NIJ project

Data extraction log (ver. 2)

# Summary

After acquisition in PiAnoS, data needed to be extracted for further analysis. Some metrics were designed to capture the information to be used in statistical analyses, and are described in this document. The (automated) overall process to produce the cooked data is also described.

# Process

Data was dumped from the PiAnoS 4 SQL database on the 10.09.2012  
(pianos4-nij-2012-09-10-090001-official.backup)

Analysis happens on a separate dataset than the live PiAnoS – frozen at the said date. Scripts (Python) were written to:

* Extract the data and create pre-processed intermediate files (.pickle, Python structures, same in essence as .m files for Matlab)
* Calculate the metrics defined below
* Create visual representations of the calculations

The process is controlled by shell scripts. Care has been taken so that the overall process (extraction, analysis, .csv files generation) can be restarted should “compile-time” parameters change.

# Output files

The output files (.csv) contain all metrics in different setups and conditions. Each user produces 3 rows: the mark during analysis, the mark during comparison and the print during comparison. These can be filtered if needed (e.g. comparison\_mark, see below).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Exercise name | User ID | Username | Status | Situation | Metric |  |
|  |  |  |  |  | Condition 1 | Condition 2 |
| Trial 01 | 517 | userXXX | 4 | analysis\_mark |  |  |
| Trial 01 | 517 | userXXX | 4 | comparison\_mark |  |  |
| Trial 01 | 517 | userXXX | 4 | comparison\_print |  |  |
| Trial 01 | 622 | userYYY | 2 | analysis\_mark |  |  |
| … | … |  |  | … | … | … |

Not all metrics are available in all situations: e.g. metrics that depend on the quality are not available on the print, the user has no data, etc. In this case the content is ‘NA’.

**Note**: the quality consensus is calculated on the mark at the analysis stage. Therefore metrics that depend on the quality consensus in ‘comparison\_mark’ should be used with caution.

# Consensus

Many metrics rely on the concept of consensus. They have been designed to correspond to an “intuitive” notion of consensus, but clearly there are assumptions here that must be kept in mind.

## Quality consensus

Each user has drawn quality zones during the analysis. The consensus is a 3-dimensional map that contains, for each pixel (x,y) and for each quality level (plus the ‘nothing’ level) the number of users that covered it with a zone.

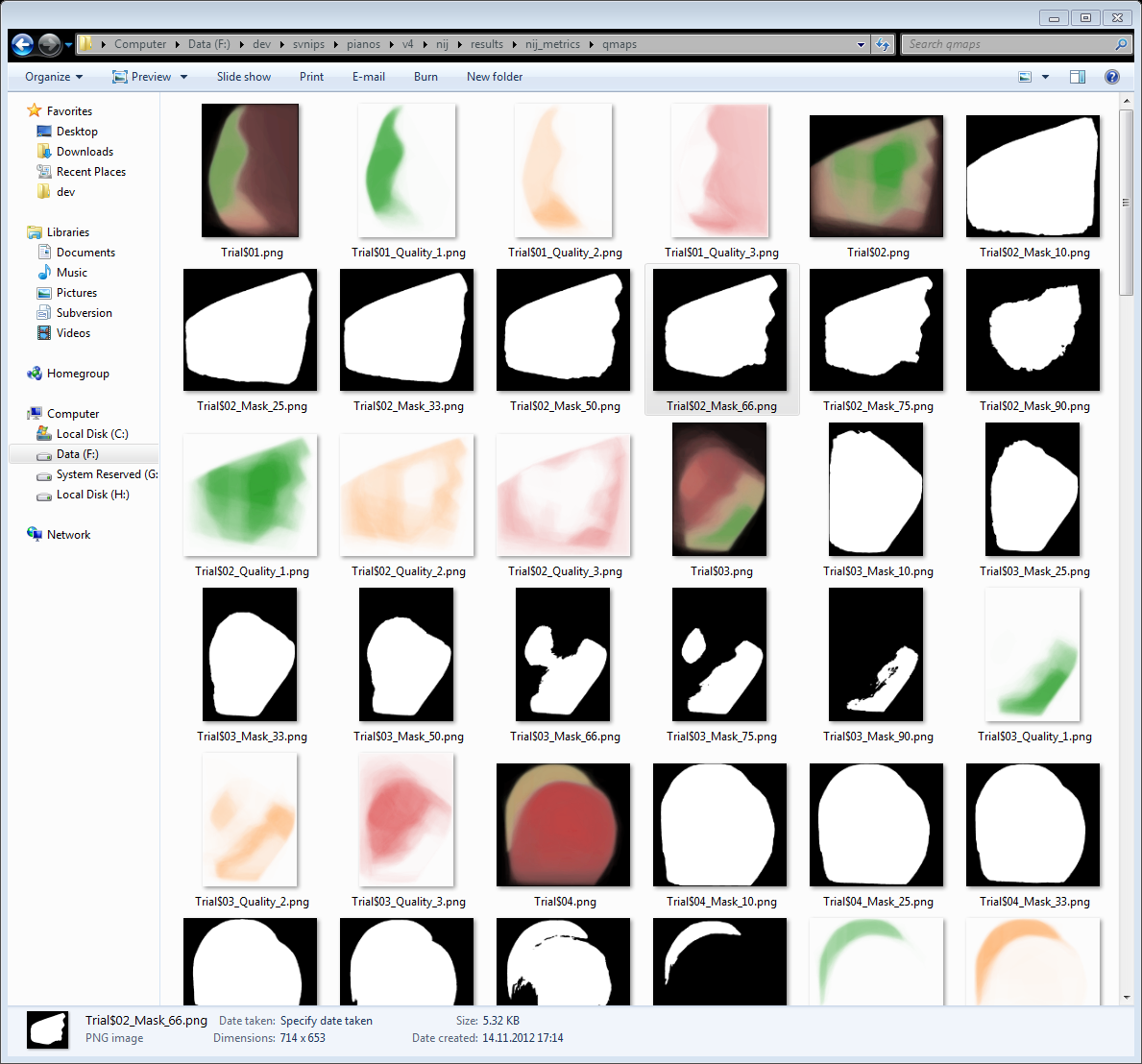


Figure 1: for Trial 01, the overall quality map and its 3 constitutive layers. Note that the colors of the overall map are “wrong” – the RGB summation used here is different than the “real” summation that happens in the quality map.

Because most users did not use the “show final quality map” feature in PiAnoS, they have a complex quality map where zones may be overlapping. The feature (calling the same code as PiAnoS would) is invoked, but this has 2 potential gotchas:

* It may simply fail, in which case the user is dropped (mainly because some users designed degenerated areas or areas with crossovers)
* It may not use the correct “primary quality level” – i.e. the color the user assumed to be background (e.g. green) and drew cuts/holes with other colors (e.g. red)  
  **This is problematic in that the issue is not addressed for the moment (difficult to estimate the consequences)**

Only users that actually contributed to the consensus are considered. This number is kept around for future use in e.g. metric M3.

## Quality masks

To restrict some metrics to a sub-portion of the print – the region of interest – a quality consensus mask is defined, using multiple thresholds (these are mirrored as conditions to the score of some metrics).

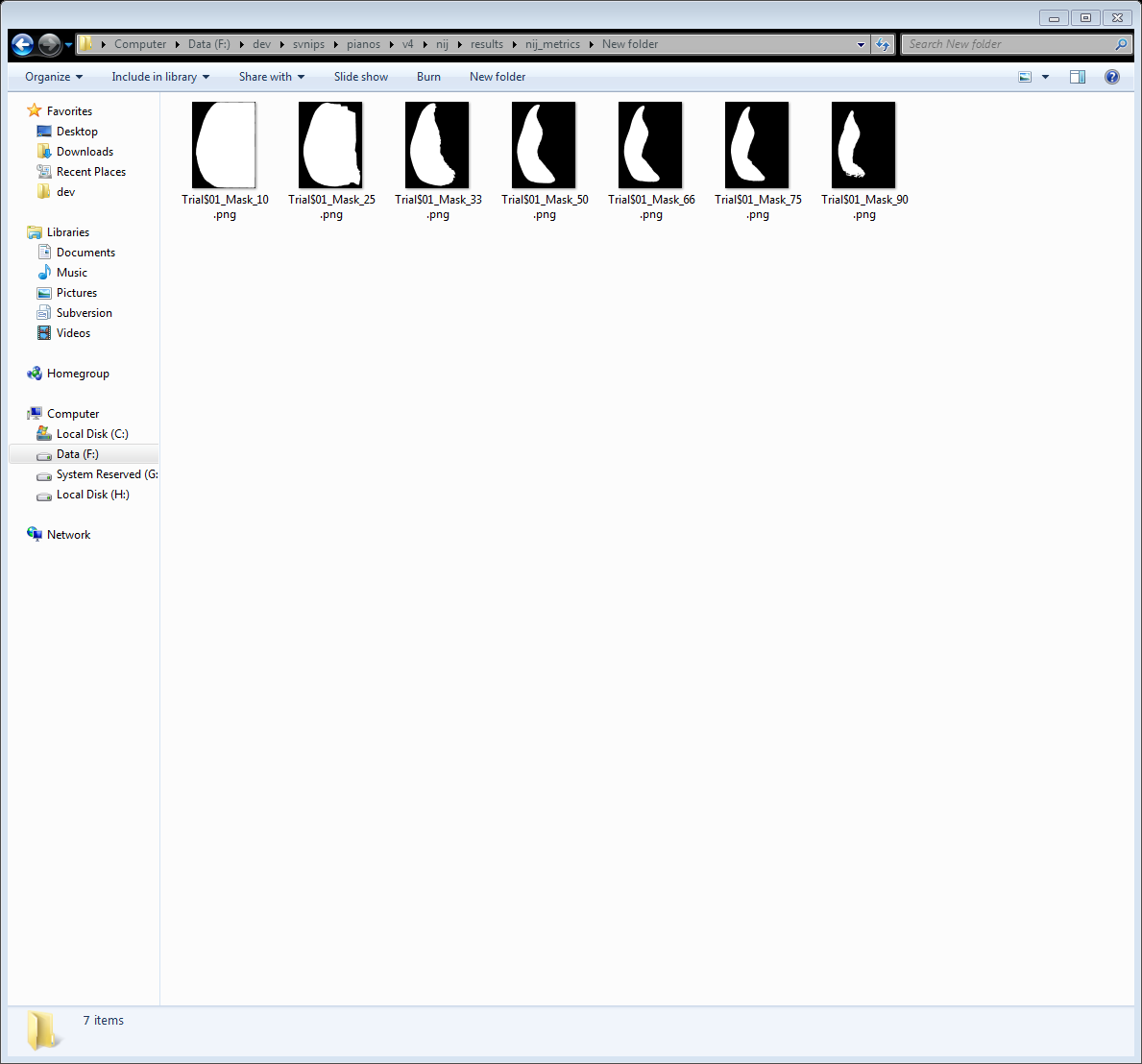


Figure 2: the quality masks for Trial 01, at 10%, 25%… up to 90%.

A pixel is said to be part of the quality consensus mask with threshold X (0<X<1) if it is covered by 100\*X percent of the contributors.

The mask is originally quite jaggy and may contain small holes so it is filtered using kernel filters (expansion + Gaussian blur + reduction + Gaussian blur, applied twice).

## Minutiae consensus

The same idea is applied to the minutiae, which are treated as ellipses whose radii depend on the confidence. Those ellipses are actually bivariate normal distributions (i.e. Gaussian ellipses), that are summed up on a common surface – the minutiae consensus map.

Like the quality consensus, the overall minutiae consensus map is the summation of 4 maps, one by minutia type (from the table below).

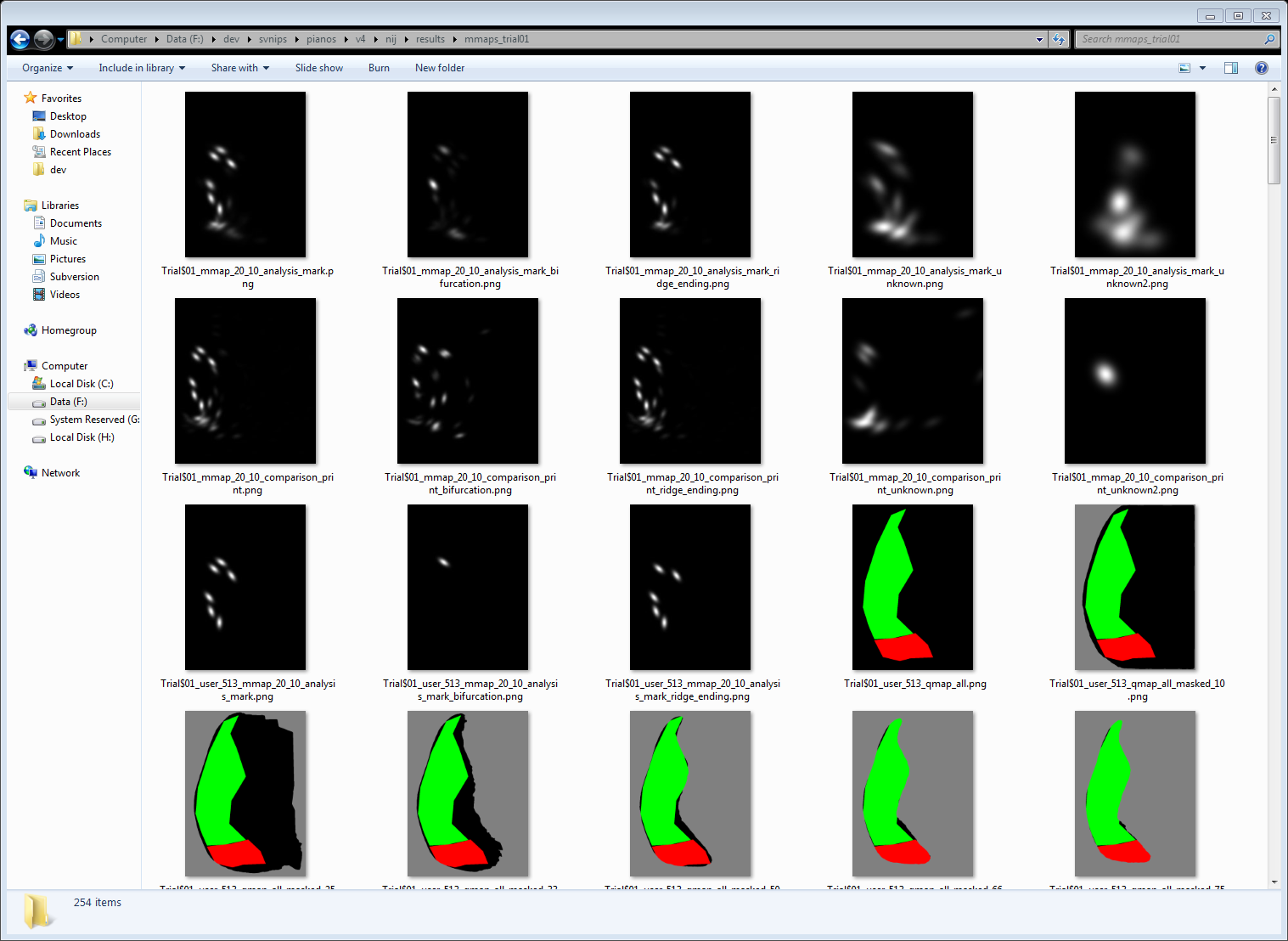


Figure 3: the minutiae consensus maps of Trial 01.   
The mark is on the first row, the print on the second.  
From left to right: overall map, bifurcations, ridge endings, type unknowns, position unknowns  
Note that intensities of all images are normalized, so the “position unknown” layer shown in upper right appears much stronger than it actually is (there may be 3-5 users here vs. 150 in the “bifurcations” layer).

Each ellipse has a density of 1, but its shape depends on its type: it is wider for minutiae with less certainty.

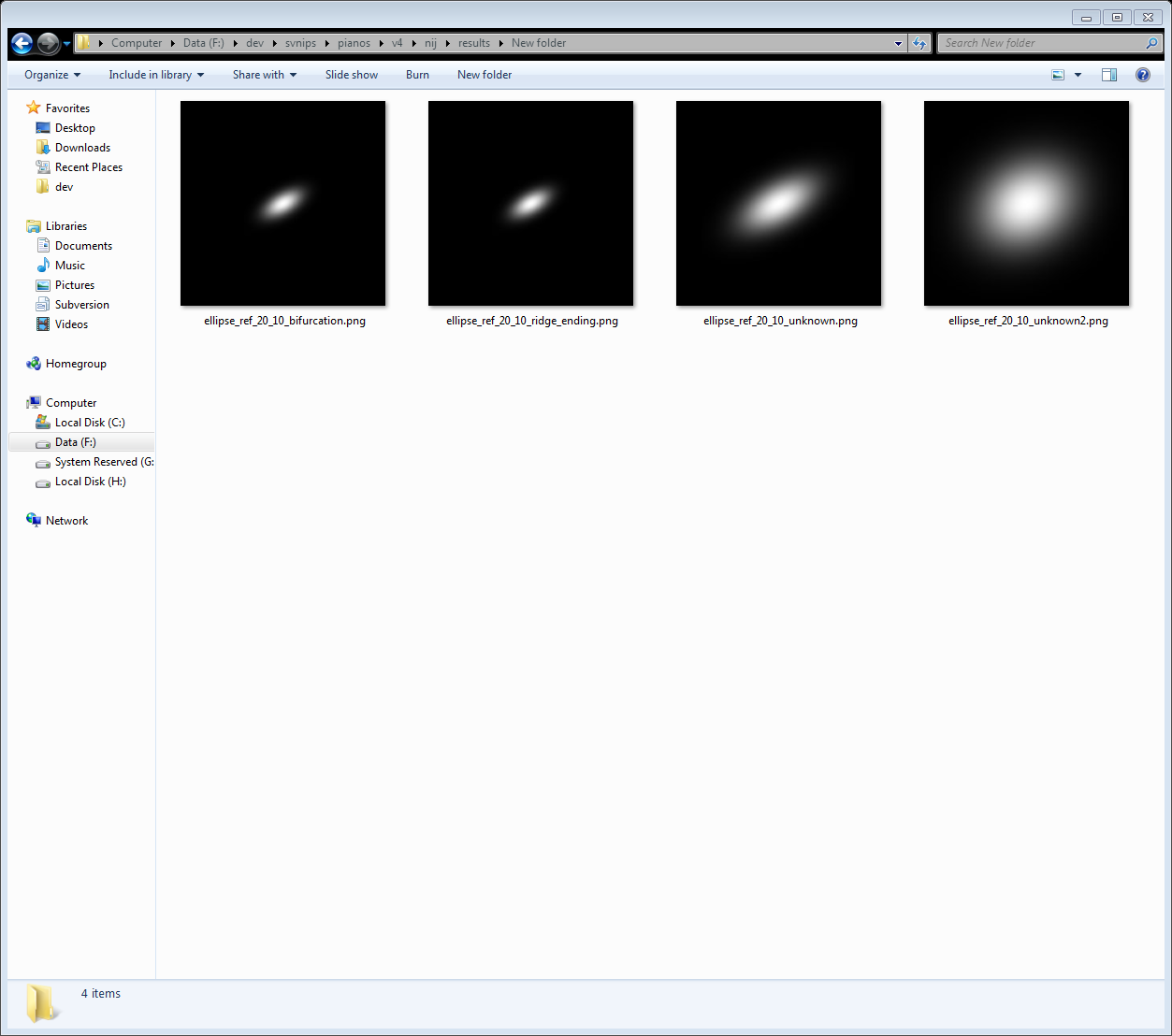


Figure 4: ellipses used for bifurcations (20x10), ridge endings (20x10), type unknown (40x20), and position unknown (50x40)

# Metrics

**Note:** metrics may have a different name in previous documents. They may be easily renamed in Excel before using them if needed.

## **M4: number of minutiae**

*# of minutiae of each type (ridge endings, bifurcations, type unknowns, position unknowns)*

## **M6: number of paired minutiae**

*# of minutiae that are paired*

Result is given for ‘comparison\_mark’ and ‘comparison\_print’. Normally both values should be equal (obviously). But if they differ it means there was a bug in PiAnoS – this is a recently discovered issue.

## M1: coherency metrics

### M1a

*# of minutiae placed outside of all quality zones*

### M1b

*(#re+#bif)/(#total\_in\_zone) for all quality zones.*

Values for high quality (q=1) and low quality (q=2) are expected to be useful. q=2 is here for the record.

### M1c

*# of minutiae in green area / surface of green area*

Values for other zones (orange, q=2 and red, q=3) are here for the record.

### M1d

*(#re+#bif)/#total*

### M1e

*surface of green area / total work surface*

Calculated for all three qualities. The surface is measured in pixel2.

Note: the “total work surface” is the total area covered by a user with quality zones, and is by definition lower than the total surface of the image. -> calculated for all 3 areas

## M2: Perceived quality

As opposed to M1x, M2x metrics use the user’s masked-out quality maps; therefore the “surface of masked green area” in M2x is a subset of the “surface of the green area” in M1x. More specifically it is an “AND” operation between the green area buffer and the mask, where both operands are lists of 0s and 1s.

## M2a

This is the overall ratio between good (green) quality zones and masked work surface.

*(surface of masked green area) / (surface of mask)*

Calculated for all thresholds of the mask.

### M2b

This is a variation of the above considering the medium quality zones as “leaning towards good” quality zones.

*(a.(surface of masked green area) + b.(surface of masked orange area)) / (surface of mask)  
 with a+b=1 and a>b*

(Values initially used: a=0.7, b=0.3)  
But **switched to a=1.0 and b=0.5** (a+b no longer equals 1 but that’s not a problem)

## M3: user's quality maps vs. the quality consensus

It is computed as follows (for each mask threshold):

* For each pixel (x,y) inside the mask
  + u = user’s quality level at (x,y)
  + For each quality level (q)
    - p = proportion (in 0..1) of contributors that used quality q to cover (x,y)
    - w = p\*M(q, u) where M is the matrix below:

M = {  
 (1,1): 0, (1,2): 1, (1,3): 2, (1,0): 3,   
 (2,1): 1, (2,2): 0, (2,3): 1, (2,0): 2,   
 (3,1): 2, (3,2): 1, (3,3): 0, (3,0): 1,   
 (0,1): 3, (0,2): 2, (0,3): 1, (0,0): 0,   
}

This algorithm measures a distance from the user’s map to the consensus. It is designed to be zero only when 100% of the contributors chose the same color as the user. As the proportion lowers, the distance augments. The coefficient matrix M penalizes colors that are “far” from the consensus (i.e. green and orange are close, ‘nothing’ and green are far).

## M5: user's minutiae vs. the minutiae consensus

The same idea is applied to the minutiae maps, but the minutia type replaces the quality level.

### M5a: full consensus map vs. full user’s map

The algorithm calculates the difference between the consensus map (using all minutiae types) and the user’s (using also all types). The consensus map is initially divided by the number of contributors. Again, to have a distance of 0 for a given minutia, all users is the consensus must have used the same minutia (confidence, position, angle) and it must perfectly match the user’s. The score increases as the user was “too confident” (placed a minutia where not all did) or “too conservative” (did not place a minutia where most did).

M5b: minutia precision/accuracy (?)

Each minutia is “projected” onto the consensus (not using ellipses, just its center). It touches the 4 different layers at 4 different densities. These are summed, using a matrix as weighting factor. The matrix is the following:

M’ = {  
 ('ridge\_ending', 'ridge\_ending'): 2.0 , ('ridge\_ending', 'bifurcation' ): 1.0 , ('ridge\_ending', 'unknown' ): 0.5, ('ridge\_ending', 'unknown2' ): 0.25,   
 ('bifurcation' , 'ridge\_ending'): 1.0 , ('bifurcation' , 'bifurcation' ): 2.0 , ('bifurcation' , 'unknown' ): 0.5, ('bifurcation' , 'unknown2' ): 0.25,   
 ('unknown' , 'ridge\_ending'): 0.5 , ('unknown' , 'bifurcation' ): 0.5 , ('unknown' , 'unknown' ): 2.0, ('unknown' , 'unknown2' ): 1.0,   
 ('unknown2' , 'ridge\_ending'): 0.25, ('unknown2' , 'bifurcation' ): 0.25, ('unknown2' , 'unknown' ): 1.0, ('unknown2' , 'unknown2' ): 2.0,  
}

This rewards users placing similar minutiae in similar places, and penalizes those that do not.

## Metric value ranges

The ranges for the metrics are not really defined – they depend on too many factors to be grasped, and therefore make normalization difficult. However, sorting by a given metric does give a satisfying ordering (visually, that is).

# Version tracking

This document always represent the latest version, but information is kept here about the modifications.

## v1: initial version

## v2: 2012-12-18

- the .csv files contain the username (column 3) and the exercise status (column 4). The rest is shifted right by 2 columns  
  
- the overall mask sizes are reported at the end of each line (all lines are equal, but this eases the integration of the data into R as the sizes are not in a separate file)  
  
-  M1b: made sure the color scale is converted first. I don't know how CC and CN spotted the problem, therefore I'll let you check you get the "good" values in the file  
  
- "None" values have been replaced by "NA

- M2a: fixed the "integer calculation" bug.   
  
- M2a: the text was in agreement with the code, but overall, I was wrong. The mask was erroneously not used. The M2a metric has been changed to "(surface of masked green area) / (surface of mask)", taking the mask into account.  
  
- M2b: weights have been set to 1 and 0.5 (instead of 0.7 and 0.3)  
  
- M5: the orientation of the ellipses was wrong, and this has been fixed (pianos uses a clockwise system, the ellipses were drawn using a CCW system). The impact is expected to be small, though, because all ellipses were (badly) rotated the same way.  
  
- M6: this is a new metric: the number of paired minutiae