MASTER MVA
REMOTE SENSING

ENS Paris Saclay

POINTING ERROR CORRECTION

Jules Kozolinsky, Vincent Matthys
jules.kozolinsky@ens-cachan.fr
vincent.matthys@ens-paris-saclay.fr

Contents

1 Attitude of a Satellite **1**
1.1 Localization and projection functions . . . 2
1.2 Epipolar curves . . . 3
1.3 Affine fundamental matrix approximation . . . 3

2 Satellite attitude error effects on stereo images **3**
2.1 Pointing error . . . 3
2.2 Sensibility . . . 3

3 Correction of Relative Pointing Error **3**
3.1 Roll and Pitch Angles . . . 4
3.2 Roll, Pitch and Yaw Angles . . . 5
3.2.a) Approximation of the Rotation . . . 5
3.2.b) Solving Numerically an Optimization Problem . . . 6
3.2.b).1 Limited-memory BFGS . . . 6

4 Planet data **8**
4.1 Structure . . . 8
4.2 Geolocalisation . . . 9

A Rotations and translation **10**
A.1 Rotation followed by a translation . . . 10
A.2 Translation followed by a rotation . . . 10

Abstract

1 Attitude of a Satellite

Roll, pitch and yaw angles. (See Figure 1).

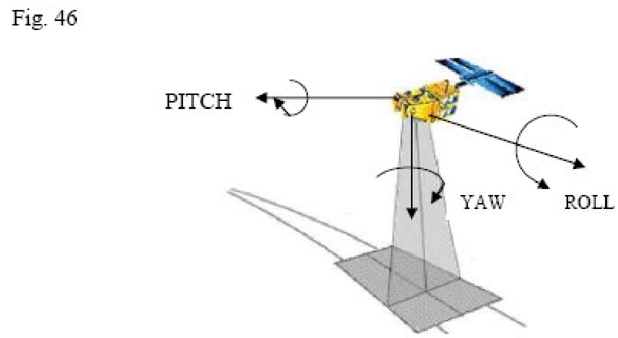

Figure 1: Roll, pitch and yaw angles.

1.1 Localization and projection functions

Localization and projection functions allow converting from image coordinates to coordinates on the globe and back (See Figure 2).

The projection function returns the image coordinates, in pixels, of a given 3-space point.

$$\begin{aligned} P : \mathbb{R}^3 &\to \mathbb{R}^2 \\ (\lambda, \theta, h) &\mapsto \mathbf{x} \end{aligned}$$

In that system a point of 3-space is identified by its longitude $\lambda \in [-180, 180]$, latitude $\theta \in [-90, 90]$ and altitude h, in meters, above the reference ellipsoid.

The localization function is its inverse with respect to the first two components. It takes a point $\mathbf{x} = (x, y)^\top$ in the image domain together with an altitude h, and returns the geographic coordinates of the unique 3-space point $\mathbf{X} = (\lambda, \theta, h)$ whose altitude is h and whose image is $\mathbf{x}$.

$$\begin{aligned} L : \mathbb{R}^3 &\to \mathbb{R}^2 \\ (\mathbf{x}, h) &\mapsto (\lambda, \theta) \end{aligned}$$

We use the Rational Polynomial Coefficient (RPC) camera model which is an analytic description of the projection and localization functions (cite). In the RPC model, the projection and localization functions are expressed as ratio of multivariate cubic polynomials.

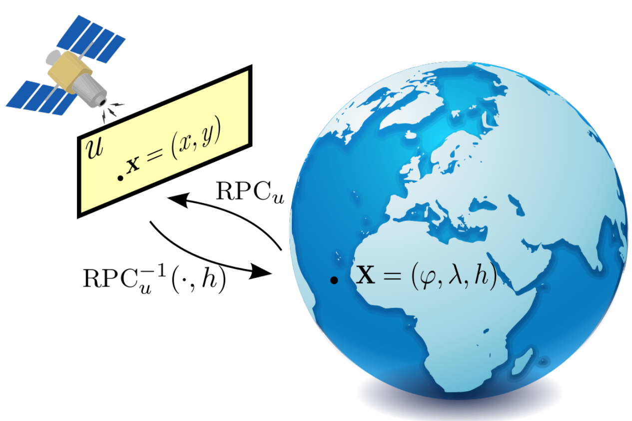

Figure 2: Projection (RPC) and localization (RPC^{-1}) functions.

1.2 Epipolar curves

For a point $\mathbf{x}$ in the image u, the epipolar is the curve pararametrized by the alitude h and defined as:

$$\mathrm{epi}_{uv}^{x} : h \to \mathrm{RPC}_u(\mathrm{RPC}_v^{-1}(\mathbf{x}, h))$$

where RPC and RPC^{-1} are the projection and localization functions. In pratice we observe this curve as straight parallel lines.

1.3 Affine fundamental matrix approximation

From [?], we know that, for each correspondence i, the epipolar curve $\mathrm{epi}_{uv}^{\mathbf{x}_i}(R)$ is approximated up to 0.05 pixels by the straight line $F\mathbf{x}_i$, where F is the affine fundamental matrix between the two views for the considered tile. Since the fundamental matrix, due to the approximation, is *affine*, all the lines $(F\mathbf{x}_i)_{i=1...N}$ are parallel.

2 Satellite attitude error effects on stereo images

2.1 Pointing error

However, due to the limited precision of the camera calibration, there is a bias of a few pixels in the RPC functions. For stereo matching, we can not ignore this error since the epipolar constraints are derived from the parameters of the cameras. Therefore this bias has to be corrected before applying rectification.

This error, often of the order of a few pixels, is the sum, for all correspondences, of the distance between a correspondance on an image and the epipolar curve on the same image computed with the correspondance on the other image. Formally, given two images u, v and a set of correspondences $(\mathbf{x}_i, \mathbf{x}'_i)_{i=1...N}$, the pointing error between u and v is defined by:

$$\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}'_i, \mathrm{epi}_{uv}^{\mathbf{x}_i})$$

which can be approximated, using the affine fundamental matrix F, to:

$$\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}'_i, F\mathbf{x}_i)$$

2.2 Sensibility

The effect of attitude errors on the localization function RPC^{-1} can be computed simply with zero roll and pitch angles. For roll or pitch angles a small error ε induces, to first order, a translation of $a\varepsilon$ on the ground, where a is the distance between the satellite and the ground. For the yaw, a small error ε induces a rotation of at most $D\varepsilon/2$ on the ground, where D is the width of the projected pushbroom sensor of the satellite (See Figure 3).

For SkySat satellites, we can have an error $\varepsilon \simeq 50$ μrad with leads to an error of about 10 degree knowing that $D \simeq 9$km.

3 Correction of Relative Pointing Error

A simple and automatic way to correct the pointing error is to transform one of the two images, in such a way that the corresponding points fall on the respective epipolar curves. More formally, given two images u, v and a set of correspondences $(\mathbf{x}_i, \mathbf{x}'_i)_{i=1...N}$, we search for a transformation f such that, for all i, the

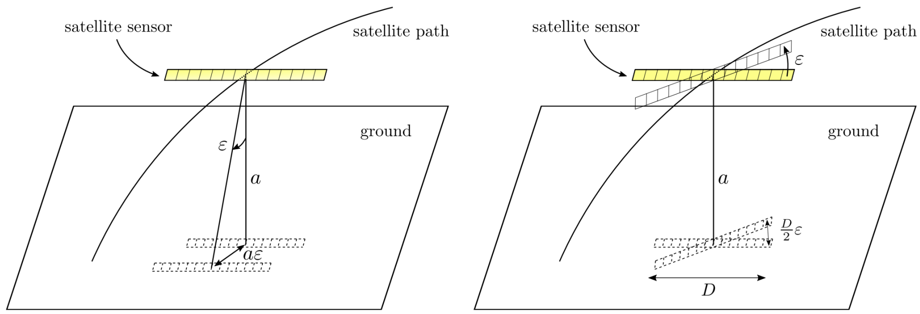

Figure 3: Sensibility for roll, pitch and yaw angles.

transformed point $f(\mathbf{x}'_i)$ lies on the epipolar curve $\text{epi}_{uv}^{\mathbf{x}_i}(R)$. The desired transformation f^* minimises the relative pointing error:

$$f^* = \underset{f}{\text{argmin}}\, \frac{1}{N}\sum_{i=1}^{N} d(f(\mathbf{x}'_i), \text{epi}_{uv}^{\mathbf{x}_i}(R)) \tag{1}$$

3.1 Roll and Pitch Angles

Because of sensitivity issues, we first can take only roll and pitch error into account. Therefore, according to section 2.2, we search for a transformation f such that

$$f(\mathbf{x}) = T\mathbf{x}$$

where T is a translation:

$$T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

We can write the transformation f, for $\mathbf{x} = (x\ y\ 1)^T$, as:

$$f(\mathbf{x}) = T\mathbf{x} = \begin{pmatrix} x + t_1 \\ y + t_2 \\ 1 \end{pmatrix}$$

Without any additional restriction, we may assume that these lines are horizontal (otherwise just do a change of coordinates). This change of coordinates is called *rectification*. We find

After rectification, for each point i, the horizontal line $F\mathbf{x}_i$ can be written as

$$F\mathbf{x}_i = [0\ 1\ c_i]$$

With these notations, for each point correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$, the pointing error e is:

$$e(\mathbf{x}_i, \mathbf{x}'_i) = d(\mathbf{x}'_i, epi_{uv}^{\mathbf{x}_i}(R)) = d(\mathbf{x}'_i, F\mathbf{x}_i) = |y'_i + c_i|$$

Here the error e is invariant to any horizontal translation, thus the search for a translation minimizing the relative pointing error of formula (1) can be restricted to vertical translations only. With a vertical translation of parameter t, the global pointing error becomes

$$E = \frac{1}{N}\sum_{i=1}^{N} d(T\mathbf{x}'_i, F\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N} |y'_i + t + c_i|$$

The translation that minimizes this sum is given by the geometric median (Weiszfeld, 1937) of the vector $(-y_i' - c_i)_{i=1\ldots N}$. The pointing error can thus be minimized by applying a translation to one of the images. Note that the median is robust against outliers, thus this correction procedure works well even in the presence of false matches.

3.2 Roll, Pitch and Yaw Angles

If we assume that the scene is located at infinity with respect to the satellite, an error in the sensor attitude measurement can be modeled in image space as a translation composed with a rotation. Therefore we have

$$f(\mathbf{x}) = TR\mathbf{x}$$

where R is a rotation and T a translation:

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

3.2.a) Approximation of the Rotation

To correct the pointing error in the rectified setting we find R^* and T^* minimizing the pointing error defined as follow :

$$(R^*, T^*) = \underset{R,T}{\operatorname{argmin}} \sum_{i=1}^{N} d(TR\boldsymbol{x_i'}, F\boldsymbol{x}_i)^2 \tag{2}$$

As in [?]

After rectification, the horizontal line $F\boldsymbol{x}_i$ can be written, in homogeneous coordinates, as

$$F\boldsymbol{x}_i = \begin{bmatrix} 0 & 1 & y_i \end{bmatrix}$$

With these notations, if the model now includes a rotation R, for each point correspondence $(\boldsymbol{x}_i, \boldsymbol{x}_i')$, we have

$$e(\mathbf{x}_i, \mathbf{x}_i') = d(TR\boldsymbol{x}_i', \mathrm{epi}_{uv}^{\boldsymbol{x}_i}(R)) = d(TR\boldsymbol{x}_i', F\boldsymbol{x}_i) = |x_i' \sin\theta + y_i' \cos\theta + t - y_i|$$

where θ is the angle of the rotation.

We can consider that θ is small enough compare to 2π. In fact on satellites such as Sentinel or Pleiades (the precision of the yaw angle is around 50 μrad. Even if Planet's sensor would be 100 less precise, we will still have an yaw angle less than 1 degree. With this approximation, we can write:

$$e(\mathbf{x}_i, \mathbf{x}_i') = |x_i'\theta + y_i' + t - y_i|$$

Correcting the global pointing error is to minimize:

$$(\theta^*, t^*) = \underset{\theta,t}{\operatorname{argmin}} \sum_{i=1}^{N} d(TR\boldsymbol{x_i'}, F\boldsymbol{x}_i)^2 = \sum_{i=1}^{N} (x_i'\theta + y_i' + t - y_i)^2$$

We can write this optimization problem as :

$$X^* = \underset{X}{\operatorname{argmin}} \|AX + b\|^2$$

where

$$A = \begin{pmatrix} x'_1 & 1 \\ x'_2 & 1 \\ \vdots & \vdots \\ x'_n & 1 \end{pmatrix} \quad X = \begin{pmatrix} \theta \\ t \end{pmatrix} \quad b = \begin{pmatrix} y'_1 - y_1 \\ y'_2 - y_2 \\ \vdots \\ y'_1 - y_1 \end{pmatrix}$$

The solution of this optimization problem is the solution of the normal equation

$$A^T AX = -A^T b$$

Here,

$$A^T A = \begin{pmatrix} \sum_{i=1}^{N} x_i'^2 & \sum_{i=1}^{N} x'_i \\ \sum_{i=1}^{N} x'_i & N \end{pmatrix}$$

So $A^T A$ is inversible, and the optimal correction in the rectified images is:

$$\begin{pmatrix} \theta^* \\ t^* \end{pmatrix} = X^* = -(A^T A)^{-1} A^T b$$

3.2.b) Solving Numerically an Optimization Problem

To correct the pointing error in the rectified setting we find R^* and T^* minimizing the pointing error defined as follow :

$$(R^*, T^*) = \underset{R,T}{\operatorname{argmin}} \sum_{i=1}^{N} d(RT\boldsymbol{x_i'}, F\boldsymbol{x}_i) \tag{3}$$

Since we are not able to solve this optimization problem analytically, we solve it numerically thanks to the L-BFGS algorithm. As input are the correspondences on the images and the affine approximation matrix and the optimization algorithm outputs the optimal angle $\theta \in [0, 2\pi]$ corresponding to the error on the yaw and a translation $t \in \mathbb{R}$ corresponding to a translation in the normal direction w.r.t. the epipolar lines.

3.2.b).1 Limited-memory BFGS

Limited-memory BFGS is an optimization algorithm in the family of quasi-Newton methods that approximates the BFGS algorithm using a limited amount of computer memory. In our case we do not need to compute the gradient analytically since the algorithm approximates it using finite difference. We use this algorithm as a black box to solve the minimization problem.

4 Planet data

4.1 Structure

- data
  - s03_20161003T161107Z
    - panchromatic
      - s03_20161003T161107Z_pan_d1_0001_rpc.txt
      - s03_20161003T161107Z_pan_d1_0001.tif
    - pansharp
      - s03_20161003T161107Z_pansharp_bgrn_d1_0001_rpc.txt
      - s03_20161003T161107Z_pansharp_bgrn_d1_0001.tif
  - s03_20161003T161148Z
    - panchromatic
      - s03_20161003T161148Z_pan_d1_0001_rpc.txt
      - s03_20161003T161148Z_pan_d1_0001.tif
    - pansharp
      - s03_20161003T161148Z_pansharp_bgrn_d1_0001_rpc.txt
      - s03_20161003T161148Z_pansharp_bgrn_d1_0001.tif
  - s03_20161003T161231Z
    - panchromatic
      - s03_20161003T161231Z_pan_d1_000_rpc.txt
      - s03_20161003T161231Z_pan_d1_0001.tif
    - pansharp
      - s03_20161003T161231Z_pansharp_bgrn_d1_0001_rpc.txt
      - s03_20161003T161231Z_pansharp_bgrn_d1_0001.tif

4.2 Geolocalisation

A Rotations and translation

In this part, we consider any point $(x, y) \in \mathbb{R}^2$. As translation and rotation do not commute, we will consider two cases: rotation followed by translation and translation followed by rotation, with R denoting the rotation of angle θ and T denoting the translation of vector (t_x, t_y). Naming f the resulting function.

$$
\begin{aligned}
T &= \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \\
R &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

A.1 Rotation followed by a translation

This happens when correcting the pointing error first with the rotation.

$$
\begin{aligned}
f(\boldsymbol{x}) &= TR\boldsymbol{x} \\
&= \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta x - \sin\theta y \\ \sin\theta x + \cos\theta y \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \cos\theta x - \sin\theta y + t_x \\ \sin\theta x + \cos\theta y + t_y \\ 1 \end{pmatrix}
\end{aligned}
$$

A.2 Translation followed by a rotation

This happens when correcting the pointing error first with the translation

$$
\begin{aligned}
f(\boldsymbol{x}) &= RT\boldsymbol{x} \\
&= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \cos\theta(x + t_x) - \sin\theta(y + t_y) \\ \sin\theta(x + t_x) + \cos\theta(y + t_y) \\ 1 \end{pmatrix}
\end{aligned}
$$

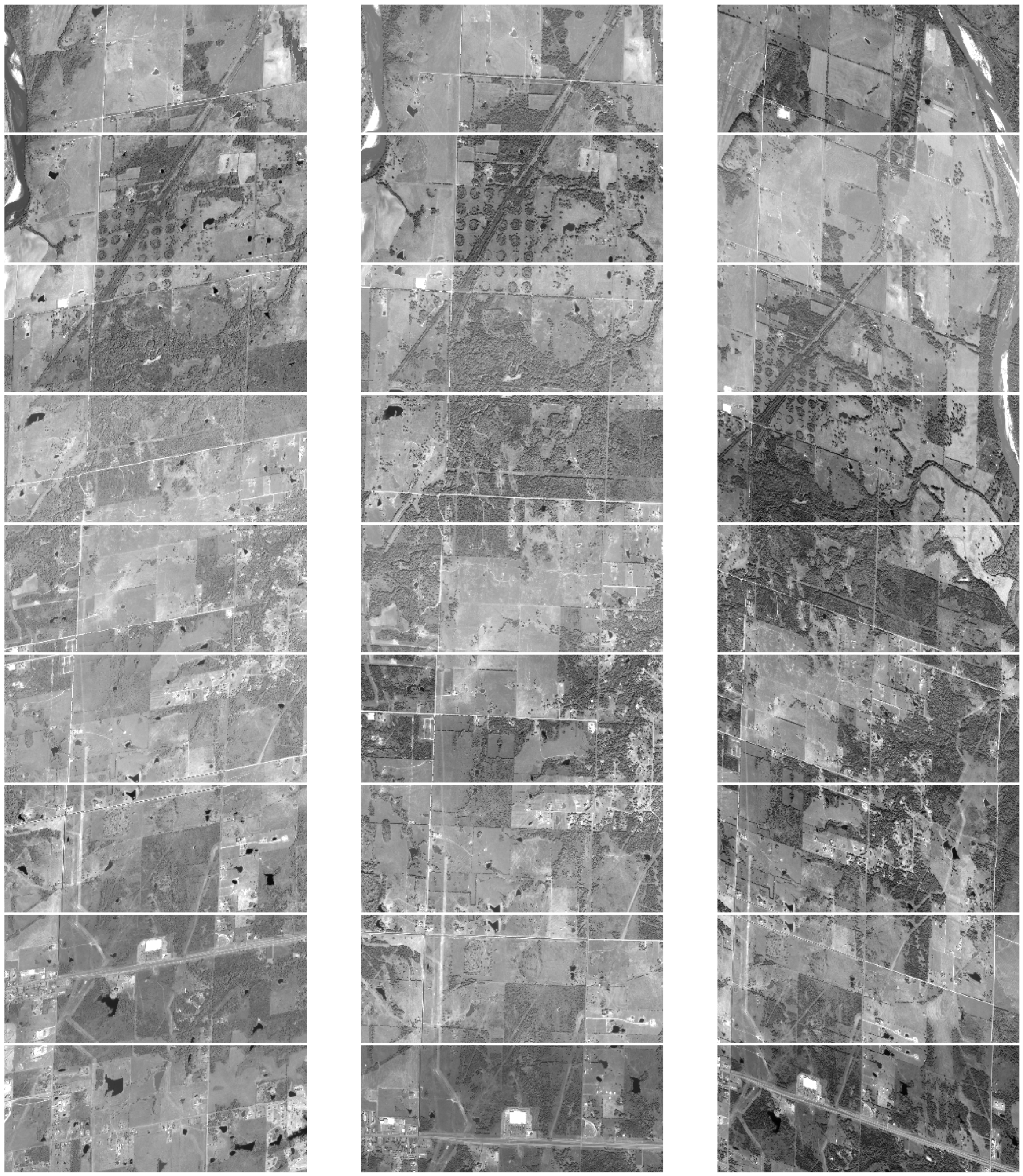

Figure 4: Mosaic of d1. From left to right: 1Z, 7Z, 8Z

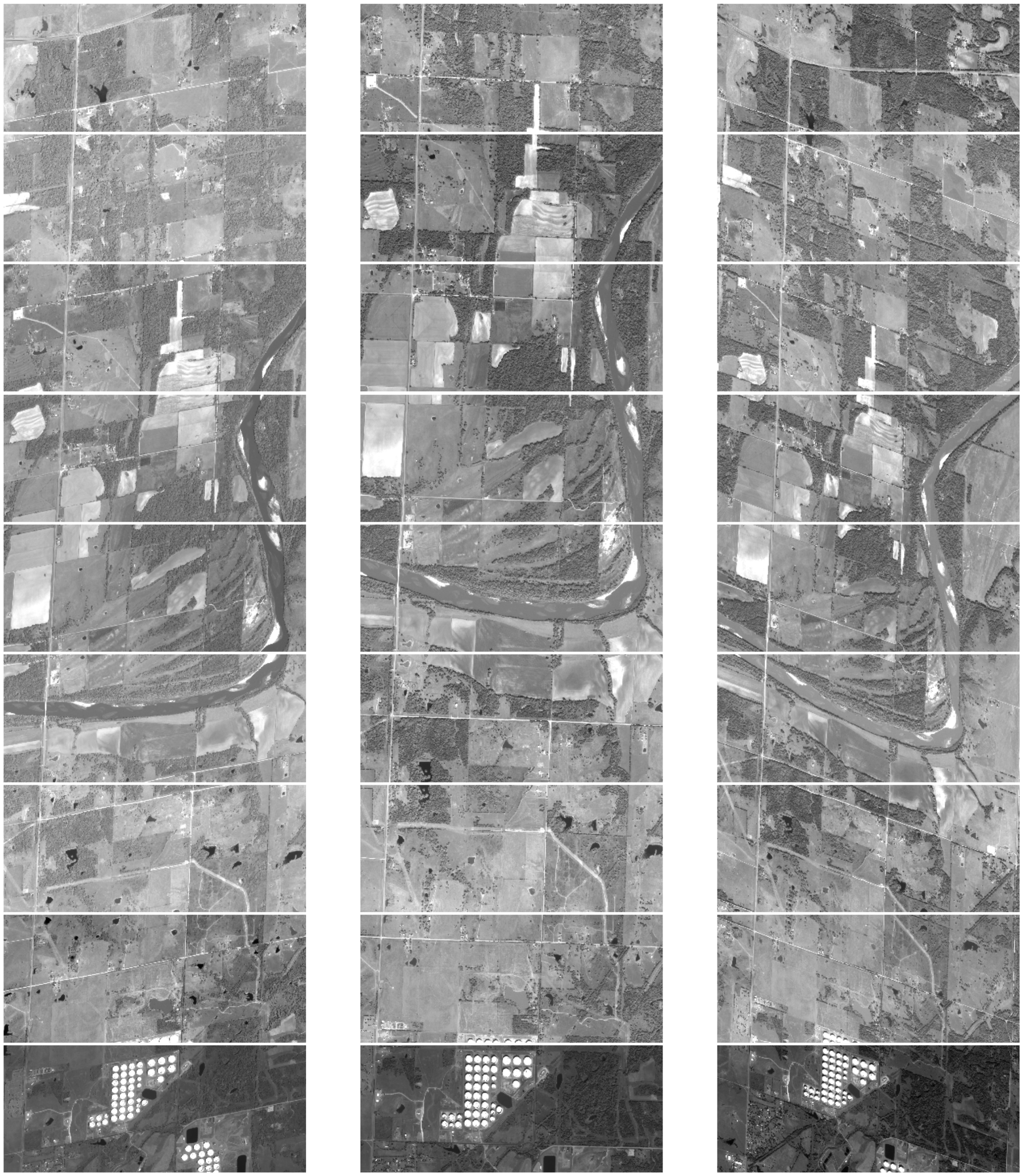

Figure 5: Mosaic of d2. From left to right: 1Z, 7Z, 8Z

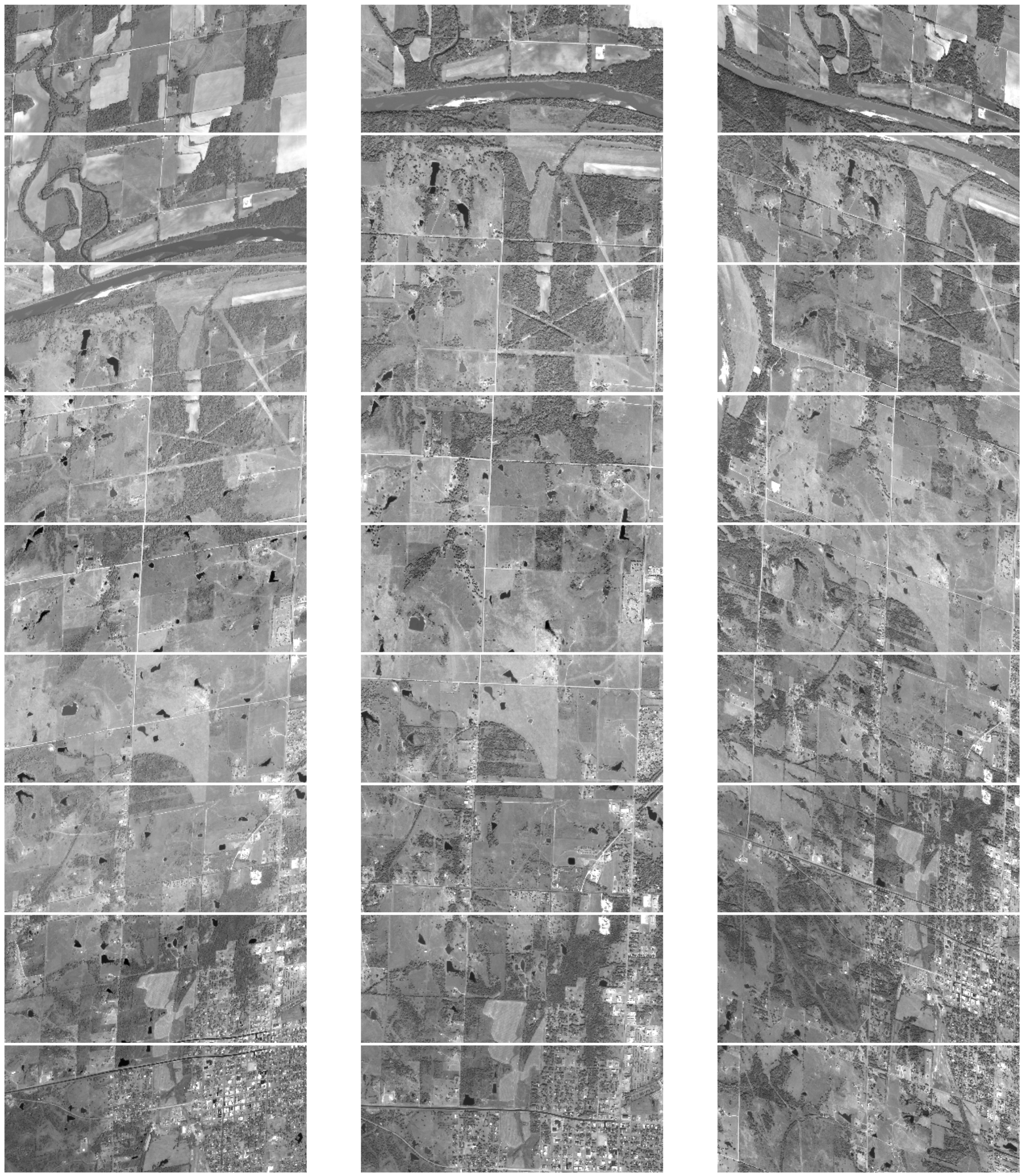

Figure 6: Mosaic of d3. From left to right: 1Z, 7Z, 8Z