Renewal of the Cadastral Map of The Netherlands, an iterative adjustment approach

Frank van den HEUVEL[1], Juriaan LUCASSEN[2], Mark van den BROEK[2], Lense SWAENEN[2], Gerbrand VESTJENS[1], Wim FLORIJN[1], Stefan BUSSEMAKER[1], Eric HAGEMANS[1]
[1] Dutch Cadastre, [2] Sioux Mathware, The Netherlands

Key words: cadastral map, geometric quality, network adjustment, statistical testing

SUMMARY

The current positional accuracy of the Dutch cadastral map is considered not to be sufficient for future use. Therefore, the Cadastre started a project aiming at renewal of the cadastral map. The information needed is available in the form of more than 5 million historical field sketches containing the original survey information. Software has been developed to semi-automatically read and vectorize these field sketches. Large survey networks are set up and adjusted using the collected measurements. The goal of the adjustment and testing is the validation of this information before it is used for improving the current cadastral map. The focus of this paper is on the last step of this process: an iterative approach of adjustment in which the cadastral map is improved step by step by processing sets of validated measurements. Currently, the complete process for the production of this so-called reconstruction map is being tested. Preliminary results of this test are presented.

The approach adopted for the production of the reconstruction map is based on the Delft method of testing where quality control is performed in all steps of the process. We focus on the calculation of the improved geometry of the cadastral map. For this final step in the production process we developed dedicated adjustment software in corporation with Sioux Technologies. This software allows to efficiently update the cadastral map in an iterative way by connecting two point fields while accounting for the precision of both point sets represented by their full covariance matrices. As adjustments with full covariance matrices are very computation-expensive, it is obvious that this adjustment cannot be performed for the whole country in one calculation; we therefore propose the iterative approach presented in this paper. Even with this approach, however, processing speed is still one of the main challenges.

Renewal of the Cadastral Map of The Netherlands, an iterative adjustment approach

Frank van den HEUVEL[1], Juriaan LUCASSEN[2], Mark van den BROEK[2], Lense SWAENEN[2], Gerbrand VESTJENS[1], Wim FLORIJN[1], Stefan BUSSEMAKER[1], Eric HAGEMANS[1]
[1] Dutch Cadastre, [2] Sioux Mathware, The Netherlands

Key words: cadastral map, geometric quality, network adjustment, statistical testing

1. INTRODUCTION

The goal of the research project “Rebuilding the Cadastral Map of The Netherlands” is to build the so-called “reconstruction map” from the survey measurements that have been gathered since the start of the Dutch Cadastre in 1832 (Hagemans, et al., 2021). These survey measurements have been registered in more than 5 million field sketches that all have been scanned several years ago. An example of a part of a field sketch is shown in Figure 1.

Figure 1: Sample of a part of a historic field sketch.

The reconstruction map is the improved version of the cadastral map and will replace the official cadastral map in the future. A schematic overview of the envisaged process is depicted in Figure 2 and consists of three main steps:

1. Collecting the historic survey measurements by vectorizing the field sketches
2. Validating the measurements by building and adjusting survey networks using the vectorized measurements
3. Improving the cadastral map using networks built from validated survey measurements

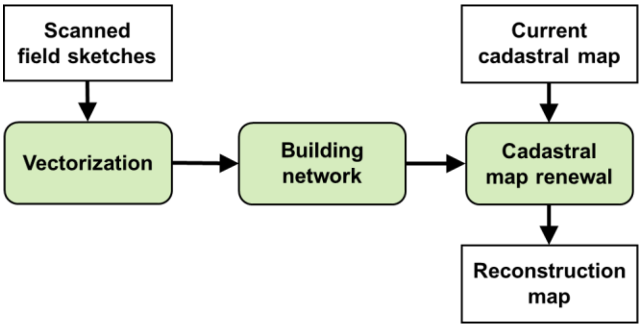

Figure 2: Overview of the approach to renewal of the cadastral map.

In the next section we discuss these steps in more detail. In section 3 the focus is on network adjustment and validation of the vectorized measurements by applying the Delft method of adjustment and testing (Teunissen, 2006). Section 4 describes the method for updating the cadastral map, in particular the mathematical model adopted, implementation aspects, and the statistical testing involved. Performance is assessed and verified with experiments of different size and nature as presented in section 5. Conclusions are drawn in section 6.

2. OVERVIEW OF THE APPROACH

2.1 Vectorizing the field sketches

The first step in the process of cadastral map renewal is the vectorization of over 5 million scanned field sketches that contain the historic measurements of the legal boundaries. In Figure 3 a sample field sketch is shown, with the vectorization result in overlay on the right. Shown is the oldest type of sketch. In the last decades sketches have a different form due to the use of modern techniques applied in cadastral surveying, such as the use of total stations, Global Navigation Satellite Systems (GNSS), and digital mapping tools. As this information is often in digital form, vectorization effort needed for more recent field sketches is limited.

Figure 3: Partial field sketches, with vectorization result in overlay (right).

That it is feasible to extract the original survey measurements from the many millions of historic survey documents or field sketches is a credit to the developments in artificial intelligence in recent years. Artificial intelligence is essential in automating the digitization of the field sketches. See (Franken, et al., 2021) for more details on the vectorization process and the machine learning involved.

2.2 Building geodetic networks

The measurements in the field sketches build a survey network for each field sketch. These networks are adjusted, and statistical testing is used to find possible errors in the vectorization (see section 3). However, the probability of finding errors (reliability) is low for many measurements as the redundancy of the individual survey networks is relatively low. Therefore, larger geodetic networks are constructed in 2 steps:

1. Finding the location of the field sketches
2. Linking them together by identifying common points.

Firstly, each sketch is positioned on the national large-scale base map of the Netherlands (BGT), and linked to points of the map, often corners of buildings. Of course this is only possible for buildings that have not been demolished or modified since the mapping took place. Secondly, field sketches are linked by identifying identical points in different sketches.

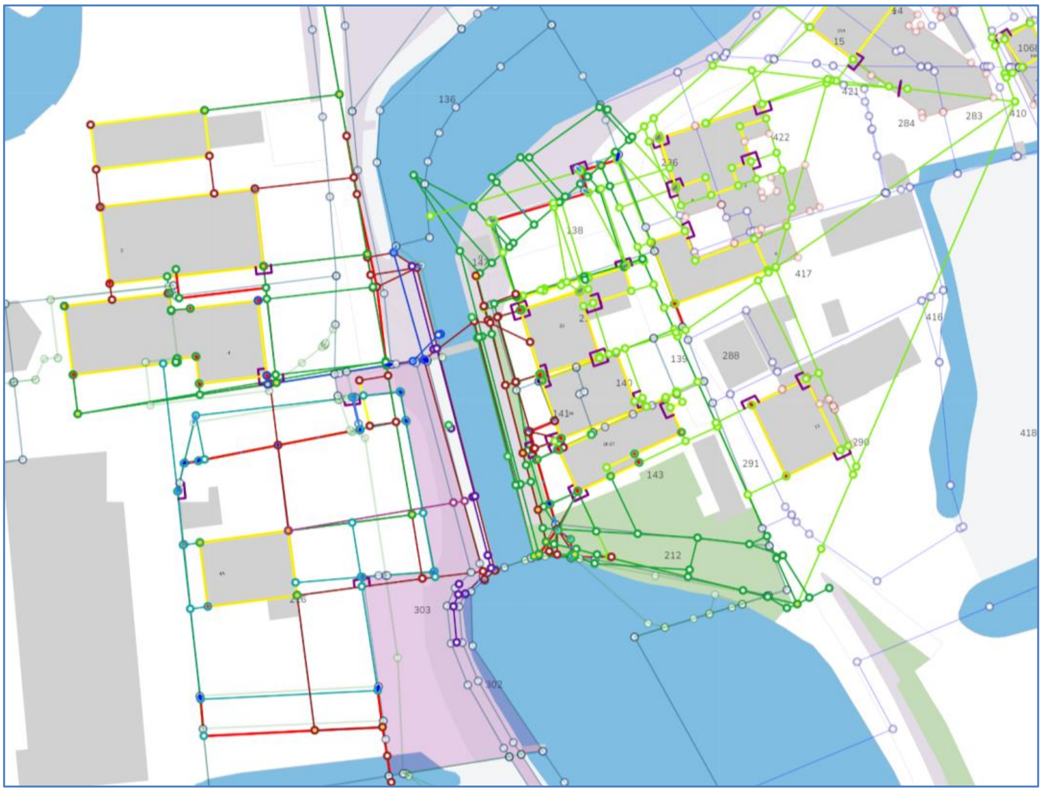

Figure 4: Field sketches linked together and linked to the large-scale base map of The Netherlands. Each sketch has a different colour, building outlines are in yellow, new legal boundaries in red.

The links are often found in points of the base map, and in cadastral survey marks that have been identified in the vectorization. The linking of field sketches to its neighbours is performed for each field sketch, so for each sketch an extended survey network is established, adjusted, and tested.

The linking process is performed in a multi-user environment with all data stored centrally. As a result, no double work is done, and after investigating possible links for each field sketch the geodetic network is complete: in principle one nation-wide network is established using the vectorized measurements of all field sketches. An example of a network of connected field sketches is shown in Figure 4.

2.3 Cadastral map renewal

To be able to improve the cadastral map, the linking process described in the last section is extended with linking to the cadastral map. There is no fundamental difference with the linking to the large-scale base map: in the adjustment and quality control stage the links with the cadastral map are tested just like the links to the large-scale base map. However, the focus for linking to the large-scale base map is on points, while focus for linking the field sketches to the cadastral map is on lines. The reason is that frequently the corner points of parcels are not measured directly but constructed as an intersection of partly measured boundaries.

Before starting the coordinate computation for updating the cadastral map, quality control is performed through integral adjustment and testing (Figure 5). In the quality control step geodetic networks are setup optimized for this goal. After validation of all information gathered, validated field sketches are stored, including their links to neighbouring sketches and links with the two maps.

Validated field sketch networks are combined into large-scale networks for the final step: the coordinate computation of the reconstruction map. Now field sketches are selected to facilitate building the reconstruction map. One of the requirements is that every field sketch is used once for building the reconstruction map. The quality control step is the topic of section 3, while coordinate computation is elaborated upon in section 4.

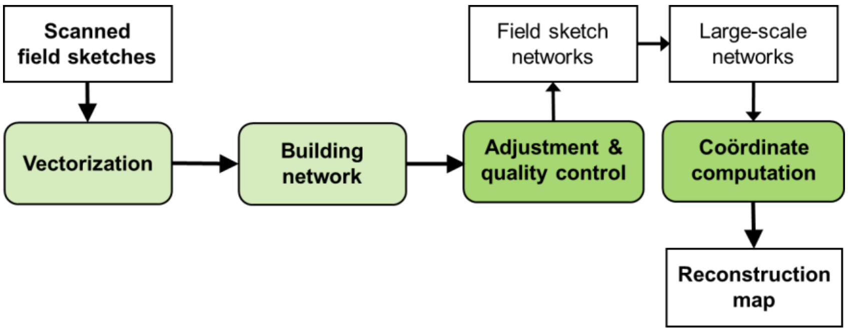

Figure 5: The renewal process of the cadastral map in more detail.

3. NETWORK ADJUSTMENT AND QUALITY CONTROL

3.1 Validation of individual field sketches

For the adjustment of the measurements of individual field sketches no control information is available, and therefore a so-called free network adjustment is to be performed. However, because of the limited redundancy, some points of the network cannot be positioned based on the measurements alone. Therefore, all points are processed as control with a very low a priori precision of 20m standard deviation. This ensures that adjustment does not fail due to a (local) lack of observations. After acceptance of the statistical tests, i.e. the overall F-test and the w-tests for the observations (Teunissen, 2000), the field sketch is accepted for the next step of processing: positioning and linking. In Figure 6 an accepted field sketch is shown next to its vectorization result in the form of a so-called on-scale drawing. The field sketch itself is not to scale. More information on the adjustment and testing for validation of field sketches can be found in (Van den Heuvel, et al., 2021).

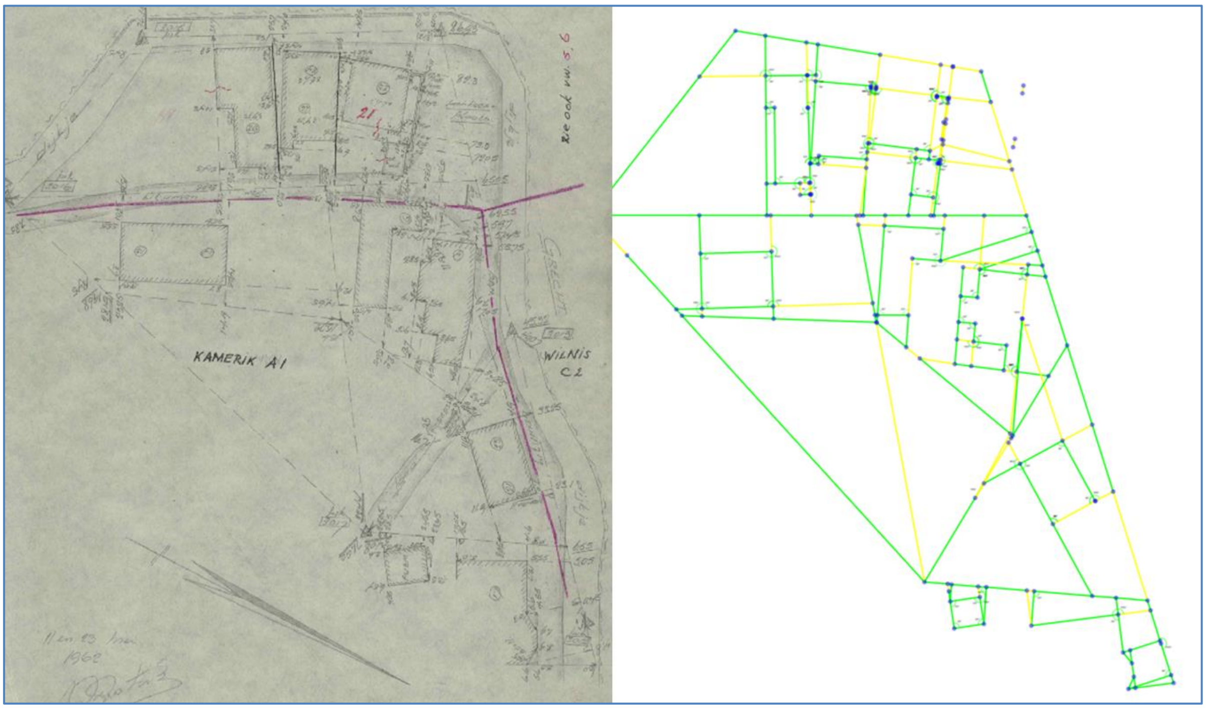

Figure 6: Field sketch (left) and on-scale drawing of associated network (right). Yellow edges indicate measurements without redundancy, green lines relate to measurements for which the w-test is accepted.

3.2 Adjustment of large geodetic networks

After linking the field sketches to each other, to the large-scale base map, and to the cadastral map, network adjustments are performed with testing of the observations as the main goal. The adjustment and testing approach applied is the so-called “Delft school” of mathematical geodesy (Teunissen, 2006) (Baarda, 1968). Firstly, adjustment is done for each field sketch with a combination of its measurements and those of the neighbouring field sketches linked. Secondly, large-scale adjustments are performed for combinations of around 200 field sketches, a network size that is processed within minutes. It is feasible to process much larger networks with the software DynAdjust (Fraser, et al., 2022), however, with the proposed approach for map renewal there is no benefit in processing very large networks.

Two large-scale adjustments are executed: a “free” network adjustment (for all points an a priori standard deviation of 20m is used), and a connected network adjustment with control points. In the first adjustment the links to the large-scale base map and the links to the cadastral map are not used: the focus is on validation of the measurements and the links between the field sketches. In the second adjustment the links to the maps are added and then all information is combined and tested in one adjustment. In both adjustments the links between field sketch points are treated as observations and can thus be tested. In Figure 7 part of the output of such as an adjustment is shown.

The goal of the large-scale network adjustments is the validation of field sketch measurements and links. On the borders of these networks the redundancy can be low. Therefore, networks for quality control will be setup in overlap in order to be able to test each observation with maximum redundancy. For coordinate computation new networks are created that do not overlap as each field sketch is to be used only once for updating the reconstruction map. Furthermore, the network for coordinate computation is connected to high quality reference points only, i.e. points that have been measured with GNSS and transformed to the national coordinate system.

Statistics:

F-test:	0.11
Number of stations:	491
Number of observations:	2235
Degrees of freedom:	1247

ADVANCED

Observations

Type	Record	Station	Target	Value	W-test	Est. err	GW	Resid	Stdev	Redun
CH	NKP01G00033_t131_791534	p_NKP01G00033_p346_1324333	p_NKP01G00033_p271_1324424 p_NKP01G00033_p3_1324420	57.45	3.28	14.64	0.48	0.019	0.03	5.04
PL	NKP01G00033_t131_791534	p_NKP01G00033_p346_1324333	p_NKP01G00033_p271_1324424 p_NKP01G00033_p3_1324420	0.00	3.28	6.93	0.19	0.033	0.02	22.44
TD	NKP01G00033_t122_791547	p_NKP01G00033_p34_1324371	p_NKP01G00033_p346_1324333	2.10	3.28	7.72	0.17	0.025	0.02	18.10
CH	NKP01G00033_t44_791558	p_NKP01G00033_p70_1324433	p_NKP01G00033_p159_1324453 p_NKP01G00033_p72_1324358	2.65	-2.85	-5.54	0.17	-0.031	0.02	26.53
AN	NKP01G00033_s37_405483	p_NKP01G00033_p132_1324458	p_NKP01G00033_p163_1324411 p_NKP01G00033_p205_1324314	100.00	-2.71	-5.12	3.27	-0.601	0.42	27.99
CH	NKP01G00033_t46_791542	p_NKP01G00033_p131_1324289	p_NKP01G00033_p39_1324431 p_NKP01G00033_p71_1324459	3.60	2.67	6.57	0.21	0.023	0.02	16.56
CL	NKP01G00033_s82_405530	p_NKP01G00033_p132_1324458	p_NKP01G00033_p131_1324289 p_NKP01G00033_p205_1324314		2.67	4.31	0.14	0.035	0.02	38.45
CH	NKP01G00067_t45_781918	p_NKP01G00067_p33_1304962	p_NKP01G00067_p19_1304957 p_NKP01G00067_p5_1304964	2.81	2.63	11.40	0.38	0.013	0.02	5.32
AN	NKP01G00067_s1_400570	p_NKP01G00067_p33_1304962	p_NKP01G00067_p20_1304956 p_NKP01G00067_p3_1304972	100.00	-2.63	-12.11	4.03	-0.120	0.21	4.70
PL	NKP01G00067_t4_781919	p_NKP01G00067_p3_1304972	p_NKP01G00067_p19_1304957 p_NKP01G00067_p5_1304964	0.00	-2.53	-12.71	0.44	-0.011	0.02	3.96
PL	NKP01G00033_t167_791619	p_NKP01G00033_p184_1324419	p_NKP01G00033_p322_1324324 p_NKP01G00033_p91_1324337	0.00	-2.44	-3.86	0.14	-0.033	0.02	40.09
SV	p_link_242385	p_NKP01G00088_p44_1324137	p_NKP01G00134_p12_1318121		2.38	7.15	0.12	0.008	0.01	11.08
TD	NKP01G00033_t237_791694	p_NKP01G00033_p322_1324324	p_NKP01G00033_p311_1324454	0.50	2.33	7.82	0.25	0.012	0.02	8.85
CH	NKP01G00067_t13_781912	p_NKP01G00067_p19_1304957	p_NKP01G00067_p21_1304982	3.75	2.31	4.78	0.18	0.024	0.02	23.46

Figure 7: Screenshot of adjustment output: a record for each observation containing w-test, estimated error, Minimal Detectable Bias (GW), residual, standard deviation, and redundancy. The blue links will show the user the measurement in the field sketch, or zoom to the station on the base map.

4. COORDINATE COMPUTATION: AN ITERATIVE APPROACH

4.1 Introduction

The integration of the validated field sketches into the reconstruction map follows the procedure outlined in section 2.3. The integration is realized through an adjustment that connects two point fields: the first point field results from a large-scale adjustment of the data of around 200 field sketches (we name this point field: RV), the second point field is the reconstruction map (RK), see Figure 8. RK is initiated with the current cadastral map. For both point fields a full covariance matrix is processed: for RV this covariance matrix is computed in the large-scale adjustment of the field sketch data, for RK an artificial covariance matrix is setup based on limited information on the quality of the current cadastral map.

After the connection of the two point fields, the result is an updated set of RK coordinates and an updated covariance matrix, which is used as a starting point for the next connection of RV points to the RK points. This is an iterative process which implies that we can adjust the reconstruction map piecewise using a succession of smaller sets of RV points. Furthermore, it allows sets of RV points to overlap geographically, so different sets of RV points can affect the same RK points. Other advantages of this approach are its scalability and the built-in interpolation of RK points that are not linked to RV points. This interpolation is controlled by the artificial covariance matrix of RK that reflects an adjustable distance dependent correlation between the coordinates of RK.

The connection is made by defining relations between the two point fields. Three types of relations are distinguished:

1. Point-point (PP) relations: two points represent the same point in the terrain;
2. Point-line (PL) relations, which describe a case in which a point of the RV should lie on a line defined by two points in RK, a so-called collinearity relation;
3. Line-line (LL) relations, which means that 2 points in RK and 2 points in RV should all lie on the same line. LL relations are implemented as 2 separate PL relations and will therefore not be discussed separately.

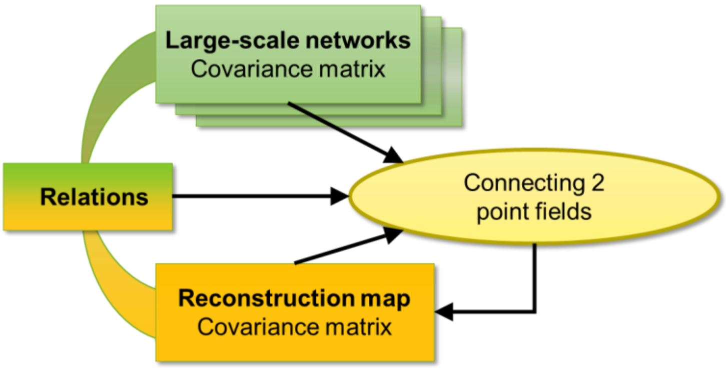

Figure 8: Iterative updating of the reconstruction map by connecting 2 point fields.

4.2 The mathematical model

In this section we will describe two ways in which we can interpret the problem on a more mathematical basis. Let's start with some definitions: there are observations $\boldsymbol{Y}$ and those coordinates for which we want to solve: $\boldsymbol{X}$. Their relationship is described as

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{r},$$

where $\boldsymbol{r}$ is the residual vector, and $\boldsymbol{f}$ is a vector valued function that describes the expected relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$. In general, we are looking to solve the following (non-linear) least-squares problem for $\boldsymbol{X}$: (Nocedal & Wright, 2006; Polman & Salzmann, 1996)

$$\min_{\mathrm{X}} \boldsymbol{\Phi} = \min_{\boldsymbol{X}} \boldsymbol{r}^T \sigma_y^{-1} \boldsymbol{r},$$

where σ_y is the covariance matrix of the observations, and $\boldsymbol{\Phi}$ the cost function. The normal-equations to be solved iteratively to find the minimum of this cost function are then given by:

$$\boldsymbol{J}^T \sigma_y^{-1} \boldsymbol{J}\, \Delta\boldsymbol{X} = \boldsymbol{J}^T \sigma_y^{-1} \Delta\boldsymbol{Y},$$

where $\boldsymbol{\Delta X}$ is the step taken during an iteration and $\boldsymbol{J}$ is the Jacobian $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{X}}$. The updated covariance matrix is given by:

$$\sigma_x^{-1} = \boldsymbol{J}^T \sigma_y^{-1} \boldsymbol{J}.$$

View 1

To make sure the PP and PL relations are satisfied exactly we use Lagrange multipliers. (Nocedal & Wright, 2006) We define $\boldsymbol{X} = [\boldsymbol{X}_{\mathbf{RK}}, \boldsymbol{X}_{\mathbf{RV}}]$ and $\boldsymbol{Y} = [\boldsymbol{Y}_{\boldsymbol{RK}}, \boldsymbol{Y}_{\boldsymbol{RV}}]$, and:

$$\sigma_y = \begin{pmatrix} \sigma_{RK} & 0 \\ 0 & \sigma_{RV} \end{pmatrix}$$

If we introduce Lagrange multipliers $\boldsymbol{\lambda}_{PP}$ and $\boldsymbol{\lambda}_{PL}$ to the cost-function the result is:

$$\boldsymbol{\Phi} = \mathbf{r}^{\boldsymbol{T}} \boldsymbol{\sigma}_{\boldsymbol{y}}^{-\mathbf{1}} \mathbf{r} + \boldsymbol{\lambda}_{\mathbf{PP}}^{\boldsymbol{T}} \boldsymbol{r}_{\mathbf{PP}} + \boldsymbol{\lambda}_{\mathbf{PL}}^{\boldsymbol{T}} \boldsymbol{r}_{\mathbf{PL}},$$

with $\boldsymbol{r} = [\boldsymbol{r}_{\mathbf{RK}}, \boldsymbol{r}_{\mathbf{RV}}]$ and the individual components:

$$\boldsymbol{r}_{\mathbf{RK}} = \boldsymbol{Y}_{\mathbf{RK}} - \boldsymbol{X}_{\mathbf{RK}},$$
$$\boldsymbol{r}_{\mathbf{RV}} = \boldsymbol{Y}_{\mathbf{RV}} - \boldsymbol{X}_{\mathbf{RV}},$$
$$\boldsymbol{r}_{\mathbf{PP},\boldsymbol{i}} = \boldsymbol{X}_{\mathbf{RK,j}} - \boldsymbol{X}_{\mathbf{RV,k}},$$

with i the i^{th} point-point relation between point j of the RK and point k of the RV. And last,

$$\mathbf{r}_{\mathbf{PL}.\boldsymbol{i}} = (\boldsymbol{X}_{\mathbf{RK,j,x}} - \boldsymbol{X}_{\mathbf{RK,k,x}})(\boldsymbol{X}_{\mathbf{RK,k,y}} - \boldsymbol{X}_{\mathbf{RV,l,y}}) - (\boldsymbol{X}_{\mathbf{RK,k,x}} - \boldsymbol{X}_{\mathbf{RV,l,x}})(\boldsymbol{X}_{\mathbf{RK,j,y}} - \boldsymbol{X}_{\mathbf{RV,k,y}}),$$

with i the i^{th} point-line relation between the line through points j and k of the RK and point l of the RV (where we now add additional subscripts to indicate which coordinate component is required). This is a way to formulate point-line distances. Typically, however, when computing-line distances there is also a denominator involved. However, this does not affect the equality and leads to more complex formula Note that the PL relations are the only source of non-linearity in the coordinate computation. The normal equations for this cost function should include the partial derivatives with respect to the Lagrange multipliers. By solving these normal equations iteratively, we will find a solution where the relations are satisfied exactly. This view, however, has some disadvantages: if any of the constraints are linearly dependent the system becomes singular. As this happens regularly with realistic data, we chose to implement View 2 instead.

View 2

In this view or formulation of the mathematical model we solve for both the new coordinates of the RV and the RK by including the relations as observations such that $\boldsymbol{X} = [\boldsymbol{X}_{\mathbf{RK}}, \boldsymbol{X}_{\mathbf{RV}}]$, $\boldsymbol{Y} = [\boldsymbol{Y}_{\mathbf{RK}}, \boldsymbol{Y}_{\mathbf{RV}}, \boldsymbol{Y}_{\mathbf{PP}}, \boldsymbol{Y}_{\mathbf{PL}}]$, where $\boldsymbol{Y}_{\mathbf{PP}}, \boldsymbol{Y}_{\mathbf{PL}}$ are the point-point observations and point-line observations (which are equal to zero vectors because they are given by the predefined relations), and $\boldsymbol{r} = [\boldsymbol{r}_{\mathbf{RK}}, \boldsymbol{r}_{\mathbf{RV}}, \boldsymbol{r}_{\mathbf{PP}}, \boldsymbol{r}_{\mathbf{PL}}]$. The covariances for $\boldsymbol{Y}$ are now defined as:

$$\sigma_y = \begin{pmatrix} \sigma_{RK} & 0 & 0 & 0 \\ 0 & \sigma_{RV} & 0 & 0 \\ 0 & 0 & \epsilon\, I & 0 \\ 0 & 0 & 0 & \epsilon\, I \end{pmatrix},$$

with ϵ some small number and I the identity matrix. Because the weighting term for the PP and PL relations are given by the very small number ϵ we force these residuals to be close to zero. If this ϵ is taken small enough this view will yield (approximately) the same solution as view 1. Compared to view 1, we also don't suffer from linearly dependent constraints, as each one has its own unique observation. A further benefit is the fact that the system we must solve during iterative updating of $\boldsymbol{X}$ is significantly smaller because of the absence of nuisance parameters in the form of the Lagrange multipliers of View 1 and should thus be solved faster (see 5.2).

4.3 Implementation aspects: speed and memory usage

The iterative update of $\boldsymbol{X}$ through the solving of the normal equations for $\boldsymbol{\Delta X}$ is done using a custom implementation of a Levenberg-Marquardt algorithm. This is an almost exact copy of the solver used for the large-scale network adjustment (section 3.2). However, one major difference is the sparsity of the problem. Because we now have dense covariance matrices of both RK and RV, we can no longer use the (sparse) Pardiso (PARDISO, sd) solver. Instead, we must now use a dense solver. As the system is positive-definite, we can use Lapack's dposv. The whole solver is implemented in Python, where we make use of a version of NumPy and SciPy (Harris, et al., 2020; Virtanen, et al., 2020) that is linked against Intel's MLK Lapack implementation (Intel, sd).
Because the goal is to work with as large a network as possible, speed and memory consumption are critical, especially in this solver where we are forced to work with dense matrices. It is therefore crucial that we avoid doing unnecessary work. For example:

- Inversion of σ_y^{-1} needs to be done only once, as it is constant during iteration.
- The Jacobian is sparse, and only a few elements change per iteration. This means we can save on memory usage by saving it as a sparse matrix.
- Per iteration only a few normal equations change. By writing out the normal equations explicitly, we find that only a few elements change, defined by the sparse-subsets of the Jacobian. Thus, setting up the normal equations can be done quickly if we reuse the normal equations from the previous iteration and do explicit sparse-sparse multiplications before inserting the updates of the Jacobian in the normal equations.
- The matrices are positive-definite; not only the solver but also the inversion can be sped up using algorithms tailored for positive-definiteness.

However, improvements are still possible. For example, we have not yet implemented additional memory-saving techniques, such as using only upper or lower triangular parts of the matrices (matrices are positive-definite and thus symmetric). Additionally, the current implementation uses CPU-based linear algebra, which can be sped up significantly when transferring it to a GPU (see for example the MAGMA project (Tomov, et al., 2010)). And, last, we could try to find sparse approximations to the normal equations such that we are once again able to use sparse solvers, which come with a considerable speed-gain (PARDISO, sd).

4.4 Statistical testing: 3 different tests for error detection

We have also implemented statistical tests to verify if the input data matches the model. Specifically, we have implemented the F-test (or goodness of fit), the W-test, and the T-test. (Polman & Salzmann, 1996) It is beyond the scope of this paper to discuss these tests further, but in terms of speed and memory usage one needs to be very careful here. These tests usually involve explicit inversion of large matrices, which is the most time-consuming step of the solution process. We have therefore analysed the formulas in detail and optimized the inversion process by reusing information, only inverting submatrices (typically, we only require several elements of the inverted matrices), and utilizing sparsity of these sub-matrices.

Doing this reduced the calculation time by 1-2 orders of magnitude such that calculating the 3 different statistical test is no longer the dominant contribution to the solve time.

5. EXPERIMENTS AND RESULTS

5.1 Introduction

Using the implementation of the solver we ran some realistic test cases to gauge performance. That is the topic of the next section: the performance using quasi-realistic test cases without PL relations to investigate both time and memory consumption as well as scaling behaviour. This should pave the way to further investigations of the behaviour under iterative application of the solver, as well as full integration tests that include vectorization and large-scale adjustments for an automated update of the cadastral map. First results of a test of the complete production chain are presented in section 5.3.

5.2 Performance and scaling

We can subdivide the performance of the solver into three distinct phases (where $n_{RK/RV}$ are the number of RK/RV points):

1. **Loading in and setting up problem**. Including the inversion of the covariance matrix of the observations, this scales approximately as $\max(n_{RK}, n_{RV})^3$, as the 2 point fields are assumed to be uncorrelated.
2. **(Multiple) Levenberg-Marquardt iterations**. Expected to scale as $(n_{RK} + n_{RV})^3$ because this step is dominated by the solving of the system of normal equations.
3. **Calculating statistics and generating output.** Combination of calculating the new covariance matrix and additional output statistics. The scaling here is a little more complicated, but typically it should scale as $(n_{RK} + n_{RV})^3$ because the most time is spend inverting the normal equations to get an updated covariance matrix.

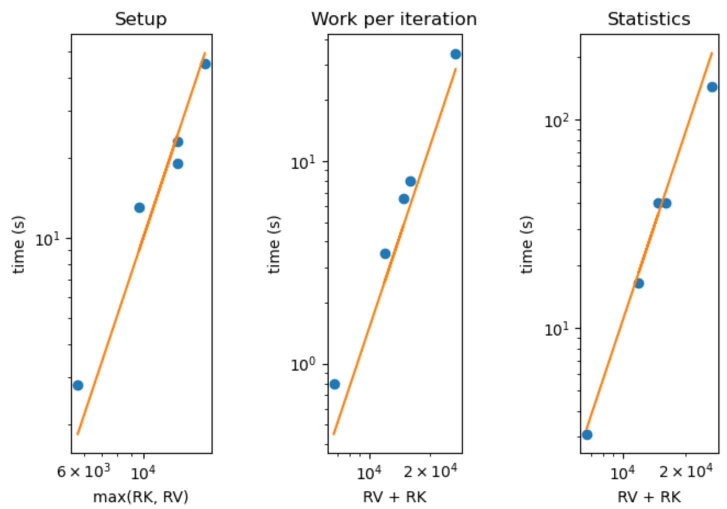

Figure 9. Scaling behavior of the solver. For the three different phases of the solver we plot the time taken as a function of relevant scaling parameters. It includes all data from ***Error! Reference source not found.*** *as well as an additional test case. This a log-log plot, where the orange line indicates what you would expect for cubic scaling.*

We have benchmarked each of these steps for realistic test cases (using PP relations only), and the results are shown in Table 1 and Figure 9, where we indeed observe this cubic scaling of the different terms of the adjustment. The scaling of the memory consumption will scale as $(n_{rk} + n_{rv})^2$ in most normal circumstances.

If we put all of this together (assuming a ratio of 5:1 for n_{rk}: n_{rv}, 5 iterations and as many relations as RV points) we estimate the capabilities of the solver as in Table 2. Note that only the RV points that have a relation with the RK points need to be processed. Normally, many more points have been processed in the large-scale adjustment (section 3.2).

Table 1: Typical performance of the solver on an Intel i9-9880H.

n_{rk}	n_{rv}	n_{pp}	Setup time (s)	Work per iteration (s)	Statistics and output (s)
2842	505	508	2.8	0.8	3.1
6718	669	671	19	6.5	40
4816	1126	1131	13	3.5	16.5
6706	1287	1288	23	8	40

Table 2: Estimated capabilities of the solver on an Intel i9-9880H.

n_{rk}	n_{rv}	n_{pp}	Total time (minutes)	Memory (GB)
4000	800	800	0.5	4.5
5000	1000	1000	1	10
8600	1720	1720	5	28
10800	2160	2160	10	40

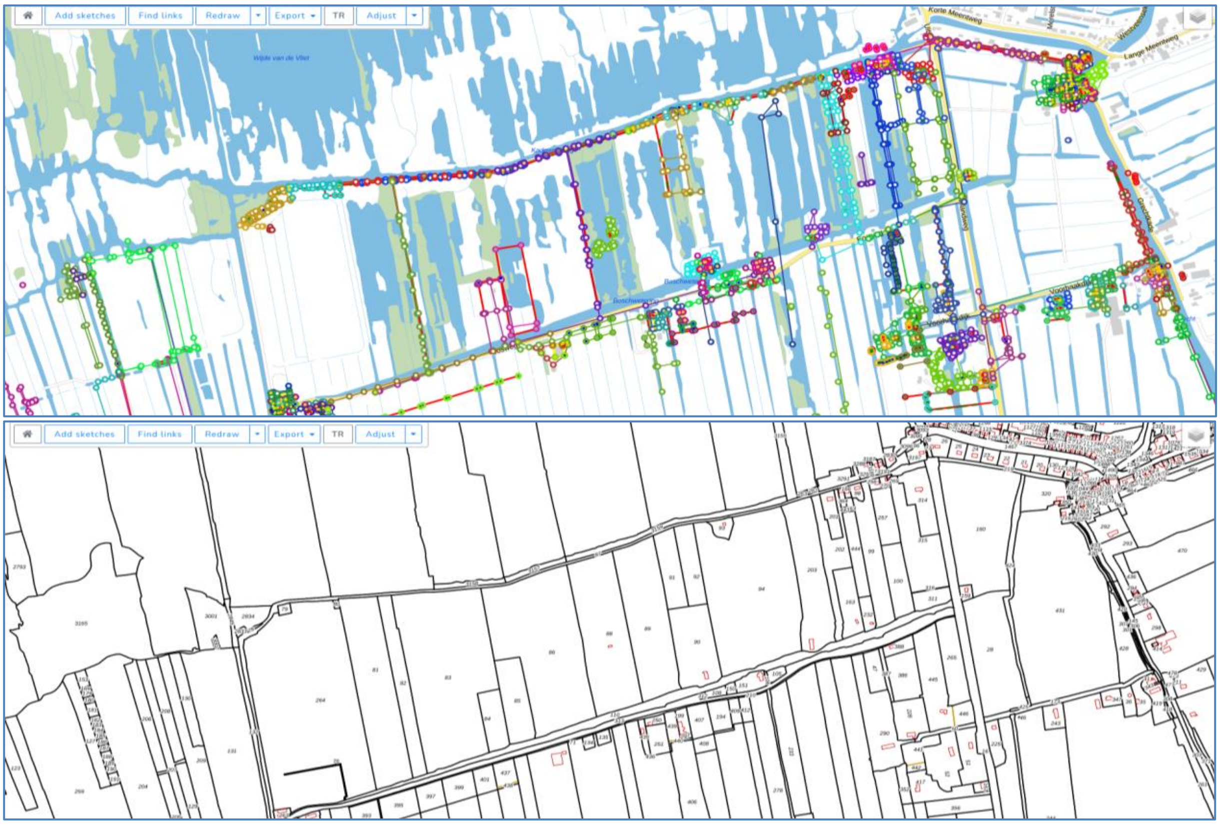

Figure 10: Networks resulting from vectorization plotted on a map (top) and cadastral map. Coloring is per field sketch. Note that not all borders are found in the field sketches: in this area a map renewal project took place about 50 years ago.

5.3 First test of the reconstruction map production process

To test the production process using real data, a small section of the cadastral map was selected in a rural area. The 196 field sketches were vectorized, positioned on the base map, and linked to each other, to the base map, and to the cadastral map. Small field sketch network adjustments, as well as a large-scale adjustment with the data of all field sketches were used to find and correct vectorization and linking errors. An overview of a part of the area and the field sketch networks is shown in Figure 10.

With the results of the large-scale adjustment and the related covariance matrix the point field RV is ready for updating the cadastral map. For the cadastral map (the initial version of the reconstruction map RK) we choose an artificial covariance matrix in the form of a modified Baarda-Alberda matrix (Polman & Salzmann, 1996) with a priori standard deviation for all coordinates of 2m, and a maximum correlation distance of 100m. This standard deviation does not reflect the actual precision of the current map, but minimizes the effect of the current map on the reconstruction map.

The result of the coordinate computation is shown for a few points in Figure 11. The new updated covariance map shows very small uncertainty ellipses where relations between RK and RV are given because the uncertainty of RV is typically small, i.e. at the centimetre level. This then translates to a small uncertainty in the reconstruction map. Because of the distance dependent correlation in the artificial covariance matrix of RK, we find an increase in the radius of the uncertainty ellipse the further away we move from links between RV and RK. In this test correlation is proportional to the inverse of the squared distance.

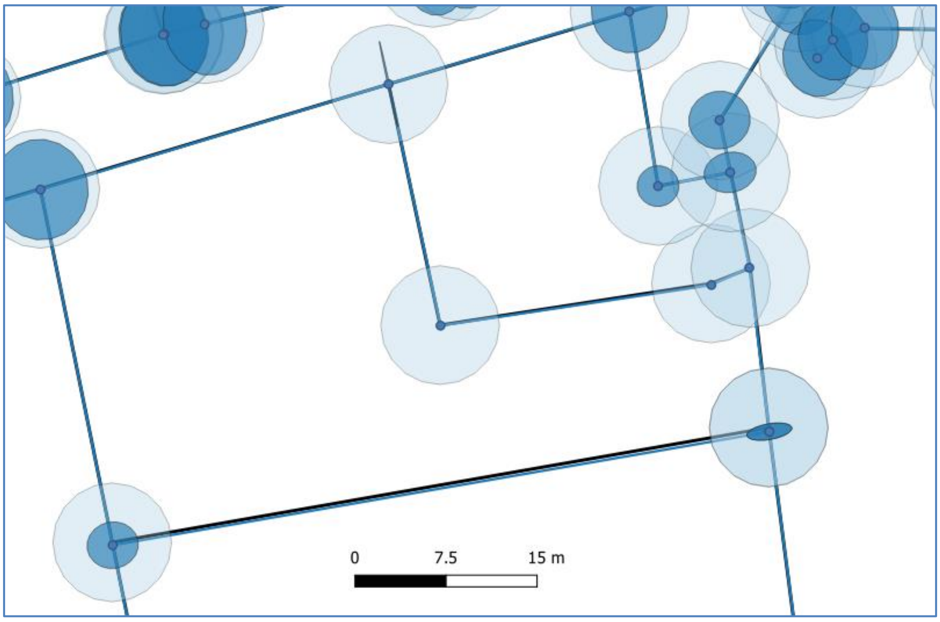

Figure 11. Updated reconstruction map (original cadastral map in black, overlayed by RK in blue). The ellipses show a 95% confidence area and illustrate the gain in precision: light blue circles for the original cadastral map, dark blue ellipses for the reconstruction map. Dark blue ellipses remain large where improvement is limited due to the absence of links between RK and RV.

6. CONCLUSIONS

The paper gives an overview of the process developed for renewal of the cadastral map of The Netherlands with a focus on the least-squares adjustments involved. Two types of adjustment can be distinguished: firstly the adjustment of vectorized survey measurements that facilitate validation of the input data for the map renewal process, and secondly the adjustment needed for the map renewal itself: the coordinate computation of the reconstruction map. The latter type of adjustment is the main topic here and described in more detail, including results of experiments used to verify the envisaged performance in terms of CPU and memory usage. The conclusion is that it is possible to iteratively update parts of the cadastral map with more than 5000 points in minutes.

First results of a test of the complete production process are presented, demonstrating the feasibility of the approach and the potential gain in quality of the improved cadastral map. Furthermore, advantages of the approach are emphasized such as its scalability, and the built-in interpolation of points of the cadastral map for which no measurements were vectorized. Future research aims at the segmentation of the cadastral map of The Netherlands in parts that can be processed in one adjustment, in combination with the selection of parameters for the artificial covariance matrix such as the maximum correlation distance.

References

Baarda, W., 1968. *A testing procedure used in geodetic networks.* Delft: Netherlands Geodetic Commission, Publ. on Geodesy, New Series 2(5).

Franken, J., Florijn, W., Hagemans, E. & Hoekstra, M., 2021. *Rebuilding the cadastral map of The Netherlands, the artificial intelligence solution.* Amsterdam, s.n.

Fraser, R., Leahy, F. & Collier, P., 2022. *DynAdjust User's Guide - version 1.2,* Melbourne, Australia: ICSM open source: github.com/icsm-au/DynAdjust.

Hagemans, E., Busink, R., Grift, J. & Schouten, F., 2021. *Rebuilding the cadastral map of The Netherlands, the overall concpept.* Amsterdam, s.n.

Harris, C. R. et al., 2020. Array programming with Numpy. *Nature,* 585(7825), pp. 357--362.

Intel, n.d. *Math Kernel Library (MKL).* [Online]
Available at: https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html
[Accessed 25 March 2022].

Nocedal, J. & Wright, S. J., 2006. *Numerical Optimization.* New York: Springer.

PARDISO, n.d. *PARDISO.* [Online]
Available at: https://pardiso-project.org/
[Accessed 25 March 2022].

Polman, J. & Salzmann, M. A. eds., 1996. *Handleiding voor de Technische Werkzaamheden van het Kadaster.* Apeldoorn: Kadaster.

Teunissen, P., 2000. *Testing theory, an introduction.* Delft: Delft University Press.

Teunissen, P., 2006. *Network quality control.* Delft: Delft University Press.

Tomov, S., Dongarra, J. & Baboulin, M., 2010. Towards dense linear algebra for hybrid GPU accelerated manycore systems. *Parallel Computing,* 36(5-6), pp. 232--240.

Van den Heuvel, F., vestjens, G., Verkuijl, G. & Van den Broek, M., 2021. *Rebuilding the cadastral map of The Netherlands, the geodetic concept.* Amsterdam, s.n.

Virtanen, P. et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods,* Volume 17, pp. 261--272.

BIOGRAPHICAL NOTES

Frank van den Heuvel is working as a geodetic specialist at the Dutch Cadastre. He worked as an assistant professor at the Delft University of Technology and obtained a PhD-degree in 2003. He specialized in photogrammetry and worked for several companies before joining the Cadastre in 2018. He is working on quality assurance in large-scale photogrammetric projects and research programs, specifically on renewal of cadastral map of The Netherlands.

Juriaan Lucassen works as a Mathware engineer at Sioux technologies since the end of 2020. Prior to starting at Sioux Mathware he obtained a PhD degree in experimental condensed matter physics in 2020. At Sioux technologies, Juriaan has been mostly involved in projects with a focus on both discrete and continuous optimization.

Mark van den Broek has been involved in the research program for renewal of the cadastral map of The Netherlands with key contributions in machine learning and optimization algorithm development. Mark has over 15 years of experience in analytics and algorithm development in various application domains. Mark studied in parallel Mathematics at Eindhoven University and Econometrics at Tilburg University.

Lense Swaenen is Mathware Architect and Technology Manager Optimization at the Mathware department of Sioux Technologies. He obtained M.Sc. degrees in Theoretical Physics and Mathematical Engineering at the KU Leuven. His current focus is on mathematical optimization applications. He is industrial partner representative for Sioux Technologies in the EU Horizon 2020 BigMath project.

Gerbrand Vestjens is working as a geodetic specialist at the Dutch Cadastre. After obtaining his M.Sc. degree in Geodesy from the Delft University of Technology he worked at Ingenieursbureau Geodelta until 2016. He is experienced in drafting technical specifications for nationwide geodetic data collection, currently working on large-scale photogrammetric projects, and the research program for renewal of the cadastral map of The Netherlands.

Wim Florijn has been working since 2019 as a machine learning engineer and software developer at Kadaster. He is working on the entire KKN application stack, and is specifically interested in creating and improving its machine learning components. After receiving his master’s degree in computer science specializing in machine learning and data science, he has previously worked as a machine learning engineer at a private company.

Stefan Bussemaker obtained his master’s degree in artificial intelligence from the university of Groningen in 2018 and started in 2019 as a machine learning engineer at Kadaster. There he contributed to the cadastral map renewal project: training Machine Learning models, using computer vision techniques to extract data from field sketches, and building pipelines to run the models in a cloud production environment.

Eric Hagemans has been working since 2014 as a geodetic specialist and innovation advisor at Kadaster in The Netherlands. He is responsible for the content of the KKN program and working on the innovation of cadastral surveying. Before he has worked as teacher and manager at the University of Applied Science in Utrecht and as geodetic engineer at two engineering companies. He studied geodesy at the Delft University of Technology.

CONTACTS

Dr. Frank van den Heuvel
Dutch Cadastre, Land Registry and Mapping Agency
Postbus 9046, 7300 GH Apeldoorn, The Netherlands
Tel: +31 88 183 2200
Email: frank.vandenheuvel@kadaster.nl
Website: http://www.kadaster.nl

Dr. Juriaan Lucassen
Sioux Technologies
Esp 130, 5633 AA Eindhoven, The Netherlands
Tel: +31 40 267 7100
Email: juriaan.lucassen@sioux.eu
Website: www.sioux.eu