# PhotoScan Analysis

Mit dem Python Skript lassen sich einige Analysen zur Inneren Genauigkeit eines Bildverbandes durchführen. Zum einen kann ein Report zur Bildmessgenauigkeit erstellt werden. Weiter lassen sich die Bildmessungen visualisieren. Um die Aufnahmekonfiguration zu bewerten dient ein 3D-Plot der Fehlerellipsoide der Objektpunkte (nicht metrisch).

## Report zur Bildmessgenauigkeit

Camera Name Projections SIGMA x SIGMA y SIGMA P MAX x MAX y

IMG\_4287.JPG 297 0.32126 0.35394 0.47799 1.26451 1.10858

IMG\_4288.JPG 231 0.36953 0.39133 0.53823 1.23350 1.23678

.......

IMG\_4335.JPG 612 0.19908 0.28742 0.34963 1.08133 1.27099

IMG\_4336.JPG 367 0.27735 0.35559 0.45096 1.09968 1.40987

RMS: 0.39662 0.52912

* Projections (Anzahl der in das Bild projizierten Objektpunkte)
* SIGMA x, y (Standardabweichung der Bildmessungen in x und y)
  + Berechnung:

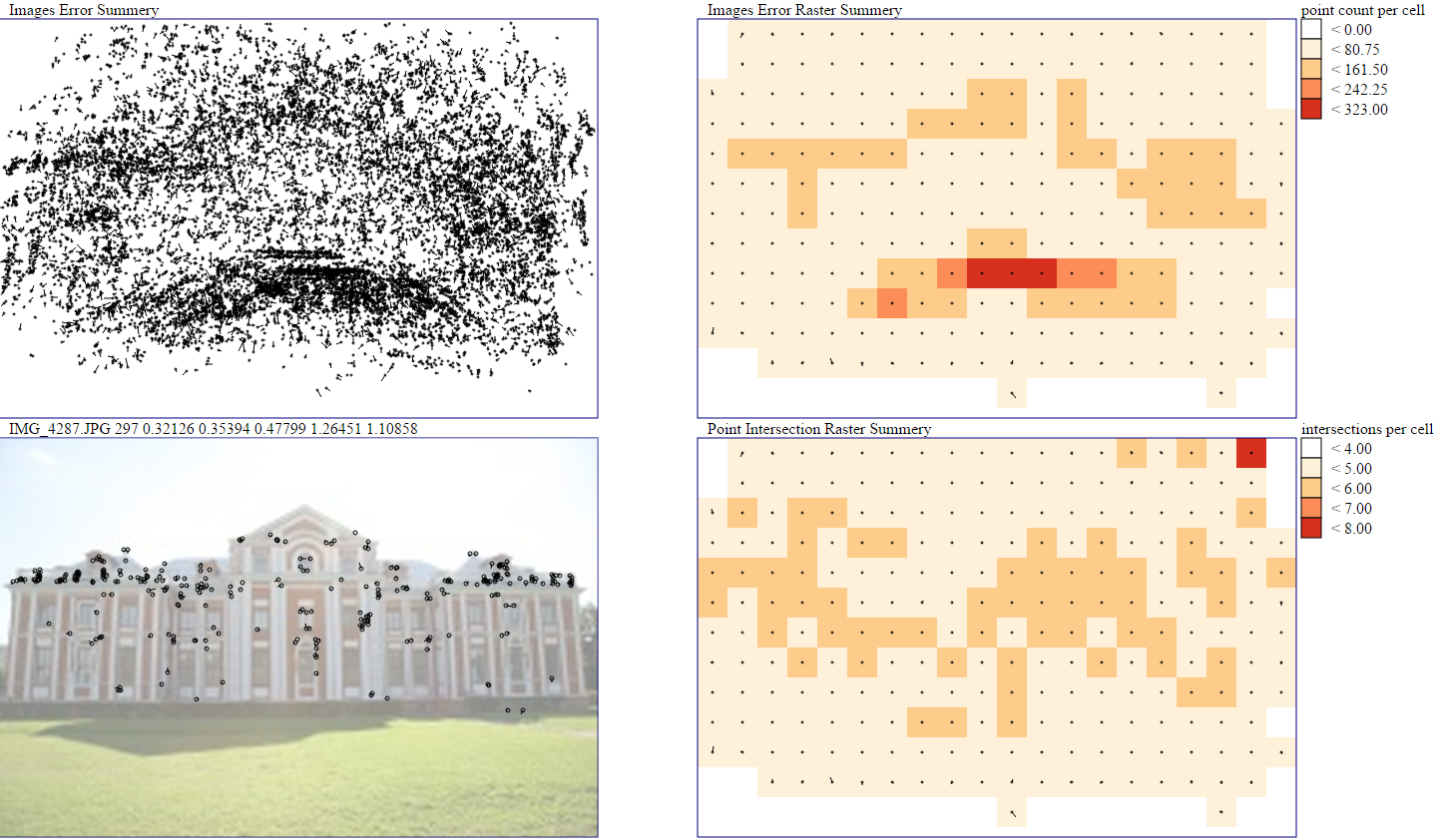
* SIGMA p (Punktstandardabweichung
* MAX x,y (Maximaler Bildmessfehler in x und y)
* RMS (Root Mean Square: )

## Visualisierung der Bildmessungen

1

2

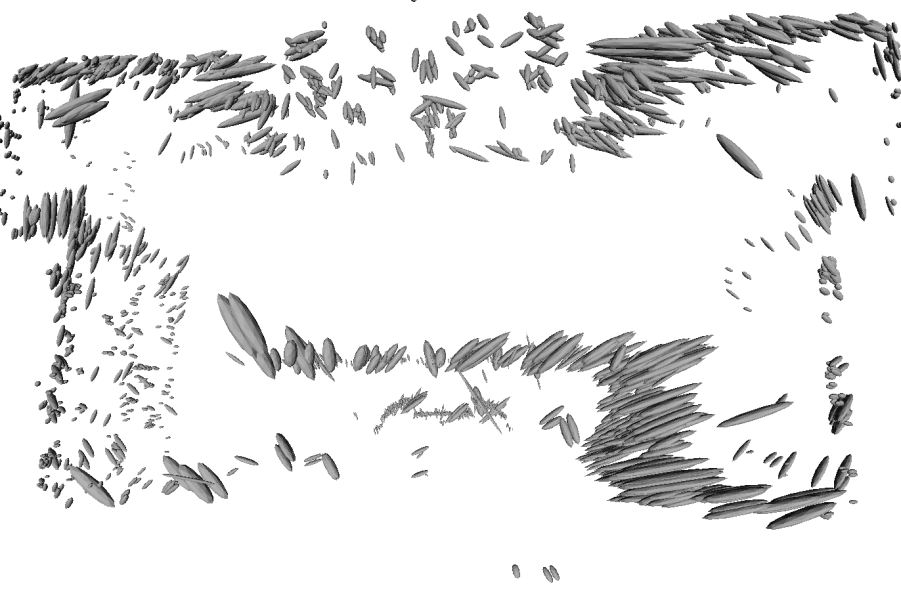
3



4

1. Zusammenfassung aller Bildmessung mit Fehlervektoren
2. Bildmessungen mit Fehlervektoren pro Bild
3. Zusammenfassung aller Bildmessung in regelmäßigen Rastern. Der Mittelpunkt einer Zelle zeigt den Fehlervektor aller in der Zelle liegenden Messungen. Über die Farbskalierung lässt sich die Anzahl der Messungen pro Zeller ermitteln.
4. wie 3. Die Farbskalierung entspricht der durchschnittlichen Anzahl an Strahlenschnitte aller Punkte einer Rasterzeller.

## Fehlerellipsoide der Objektpunkte



Die Fehlerellipsoide geben Aufschluss über die Netzkonfiguration des Bildverbandes. Eventuelle grobe Fehler lassen sich durch verhältnismäßig große Ellipsoide erkennen.

Die Berechnung der Ellipsoide erfolgt über die Bestimmung der Normalgleichungsmatrix pro Objektpunkt

und der daraus resultierenden Kovarianzmatrix . Die Jakobimatrix wird definiert über den Parametervektor ,welcher nur die Koordinaten eines Objektpunktes enthält (Alle anderen Parameter werden als fest angenommen). Der Beobachtungsvektor enthält die Bildmessungen aller Kameras. Die genaue Aufstellung der Jakobimatrix findet sich im Luhmann S. 244 und 308. Die für die Gewischtsmatrix notwendigen Standardabweichungen der Beobachtungen sind die bereits für den Report bestimmten Standardabweichungen pro Bild.

Durch Bestimmung der Eigenwerte (durch Singulärwertzerlegung) der Kovarianzmatrix lassen sich die Halbachsen der Fehlerellipsoide bestimmen (Luhmann S. 345).

Um Ellipsoide im STL-Format zu erzeugen, wird zunächst eine Einheitskugel (R=1, X0,Y0,Z0 = 0) zu einem Ellipsoid transformiert. Zur Skalierung dienen die Längen der Halbachsen. Die Rotation erfolgt über eine Rotationsmatrix


R_z(\alpha) = \begin{pmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha &  \cos \alpha & 0 \\         
   0        &  0           & 1
\end{pmatrix}

R_y(\alpha) = \begin{pmatrix}
\cos \alpha  & 0 & \sin \alpha \\
   0         & 1 &  0          \\
-\sin \alpha & 0 & \cos \alpha
\end{pmatrix}

R_x(\alpha) = \begin{pmatrix}
1 &   0         & 0           \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha &  \cos \alpha
\end{pmatrix}


# Nutzung

Um das Skript auszuführen wählt man in PhotoScan den Menüpunkt Tools-> Run Script … und wählt die Datei **analysis.py** aus. Wird das Skript ohne Argumente benutzt erscheint der folgende Hilfetext:

PhotoScan Analysis v0.2

HowTo:

Command Line Arguments:

-report\_out [filename] Creates a report file. Options: filename (default: report)

-svgout [filename] Creates a SVG-Image with image-measurements Option: filename (default: image\_measurements

-svgfactor [factor] Magnification factor of the error-vector for the SVG-File (default: 40)

-svgcols [columns] The number of columns used to generate the overview image (default: 20)

-stlout [filename] Create a STL-Mesh with Point-Error-Ellipsoids. Option: filename (default: stl\_export)

-stlfactor [factor] Magnification factor of the ellipsoid-axis (default: 100)

-export\_ellipsoid [filename] Export a ellipsoid file

Sample:

-rout reportname -svgout svgname -svgfactor 12 -svgcols 10 -stlout stlname -stlfactor 12

GUI

You can also use the GUI by choosing the argument '-useGUI'

Please open a Project with completed photo alignment

Besonders die Vergrößerungsfaktoren sind für jedes Projekt oder Analysefall unterschiedlich und müssen iterativ gefunden werden. Dazu muss das Skript mehrmals, mit veränderten Werten, gestartet werden, bis das gewünschte Ergebnis erzielt ist. Um die Rechenzeit zu beschleunigen müssen nicht immer alle drei Ausgaben erzeugt werden.

### Ergebnis

Die erzeugten Dateien liegen im PhotoScan Projektverzeichnis.

reportname.txt

svgname.svg SVG-Dateien können im Web-Browser (z.B. Firefox) geöffnet werden

stlname.stl STL-Dateien können in der Open-Source Software MeshLab geöffnet werden

ellname.ell Datei mit Ellipsoidparametern zur weiteren Nutzung (z.B. Matlab)

## Verwendete Software-Quellen:

* svd.py Python-Skript zur Singulärwertzerlegung von Thomas R. Metcalf unter der

GNU-Lizenz (<http://stitchpanorama.sourceforge.net/Python/svd.py>)

* STL\_Writer.py Python-Skript zur Erzeugung binären STL-Dateien. MIT-Lizenz

(https://github.com/jv4779/spin\_mirror/blob/master/STL\_Writer.py)

* pysvg Python-Modul zur Erzeugung von SVG-Dateien. BSD-Lizenz

(http://codeboje.de/pysvg/)

**Changelog**

v0.2 [NEU] Export der Ellipsoide im .ell Format.

v0.3 [NEU] Übersichtsbild mit durchschnittlicher Anzahl der Strahlenschnitte

v0.4 Bugfix Matrixmultiplikation