tested, the changed coefficients were always either right on the edge of the subband, or one coefficient in.

Observations Continued

- Compression Ratios Achieved
  - Depend upon original compression and % background
  - Smart E/C/R & Crop WSQ Coeff generate files of very similar in size
    - Smart E/C/R files are just a touch larger

Original Ratio	26	21	18	19	20	19	21
Crop WSQ Coeff	12	11	10	10	10	12	9
Smart E/C/R	12	11	10	10	10	12	9

MITRE

Given the improved performance of smart expand/crop/recompress, it is natural to wonder what compression ratio is being achieved, especially relative to other techniques. The file sizes achieved by cropping WSQ coefficients and smart expand/crop/recompress are very similar with smart expand/crop/recompress being ever so slightly larger. The simple expand/crop/recompress sizes with parameter 0.93 were also similar, so the perceived quality differences are due to a differences in methodology rather than changes in compression ratio.

Notice that the achieved compression ratio is quite different between the original 4-finger file and the cropped 1-finger file. This occurs because due to the greatly reduced amount of background area in the cropped image. Consistent background is very efficiently compressed using wavelets, so higher compression ratios appear. In addition, the quantization parameters are originally set using statistics with a high percentage of background. When quantized and encoded, the background areas will consume only a small percentage of the 4-finger file, and the fingerprint data itself will use the bulk of the space available. Once the background is cropped out, the true compression ratio in the fingerprint areas is revealed.

Result Image Comparison

Crop WSQ Coeff

Smart Exp/Crop/Recompress

MITRE

Here are the result images from both cropping WSQ coefficients and smart expand/crop/recompress. Smart expand/crop/recompress is much cleaner in the border background area, and doesn’t bend ridges near the edge.

Result Image Comparison

Reference Image

Smart Exp/Crop/Recompress

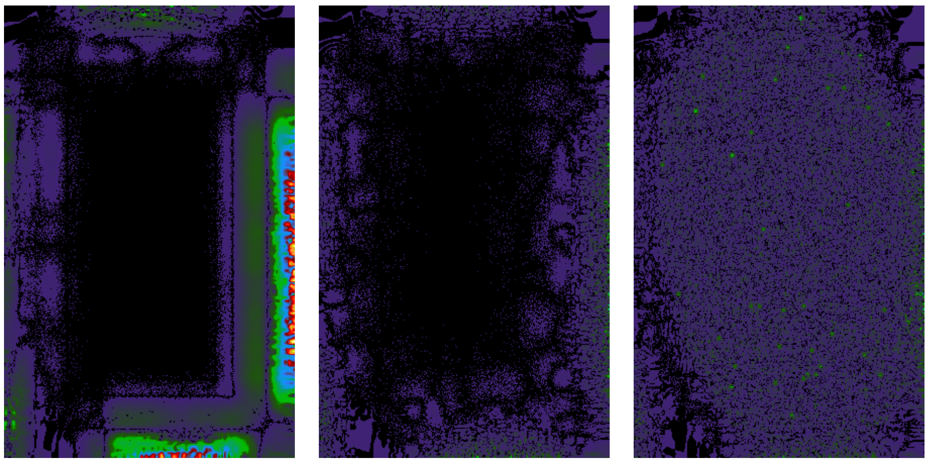
MITRE

Here the reference image cropped directly from the expanded 4-finger WSQ is shown side-by-side with smart expand/crop/recompress. Differences are not extremely hard to see.

If you flip back and forth between this slide and the previous one on a display flicker differences will draw attention to cropping WSQ coefficient changes near the border.

Difference Images for All 3 Methods: On Grid

Crop WSQ Coeff | Smart E/C/R | Simple E/C/R 0.93

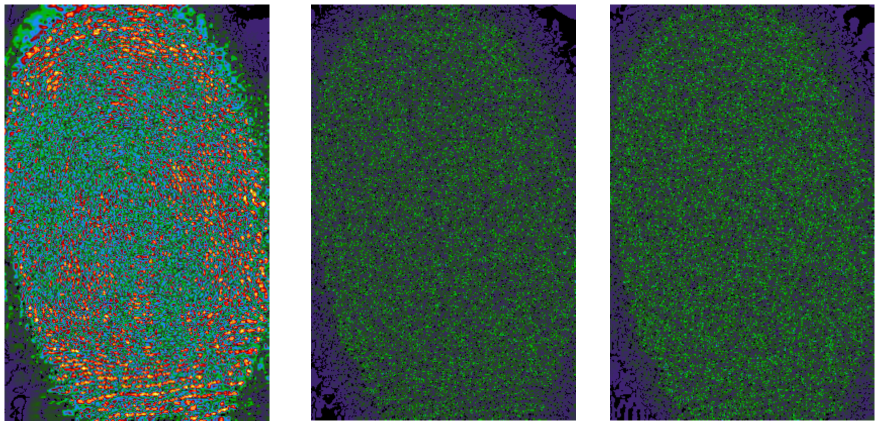


MITRE

This is a side-by-side comparison of the best difference images from each of the methods. The cropped area is aligned with the 32x32 grid. The dark center indicating no change in cropping WSQ coefficients and smart expand/crop/recompress is highly desirable. The brightly colored border areas when cropping WSQ coefficients aren't so good.

Difference Images for All 3 Methods: ROI only (off grid)

Crop WSQ Coeff | Smart E/C/R | Simple E/C/R 0.93



This is a side-by-side comparison of the highly discouraged Off grid cropping (originally requested ROI) difference images from each of the methods. The crop WSQ coefficients method must not used in these conditions, since it produces an image where the ridges are highly distorted. If a crop with UL corner off the 32x32 grid is absolutely required, then the simple expand/crop/recompress (with high bitrate) gives similar quality to smart expand/crop/recompress, though the smart version still has a very slight edge.

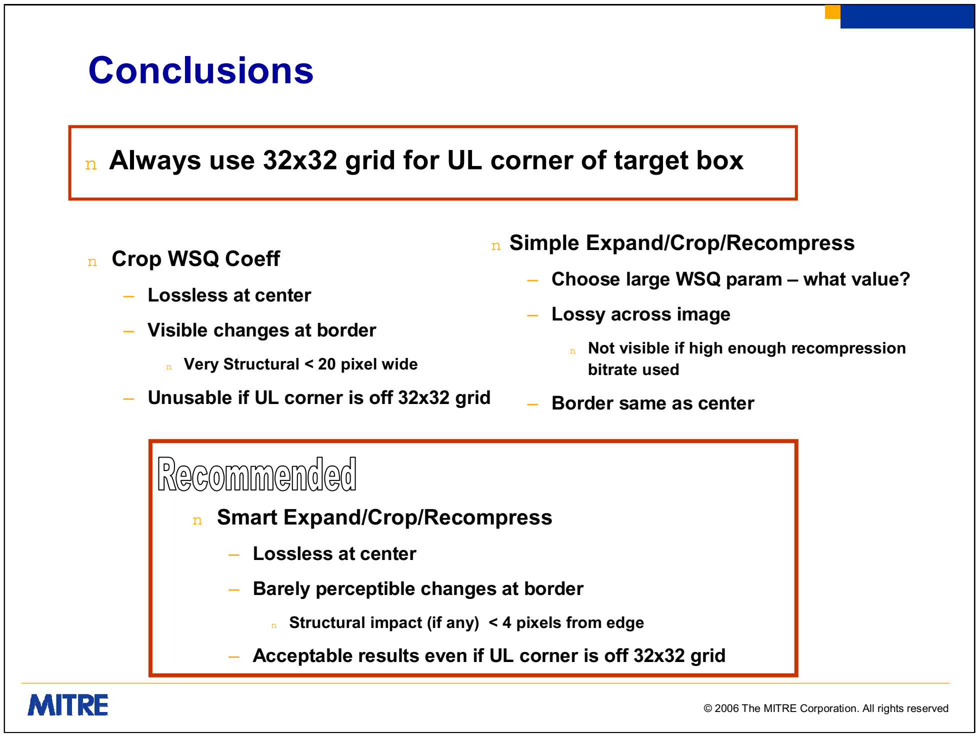
Zoomed in Comparison: On Grid Crop

- **Near Boundary**

Reference | Crop WSQ Coeff | Smart E/C/R | Simple E/C/R 0.93

MITRE

Here's a high res view of a border area containing ridges when using an on grid crop area. This is a segment of one of the test livescan images. Flickering (available in slideshow mode) shows that cropping WSQ coefficients alters ridge direction near the edge, while smart and simple expand/crop/recompress make very small changes in the border areas (~4 pixel border area has visible flicker).



A variety of things have been learned about the different cropping methods. But the ultimate best choice in terms of accuracy and ease of use is smart expand/crop/recompress, making sure to snap at least the UL crop corner to the 32x32 grid.

Extrapolating these results to other wavelet based compression formats:

a. 32x32 grid placement of the Upper Left crop corner is necessary if the format does not allow transform offsets.

b. For best quality results, coefficients in the neighborhood of the border should be recomputed using the appropriate edge extension. Although such changes will cause a few lossy changes closer to the image interior, they are needed if the border area is to remain fairly true to the original.

c. Quantization settings should be maintained as much as possible. Any quantization changes should not reduce accuracy (larger step size). This is most reliably achieved at least extra expansion by using the original quantization parameters.

It is recognized that in most instances users will still tend to use the simple expand/crop/recompress since those tools are easily accessible. It should be noted that the 0.93 bitrate that worked reasonably well in the examples shown here was manually hand tuned for that image. No inferences should be drawn about its appropriateness for general use.

References

1. IAFIS-IC-0010(V3), IAFIS Wavelet Scalar Quantization (WSQ) Grayscale Fingerprint Image Compression Specification, 1997

2. ANSI/NIST-CSL 1-1993, "Data Format For The Interchange of Fingerprint, Facial and Scar/Mark/Tattoo Information"

3. NIST Fingerprint Image Software 2 – WSQ code
   - http://fingerprint.nist.gov/NFIS/index.html

4. Uytterhoeven et al, WAILI: A software library for image processing using integer wavelet transforms. *Image Processing*, v 3338 of *SPIE Proceedings*

5. ISO IS 15444-1: JPEG2000 Image Coding System

MITRE

Implementation

- Code for Smart E/C/R
  - Mod/addition to NIST WSQ code
    - Source available on request from mlepley@mitre.org
  - Compilation requires NIST Fingerprint Image Software 2

MITRE

The small modifications made to the NIST WSQ code to allow it to perform the Smart expand/crop/recompress function are available by request. This code is directly derived from NIST Fingerprint Image Software 2 (NFIS2) and will not compile without having the larger package available. See reference 3 on the previous page for information on obtaining the NFIS2 package.