



# Efficient Merging of Maps and Detection of Changes

Gabrielle Flood<sup>(✉)</sup>, David Gillsjö, Anders Heyden, and Kalle Åström

Centre for Mathematical Sciences, Lund University, Lund, Sweden  
{gabrielle,davidg,heyden,kalle}@maths.lth.se

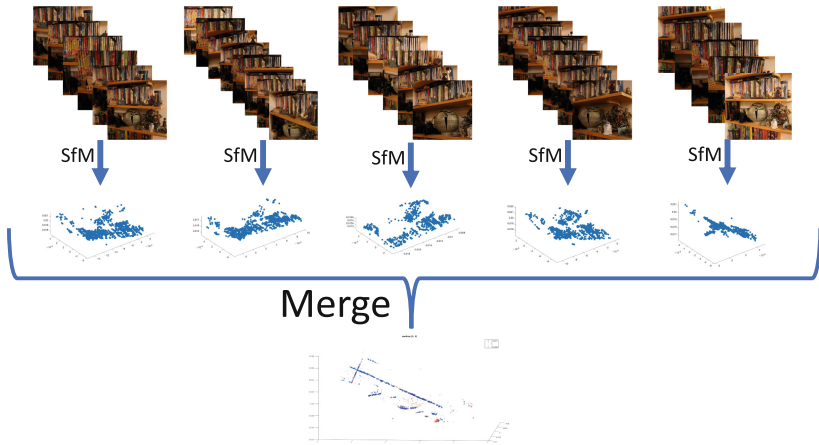
**Abstract.** With the advent of cheap sensors and computing capabilities as well as better algorithms it is now possible to do structure from motion using crowd sourced data. Individual estimates of a map can be obtained using structure from motion (SfM) or simultaneous localization and mapping (SLAM) using e.g. images, sound or radio. However the problem of map merging as used for collaborative SLAM needs further attention. In this paper we study the basic principles behind map merging and collaborative SLAM. We develop a method for merging maps – based on a small memory footprint representation of individual maps – in a way that is computationally efficient. We also demonstrate how the same framework can be used to detect changes in the map. This makes it possible to remove inconsistent parts before merging the maps. The methods are tested on both simulated and real data, using both sensor data from radio sensors and from cameras.

**Keywords:** Map merging · Change detection · Collaborative SLAM · SfM

## 1 Introduction

Structure from motion [5], is the problem of estimating the parameters of a map and of sensor motion using only sensor data. The map is typically a set of 2D or 3D points each consisting of a position and a feature vector. Assuming that feature errors are zero-mean Gaussian, the maximum likelihood estimate is that of minimising the sum of squares of the residuals. Within the field of computer vision this process is denoted bundle adjustment, where *bundle* refers to the bundle of light rays connecting each camera with each 3D point. For an overview of the literature and theory, see [13].

These optimization techniques are applicable not only to vision, but also to other types of sensors, such as audio, [9, 14] and radio [1]. With the advent of cheaper sensors and computing capabilities as well as better algorithms, it is now possible to gather and use much larger datasets. Instead of mapping a city every 5 years using special measurement cars or aerial photography, it is in principle possible for every car to add to the map of cities as they drive through them. Thus there is an additional need for research on map merging, including



**Fig. 1.** Structure from motion (SfM) is used to estimate a 3D map of scene features using images (or other sensors). In this paper we study the problem of detecting changes and merging maps, given multiple maps estimated by SfM from datasets collected at different occasions.

the problem of determining what has changed in a map. In this paper we study the basic principles behind map merging and collaborative SLAM. A straightforward method to merge several individual maps is to take all measurements into account simultaneously. However, non-linear optimization using all data can be prohibitively slow. We will study how a small memory footprint representation of a map can be generated and used to merge maps in a way that is computationally efficient, while still retaining most of the information from each individual bundle adjustment. We also demonstrate how the same framework can be used to detect changes in the map. This makes it possible to remove changing parts before merging the stationary parts of the map. The idea is demonstrated in Fig. 1.

The idea of approximating the result from parts of the data has previously been used in the rotation averaging literature, cf. [2]. These approximate methods can give satisfactory results at a much increased speed. Another example of this idea is the approach of Global Epipolar Adjustment [12], in which a simplified error metric is based on the linear epipolar constraints for image pairs. Another approach is incremental light bundle adjustment, iBLA, [6] in which an error metric based on a combination of epipolar constraints and a variant of the trifocal constraint is used.

The main contributions of this paper are a novel method for computationally efficiently merging of individual maps obtained from bundle adjustment, utilizing a compact representation of the Jacobian matrix, and a change detection method based on a statistical analysis of the residuals.

## 2 The Separate Bundles - for TOA and Images

Before different maps are merged, the individual map estimates have to be created. In this section we present some of the notations used to understand how the raw data relates to the quality of the map estimates.

For the case of time of arrival (TOA) measurements the feature map consists of a number of receiver positions. Initially, TOA measures between  $m$  receivers at positions  $x_i \in \mathcal{R}^3$  and  $n$  sender positions  $y_j \in \mathcal{R}^3$  are given. For each sender-receiver pair this measure can be translated into a distance estimate  $d_{ij} = |x_i - y_j| + \varepsilon_{ij}$ , where  $1 \leq i \leq m$  and  $1 \leq j \leq n$  and where  $|\cdot|$  denotes the Euclidean norm of a vector in  $\mathcal{R}^3$ . The measurements errors  $\varepsilon_{ij}$  are assumed to be independent, Gaussian with mean zero and standard deviation  $\sigma$ .

The final map estimate for a TOA or structure from motion system is usually obtained by non-linear least squares minimization over inlier measurements; this process is referred to as bundle adjustment in computer vision. Here, a few key components from the optimization are presented.

For the TOA data, let  $\mathbf{r}$  denote the measurements residuals,

$$\mathbf{r} = [r_{11} \dots r_{1n} \ r_{21} \dots r_{2n} \ r_{m1} \dots r_{mn}]^T, \quad r_{ij} = d_{ij} - |x_i - y_j|, \quad (1)$$

and denote the parameters of interest, which are optimized, by  $\mathbf{z}$ . This would typically be the receiver and the sender positions,

$$\mathbf{z} = (x_1, x_2, \dots, x_m, y_1, \dots, y_n). \quad (2)$$

The computer vision case is analogous. Denoting the camera matrices  $P_i$  and the 3D points  $U_j$ , each image point  $u_{ij}$  gives a residual  $r_{ij}$ . The residual vector  $\mathbf{r}$  is found by stacking all image feature residuals  $r_{ij}$  and the parameters are collected in a parameter vector

$$\mathbf{z} = (P_1, P_2, \dots, P_m, U_1, \dots, U_n). \quad (3)$$

The maximum likelihood estimate of  $\mathbf{z}$  is found by minimizing the sum of the squares of the residuals, i.e.

$$\mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}} \mathbf{r}^T \mathbf{r}, \quad (4)$$

which gives the optimal parameter update

$$\Delta \mathbf{z} = -(J^T J)^{-1} J^T \mathbf{r}. \quad (5)$$

For more details on the optimization, see [13]. For the analysis, the estimate of the matrix  $J$  (the Jacobian) is containing the derivatives of the residuals with respect to the parameters is of interest, i.e.  $\mathbf{r}$  with respect to  $\mathbf{z}$ , further on denoted  $\partial \mathbf{r} / \partial \mathbf{z}$ .

The map points can only be estimated up to a choice of coordinate system. For simplicity we will in the TOA case normalize the coordinate system so that the first receiver is placed in the origin, the second along the x-axis, the third

in the xy-plane and so forth. By removing this gauge freedom with dimension  $\phi$  we see that the effective number of degrees of freedom in the problem is  $d_{dof} = (m + n)\rho - \phi$ , where  $\rho$  denotes the dimension. For TOA problems in 3D we have  $\rho = 3$  and  $\phi = 6$ . The effective degrees of freedom for the computer vision case becomes  $d_{dof} = (6m + 3n) - \phi$ , with gauge freedom  $\phi = 7$  since we are free to choose position, orientation and scale of the coordinate system.

### 3 Merging Separate Maps

Once the  $N$  separate maps are obtained they can be merged to get a single more accurate map. We have investigated three different ways to do this.

#### 3.1 The Full Bundle

One way to add the maps is do one large bundle where all the individual measurements are used simultaneously. Merging all maps through a large bundle is a good way to get an accurate map. However, the method is time consuming and if a new measurement is made after the original merge, the whole map has to be re-bundled. In that sense, there is no way to *add* new information to the existing, which makes this method unsuitable for online applications.

#### 3.2 The Kalman Filter

A traditional method designed to update parameters gradually is the Kalman filter [8]. The algorithm for the Kalman filter looks as follows:

Priori estimate update: $x_1 = A \cdot x_0$ (6)	Measurement update: $K = P_1 \cdot H^T \cdot (H \cdot P_1 \cdot H^T + R)^{-1}$ (8)
$P_1 = A \cdot P_0 \cdot A^T + Q$ (7)	$x_2 = x_1 + K \cdot (u - H \cdot x_1)$ (9)
	$P_2 = (I - K \cdot H) \cdot P_1.$ (10)

Then,  $H \cdot x_2$  is the new state prediction, and  $x_2$  and  $P_2$  are the new estimates replacing  $x_0$  and  $P_0$  for the next iteration. In our case  $x_0$  will be the receivers from the first measurement occasion,  $x_0 = \mathbf{q}^{(1)}$  (superscript denoting measurement occasion), while the observation  $u$  will be the receiver values from the following  $N - 1$  measurements s.t.  $u_{k-1} = \mathbf{q}^{(k)}$ ,  $2 \leq k \leq N$ . Both the update matrix and the observation matrix are identity matrices,  $A = I$ ,  $H = I$  and the covariance of the random excitation is set to  $Q = 0.1 \cdot I$ . Finally,  $P_0$  and  $R$  are measurement uncertainties,  $P_0 = \mathbf{C}[\Delta \mathbf{q}^{(1)}]$  and  $R_{k-1} = \mathbf{C}[\Delta \mathbf{q}^{(k)}]$ ,  $2 \leq k \leq N$ . The covariance  $\mathbf{C}[\Delta \mathbf{q}]$  can be extracted from the covariance of  $\Delta \mathbf{z}$  from Eq. (5). This is given by

$$\mathbf{C}[\Delta \mathbf{z}] = (J^T J)^{-1} J^T \cdot \mathbf{E}[\mathbf{r}^T \mathbf{r}] \cdot J (J^T J)^{-1} = \sigma^2 (J^T J)^{-1}. \quad (11)$$

The covariance of the map,  $\mathbf{C}[\Delta\mathbf{q}]$ , can be retrieved by picking the rows and columns in  $\mathbf{C}[\Delta\mathbf{z}]$  that correspond to  $\mathbf{q}$  and the variance of  $\mathbf{r}$  can be approximated by [7, p. 148]

$$\sigma^2 \approx \frac{1}{m \cdot n - d_{dof}} \cdot \mathbf{r}^T \mathbf{r} = \frac{1}{m \cdot n - d_{dof}} \cdot \sum_{i=1}^{m \cdot n} r_i^2. \quad (12)$$

The Kalman filter is a computationally cheap method. However, it is not as accurate as the full bundle. Also, the parameters need to be tuned for the specific problem and it is not evident either how to detect and handle changes in the map.

### 3.3 The Linearized Method

The idea of this method is that the optimal residuals from the separate bundles can be linearized – such that all that needs to be saved is a small memory footprint representation – to avoid the large bundles. Having the optimal residuals  $\mathbf{r}^{(k)}$  and the optimal Jacobians  $J^{(k)}$  from each run  $k$ , the residuals can be linearized using a first order Taylor approximation. A key idea here is to divide the unknown parameters in  $\mathbf{z}$  into two parts  $\mathbf{q}$  and  $\mathbf{s}$ , where  $\mathbf{q}$  are the parameters that exist in several SLAM sessions. The parameters  $\mathbf{s}$  can be thought of as auxiliary parameters, e.g. those that are relevant only for one specific bundle session. In the time-of-arrival case, some of the 3D anchors might be constant over several SLAM sessions whereas the measurement points and some of the anchors might be different. For vision based structure from motion, some of the 3D points are the same (these go into  $\mathbf{q}$ ) whereas the rest of the points and camera matrices go into  $\mathbf{s}$ .

**The Compressed Residual.** First, the Jacobian is divided into two blocks

$$J = [J_a \ J_b], \quad (13)$$

where  $J_a$  contains the columns that correspond to the main parameters  $\mathbf{q}$  and  $J_b$  contains the columns corresponding to the auxiliary parameters  $\mathbf{s}$ . The squared Jacobian is

$$J^T J = \begin{bmatrix} J_a^T \\ J_b^T \end{bmatrix} \cdot [J_a \ J_b] = \begin{bmatrix} J_a^T J_a & J_a^T J_b \\ J_b^T J_a & J_b^T J_b \end{bmatrix} = \begin{bmatrix} U & W \\ W^T & V \end{bmatrix}. \quad (14)$$

Furthermore, if we insert this in the equation for the optimal update from (5) we get

$$\Delta\mathbf{z} = \begin{bmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{s} \end{bmatrix} = -(J^T J)^{-1} J^T \mathbf{r} \quad \Leftrightarrow \quad \begin{bmatrix} U & W \\ W^T & V \end{bmatrix} \begin{bmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{s} \end{bmatrix} = -J^T \mathbf{r}. \quad (15)$$

The product  $-J^T \mathbf{r}$  is zero in an optimal point and so the second row provides a connection between  $\mathbf{q}$  and  $\mathbf{s}$ . This gives a linear constraint on how to adjust the

auxiliary parameters  $\mathbf{s}$  when the main parameters  $\mathbf{q}$  change. Thus the partial derivatives of  $\mathbf{s}$  with respect to  $\mathbf{q}$  is

$$W^T \Delta \mathbf{q} + V \Delta \mathbf{s} = 0 \quad \Leftrightarrow \quad \Delta \mathbf{s} = -V^{-1} W^T \Delta \mathbf{q} \quad \Rightarrow \quad \frac{\partial \mathbf{s}}{\partial \mathbf{q}} = -V^{-1} W^T. \quad (16)$$

We can use this together with the definition  $J = \partial \mathbf{r} / \partial \mathbf{z}$  to find how the residuals change if we change the receiver map

$$\Delta \mathbf{r} = [J_a \ J_b] \begin{bmatrix} \Delta \mathbf{q} \\ \Delta \mathbf{s} \end{bmatrix} = \left( J_a + J_b \cdot \frac{\partial \mathbf{s}}{\partial \mathbf{q}} \right) \Delta \mathbf{q}. \quad (17)$$

Thus,  $J_a + J_b \frac{\partial \mathbf{s}}{\partial \mathbf{q}}$  will be the Jacobian for the map, further on denoted  $J_q$ .

Now, denote the residuals as a function of  $\Delta \mathbf{q}$ . A first order Taylor expansion gives

$$\mathbf{r}(\Delta \mathbf{q}) \approx \mathbf{r}|_o + \mathbf{r}'_{\Delta \mathbf{q}}|_o \Delta \mathbf{q} = \mathbf{r}|_o + J_q|_o \Delta \mathbf{q}. \quad (18)$$

Here  $o$  denotes an optimal point and  $|_o$  denotes evaluating an expression at the point  $o$ . Then, the square of these residuals will be

$$\mathbf{r}^T \mathbf{r} \approx (\mathbf{r}|_o + J_q|_o \Delta \mathbf{q})^T (\mathbf{r}|_o + J_q|_o \Delta \mathbf{q}) = \mathbf{r}|_o^T \mathbf{r}|_o + 2\mathbf{r}|_o^T J_q|_o \Delta \mathbf{q} + \Delta \mathbf{q}^T J_q^T|_o J_q|_o \Delta \mathbf{q}. \quad (19)$$

In a minimum point  $\mathbf{r}|_o^T J_q|_o$  is zero. Furthermore, using the QR-decomposition of the Jacobian we get

$$\Delta \mathbf{q}^T J_q^T J_q \Delta \mathbf{q} = \Delta \mathbf{q}^T (QR)^T QR \Delta \mathbf{q} = \Delta \mathbf{q}^T R^T Q^T QR \Delta \mathbf{q} = \Delta \mathbf{q}^T R^T R \Delta \mathbf{q}. \quad (20)$$

Introducing the notation  $a = (\mathbf{r}|_o^T \mathbf{r}|_o)^{1/2}$ , the squared residuals from (19) can be written shorter as

$$\mathbf{r}^T \mathbf{r} \approx a + \Delta \mathbf{q}^T R^T R \Delta \mathbf{q}, \quad (21)$$

and this is our compressed expression for the residuals.

**The Merge.** Furthermore, this compressed expression can be used to add two separate maps. Assume that we have the residuals for the two maps,

$$(\mathbf{r}^{(i)})^T (\mathbf{r}^{(i)}) = (a^{(i)})^2 + (\Delta \mathbf{q}^{(i)})^T (R^{(i)})^T R^{(i)} \Delta \mathbf{q}^{(i)}, \quad i = 1, 2. \quad (22)$$

Adding the two equations and writing  $\Delta \mathbf{q}^{(i)} = \mathbf{q} - \mathbf{q}^{(i)}$  for an arbitrary  $\mathbf{q}$  gives

$$\begin{aligned} \sum_{i=1}^2 (\mathbf{r}^{(i)})^T (\mathbf{r}^{(i)}) &= \sum_{i=1}^2 (a^{(i)})^2 + (\Delta \mathbf{q}^{(i)})^T (R^{(i)})^T R^{(i)} \Delta \mathbf{q}^{(i)} = (a^{(1)})^2 + (a^{(2)})^2 \\ &+ \begin{bmatrix} R^{(1)}(\mathbf{q} - \mathbf{q}^{(1)}) \\ R^{(2)}(\mathbf{q} - \mathbf{q}^{(2)}) \end{bmatrix}^T \begin{bmatrix} R^{(1)}(\mathbf{q} - \mathbf{q}^{(1)}) \\ R^{(2)}(\mathbf{q} - \mathbf{q}^{(2)}) \end{bmatrix} = (a^{(1)})^2 + (a^{(2)})^2 + \hat{\mathbf{r}}^T \hat{\mathbf{r}}. \end{aligned} \quad (23)$$

The terms  $(a^{(1)})^2$  and  $(a^{(2)})^2$  are fixed while the third term  $\hat{\mathbf{r}}^T \hat{\mathbf{r}}$  can be minimized to minimize the sum of the residuals. Introducing new notations  $M$  and  $b$ ,  $\hat{\mathbf{r}}$  can be written

$$\hat{\mathbf{r}} = \begin{bmatrix} R^{(1)}(\mathbf{q} - \mathbf{q}^{(1)}) \\ R^{(2)}(\mathbf{q} - \mathbf{q}^{(2)}) \end{bmatrix} = \begin{bmatrix} R^{(1)} \\ R^{(2)} \end{bmatrix} \mathbf{q} - \begin{bmatrix} R^{(1)} \mathbf{q}^{(1)} \\ R^{(2)} \mathbf{q}^{(2)} \end{bmatrix} = M \mathbf{q} - b. \quad (24)$$

To minimize  $\hat{\mathbf{r}}$  and thus  $\hat{\mathbf{r}}^T \hat{\mathbf{r}}$  is a least squares problem which can be solved using the pseudo inverse. Denoting the merged map  $\mathbf{q}^*$  gives

$$\mathbf{q}^{(*)} = (M^T M)^{-1} M^T b. \tag{25}$$

We can also compress the final result. Using that a general  $\mathbf{q}$  can be written  $\mathbf{q} = \Delta\mathbf{q}^{(*)} + \mathbf{q}^{(*)}$ , the third term in (23) can be expressed

$$\begin{aligned} \hat{\mathbf{r}}^T \hat{\mathbf{r}} &= (M\mathbf{q} - b)^T (M\mathbf{q} - b) = (M\mathbf{q}^{(*)} - b + M\Delta\mathbf{q}^{(*)})^T (M\mathbf{q}^{(*)} - b + M\Delta\mathbf{q}^{(*)}) \\ &= (M\mathbf{q}^{(*)} - b)^T (M\mathbf{q}^{(*)} - b) + (\Delta\mathbf{q}^{(*)})^T M^T M \Delta\mathbf{q}^{(*)}, \end{aligned} \tag{26}$$

where the linear term vanishes due to orthogonality. Using this in Eq. (23) gives

$$(\mathbf{r}^{(*)})^T \mathbf{r}^{(*)} = (a^{(1)})^2 + (a^{(2)})^2 + (M\mathbf{q}^{(*)} - b)^T (M\mathbf{q}^{(*)} - b) + (\Delta\mathbf{q}^{(*)})^T M^T M \Delta\mathbf{q}^{(*)}. \tag{27}$$

If  $M$  is QR-decomposed in a similar manner as  $J_q$  was in (20) this total result can be compressed as

$$(\mathbf{r}^{(*)})^T \mathbf{r}^{(*)} = (a^{(*)})^2 + (\Delta\mathbf{q}^{(*)})^T (R^{(*)})^T R^{(*)} \Delta\mathbf{q}^{(*)}, \tag{28}$$

with  $R^{(*)}$  being the triangular matrix from the QR-decomposition of  $M$  and

$$a^{(*)} = \left( (a^{(1)})^2 + (a^{(2)})^2 + (M\mathbf{q}^{(*)} - b)^T (M\mathbf{q}^{(*)} - b) \right)^{\frac{1}{2}}. \tag{29}$$

By this, the representation of the final map is the same as in (21) and the merged map can be treated as one of the original. Furthermore, more maps can be added using the algorithm described above. Thus, to add maps, all we need to save from the separate bundles are the maps  $\mathbf{q}^{(i)}$ , the squared residuals  $a^{(i)}$ , and the triangular matrices  $R^{(i)}$  from the QR-decompositions of the Jacobians.

In some cases the linearized method is similar to the Kalman filter. However, several maps can be added at once using the linearized model and it also allows for better control. We will also show that this method can be developed to detect map changes.

## 4 Detection of Changes

Once we know how to merge two or more maps we can also use this to detect whether the map has changed between the measurement occasions. For this, assume that we have two maps  $\mathbf{q}^{(1)}$  and  $\mathbf{q}^{(2)}$  and their merge  $\mathbf{q}^{(*)}$ . Furthermore, we have the norms of their residuals,  $a^{(1)}$ ,  $a^{(2)}$  and  $a^{(*)}$ . An approximation for the residual variance is derived in (12). This can be used to find the estimated value of how the squared residuals change when we add maps. Rearranging terms from (12), we get

$$\mathbf{E}[(a^{(i)})^2] = \mathbf{E}[(\mathbf{r}^{(i)})^T (\mathbf{r}^{(i)})] = \sigma^2(mn - (n\rho + m\rho - \phi)), \quad i = 1, 2 \tag{30}$$

$$\mathbf{E}[(a^{(*)})^2] = \mathbf{E}[(\mathbf{r}^{(*)})^T (\mathbf{r}^{(*)})] = \sigma^2(mn - (Nn\rho + m\rho - \phi)), \tag{31}$$

and subtracting these – in this case with  $N = 2$  maps – gives

$$\mathbf{E}[(a^{(*)})^2 - (a^{(1)})^2 - (a^{(2)})^2] = \sigma^2(N - 1)(m\rho - \phi). \quad (32)$$

If we use real data,  $\sigma$  is unknown, but it can be estimated from the separate bundles using (12), s.t.  $\hat{\sigma}^2 = ((\sigma^{(1)})^2 + (\sigma^{(2)})^2)/2$ .

The values in (32) can be seen as a sum of  $(N - 1)(m\rho - \phi)$  Gaussian variables, and a sum of  $2\nu$  independent Gaussian distributed variables with mean zero and standard deviation  $\sigma_n$  has a  $\Gamma$  distribution with density [3, p. 47]

$$f_{\alpha,\nu}(x) = \frac{1}{\Gamma(\nu)} \alpha^\nu x^{\nu-1} e^{-\alpha x}, \quad (33)$$

with  $\alpha = 1/(2\sigma_n^2)$  and  $\Gamma$  being the gamma function. This density will be denoted  $\Gamma(\alpha, \nu)$  (two parameters). Furthermore, using  $\tilde{a} = (a^{(1)})^2 + (a^{(2)})^2 - (a^{(*)})^2$  and  $\gamma = (N - 1)(m\rho - \phi)$  we get that  $\tilde{a} \sim \Gamma(1/(2\sigma^2), \gamma/2)$ . Thus, to know whether a map has changed we can compare the estimated  $\tilde{a}$  to the distribution. A reasonable choice is that if the difference  $\tilde{a}$  lies within the 99 percentile of  $\Gamma(1/(2\sigma^2), \gamma/2)$  there has not been any change in the map, but if  $\tilde{a}$  is higher than this limit, a change has probably occurred.

If a change between two maps is discovered, we further investigate those maps. By comparing the positions for each map point, we say that if the distance between them is larger than  $3\hat{\sigma}$  the map point has probably moved. This could also be used to decrease the variance even further for the receivers that have not changed, by using information from all maps for these receivers.

## 5 Experimental Validation

To validate the method suggested in this paper, experiments on simulated TOA data as well as real ultra-wideband (UWB) data have been performed. We have also developed the method to work for, and tried it on, 3D-reconstructions from image data.

### 5.1 Time of Arrival – Simulated Data

For each of the simulated experiments  $m$  receivers in 3D were generated from a uniform distribution,  $\mathbf{q}^{(t)} \sim \mathcal{U}(0, 10)$ , superscript ( $t$ ) denoting the true value. We simulated  $N$  different measurement occasions with  $n$  sender positions  $\mathbf{s}^{(t)} \sim \mathcal{U}(0, 10)$  each and calculated the  $mn$  sender-receiver distances. Gaussian noise with standard deviation  $\sigma$  was added to achieve distance measurements. For each measure we performed a separate bundle to get the  $N$  maps  $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N)}$  and the compressed representation explained in Sect. 3.3 and more specifically in (21).

**Test of Time and Accuracy.** For the first experiment  $m = 10$ ,  $\sigma_n = 0.3$ ,  $N = 2$  and no change occurred in the true map. The experiments were run four times with  $n = 10, 100, 1000, 4000$  respectively. For each case, the merge was



computed using the three methods presented in this paper and the runtimes were measured. We computed the error norm  $\sqrt{\sum_{i=1}^m |q_i^{(t)} - q_i|^2}$  and for the full bundle and the linearized method, we also computed the squared distance residuals per residual  $\mathbf{r}^T \mathbf{r} / (mn) = a^2 / (mn)$ . The results can be seen in Table 1.

**Table 1.** The results from the experiment explained in Sect. 5.1. The values come from a merge of two maps between which no change has occurred. These values are the mean of 10 similar runs.

	$n$	10	100	1000	4000
Runtime [s]	Full bundle	$2.3 \cdot 10^{-2}$	0.19	3.8	54.7
	Linearized	$1.9 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$4.7 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$
	Kalman	$2.4 \cdot 10^{-3}$	$2.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$
$\   \mathbf{q}^{(t)} - \mathbf{q}  \ $	Full bundle	1.20	0.11	$1.6 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$
	Linearized	1.34	0.11	$1.6 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$
	Kalman	1.48	0.12	$2.2 \cdot 10^{-2}$	$5.8 \cdot 10^{-3}$
$\frac{\mathbf{r}^T \mathbf{r}}{mn} = \frac{a^2}{mn}$	Full bundle	0.11	0.12	0.13	0.13
	Linearized	0.11	0.12	0.13	0.13

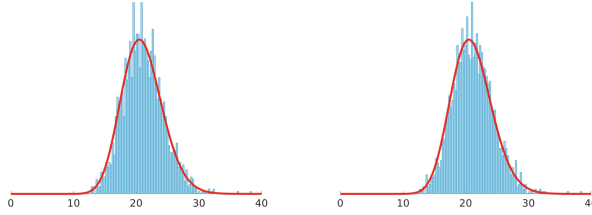
Even if the runtime is highly dependent on the implementations, the table gives a valid comparison between the methods. The linearized method is almost as accurate as the full bundle. Moreover, when only the sender positions increase, and thus also the number of distances, the runtime for the linearized method and the Kalman filter does not increase notably, while the runtime for the full bundle does. Hence, the linearized method is faster than the full bundle and more accurate than the Kalman filter.

**Validating the Detection Threshold.** To validate the threshold for detection of changes described in Sect. 4, we tested the distribution of  $\tilde{a}$  empirically. Using  $m = 30, n = 200, N = 2$  and  $\sigma_n = 0.5$  the distances were computed. The separate bundles as well as the merge using both the full bundle and the linearized method were then conducted. For all of the different maps we computed the compressed representations from (21). We then computed

$$\tilde{a}_{full} = (a^{(1)})^2 + (a^{(2)})^2 - (a_{full}^{(*)})^2, \quad \text{and} \quad \tilde{a}_{lin} = (a^{(1)})^2 + (a^{(2)})^2 - (a_{lin}^{(*)})^2, \tag{34}$$

where subscript index *full* and *lin* denotes the full bundle and the linearized method respectively. This was re-made 2000 times with different noise. The total degrees of freedom were  $\gamma = (N - 1)(m \cdot \rho - \phi) = 30 \cdot 3 - 6 = 84$ . The results of  $\tilde{a}_{full}$  and  $\tilde{a}_{lin}$  were then plotted in a histogram together with a  $\Gamma(2, 42)$  distribution in Fig. 2. The histograms agree well with the gamma distribution in both cases; hence, this can be used to test the significance.

**Detection of Changed Maps.** Furthermore, we did an experiment where the map actually had changed. This time we used  $m = 10, n = 30, N = 3$  and



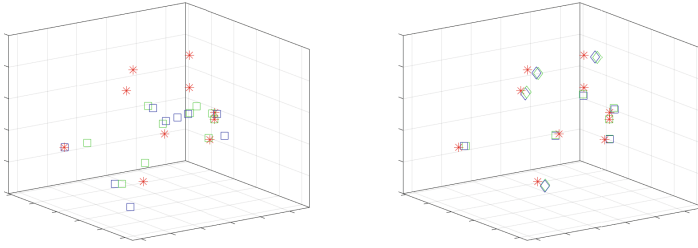
**Fig. 2.** The plots show histograms of the residuals  $\tilde{a}_{full}$  (to the left) and  $\tilde{a}_{lin}$  (to the right) computed using the full bundle and the linearized method respectively. The curve (–) shows the  $\Gamma$  distribution which we expect  $\tilde{a}$  to belong to.

$\sigma_n = 0.5$ . Four of the ten receivers moved before the last measurement. After running the separate bundles and merging the maps both using a full bundle and our linearized method we investigated the differences in the residuals. The system had  $\gamma = 2 \cdot (10 \cdot 3 - 6) = 48$  degrees of freedom and thus  $\tilde{a}$  should be such that it could come from a  $\Gamma(1/(2\hat{\sigma}^2), 24)$  distribution if no changes has occurred. Using the estimate  $\hat{\sigma}^2$  the 99-percentile of this was  $\tilde{a} = 17.7$ . In this specific case, the results from the merge gave  $\tilde{a}_{full} = 603$  and  $\tilde{a}_{lin} = 749$  and this clearly showed that something had changed. The results from the unsuccessful merge can be seen to the left in Fig. 3. To the right in Fig. 3 are the results from the merge between the first and second map, after the system successfully had detected the change.

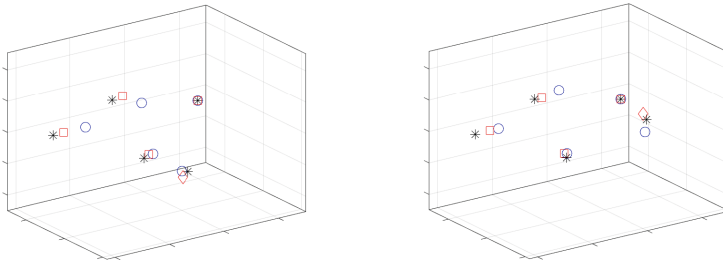
## 5.2 Time of Arrival – Real Data

To test our method  $N = 9$  experiments were conducted using a Bitcraze Crazyflie quadcopter and their Loco-positioning system which consists of  $m = 5$  anchors with UWB chips and a flying quadcopter with a mounted UWB chip, giving approximately  $n = 600$  sender positions for each measurement. The five anchors were positioned around the room and one of them was moved before the last three runs. The experiment was conducted in a MOCAP studio to record the ground truth flightpath as well as the anchor positions. Distance measurements from the quadcopter (sender) to all the anchors (receivers) were measured at a frequency of 30 Hz.

The problem was solved as explained in previous sections, except that the threshold for  $\tilde{a}$  now was 10 times the 99 percentile for the  $\Gamma$  distribution. This threshold was used for all real data experiments. In Fig. 4 the results from the Kalman filter and the linearized method are shown. While the dynamics of the Kalman filter makes the estimated receivers end up further away from the true positions – *on their way* to the correct position – for some of the measurements, the linearized method correctly detects when a change has occurred. Thereafter, only the similar maps are merged.



**Fig. 3.** An unsuccessful merge of map 1,2 and 3 (left) and a successful merge of map 1 and 2 (right). The stars (\*) show the true receiver positions, the squares the results from full bundle ( $\square$ ) and the linearized method ( $\square$ ). In the right figure, the points for which a change has been detected are (correctly) marked by a diamond ( $\diamond, \diamond$ ). (Color figure online)



**Fig. 4.** Results from two of the maps from the experiments with UWB data. The stars (\*) show the true receiver positions, the circles ( $\circ$ ) the results from the Kalman filter and the squares ( $\square$ ) from the linearized method. The change between the maps has been correctly detected by the linearized method and changed receivers are marked with a diamond ( $\diamond$ ). (Color figure online)

### 5.3 Images – Real Data

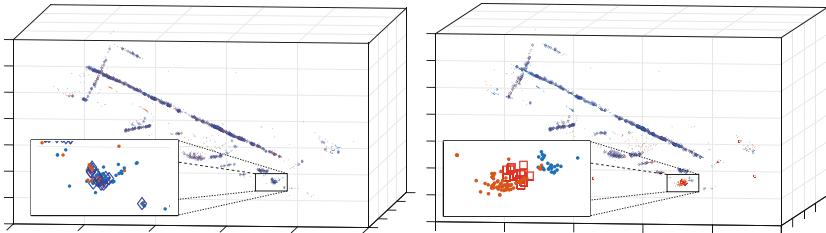
In this experiment,  $N = 5$  sets of images were taken of an indoor scene, a bookshelf with a number of toy models, as depicted in Fig. 1. In between set 2 and 3 an R2D2 model was moved, which we wanted to detect. As a first step we used a structure from motion pipeline [11] to obtain a 3D reconstruction for each set. The points in this reconstruction are the feature points in the map, corresponding to the receivers in the TOA experiments.

Unlike the TOA experiments, correspondence between 3D points in the different datasets are not given. Prior to merging, we performed data association by SIFT [10] feature matching and geometric alignment in a RANSAC [4] framework. After this the maps were also in the same coordinate system, which is required for the linearized method and speeds up the full bundling method.



**Fig. 5.** Changes detected in merge between dataset 2 and 3. Feature points are marked with blue dots and changed features are circled in cyan. (Color figure online)

Using the same method as in Sect. 5.2 – with detection based on a  $\Gamma$  distribution and the feature point distances – the algorithm detected change during the merge of dataset 2 and 3, which is correct. In Fig. 5 we see that the feature points on R2D2 are correctly detected as changed. Note that some features are not present in both datasets and therefore these features on the R2D2 are not marked as changed. Figure 6 shows the 3D reconstruction from above. Here we see that the merged points on R2D2 does not align with either dataset 2 or 3.



**Fig. 6.** To the left, the merge between dataset 1 and 2 where no change was detected. The separate maps are marked with dots ( $\bullet$ ,  $\bullet$ ) and the merge by diamonds ( $\diamond$ ). To the right, the merge between dataset 2 and 3, where a change was detected. The points for which a change was detected are marked by squares ( $\square$ ). (Color figure online)

## 6 Conclusions

We have presented a novel and efficient method, with small memory footprint, for merging individual maps obtained from bundle adjustment optimization along with a statistically motivated method for detecting changes in the map. The method has been compared favorably to using full bundle adjustment and the Kalman filter and is shown to be a good compromise between performance and time efficiency. This makes the method suitable for online applications as well as the use of crowd sourced data. The performance has been confirmed on both TOA and vision problems for both simulated and real data. One limitation is

that the map points used for the coordinate system normalization need to be consistent for all maps. However, if this problem is solved, we believe that the method could be further developed to a full collaborative SLAM system.

**Acknowledgments.** This work was partially supported by the strategic research projects ELLIIT and eSENCE, the Swedish Foundation for Strategic Research project, Semantic Mapping and Visual Navigation for Smart Robots (grant no. RIT15-0038), and Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

## References

1. Batstone, K., Oskarsson, M., Åström, K.: Robust time-of-arrival self calibration and indoor localization using Wi-Fi round-trip time measurements. In: Proceedings of International Conference on Communication (2016)
2. Enqvist, O., Olsson, C., Kahl, F.: Non-sequential structure from motion. In: Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS), Barcelona, Spain (2011)
3. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 2. Wiley, Hoboken (1968)
4. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
6. Indelman, V., Roberts, R., Beall, C., Dellaert, F.: Incremental light bundle adjustment. In: 2012 Electronic Proceedings of the British Machine Vision Conference, BMVC 2012, January 2012
7. Jakobsson, A.: An Introduction to Time Series Modeling, 1st edn. Studentlitteratur AB, Lund (2013)
8. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960)
9. Kuang, Y., Burgess, S., Torstensson, A., Åström, K.: A complete characterization and solution to the microphone position self-calibration problem. In: ICASSP (2013)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
11. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Heyden, A., Kahl, F. (eds.) SCIA 2011. LNCS, vol. 6688, pp. 524–535. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21227-7\\_49](https://doi.org/10.1007/978-3-642-21227-7_49)
12. Rodriguez, A.L., López-de Teruel, P.E., Ruiz, A.: Reduced epipolar cost for accelerated incremental SfM. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3097–3104. IEEE (2011)
13. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) IWVA 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-44480-7\\_21](https://doi.org/10.1007/3-540-44480-7_21)
14. Zhayida, S., Andersson, F., Kuang, Y., Åström, K.: An automatic system for microphone self-localization using ambient sound. In: 22nd European Signal Processing Conference (2014)