Reading QR codes on challenging surfaces

David Martínez Carpena

Ismael Benito Altamirano i Dr. Juan Daniel Prades García

Departament d'Enginyeria Electrònica i Biomèdica

Table of contents

Deformations on QR Codes

QR codes

- A common way of representing information in a machine-readable format, thanks to its fast reading speed and reliability.

QR codes

- A common way of representing information in a machine-readable format, thanks to its fast reading speed and reliability.
- A type of 2D (or matrix) barcode.

QR codes

- A common way of representing information in a machine-readable format, thanks to its fast reading speed and reliability.
- A type of 2D (or matrix) barcode.
- Created by the company Denso Wave in the decade of the 90s.

QR codes

- A common way of representing information in a machine-readable format, thanks to its fast reading speed and reliability.
- A type of 2D (or matrix) barcode.
- Created by the company Denso Wave in the decade of the 90s.
- Internationally standardized in the 2000, and has been updated in two occasions, with the current standard from 2015.

Version of QR codes

Version 2

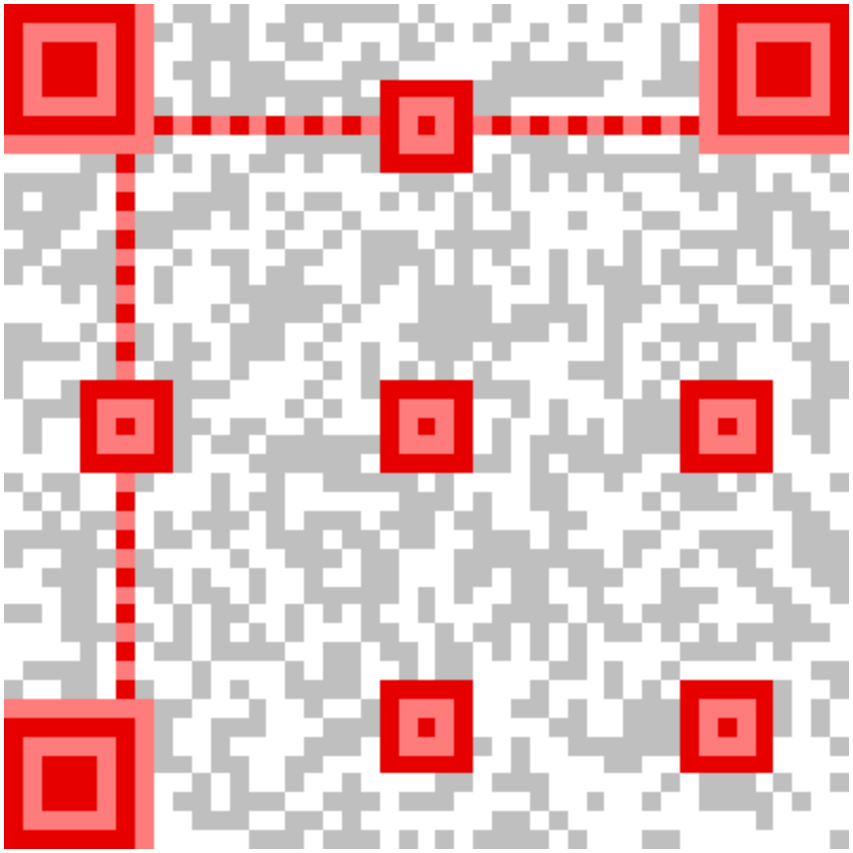
Version of QR codes

Version 7

Structure of QR codes

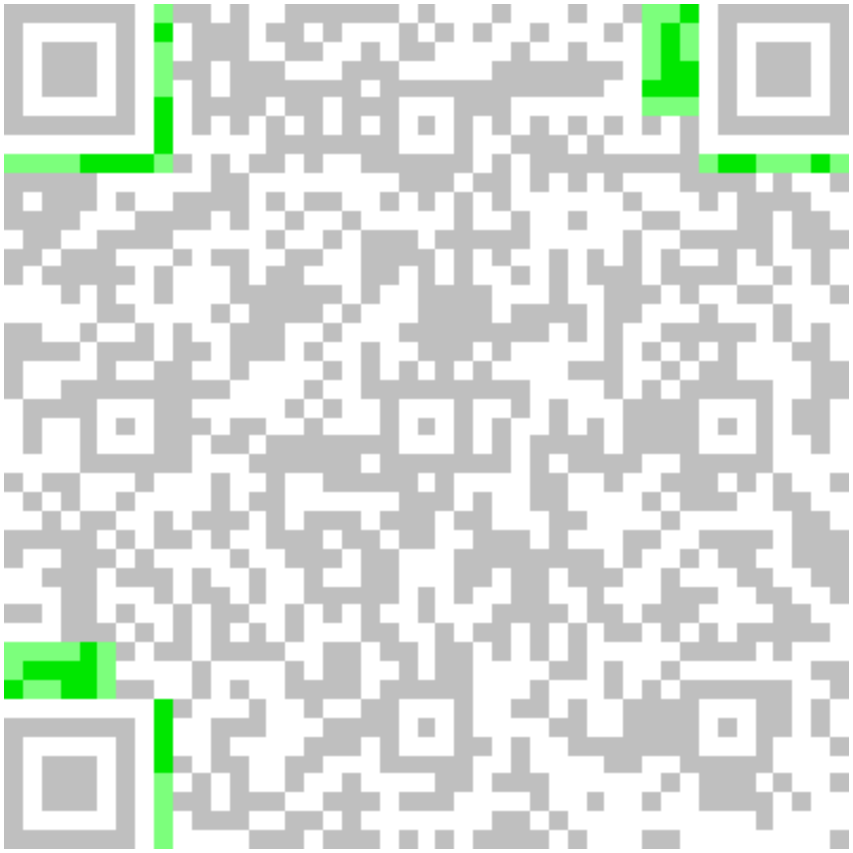
Structure of QR codes

Function patterns



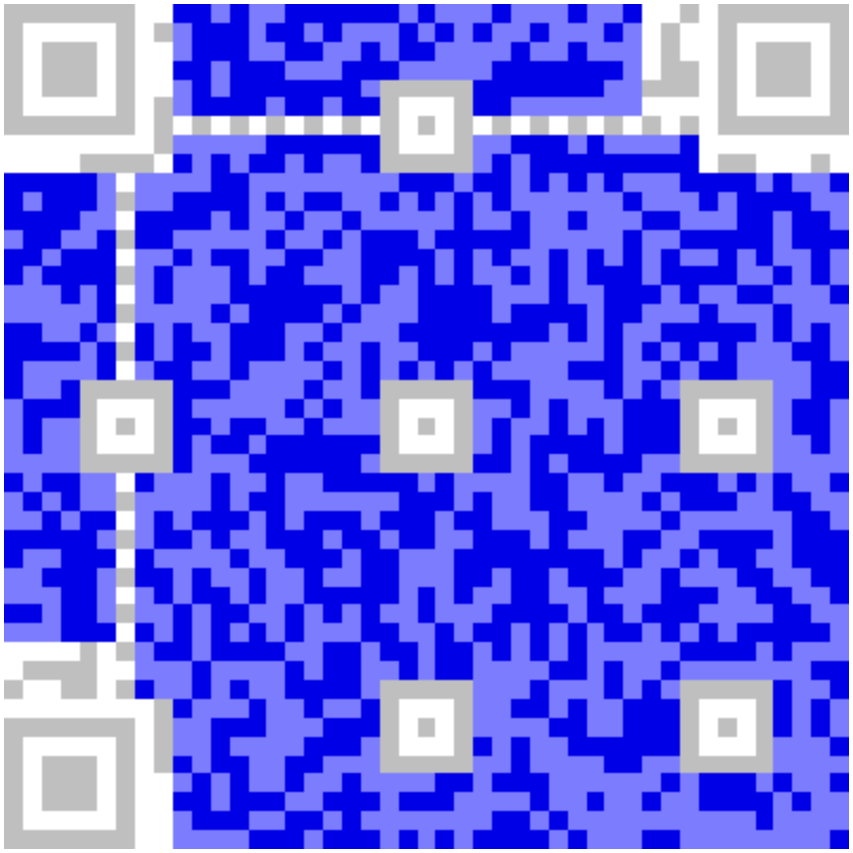
Structure of QR codes

Encoding region

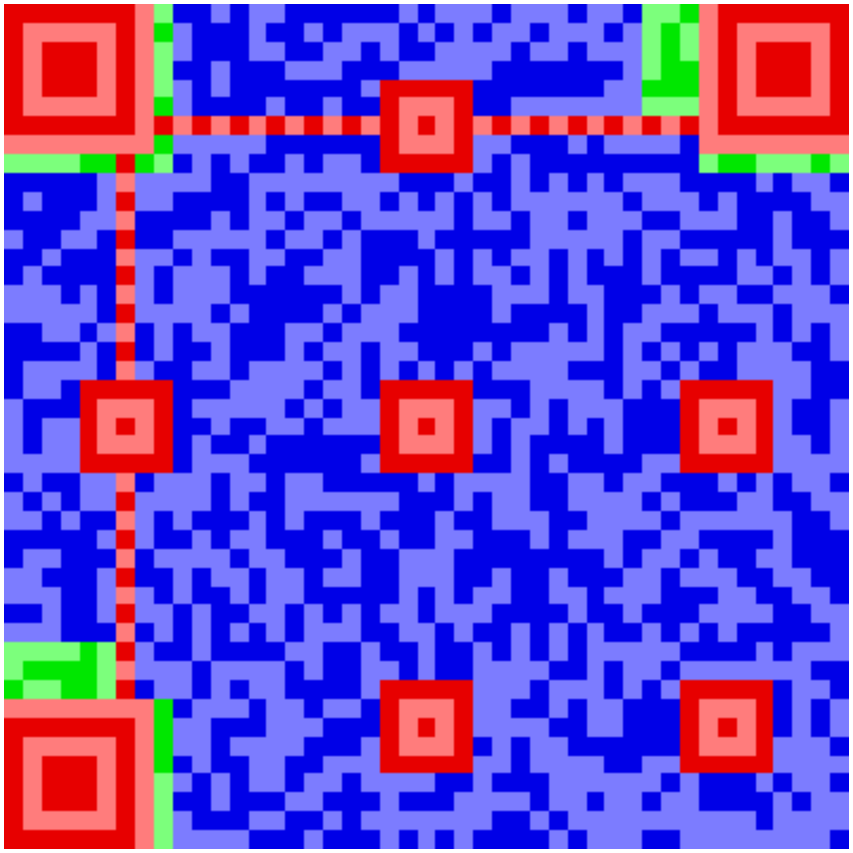


Structure of QR codes

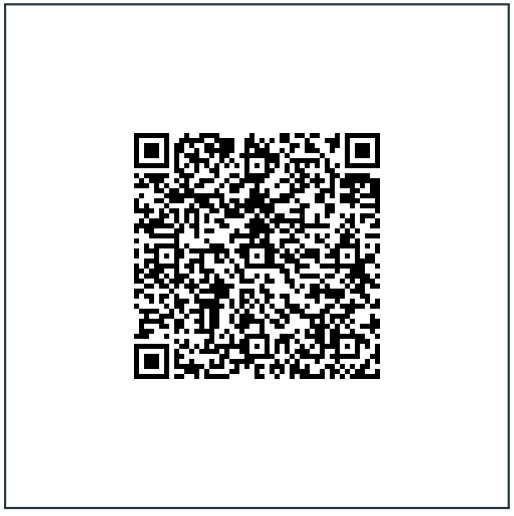
Content region



Structure of QR codes



The problem: Deformations

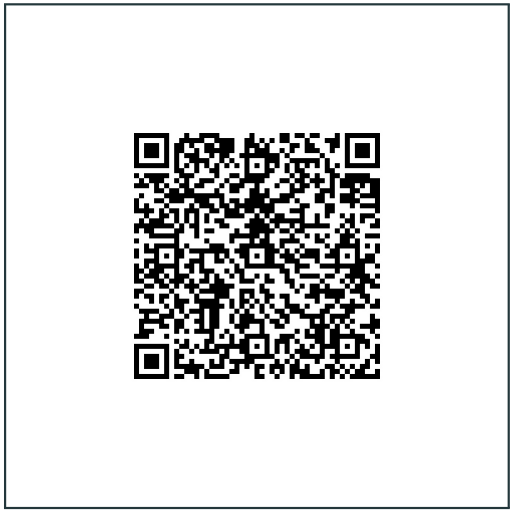


(1)

(2)

(3)

The problem: Deformations



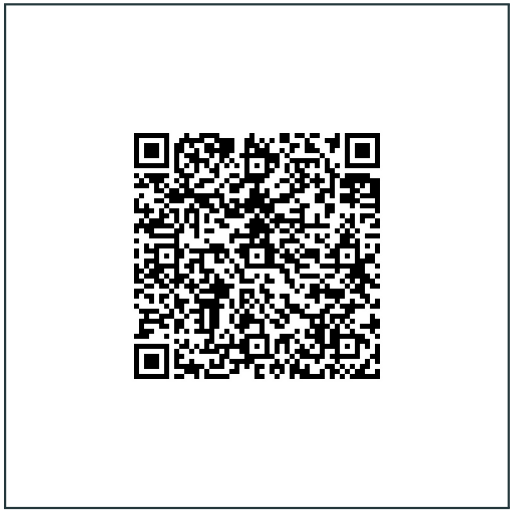
(1)

(2)

(3)

(1) No deformation

The problem: Deformations



(1)

(2)

(3)

(1) No deformation

(2) Perspective deformation from camera view

The problem: Deformations

(1)

(2)

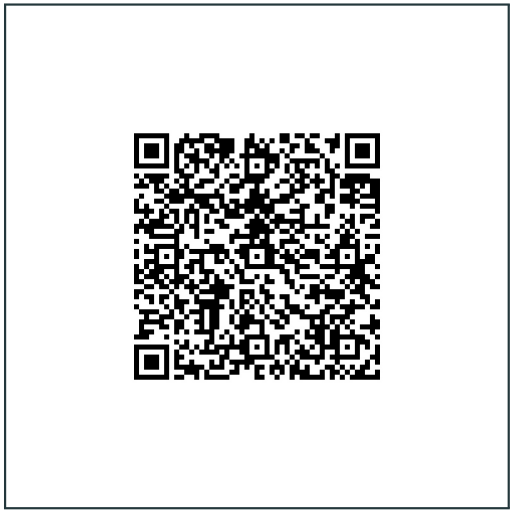
(3)

(1) No deformation

(2) Perspective deformation from camera view

(3) Intrinsic deformation from the surface

The problem: Deformations



(1)

(2)

(3)

(1) No deformation

(2) Perspective deformation from camera view

(3) Intrinsic deformation from the surface

Objectives

- Implementation and a comparison of four correction methods for deformations of QR Codes.

Objectives

- Implementation and a comparison of four correction methods for deformations of QR Codes.
- Make a custom modular QR decoding library.

Objectives

- Implementation and a comparison of four correction methods for deformations of QR Codes.
- Make a custom modular QR decoding library.
- Implement a localizer of QR codes from scratch.

Objectives

- Implementation and a comparison of four correction methods for deformations of QR Codes.
- Make a custom modular QR decoding library.
- Implement a localizer of QR codes from scratch.
- Integrate a third party decoder for the reading of the data.

Objectives

- Implementation and a comparison of four correction methods for deformations of QR Codes.
- Make a custom modular QR decoding library.
- Implement a localizer of QR codes from scratch.
- Integrate a third party decoder for the reading of the data.
- Create datasets to extract some results from the comparison of corrections.

Localization

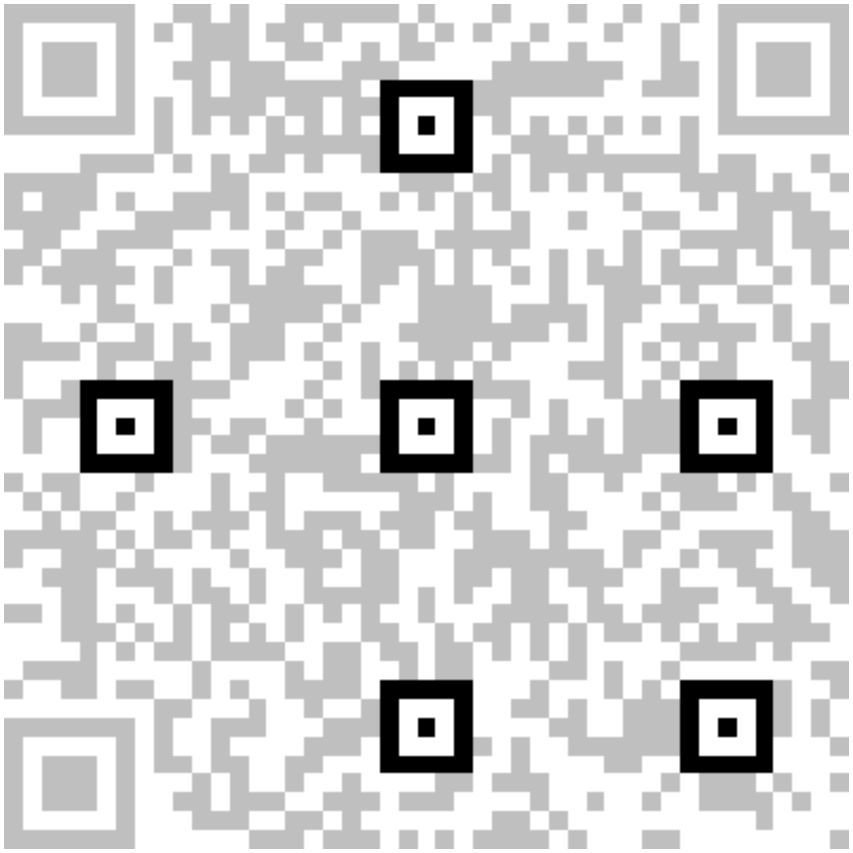
How we localize a QR code?

How we localize a QR code?

Finder patterns

How we localize a QR code?

Alignment patterns

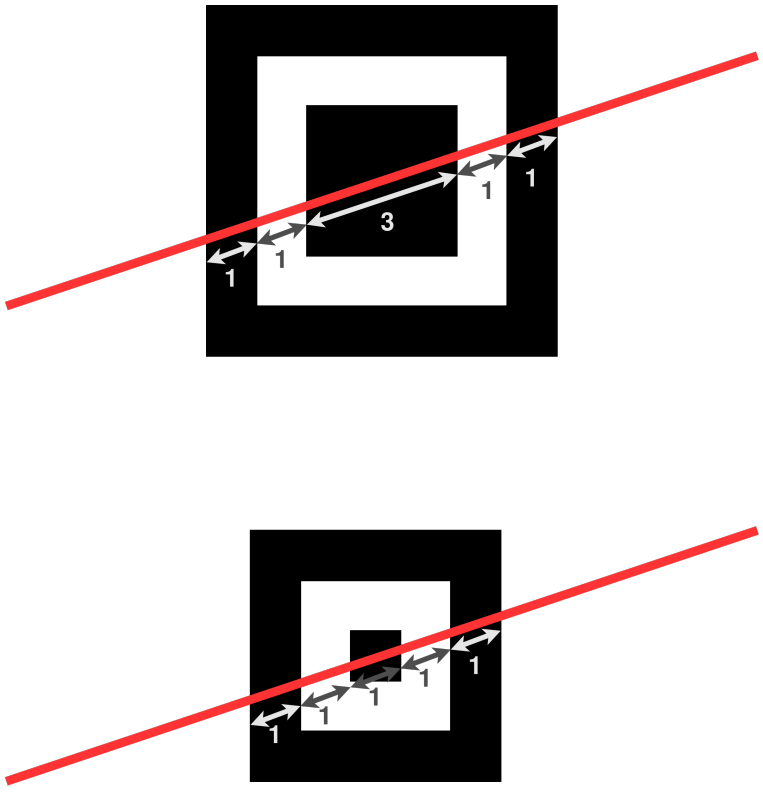


How we localize a QR code?

Timing patterns

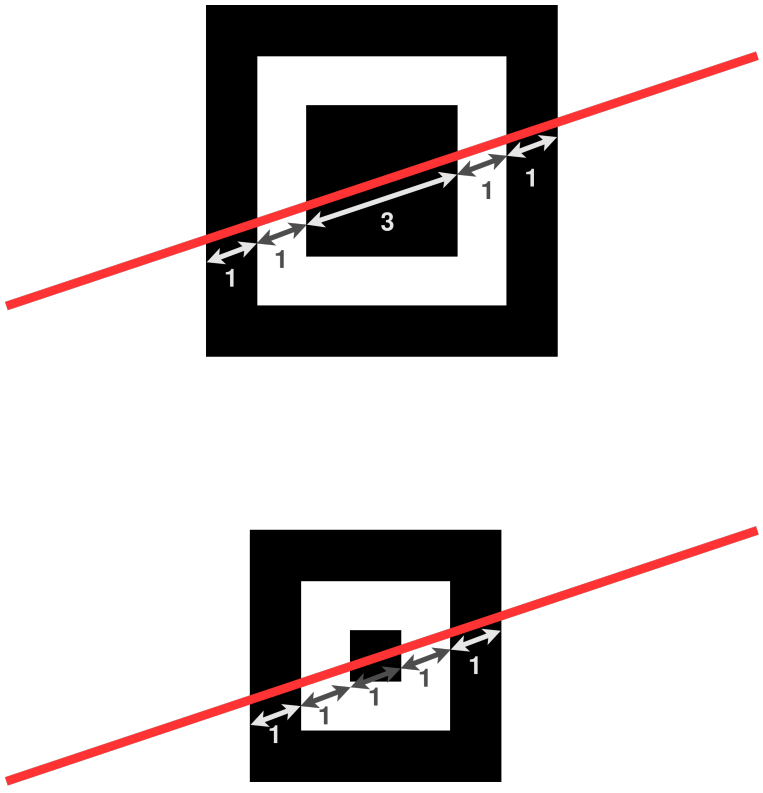
Finder and alignment patterns

- The standard ratio-based algorithm.



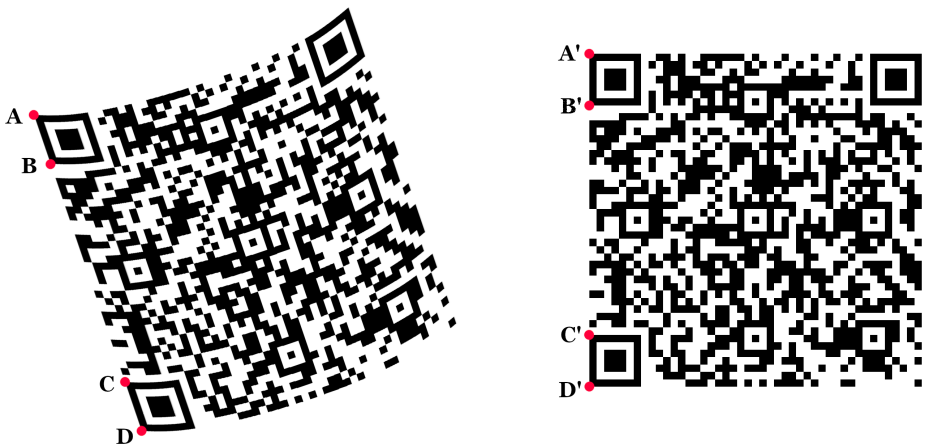
Finder and alignment patterns

- The standard ratio-based algorithm.
- Based on the property that all the lines crossing the pattern by the center follow a constant ratio of black and white pixels.



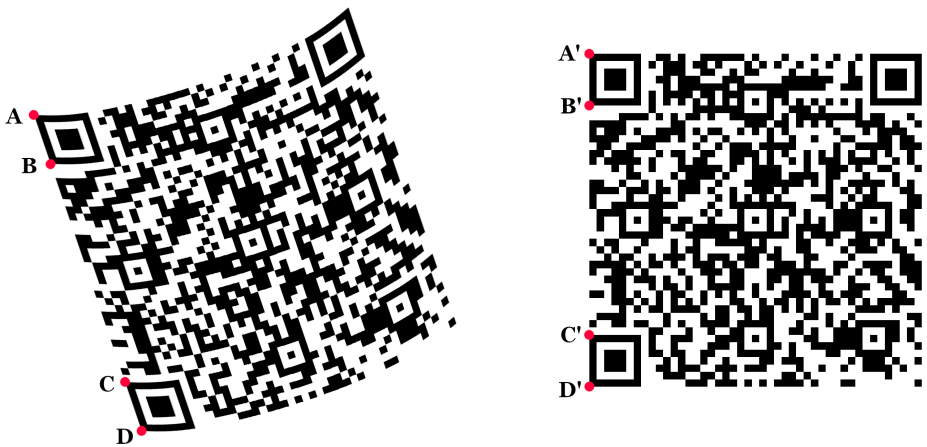
Version

- Algorithm presented for cylindrical deformation, which we will name cross ratio method.



Version

- Algorithm presented for cylindrical deformation, which we will name cross ratio method.
- Uses the assumption that at least one side of the QR Code is close to being linear.



Correction

Correction methods

We will compare four different correction methods in this work:

- Affine (AFF)
- Projective (PRO)
- Cylindrical (CYL)
- Thin Plate Spline (TPS)

Affine (AFF)

- Simplest method of correction, equivalent to the one of the standard.

Affine (AFF)

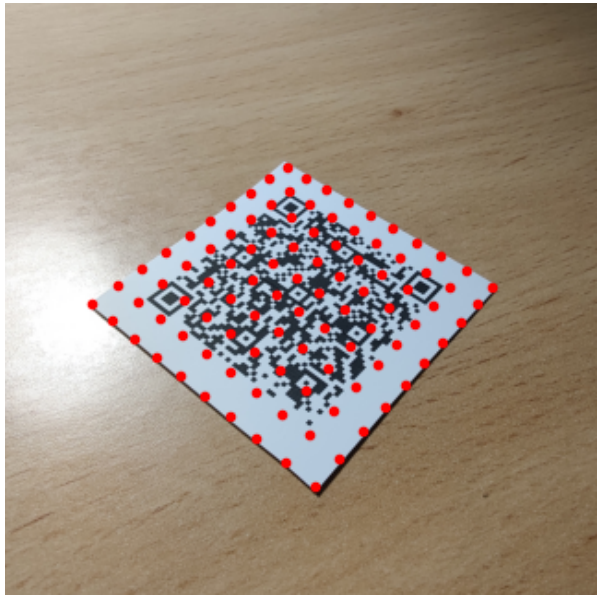
- Simplest method of correction, equivalent to the one of the standard.
- Based on constructing an affine transformation matrix.

Affine (AFF)

- Simplest method of correction, equivalent to the one of the standard.
- Based on constructing an affine transformation matrix.
- Equivalent to a linear transformation (rotation, resizing or skew) and a translation.

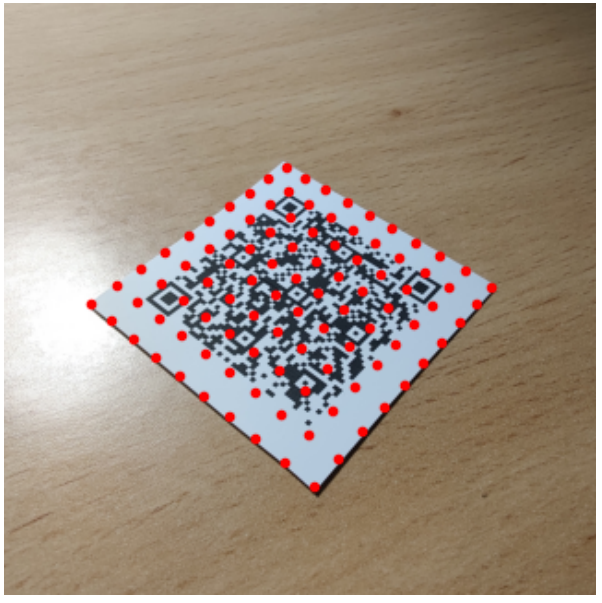
Projective (PRO)

- The most common method used for correction.



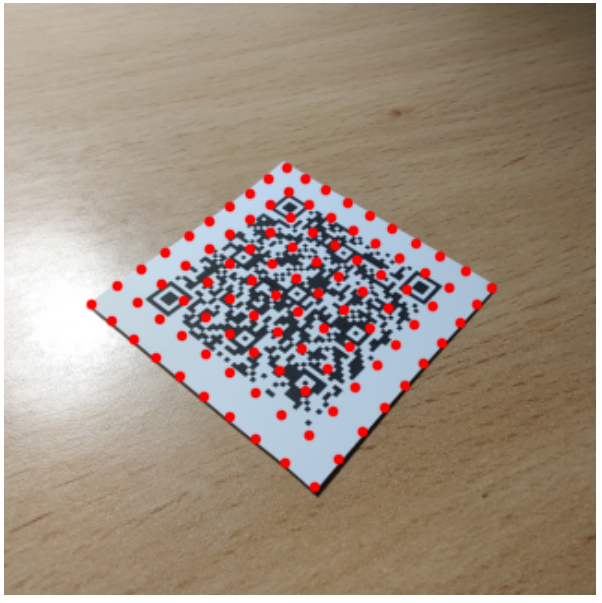
Projective (PRO)

- The most common method used for correction.
- Based on constructing a projective transformation matrix.



Projective (PRO)

- The most common method used for correction.
- Based on constructing a projective transformation matrix.
- Extension of the affine method, which can correct the perspective deformation from the camera view.



Cylindrical (CYL)

- Appeared in recent papers to solve the particular case of cylindrical deformation.

Cylindrical (CYL)

- Appeared in recent papers to solve the particular case of cylindrical deformation.
- Composition of a projective transformation and a non-linear projection to a cylinder.

Cylindrical (CYL)

- Appeared in recent papers to solve the particular case of cylindrical deformation.
- Composition of a projective transformation and a non-linear projection to a cylinder.
- When applied to a flat surface, should be equivalent to a projective transformation.

Thin Plate Spline (TPS)

- Our proposal of a surface independent correction method.

Thin Plate Spline (TPS)

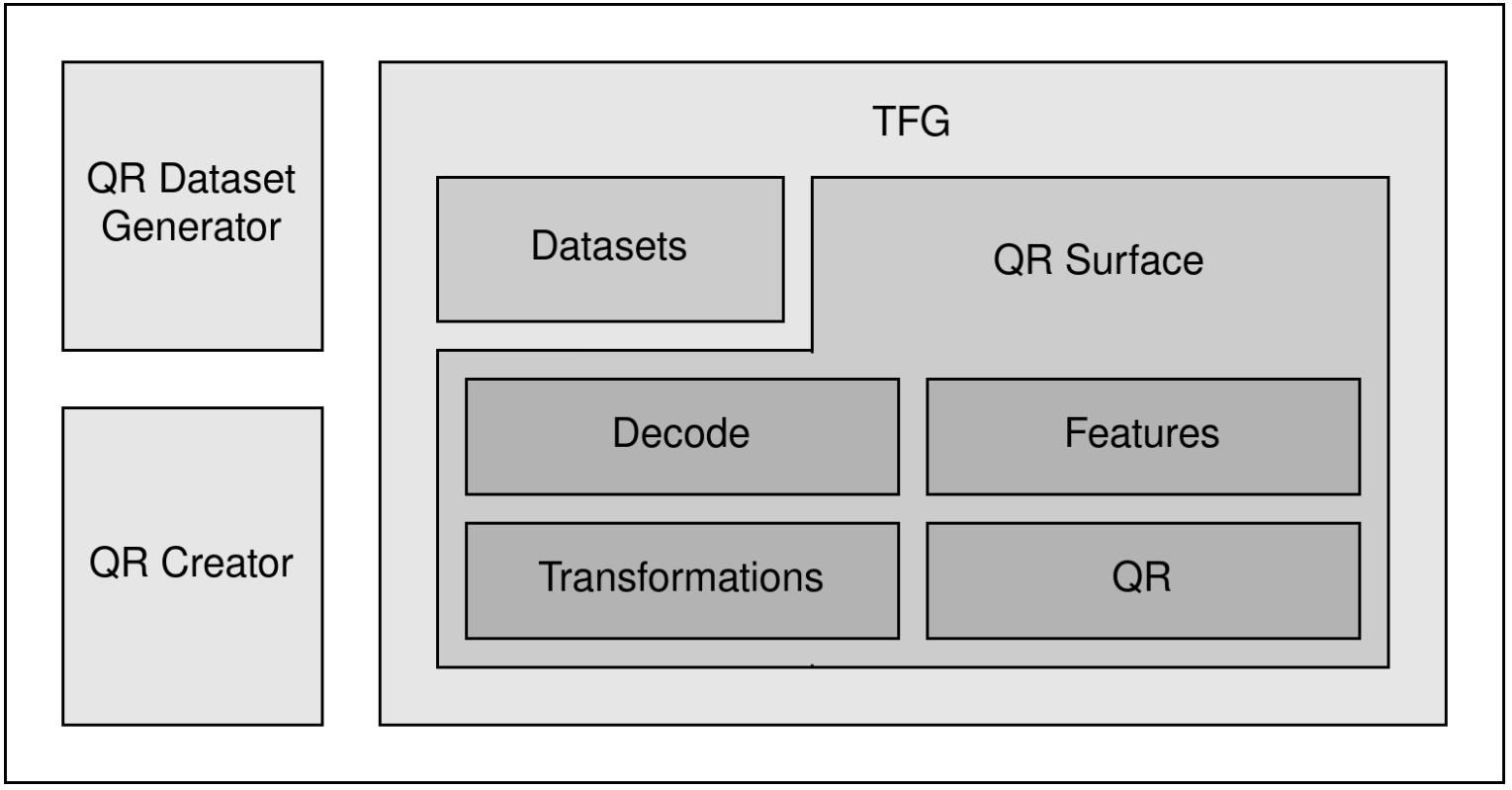
- Our proposal of a surface independent correction method.
- The Thin Plate Spline is a radial basis function with very good general interpolation properties.

Thin Plate Spline (TPS)

- Our proposal of a surface independent correction method.
- The Thin Plate Spline is a radial basis function with very good general interpolation properties.
- The method uses a set of reference points to interpolate an arbitrary non-linear transformation.

Implementation

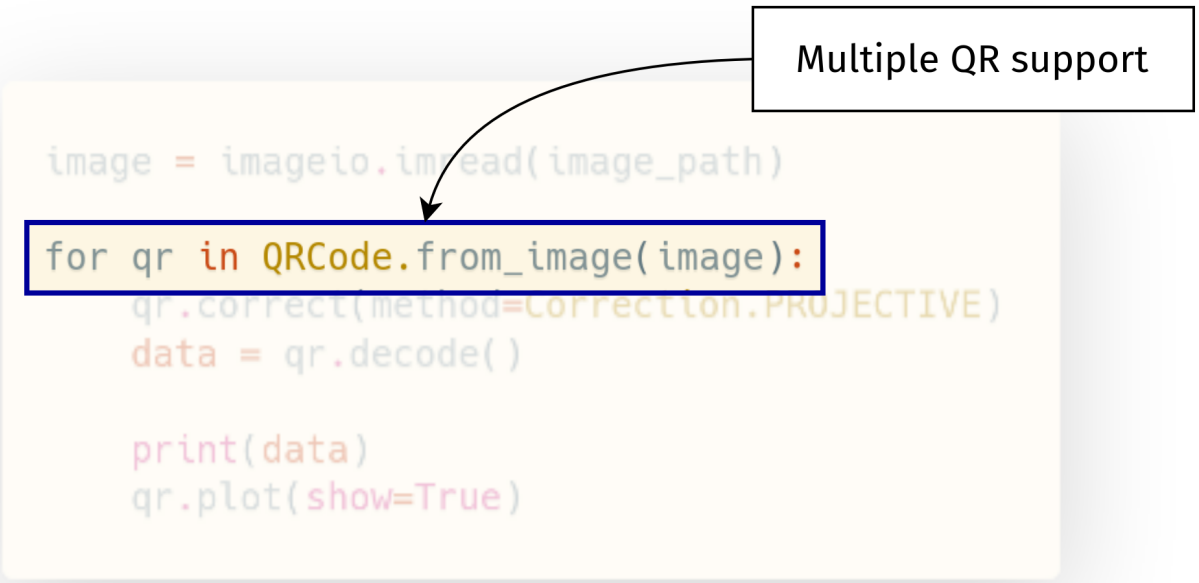
Implementation by modules



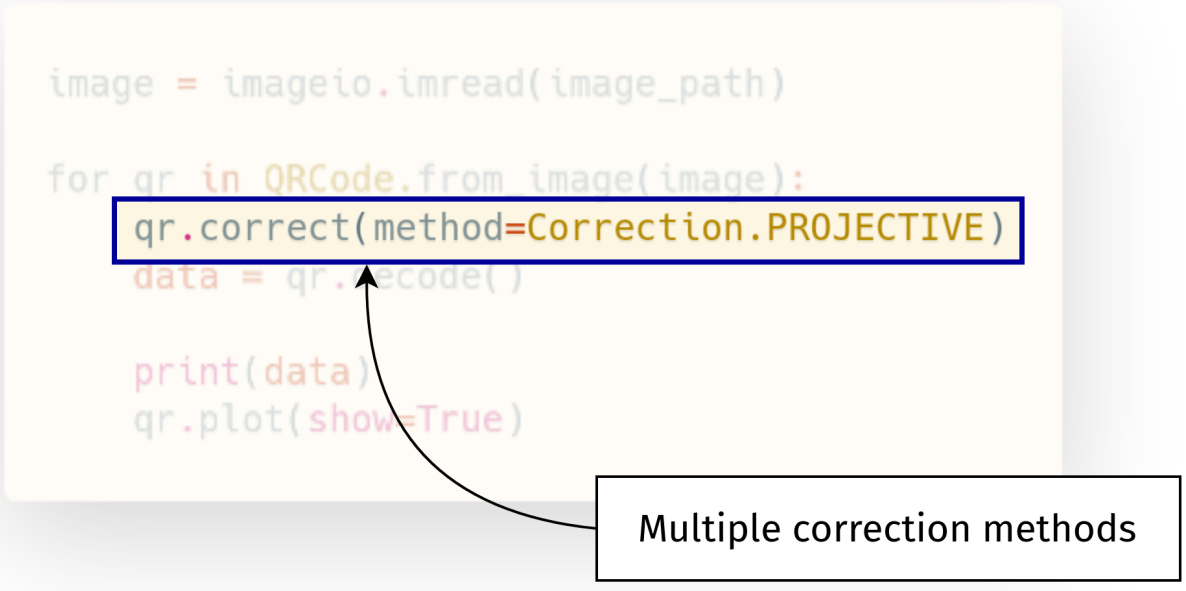
Example of QR Surface

```
image = imageio.imread(image_path)

for qr in QRCode.from_image(image):
    qr.correct(method=Correction.PROJECTIVE)
    data = qr.decode()

    print(data)
    qr.plot(show=True)
```

Example of QR Surface

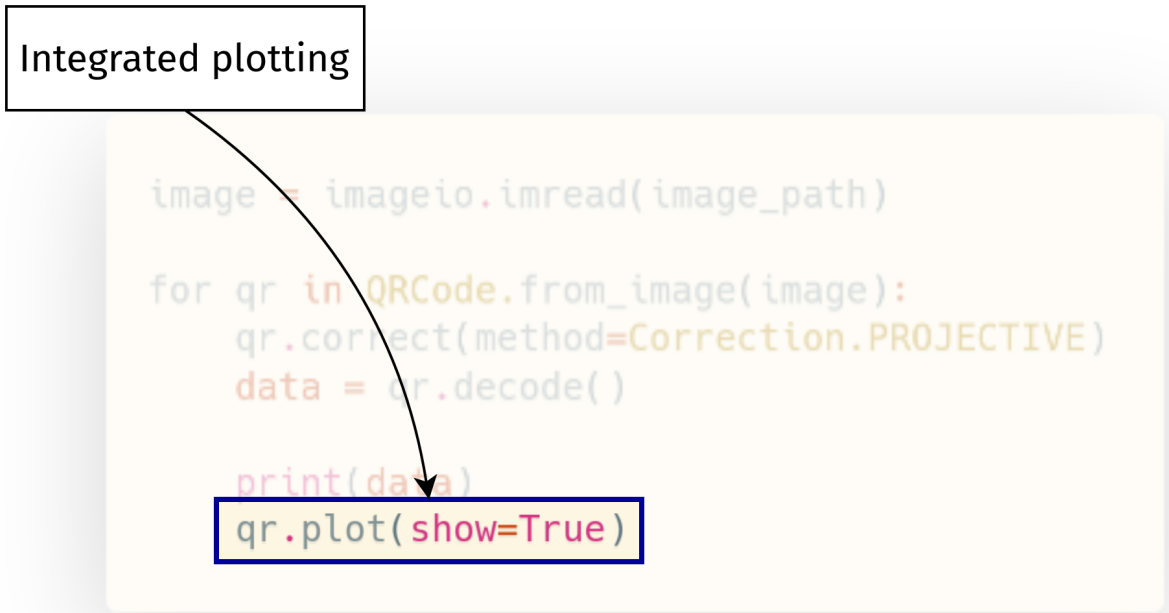


Example of QR Surface



Example of QR Surface

Example of QR Surface

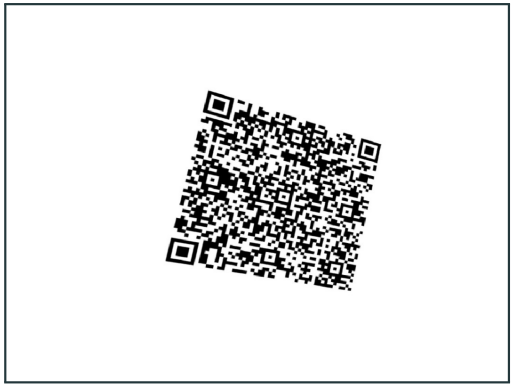


Results

Datasets

(1)

(2)

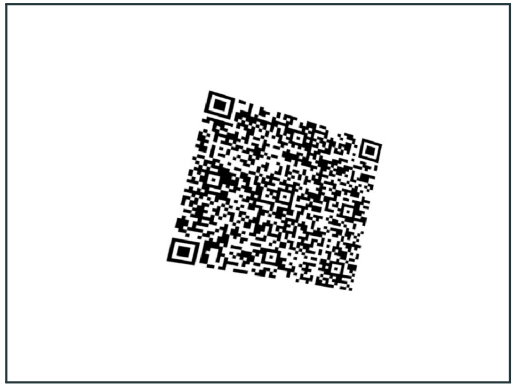


(3)

Datasets

(1)

(2)



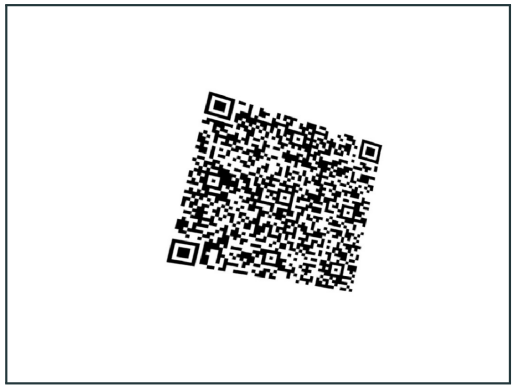
(3)

(1) 50 photos of QR codes in flat surfaces. This dataset has images with more than one QR code.

Datasets

(1)

(2)



(3)

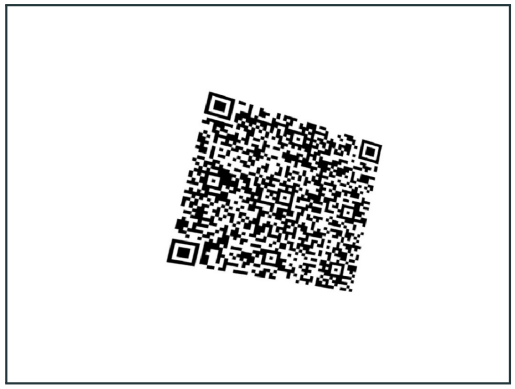
(1) 50 photos of QR codes in flat surfaces. This dataset has images with more than one QR code.

(2) 50 photos of QR codes in cylindrical and arbitrary surfaces.

Datasets

(1)

(2)



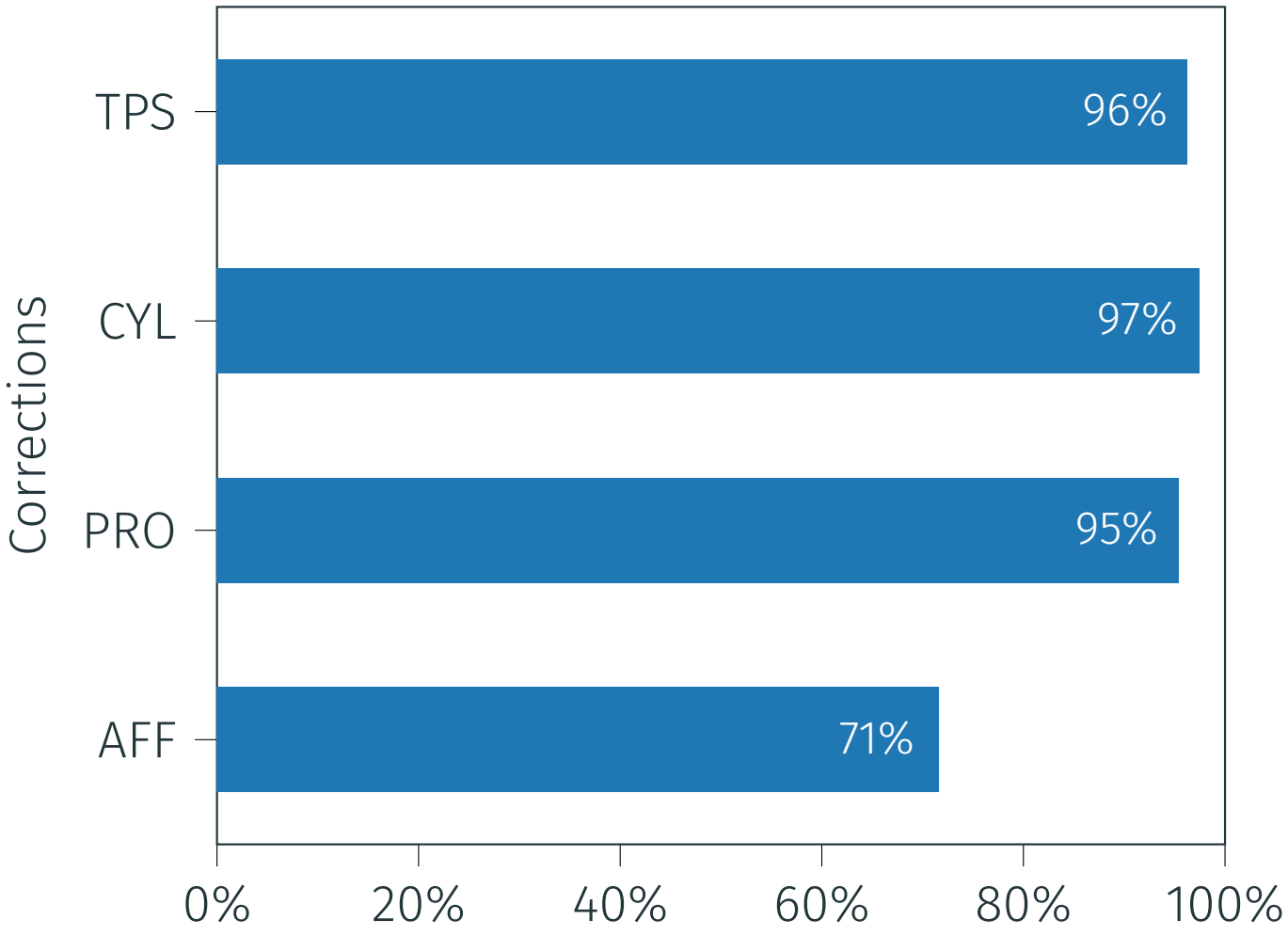
(3)

(1) 50 photos of QR codes in flat surfaces. This dataset has images with more than one QR code.

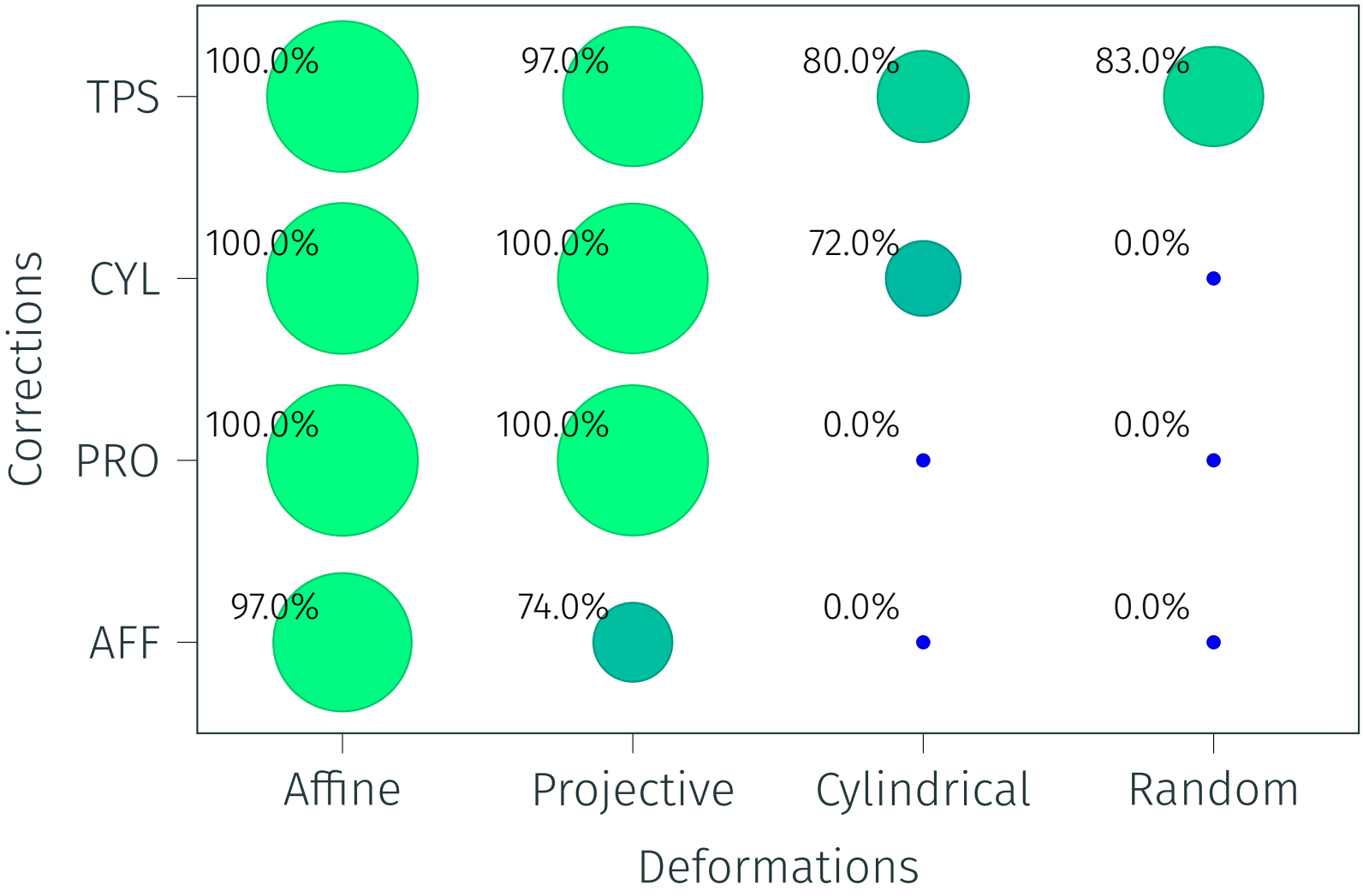
(2) 50 photos of QR codes in cylindrical and arbitrary surfaces.

(3) 819 synthetic images of QR codes with perspective deformation.

Successful decoding by correction

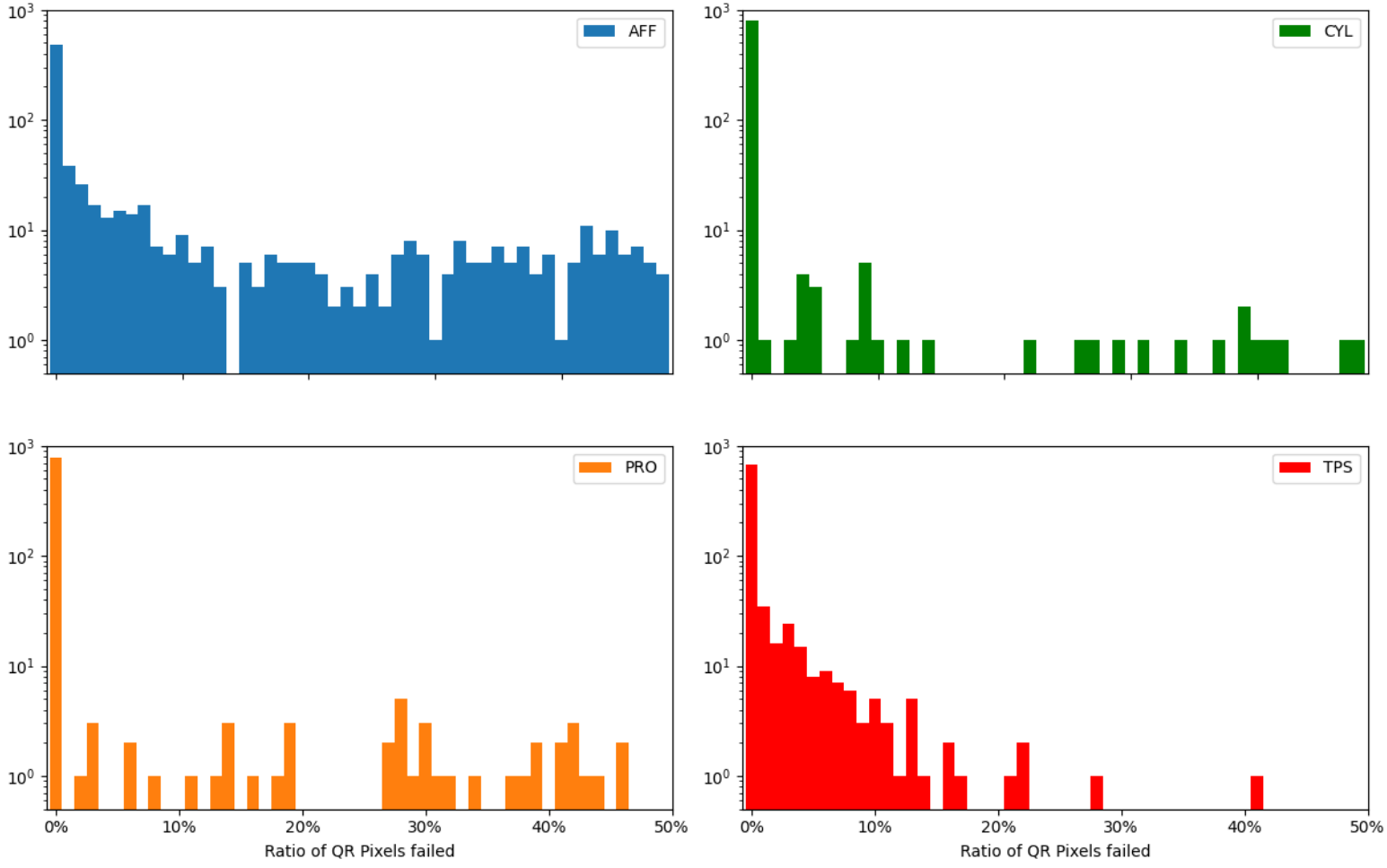


Successful decoding by deformation and correction



Ratio of QR pixels failed

Ratio of QR pixels failed by correction



Conclusion

Conclusion

- Affine and projective methods can’t correct nearly any case of challenging deformation.

Conclusion

- Affine and projective methods can’t correct nearly any case of challenging deformation.
- The cylindrical method works well with flat QR Codes, while being able to achieve good results in the cases with cylindrical deformation, but it can’t correct the arbitrary deformations.

Conclusion

- Affine and projective methods can’t correct nearly any case of challenging deformation.
- The cylindrical method works well with flat QR Codes, while being able to achieve good results in the cases with cylindrical deformation, but it can’t correct the arbitrary deformations.
- Thin Plate Spline, overcome all our expectations, being close to the other methods in flat deformation images, while being able to correct cylindrical and arbitrary deformations.

Further work

- Try to solve the problem of localization of QR Code features in arbitrary deformations.

Further work

- Try to solve the problem of localization of QR Code features in arbitrary deformations.
- Implement different decoding backends, with more commercial decoders or a handmade one.

Further work

- Try to solve the problem of localization of QR Code features in arbitrary deformations.
- Implement different decoding backends, with more commercial decoders or a handmade one.
- Create greater datasets and extract more meaningful results.

Questions?