

# ISO/IEC 29794-4 Implementation

## *NIST Fingerprint Image Quality (NFIQ) 2*

### *Updates and Weaknesses*

Greg Fiumara

greg@nist.gov | nfiq2@nist.gov

12 January 2022

ISO/IEC JTC 1/SC 37/WG 3

# New Since June 2021

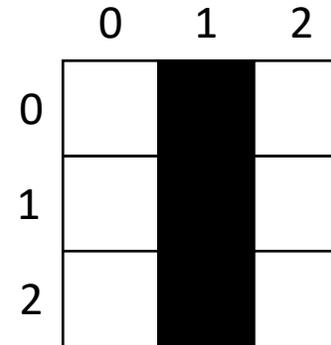
- Version 2.2
  - Pre-processing bug fixes
  - Other improvements
  - Updated conformance test set
- NIST IR 8382
- Signed Windows, macOS binaries
- Raspberry Pi OS



# Implementation Trim Off-By-One Bug

## ISO/IEC 29794-4:2017

- b) For each column  $C_i$  in  $I$ , starting from the left
  - 1) Compute the column arithmetic mean  $\mu_{col}$
  - 2) On the first occurrence where  $\mu_{col} \leq T_\mu$  set  $idx_l = i$
  - 3) On the last occurrence where  $\mu_{col} \leq T_\mu$  set  $idx_r = i$



left = 1  
 $1 \neq 0 \therefore -1 \rightarrow \text{left} = 0$   
right = 1  
 $1 \neq \text{orig.width} \therefore +1 \rightarrow \text{right} = 2$   
width = right - left + 1 = 3

## NFIQ 2.0.0 through 2.1

```
// start from left of image and find left index that is already part of
// the fingerprint image
int leftIndex = 0;
for (int j = 0; j < img.cols; j++) {
    double mu = computeMuFromColumn(j, img);
    if (mu <= MU_THRESHOLD) {
        // Mu is not > threshold anymore -> left index found
        if (j == 0) {
            leftIndex = j;
        } else {
            leftIndex = (j - 1);
        }
        break;
    }
}
...repeat on other sides...

// now crop image according to detected border indices
int width = rightIndex - leftIndex + 1;
```

## NFIQ 2.2

```
if (mu <= MU_THRESHOLD) {
    // Mu is not > threshold anymore -> left index found
    if (j == 0) {
        leftIndex = j;
    } else {
        leftIndex = (j - 1);
    }
    leftIndex = j;
    break;
}
```

left = 1  
right = 1  
width = right - left + 1 = 1

# Implementation No Trim Bug

## ISO/IEC 29794-4:2017

- b) For each column  $C_i$  in  $I$ , starting from the left
  - 1) Compute the column arithmetic mean  $\mu_{col}$
  - 2) On the first occurrence where  $\mu_{col} \leq T_\mu$  set  $idx_l = i$
  - 3) On the last occurrence where  $\mu_{col} \leq T_\mu$  set  $idx_r = i$

```
// start from left of image and find left index that is already part of
// the fingerprint image
int leftIndex { 0 }, rightIndex { img.cols - 1 };
for (; leftIndex < img.cols; ++leftIndex) {
    if (computeMuFromColumn(leftIndex, img) <= MU_THRESHOLD) {
        break;
    }
}

// If we traversed all the columns, then we don't need to check starting
// from the other side.
if (leftIndex >= img.cols) {
    // If we traversed all columns and never found data, we can stop
    throw NFIQ2::Exception { NFIQ2::ErrorCode::InvalidImageSize,
        "All image columns appear to be blank" };
} else {
    // start from right of image and find right index that is
    // already part of the fingerprint image
    for (; rightIndex >= leftIndex; --rightIndex) {
```

Bonus! Don't redo work

	0	1	2
0			
1			
2			

left = 0  
right = width - 1 = 2  
w = right - left + 1 = 3

left = width ∴ invalid

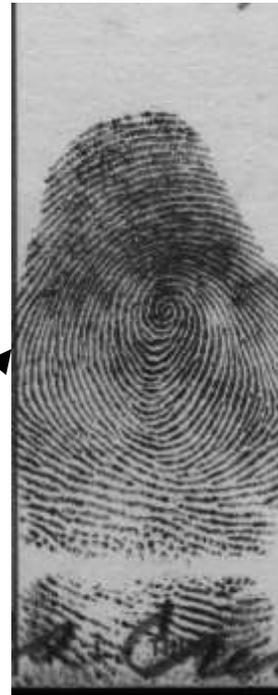
# Standard Weakness?: Off-White Constant

## ISO/IEC 29794-4:2017

- a) For each row  $R_i$  in  $I$ , starting from the top
- 1) Compute the row arithmetic mean  $\mu_{\text{row}}$
  - 2) On the first occurrence where  $\mu_{\text{row}} \leq T_\mu$  set  $\text{idx}_t = i$
  - 3) On the last occurrence where  $\mu_{\text{row}} \leq T_\mu$  set  $\text{idx}_b = i$

$$\mu \cong 220-245$$

$$T_\mu = 250$$



NFIQ 2.2: 36



NFIQ 2.2: 33



NFIQ 2.2: 39



NFIQ 2.2: 53

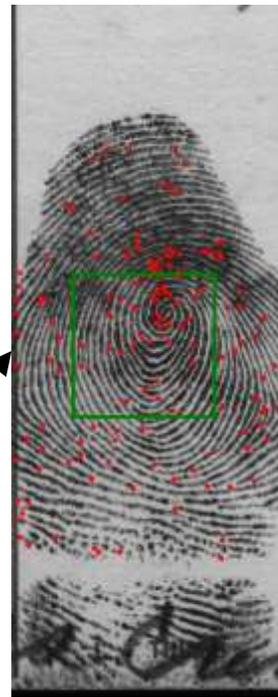
(crop likely not reasonable to expect from quality algorithm)

# Implementation Weakness?: False Minutiae **NIST**

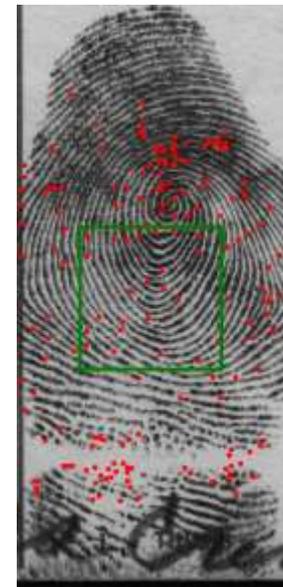
ISO/IEC 29794-4:2017

- a) For each row  $R_i$  in  $I$ , starting from the top
- 1) Compute the row arithmetic mean  $\mu_{\text{row}}$
  - 2) On the first occurrence where  $\mu_{\text{row}} \leq T_\mu$  set  $\text{idx}_t = i$
  - 3) On the last occurrence where  $\mu_{\text{row}} \leq T_\mu$  set  $\text{idx}_b = i$

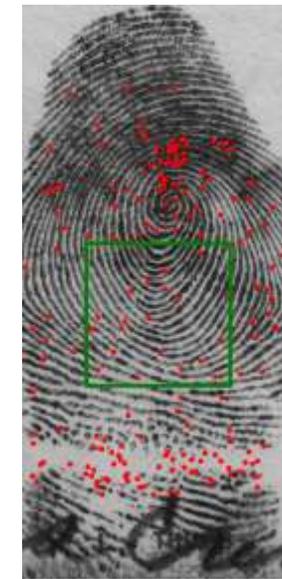
$$\mu \cong 220-245$$
$$T_\mu = 250$$



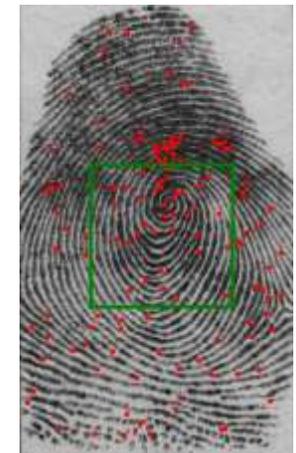
NFIQ 2.2: 36  
 $Q_{\text{MIN}}^{\text{com}}$ : 40



NFIQ 2.2: 33  
 $Q_{\text{MIN}}^{\text{com}}$ : 25



NFIQ 2.2: 39  
 $Q_{\text{MIN}}^{\text{com}}$ : 22

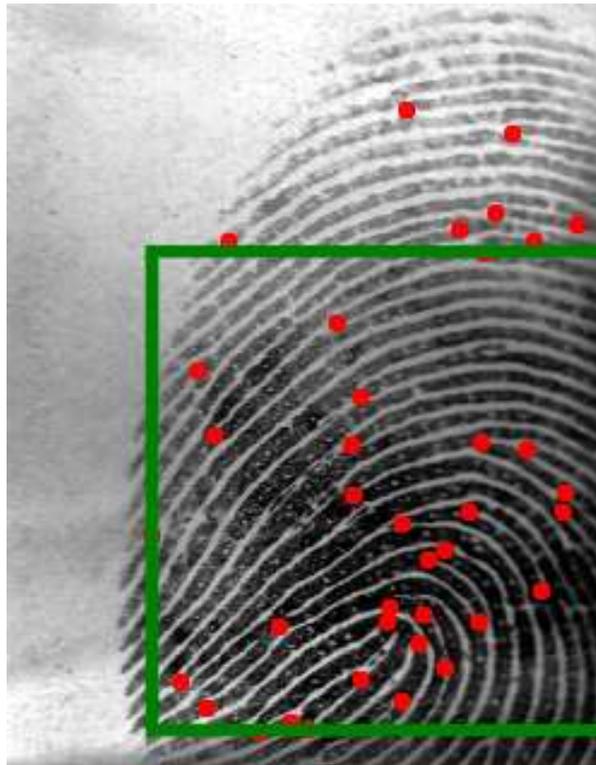


NFIQ 2.2: 53  
 $Q_{\text{MIN}}^{\text{com}}$ : 38

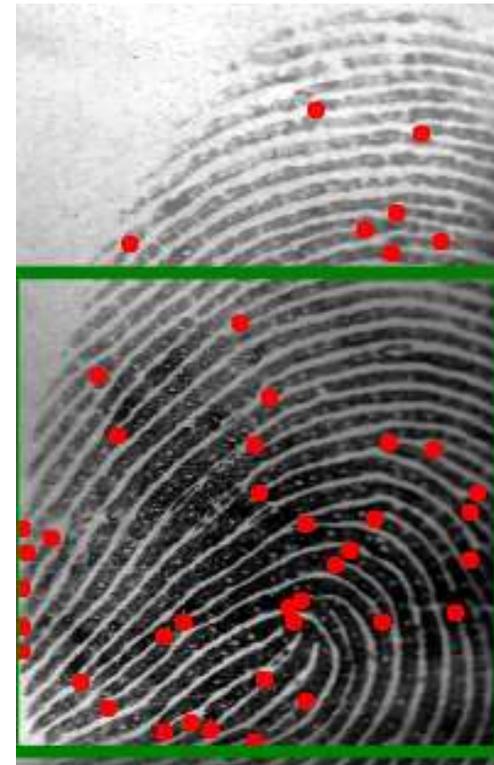
(crop likely not reasonable to expect from quality algorithm)

# Implementation Weakness?: False Minutiae **NIST**

Dirty optical platen



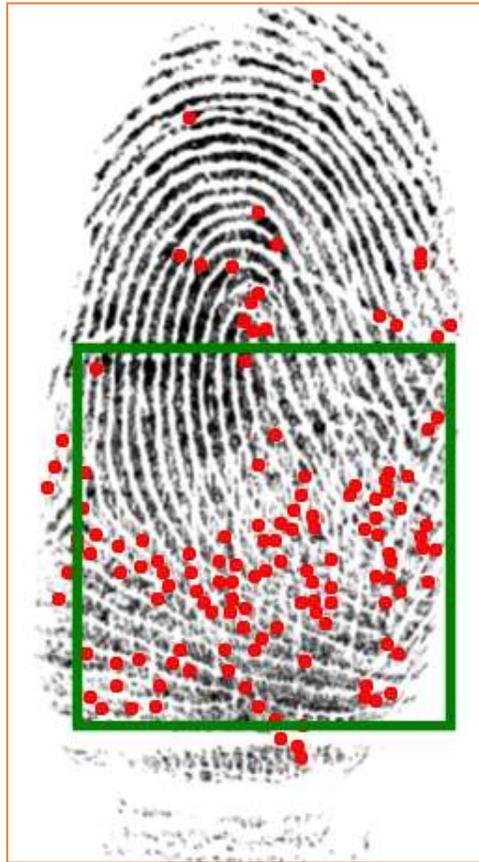
NFIQ 2.2: 35  
 $Q_{MIN}^{com}$ : 30



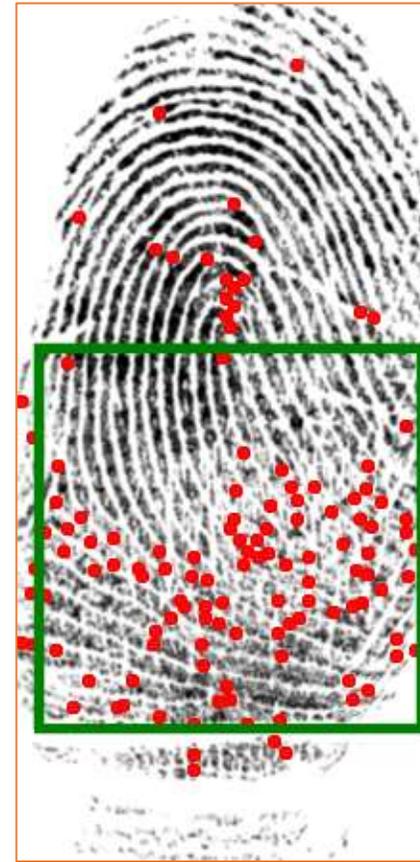
NFIQ 2.2: 39  
 $Q_{MIN}^{com}$ : 36

# Implementation Weakness?: False Minutiae **NIST**

## NFIQ 2 Pre-processing Crop

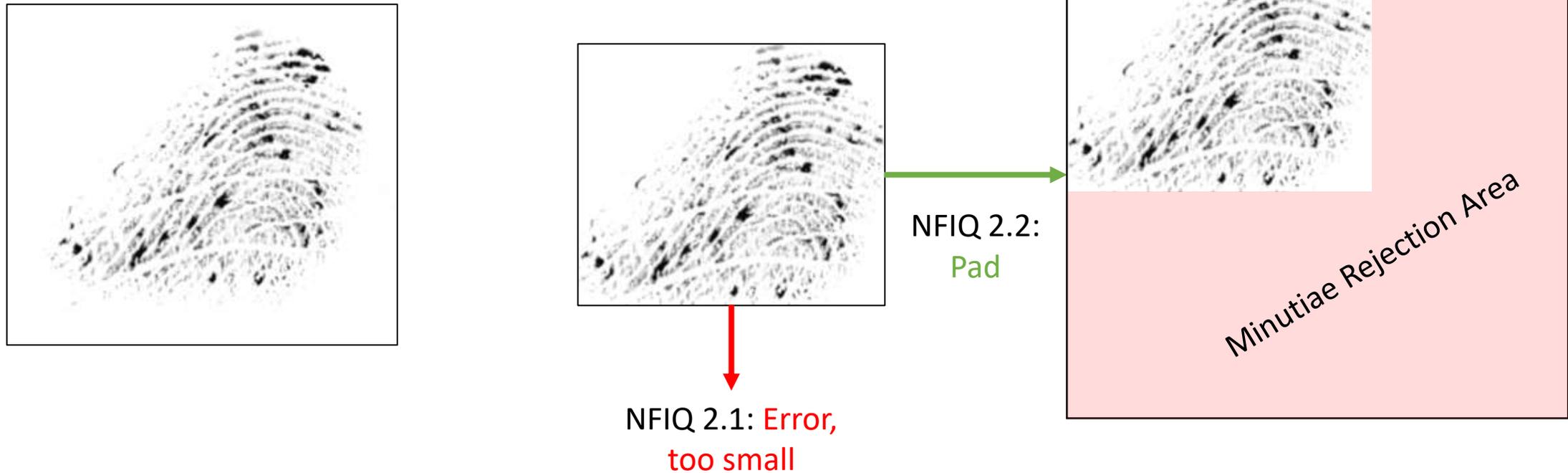


Disabled



Enabled

# Removal of Minimum Image Size



Minimum dimensions in FingerJet anticipate whitespace.

## 1 Scope

This document establishes

- terms and definitions for quantifying finger image quality,
- methods used to quantify the quality of finger images, and
- standardized encoding of finger image quality,

for finger images at 196,85 px/cm spatial sampling rate scanned or captured using optical sensors with capture dimension (width, height) of at least 1,27 cm × 1,651 cm.

(250x325 pixels)

**Not enforceable.**

Platen size cannot be inferred from image alone.

# New Conformance Images



- Restores as many  $Q_{29794-4} = [0-100]$  as possible.
- Adds sanity check images.

# What's Next?

1. (More) Build cleanup
2. Android, iOS native (top 5 request)
- ...
- ∞ Alternate impression types/sensors (top 5 request)
  - Data + algorithms are required.

Hello there! You can help! *Yes, you!*

<https://github.com/usnistgov/nfiq2/issues>

[nfiq2@nist.gov](mailto:nfiq2@nist.gov)