Real-Time Deformation Measurements Using Single Image Photogrammetry

Andrej Hideghéty
Slovak University of Technology in Bratislava, Vazovova 5, 812 43 Bratislava 1, Slovakia
andrej.hideghety@stuba.sk

Abstract. Most photogrammetric measurements are currently based on image acquisition in the field and subsequent processing in office environment with certain temporal delay. However, in some cases it is necessary to process the data real-time, or at least in-situ. Bridge load testing is an example of measurement processing directly at the place of imaging, where almost immediate information about the current state or change of the object is required. An algorithm is developed for these purposes, including a camera controlling software and a MATLAB code that identifies and quantifies the shifts of the observed points in the image plane. The observed points are in the shape of black disks on a white background. Using a horizontal camera position individual epochs are captured. Each image is immediately transferred to a computer via Wi-Fi. The MATLAB code then loads the image and binarizes it. Binarization of the image is performed by the Canny edge detector. Using normalized 2-D cross-correlation, the algorithm determines the approximate coordinates based on a target template. A function performs least squares ellipse fitting and determines the center of the target in sub-pixel accuracy, the semi-major axis, the semi-minor axis and the rotation angle of the ellipse. The target detection is executed in a while cycle loop, which compares the point coordinates from each epoch to the initial state, thus quantifying the deformations in pixels. If the next image is not yet available, the loop restarts. The deformations are calculated based on the known scale of each target. This paper presents a detailed description of the development of the algorithm, the results achieved and the proposed improvements going forward.

1. Introduction
Most photogrammetric measurements nowadays consist of image acquisition in field and data processing with certain temporal delay. This is acceptable in many applications, where the object does not undergo dynamic changes, e.g., photogrammetric scanning of cultural heritage [1]. This paper explores an approach where dynamic objects are measured, e.g., bridge load tests [2]. Although photogrammetric scanning was successfully implemented during load tests [3], this paper proposes using single image photogrammetry [4], with the aim to calculate results with minimal temporal delay, optimally in real-time.

Bridge deformations can be measured using several geodetic methods. One of the most common method in practice is precise levelling [5]. During load tests of bridges at least two surveyors are required to perform the levelling measurement, while the object is under load, thus safety of the people concerned may be compromised [6]. Levelling a bridge and providing immediate results might take up to 15-20 minutes even for an experienced surveyor, depending on the parameters of the bridge and number of measured points. Because of these reasons it is crucial to develop an algorithm using a technology that
performs measurements remotely, from a safe distance, in case the bridge collapses, while performing
the calculations as fast as possible, within seconds.

The method used in this paper according to the number of images and camera positions is single
image digital photogrammetry, while position-wise it is terrestrial [7, 8]. Measurement results are
available in real-time [9, 10]. Time wise deformation measurements are performed using time base
photogrammetry [11, 12]. This method can detect changes only in the plane parallel to the image,
however if the image plane is vertical the measured deformations represent the vertical displacements.
To achieve sub-pixel accuracy several theoretical assumptions should be fulfilled: the camera needs to
be calibrated, images should be of high resolution and high quality, the image plane should be vertical,
the internal and external orientation parameters of the camera should be constant, i.e., the camera should
be stable [13]. The aim of this work is to propose an algorithm that can perform deformation
measurements with sub-pixel accuracy, however some functions are not yet implemented.

2. Algorithm

During the development of the algorithm several issues had to be resolved. The algorithm starts with
image acquisition, followed by the data transfer, and ends with data processing. This section deals with
the detailed description of the individual parts of the software-hardware combination.

2.1. Image acquisition and data transfer

The first step should always be choosing a camera with the desired spatial and radiometric resolution
and a lens that facilitates a field of vision that captures the required targets on an object. During every
photogrammetric application it is crucial to calibrate the specific combination of the given camera and
lens [14]. The camera should be able to transfer data via Wi-Fi and to capture images in lossless format,
e.g., Tagged Image File Format (TIFF). The lossless format is important mainly in high accuracy
applications, such as in bridge load testing, to prevent noise caused by compression [15]. During the
development of this algorithm a computer was connected to the camera via Wi-Fi. The camera was
controlled using qDslrDashboard (qDD) software.

qDD is a camera controlling software, that uses the Picture Transfer Protocol (PTP) and PTP/IP [16].
Cameras can be connected over a wired or wireless network. In this case a Wi-Fi connection was used
to ensure the highest possible stability of the camera, while physically connecting via cable might have
caused a variation in the parameters of external orientation due to movement. This application facilitates
focusing the lens remotely by choosing a location on screen. Setting the aperture, shutter speed and iso
is also possible in this application. Image acquisition can be performed manually i.e., each image is
captured when the operator decides or automatically, using the interval timer function, which captures
each frame after a chosen time interval has passed. qDD also facilitates the immediate transfer of the
captured image into a specific folder in the computer. After the image acquisition and data transfer the
processing starts, which is performed using MATLAB.

2.2. The MATLAB code

MATLAB is programming environment where matrices are the key data structure in calculations [17].
Since digital images can be interpreted as 3D tensors, MATLAB seemed to be the optimal solution for
the development of this algorithm.

After starting the code, the number of the first image that enters the algorithm and the diameter of
the targets in mm is defined. The body of the algorithm takes place in a while cycle loop. The loop starts
with defining the prefix and the expected sequence number of a new image as a string and continues
with an if/else statement.
If the defined string exists in the chosen folder, the code reads the image as a $m$-by-$n$-by-3 array [17]. Subsequently the image is cropped to a size defined by the user to reduce data and is converted to greyscale. This conversion is performed by the \texttt{rgb2grey} function, which eliminates the hue and the saturation information of a true color image and retains the luminance [17]. In the next step the greyscale image is binarized using the Canny edge detector. The Canny edge detector applies a Gaussian filter to the image to reduce obvious noise and finds the intensity gradient of the image. Gradient magnitude thresholding is applied to remove false edge detections. Subsequently double thresholding is applied, strong pixels are defined, and edge detection is finalized using hysteresis [17, 18]. The binarized image will only include ones where edges were detected and zeros, where no edges were detected. A cutout of the true color image and the Canny image are shown in figure 1.

![Figure 1](image1.png)

**Figure 1.** True color image of the black disk target markers on a white background (left) and binarized image using the Canny edge detector (right).

In the next step a sample target marker is defined by choosing a submatrix of the Canny image which displays a representative marker. This sample is then compared to the rest of the Canny image using normalized 2-D cross-correlation [17]. If the correlation coefficient for a pixel is above the selected value, the given pixel is classified as a center of a marker. Since some neighboring pixels might meet the condition, in a subsequent if statement the latter is eliminated, thus only true approximate center coordinates remain. This will be fine-tuned in the later development, so the program keeps the pixels with the highest correlation. The pixels with the value 1 closest to the approximate centers represent the edges of the black disk on the target. Using the \texttt{fitEllipse} function, proposed by Richard Brown, ellipses are fit to these edge points using linear least squares method (LSM) [19], thus calculating the center coordinates of the ellipse, the major and minor axes, and the axis rotation. An ellipse fitted to a target marker in a true color image and a Canny image are displayed in figure 2. By the modification of the \texttt{fitEllipse} algorithm the accuracy of these parameters is also determined. Current test results suggest a 0.01-pixel accuracy of center coordinate estimation.

![Figure 2](image2.png)

**Figure 2.** An ellipse fitted to the edge data of the target marker with a calculated center.

After these computations are performed the algorithm calculates the deformation by subtracting the respective center coordinates of the initial epoch from the given epoch. The deformations and their accuracy are scaled based on the calculated major axis of the ellipse and known real-world size of the target. This information is stored in vectors and displayed. Lastly the sequence number is increased by one.

The else statement displays a warning, that there is no new image in the folder, waits a defined amount of time and ends without increasing the sequence number, therefore the while cycle loop restarts. The loop runs until manually stopped. During development of an algorithm, it is suitable to include a flowchart, which is included in the next figure 3.
Figure 3. Flowchart
3. Results and discussions

The proposed algorithm was tested on modified images of a bridge load test, i.e., a chosen image was edited to simulate a stable camera, while the target markers were digitally edited to resemble displacement, as displayed in figure 4. The dimension of each image, while using a Nikon D7500 20.9 Mpx camera were 5568 x 3712 and 5568 x 750 after cropping.

\[\text{Figure 4. Part of the original image with target markers (above) and digitally displaced middle marker (middle and below).}\]

While data transfer between the camera and the computer may vary, qDD is stable, no crashes were recorded during testing, in contrast with other tested applications. The speed of the Wi-Fi transfer, around 1 MB/s seems suitable, however quicker solutions will also be looked for. The code produces results in consistent time intervals, i.e., the loop finishes a whole cycle in about 5 seconds for an image with the above-mentioned dimensions with 6 markers. Only slight variations in time occurred. The marker center disks had a diameter of 15 pixels and 100 millimeters, i.e., the Ground Sampling Distance (GSD) was 6.67 mm. Slight variations occurred for each marker, these variations depend on the distance of the specific marker from the image plane. The scale for each marker was individually calculated. The detection of the marker centers is performed with a standard deviation of one hundredth of a pixel, however more extensive testing is required to verify the accuracy of the outputs. Besides the Canny edge detector, other edge detection algorithms may be tested in the future. Results are summarized in table 1.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Processing of a single image} & \text{Accuracy of marker center using least squares} & \text{Deformation measurement between two epochs} & \text{Marker disk diameter} & \text{GSD} \\
\hline
5 \text{ s} & 0.01 \text{ px} / 0.0667 \text{ mm} & 10 \text{ s} & 15 \text{ px} / 100 \text{ mm} & 6.67 \text{ mm} \\
\hline
\end{array}
\]

Two major parts need to be implemented into the algorithm: in the beginning it is necessary to include the calibration parameters of the particular lens and camera configuration. This will be implemented using the Camera Calibration Toolbox integrated in MATLAB in later development [17]. To solve the problem of camera stability the camera should be placed on heavy foundations, that absorb the possible external vibrations, however this still will not completely solve the problem, since even minimal influences may change the internal and external orientation parameters [13]. Provided the target markers are approximately in the same plane smaller changes may be corrected analytically using projective transformation [13], in other cases more complicated calculations need to be performed [20]. These two major sections need to be implemented in future versions of the algorithm.
4. Conclusions
The proposed camera-computer-code configuration proved to be highly effective in a controlled environment, i.e., when used on test images. Camera control and data transfer using qDD is practical and stable. The in-house MATLAB code performs the marker detection in only about 5 seconds, while the calculations of deformations between two epochs take twice as long. These results speed and safety-wise are incomparably better than levelling methods. This is an algorithm in development and camera calibration and analytical stability correction needs to be included. The accuracy of marker center detection according to current tests is 0.01 pixel, which corresponds to 0.0667 mm in scale for this particular configuration, however more tests need to be performed to verify this accuracy.

Acknowledgment(s)
This article was created with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic within the Slovak Research and Development Agency, project no. APVV-18-0472.

This article was created with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic within the Scientific Grant Agency, project no. VEGA 1/0584/19.

References
[1] M. Marčiš, “Atypické prípady fotogrametrického merania kultúrneho dedičstva,” Habilitation thesis, 2019 (in Slovak).
[2] V. Abolhasannejad, H. Xiaoming, and N. Namazi, “Developing an Optical Image-Based Method for Bridge Deformation Measurement Considering Camera Motion,” Sensors, 2018.
[3] M. Marčiš, M. Fraštia, M. Bajtala, and Š. Sokol, “Fotogrametrické meranie zaťažovacej skúšky betónovej dosky,” Geodézia, kartografia a geografické informačné systémy 2016 [elektronický zdroj]: zborník článkov z 9. vedecko-odbornej konferencie, Demánovská dolina, SR, 21. - 23. 9. 2016 Geodesy, Cartography and Geographic Information Systems 2016, 2016 (in Slovak).
[4] M. Fraštia, “Metódy blízkej fotogrametrie,” K. Vošková, Jesenná univerzita architektúry, pp 90-92, 2008 – 2009 (in Slovak).
[5] A. Kopáčik, P. Kyrinovič, J. Erdélyi, I. Lipták, and Š. Lukáč, “A.G.K. 2015 s tematickým zameraním Inžinierska geodezia: zbornik príspevkov,” elektronický zdroj. 1.. vyd. Bratislava: Slovenská technická univerzita v Bratislave, Stavebná fakulta, 2015 (in Slovak).
[6] J. Halvonik, V. Borzovič, M. Fraštia , M. Marčiš “Zaťažovacia skúška mosta na Bojnickej ulici v Bratislave,” R. Sonnenschine, Betonárske dni 2020 a 6. post kongresov kolokvium SNK, 2020 (in Slovak).
[7] T. Luhman, S. Robson , S. Kyle, I. Harley, “Close Range Photogrammetry: Principles, Methods and Applications,” Dunbeath: Whittless Publishing, 2006.
[8] K. B. Atkinson, “Close-Range Photogrammetry and Computer Vision,” Whittles Publishing: Dunbeath, 2001.
[9] J.J. Lee, M. Shinozuka, “A vision-based system for remote sensing of bridge displacement,” NDT E Int, 2006.
[10] A. Gruen, “Towards real-time photogrammetry,” Photogrammetria, vol 42, Issues 5–6, pp. 209-244, 1988.
[11] M. Fraštia et al., “Možnosti metódy časovej základnice v digitálnej fotogrametrii,” Geodézia, kartografia a geografické informačné systémy 2016 [elektronický zdroj]: zborník článkov z 9. vedecko-odbornej konferencie, Demánovská dolina, SR, 21. - 23. 9. 2016 Geodesy, Cartography and Geographic Information Systems 2016, 2016 (in Slovak).
[12] M. Fraštia, M. Marčiš, M. Bajtala, Š. Sokol, and M. Plakinger, “Možnosti metódy časovej základnice v digitálnej fotogrametrii,” Geodézia, kartografia a geografické informačné systémy 2016 [elektronický zdroj]: zborník článkov z 9. vedecko-odbornej konferencie,
Demänovská dolina, SR, 21. - 23. 9. 2016 Geodesy, Cartography and Geographic Information Systems 2016, 2016 (in Slovak).

[13] M. Fraštia, “Vybrané aplikácie fotogrametrie v oblasti merania posunov,” Habilitation thesis, 2017 (in Slovak).

[14] M. Fraštia, “Kalibrácia a testovanie digitálnych kamier pre aplikácie blízkej fotogrametrie,” Dissertation thesis, 2008 (in Slovak).

[15] A. Hideghéty, and M. Marčiš, “Meranie objemov pomocou RPAS fotogrametrie,” Diploma thesis, 2019 (in Slovak).

[16] qDsrlDashboard, Controlling Nikon, Canon and Sony camera [Online] 2021. Available at: https://dslrdashboard.info/.

[17] MATLAB, MathWorks [Online] 2021. Available at: https://uk.mathworks.com/products/matlab.html

[18] J. A. Canny, “Computational Approach To Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, pp. 679-698, 1986.

[19] Richard Brown. fitellipse.m, MATLAB Central File Exchange. [Online] 2021 [Accessed April 13, 2021] Available at: https://www.mathworks.com/matlabcentral/fileexchange/15125-fitellipse-m.

[20] A. Habib, I. Detchev, and E. Kwak, “Stability Analysis for a Multi-Camera Photogrammetric System,” Sensors, 2014.