

## AUTOMATIC LABEL COMPLETION FOR TEST FIELD CALIBRATION

Jan Hieronymus

Humboldt-Universität zu Berlin  
Department of Computer Science  
Computer Vision  
Unter den Linden 6, 10099 Berlin, Germany  
jan.hieronymus@informatik.hu-berlin.de  
<http://www2.informatik.hu-berlin.de/cv>

Commission V, WG V/1

**KEY WORDS:** Targets Identification Automation, Geometric Camera Calibration, Distortion, Image Processing

### ABSTRACT:

For geometric camera calibration using a test field and bundle block adjustment it is crucial to identify markers in images and label them according to a known 3D-model of the test field. The identification and labelling can become challenging, especially when the imaging system incorporates strong unknown distortion. This paper presents an algorithm, that automatically completes the labelling of unlabelled anonymous marker candidates given at least three labelled markers and a labelled 3D-model of the test field. The algorithm can be used for extracting information from images as a pre-processing step for a subsequent bundle block adjustment. It identifies an unlabelled marker candidate by referencing it to two, three or four already labelled neighbours, depending on the geometric relationship between the reference points and the candidate. This is achieved by setting up a local coordinate system, that reflects all projection properties like perspective, focal length and distortion. An unlabelled point is then represented in this local coordinate system. These local coordinates of a point are very similar in the corresponding 3D-model and in the image, which is the key idea of identifying an unlabelled point. In experiments the algorithm proofed to be robust against strong and unknown distortion, as long as the distortion does not change within a small sub-image. Furthermore no preliminary information about the focal length or exterior orientation of the camera with respect to the test field is needed.

### 1. INTRODUCTION

When a camera is used for metric applications or within a network of cameras, the interior orientation of the camera must be known. The interior orientation of a camera is not precisely predictable, so it needs to be calibrated. The result of a calibration is usually a set of parameters describing the interior orientation (Brown, 1971), (Kraus, 2007). The usual procedure for geometric camera calibration using a test field is basically the following:

1. Take a couple of pictures of a well-known test field
2. Detect/Find markers (image processing)
3. Identify/Label markers
4. Estimate initial exterior orientation for each camera picture
5. Calculate bundle block adjustment with distortion model

A marker is considered some visible feature in the image, that can be (easily) detected and its position in the image can be measured precisely. Circular markers or checker-boards are most commonly used. Let's consider a marker to be found, if it was detected and its position has been measured. A found marker is still anonymous in the sense, that it is not known, which marker was found. It is only assured, that at the determined position, there is a marker.

Doing the calibration manually can take a lot of time, so it is desirable to have fast and automatic calibration facilities. Especially the third step, the identification and labelling of markers, is time consuming when performed manually. Therefore this paper will deal with the automation of the third step, the identification or labelling of already found, but anonymous, markers. The presented algorithm is independent of the type of markers, because it

operates only the position of the markers. No information about the exterior or interior orientation or the focal length of the camera is needed. The key idea of the algorithm is to exploit the property, that distortion is approximately constant in a small sub-image. This means, if only a small sub-image is considered, then the non-linearities introduced by distortion can be neglected. The task of the algorithm is the following: Given three to four labelled image points, a set of unlabelled image points and a labelled 3D-model of the position of the markers. Determine the correct label for all other found (yet anonymous) markers.

There are already successful approaches towards this challenge. For example there is the possibility to attach some identification code to the marker (Fiala, 2004), (Atcheson et al., 2010). In (Ahn and Rauh, 1998) a collection of commonly used circular coded targets is presented. Another option is to use so called exterior orientation devices, which are basically a set of markers with a known geometric constellation, as mentioned in (Fraser and Edmundson, 2000) and (Fraser, 1997). Given a set of known points and the exterior orientation of the camera, there is the classic approach to forward-project object points into the image using collinearity equations or homographies (Hartley and Zisserman, 2004). Forward projection maps a 3D-model onto the focal plane of a virtual camera. The commercial software Australis implements such a technique under the name "driveback". To do this, information about the exterior and interior orientation of the camera is needed. The exterior orientation can be estimated by many techniques, e.g. using 2D or 3D Direct Linear Transform (DLT) (Abdel-Aziz and Karara, 1971), (Luhmann, 2000), (Kraus et al., 1997) from three or four points respectively or close-form solutions as proposed in (Horn, 1987) or (Zeng and Wang, 1992). If the imaging system contains unknown distortion, forward-projection as well as homographies will indicate

wrong positions. Furthermore the initial estimation of the camera pose can be significantly difficult, if not even impossible. It is possible to estimate local homographies for different regions of an image, which would also adapt for the local projection properties. In contrast to the aforementioned methods the presented algorithm is not restricted to 2D → 2D projections, but adapts itself to the local dimensionality of the test field. In cases where full 3D information is needed, the algorithm will eventually set up a 3D → 2D projection. On the other hand, if target points reside on a line, it is sufficient to use only two reference points, resulting in a 1D → 2D projection. Furthermore there is no need for an explicit estimation of the exterior orientation.

Identifying markers is a task similar to star constellation identification. A recent overview about this topic can be found in (Spratling and Mortari, 2009).

Following this introduction, there is a description of the key ideas and principles of the algorithm. The third section will talk about performance and characteristics as well as limits by showing some examples. In the last chapter there is a conclusion and outlook for further activities.

## 2. ALGORITHM DESCRIPTION

### 2.1 Input Data

The required input data is:

- A labelled 3D-model of the marker positions
- A set of unlabelled marker positions found from a preceding image processing step
- Three or four (in rare cases two) labelled markers

It is noteworthy, that there is absolutely no requirement towards the type of marker (circles, ellipses, crosses, corners, chess-board fragments, retro-reflectivity etc.), as only its position is used for further calculations. The number of needed initial markers depends on the dimensionality of the test field. For planar test fields, three points suffice.

To provide modularity, the presented algorithm is not designed to being able to identify the initial markers on its own. That means, it relies on external help, e.g. manual identification of markers, constellation identification (see exterior orientation devices in (Fraser and Edmundson, 2000) or for an overview about astronomical star constellation detection approaches (Spratling and Mortari, 2009)), or sophisticated image processing—possibly in conjunction with coded targets. The use of local coordinates (see section 2.3) can also be used to describe the neighbourhood of a point. Such a description can be stored in a database and searched for based on markers detected in an unlabelled image. This technique has already been successfully implemented by the author been implemented, but is not part of this paper.

### 2.2 Basis and Coordinate System

This and the following subsection will recall some basics in linear algebra and give some definitions used in this paper. A point  $P$  in an image can be described by two coordinates, let's call them  $p = (u^P, v^P)^T$ . Coordinates need to be interpreted with respect to a basis. The most commonly used (canonical) basis is:

$$B_c = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (b_1 \quad b_2) = I_2$$

It is the identity matrix of dimension  $2 \times 2$ . The variables  $b_1$  and  $b_2$  are basis vectors. Where the units of the basis vectors may be for example pixels or some metric unit like millimetres. Interpreting coordinates  $p = (u^P, v^P)^T$  of a point  $P$  with respect to a basis  $B = (b_1, b_2)$  is the following equation:

$$P = B \cdot p = B \cdot \begin{pmatrix} u^P \\ v^P \end{pmatrix} = b_1 \cdot u^P + b_2 \cdot v^P \quad (1)$$

To characterise a point  $Q$  in a three-dimensional space can be done analogously:

$$Q = B \cdot \begin{pmatrix} x^Q \\ y^Q \\ z^Q \end{pmatrix} = b_1 \cdot x^Q + b_2 \cdot y^Q + b_3 \cdot z^Q \quad (2)$$

A coordinate system consists of an origin  $O$  and a basis  $B$ . If the origin  $O = (u^O, v^O)^T$  of an image coordinate system happens not to be  $(0, 0)$ , the interpretation of coordinates given with respect to basis  $B$  shall be done regarding also the origin:

$$P = B \cdot p + O = B \cdot \begin{pmatrix} u^P \\ v^P \end{pmatrix} + O = b_1 \cdot u^P + b_2 \cdot v^P + O \quad (3)$$

That means,  $(B \cdot p)$  is a vector starting from point  $O$ .

The same applies also in the three-dimensional case.

### 2.3 Local Bases and Coordinate Systems

It is possible, to build another 2D-basis  $B_\ell$  from any two other vectors, that are linearly independent. For the 2D case, it means, that they are not collinear (parallel). The index  $\ell$  in  $B_\ell$  indicates a so called "local" basis. Such a basis can be constructed from three (reference) points, which do not reside on a straight line, let's call them  $O_\ell, R_1, R_2$ . These three points shall be chosen local to another point  $P$ , i.e. from the neighbourhood of  $P$ . One point,  $O_\ell$ , is selected as the origin of the local coordinate system. Then the difference vector between each of the points  $R_1, R_2$  and the origin  $O_\ell$  is calculated:  $b_{\ell i} := R_i - O_\ell$ . Using these two vectors  $b_1, b_2$  as basis vectors, the local basis  $B_\ell$  can be defined as  $B_\ell := (b_{\ell 1} \quad b_{\ell 2})$ . Now it is possible, to characterise any other point  $P$  with reference to either the canonical coordinate system with basis  $B_c$  and origin  $O_c = (0, 0)$  by  $p_c = (u_c^P, v_c^P)^T$  or to the local coordinate system with basis  $B_\ell$  and origin  $O_\ell$  by  $p_\ell = (u_\ell^P, v_\ell^P)^T$

$$P = B_c \cdot \begin{pmatrix} u_c^P \\ v_c^P \end{pmatrix} = B_c \cdot p_c = B_\ell \cdot p_\ell + O_\ell = B_\ell \cdot \begin{pmatrix} u_\ell^P \\ v_\ell^P \end{pmatrix} + O_\ell \quad (4)$$

Translating coordinates from one coordinate system to another can be calculated using matrix inversion of a basis:

$$B_\ell^{-1} \cdot (B_c \cdot p_c - O_\ell) = p_\ell \quad , \quad p_c = B_c^{-1} \cdot (B_\ell \cdot p_\ell + O_\ell) \quad (5)$$

Figure 1 depicts the above mentioned relationships.

One property of a local basis or a local coordinate system is, that it reflects the local properties of the image with respect to perspective and distortion. That means, the local coordinates  $p_\ell$  of a point  $P$  with respect to the local coordinate system with basis  $B_\ell$  and origin  $O_\ell$  constructed from some nearby reference points have the property, that they are mostly independent from any underlying distortion and perspective. This makes them a robust description of the position of a point  $P$  with respect to the known set of three reference points  $O_\ell, R_1, R_2$ .

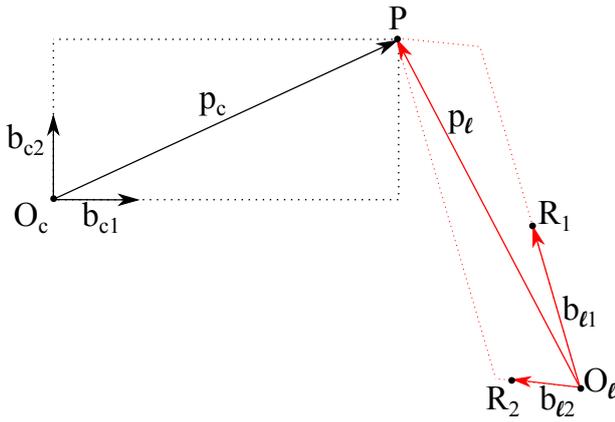


Figure 1: Representing a point  $P$  in the canonical coordinate system (black) and a local coordinate system (red).

Again the same idea can be extended to a three-dimensional space. In this case a set of four reference points  $O_\ell, R_1, R_2, R_3$  is used to construct a basis from three vectors  $b_{\ell i} := R_i - O_\ell, i = 1, 2, 3$ .

#### 2.4 Correspondence Between 3D-Model and Image

Local coordinate systems in the image and the 3D-model, that are constructed from corresponding reference points shall be called corresponding local coordinate systems. Having set up a pair of corresponding local coordinate systems, it is possible to predict the position  $p^I$  in the image from the position  $p^M$  of the corresponding point in the 3D-model. This is in fact easy through the use of corresponding local coordinate systems. As stated earlier, the local coordinates of a point are vastly independent of perspective and distortion. The main differences between the 3D-model and the image are, aside from the dimensionality, due to perspective and distortion (neglecting some noise). This means the local coordinates of the point in the 3D-model  $p_\ell^M$  are quite the same as for the image  $p_\ell^I$ . Henceforth it is straightforward, to predict the position of a point by the following steps:

1. Get canonical coordinates in the 3D-model  $p_c^M$
2. Translate to local coordinates in the 3D-model  $p_\ell^M$
3. Transfer local coordinates to the image  $p_\ell^I$
4. Translate to canonical coordinates in the image  $p_c^I$

The following formulae will show the steps 2 – 4 for the (special) case, that the unlabelled point resides on the plane spanned by three reference points. This means, that the "local basis"  $B_\ell^M$  in the 3D-model is not any longer a proper 3D-basis, but an under-determining set (see subsection 2.8) of two vectors spanning a plane that incorporates all four points (unlabelled point and three reference points).

$$\begin{aligned}
 p^M &= B_c^M \cdot \begin{pmatrix} u_c^{p^M} \\ v_c^{p^M} \\ w_c^{p^M} \end{pmatrix} = B_\ell^M \cdot \begin{pmatrix} u_\ell^{p^M} \\ v_\ell^{p^M} \end{pmatrix} + O_\ell^M = B_\ell^M \cdot p_\ell^M + O_\ell^M \\
 p_\ell^M &= \begin{pmatrix} u_\ell^{p^M} \\ v_\ell^{p^M} \end{pmatrix} \approx \begin{pmatrix} u_\ell^{p^I} \\ v_\ell^{p^I} \end{pmatrix} = p_\ell^I, \text{ fundamental assumption} \\
 B_\ell^I \cdot p_\ell^I + O_\ell^I &= B_\ell^I \cdot \begin{pmatrix} u_\ell^{p^I} \\ v_\ell^{p^I} \end{pmatrix} + O_\ell^I = B_c^I \cdot \begin{pmatrix} u_c^{p^I} \\ v_c^{p^I} \end{pmatrix} = B_c^I \cdot p_c^I = p^I
 \end{aligned} \tag{6}$$

Having the predicted coordinates  $p_c^I$  in the image space, it is straight forward to find that element from the set of unlabelled markers with minimal distance to the predicted position  $p_c^I$ . The label of  $P^M$  is used to label that nearest neighbour  $\hat{P}^I$ .

#### 2.5 Stability Considerations

To check, if coordinates expressed in local bases are numerically stable, the angle between the basis vectors is calculated. If the angle is within the range of 20 .. 160 degrees, the basis can be considered stable. The allowed range is chosen more or less arbitrarily and is not the result of detailed investigation.

To check the stability of a three-dimensional basis, it should also be tested, if the third basis vector is pointing sufficiently outside the plane formed by the first and second basis vector. If it is not, the system of three vectors would be nearly linearly dependent, which will lead to poorly defined coordinates even for nearby points. The proposed condition can be expressed having the third basis vector not being approximately perpendicular to the normal of the plane  $b_1 \times b_2$  formed by the first and second basis vector:

$$|\angle(b_3, b_1 \times b_2) - 90^\circ| > 10^\circ \tag{7}$$

Furthermore it can be useful to allow only bases, which are derived from basis vectors of similar lengths. This can be achieved by the constraint, that the ratio between the lengths of longest and the shortest basis vector may not exceed a certain number  $a$ .

$$\frac{\max_i(|b_i|_2)}{\min_i(|b_i|_2)} < a \tag{8}$$

Each pair of local bases is constructed from points of the neighbourhood of an unlabelled point. This has several positive consequences: Firstly, the projection including distortion is implicitly accounted for. Secondly, the local coordinates of the unlabelled point are small (0.5 .. 2.5) and therefore similar in the order of magnitude. The second part is advantageous in terms of stability. Any long basis vector will have the tendency to break the assumption, that the neighbourhood is local in terms of constancy of distortion. If the ratio described in (8) is large, the positional description of the unlabelled point may become unstable and a prediction might point to a wrong position.

#### 2.6 Plausibility Checks

Even though the presented method is already quite robust, the found correspondence should be checked for plausibility. Here two commonly used checks are proposed.

First there is a test, whether the distance between the predicted position  $p_c^I$  and the position of the nearest neighbour  $\hat{p}_c^I$  is less than a certain fraction, let's call that number  $f$ , of the length of the shortest basis vector of the local basis in the image.

$$|p_c^I - \hat{p}_c^I| < f \cdot \min_j(|b_{\ell j}^I|), \quad f < 1 \tag{9}$$

The shortest basis vector is a good estimator for the smallest occurring distance between markers in the considered neighbourhood. The prediction should find a marker within a small fraction of this distance to avoid ambiguities. A smaller  $f$  makes the test more reliable, whereas a larger  $f$  accepts predictions even for strongly distorted areas. In general  $f$  should be at most 0.5 to detect ambiguities.

Second there is a test, that the second nearest neighbour  $\tilde{p}_c^I$  is at least  $g$  times further away than the very nearest neighbour  $\hat{p}_c^I$ .

The number  $g$  should be greater than 1.

$$\left| \hat{p}_c^l - p_c^l \right| \cdot g < \left| \hat{p}_c^l - p_c^l \right|, \quad g > 1 \quad (10)$$

## 2.7 Final Test

As a last step all labelled markers can be tested for geometric consistency within their neighbourhood. For each point a stable local coordinate system is set up. For a quite large number of neighbouring points the image to 3D-model correspondence is verified. If expected points from the 3D-model cannot be found in the image or vice versa the label of the marker should be removed. In this way only extremely plausible labels are maintained.

## 2.8 Over- and Under-Determining Sets of Vectors

In the previous sections it was shown, how to describe the position of a point with respect to a local coordinate system constructed from a set of nearby reference points. If the set of reference points happens to contain one point more than the dimension of the space to be characterised, there is a good chance, that a valid basis (spanning the full space) can be constructed.

However if there are not enough points, obviously the full space cannot be spanned. A set of vectors spanning only a sub-space shall be called under-determining set. Only a hyper-plane or something even less dimensional can be described from an under-determining set of vectors. If a 3D-point resides on the same plane spanned by three reference points, this point can still be characterised unambiguously. Analogously it is possible to characterise a new point from two reference points, if all three points (unlabelled and reference points) lie on a common line.

On the other hand, if a set of vectors is larger than the dimensionality of the space spanned, the vectors are linearly dependent. Such a system shall be called over-determining set of vectors. Each point in the space can be described by different linear combinations of these vectors. In other words, there is not a unique set of coordinates characterising a certain point in the space. But if a set of coordinates is given, it will evaluate to exactly one point in that space.

So far in 2.4 there has been only a description on how to work with proper bases. But it is also possible to use full 3D-coordinates in the 3D-model and to use them in conjunction with an over-determining set of vectors in the image space. Analogously it is possible to describe a point in the 3D-model by only a single vector, if the unlabelled point lies on the same line as two reference points. This will then be evaluate in the image with an under-determining set of vectors.

## 2.9 Choosing the Next Unlabelled Point

As already mentioned, the fundamental assumption is, that for labelling an unlabelled point, a local coordinate system being constructed from nearby reference points. This indicates which point to choose next for labelling. A point, that has three (or four in a full 3D case) labelled neighbours and the distance to them is minimal should be investigated next for a given labelling state.

# 3. TESTS AND RESULTS

## 3.1 Accuracy Consideration and Simulations

Quantifying the power of a labelling algorithm could be done using two indicators: Coverage and Accuracy. Coverage is the ratio between detected (image pre-processing) markers and labelled

markers. Accuracy is the ratio between correctly labelled and labelled (correctly and incorrectly) markers.

It was found for this algorithm that coverage strongly depends on the marker detector (image pre-processor), which makes it a not very useful measure for this very algorithm. As can be seen from the two representative images 3 and 4 in the next subsection, the coverage is generally around 100 % (all detected markers get labelled), where the marker detection density is sufficiently high.

Concerning accuracy it is challenging to get a set of experimental ground-truth data. Creating this manually is extremely time-consuming. Using bundle-block adjustment with automatic point rejection is also difficult as a rejected point can also be rejected, if the measured coordinates are too rough, due to low image quality.

In order to test, if the algorithm works well or not, a simulator was set up. A camera with random focal length, distortion (described by the parameters as introduced by Brown and used in Australis) and exterior orientation was simulated to image markers of the targets presented in the next subsection. As mentioned earlier, only the marker positions are of interest. The main tasks of the simulator is henceforth to project the 3D-coordinates of the markers into the simulated camera using the collinearity equation. In a second step these positions were displaced according to the randomly generated distortion and some additive Gaussian noise accounting for measurement imperfections. The viewing angles were chosen to sometimes image the target only partially and sometimes totally. Furthermore about 10 % of the markers have been removed in order to simulate an (unrealistically bad) image pre-processor. As initial information one random marker and three of its neighbours were selected. This guarantees that the initial information is correct. It is noteworthy that currently the algorithm expects the given markers to be reliable.

The results are better to explain than to put into numbers. Some random distortions and exterior orientations destroyed the neighbourhood relationships completely, so that in these cases the coverage was nearly none. In contrast, if the resulting marker distribution was at least somewhat realistic, the coverage was found to be usually around 95 % or more for planar targets. If the algorithm approaches the non-planar edges of the 3D-target, it is sometimes able to hop over the edge. In general it needs to set up full 3D local coordinate system, which was not always provided by the random settings. For this reason the coverage was sometimes far less than for the planar targets, depending on the imaged parts of the target and the initial markers. It is therefore impossible to give a meaningful number of average coverage for this test. The reason is, that the parts of the target, which were imaged, had been chosen randomly as well. The probability that the algorithm steps over an edge is highly dependent on the parts of the target seen in the image.

## 3.2 Experimental Results

The algorithm was tested with different cameras and several 2D and 3D test fields, some of them can be seen in the figures 2, 3 and 4. The detection of markers (image processing) was done following an exercise held at the Albert-Ludwigs-Universität Freiburg (Rahmann and Burkhardt, n.d.). It is a basic and fast blob detector, which does not find every marker in any case. Every detected marker is indicated in the images with a red plus (+) and a number (the subsequently determined label). If there is a marker without any red plus (+) or number, this means, that no marker has been detected by the image pre-processor here. Accordingly these markers could not be labelled during the subsequent calculations. If the attached number is zero (0), no suitable label could be found. Following the blob detection, the position of the

markers is refined using an ellipse fit on the edge-pixels of each marker. This detector (blob plus ellipse fit) is quite fast and rejects some points of low image quality. That is the reason, why there are markers, which are not considered at all.

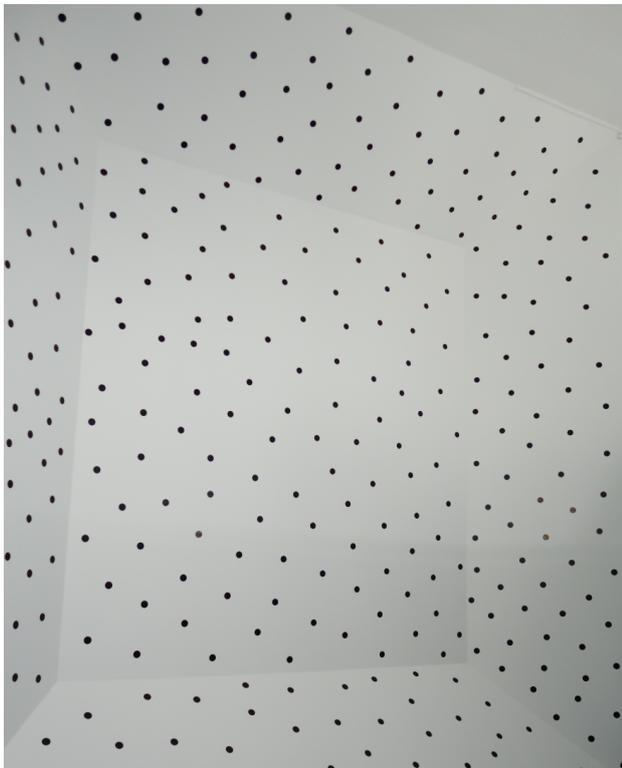


Figure 2: 3D test field imaged with surveying camera

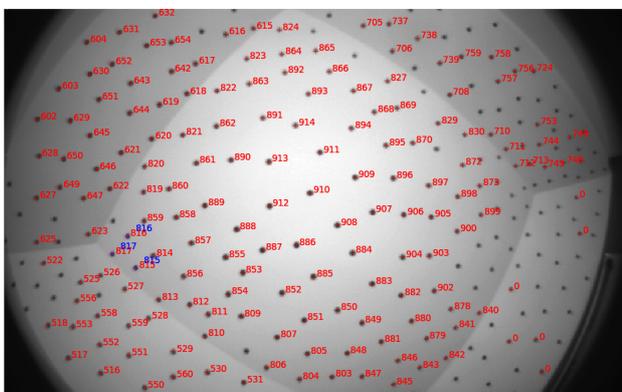


Figure 3: 3D test field imaged with a strongly distorting lens. Bottom right edge corresponds to the bottom edge in figure 2. Blue numbers were manually given as initial reference points. Red numbers were automatically found. Zero (0) indicates, that no suitable label was found. Few markers have not been identified as markers by the image pre-processor.

Labelling 2D test fields or planar parts of 3D test fields works perfectly fine as long as the markers are found during the image pre-processing. Even for images taken with a strongly distorting lens the labelling was successful as can be seen in figures 3 and 4. In early implementation it was observed in rare cases, that labels were put wrongly if several conditions—breaking the fundamental assumption of working in a small sub-image—appeared at the same time. The problem was, that in a relatively large region several markers were not detected. This created some isolated found

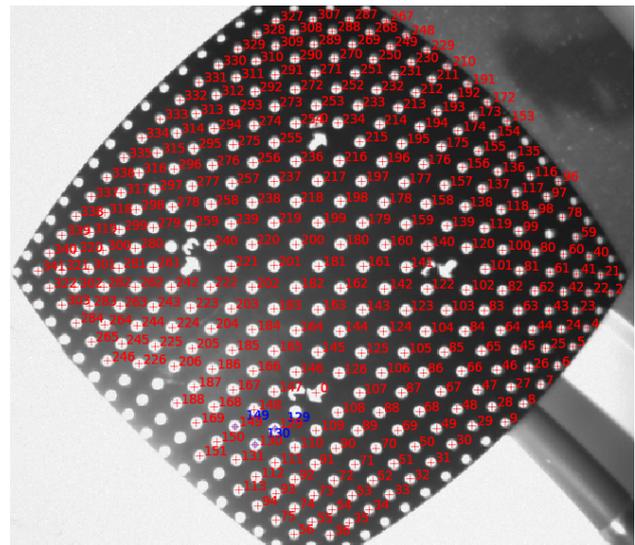


Figure 4: 2D test field imaged with a strongly distorting lens. Blue numbers were manually given as initial reference points. Red numbers were automatically found. Zero (0) indicates, that no suitable label was found. Few markers have not been identified as markers by the image pre-processor.

markers, whose neighbours were too far away for a proper prediction due to a strong distortion. This type of mislabelling can be detected and rejected by sanity checks, for example by not allowing unlabelled points to be too far away from the local origin. This example indicates, that the found markers on the test field should have a certain proximity to each other in order to work safely without explicit sanity checks. The necessary proximity is strongly dependent on the distortion of the camera. In any case, it is not essential that the markers are placed in some sort of regular grid in contrast to checker board calibration. As the algorithm works solely on the position of found markers, it is completely independent of the type of markers used, as long as there is a suitable image pre-processor.

If a 3D test field is to be labelled, the construction of a stable local basis in the 3D-model can be a bit more challenging than in 2D due to the fact, that another nearby reference point outside the plane spanned by the first two basis vectors is needed. If a fourth close-by reference point is found, everything works fine. In figure 3 the labelling result for a 3D test field is shown. There are three reference points depicted in blue. They reside on the central plane of the test field (labels starting with 8 or 9). Two interesting phenomena are visible: On the bottom right side of the target, there are a few isolated detected markers, which are outside the central plane. They could not be labelled, because no stable local 3D-coordinate system could be established from the nearby points from the central plane. In order to overcome at least small inclinations, the algorithm can be tuned to use local 2D-coordinates in environments, where the unlabelled point is "not too far away" from the plane covered by the 2D-coordinates. This enables the algorithm to label also 3D test fields that are locally quasi planar with only three initial reference points. Examples for this case are the three other sides next to the central plane.

A broad comparison to existing commercial close-range photogrammetry software-packages would be an interesting benchmark. Such a survey would be expensive and time consuming and can therefore not be part of the current study. Furthermore it is usually not known, which algorithms are implemented in closed-source software.

At last some trivia: Visualising the course of the labelling process shows, that usually neighbouring points are chosen as next point to be labelled. Indicating the reference points and the unlabelled point produces an unlabelled point being chased by three to four reference points. This looks similar to the good old game Pac-Man. For this reason, the algorithm received the nick name "Waka Waka", which is the classic sound connected with Pac-Man.

#### 4. CONCLUSION AND OUTLOOK

A new algorithm was presented, that is able to complete the labelling of anonymous markers detected by an image pre-processor with no information in advance about the camera. Tests show that the algorithm is robust also against strong distortion, which makes it a handy tool also for the calibration of strongly distorting wide-angle lenses.

A very useful extension is, to automatically detect the initial reference points. This can for example be accomplished by using coded markers. Another possibility is to geometrically describe the neighbourhood of a marker in the image and to find a corresponding description in the 3D-model. The latter has successfully been implemented, but is not subject of this paper.

It is possible to derive an algorithm, that finds corresponding markers in two images of the same scene based on sets of found anonymous markers. This would give the opportunity to label an unknown test field in two or more images consistently. This feature would facilitate initial surveys of test fields.

Making the algorithm robust against wrong initial markers would be a nice extension. If wrong initial markers are provided, the algorithm will tend to return a very small number of labels. In fact it would be better to refuse working with a distinct warning, that the initial markers might be wrong.

#### ACKNOWLEDGEMENTS

Many thanks to Martin Misgaiski-Hass who helped testing the algorithm and provided the nice graphic on coordinate systems.

#### REFERENCES

- Abdel-Aziz, Y. I. and Karara, H., 1971. Transformation from comparator coordinates into object space coordinates. ASP Symposium on Close-Range Photogrammetry, Falls Church pp. 1–18.
- Ahn, S. J. and Rauh, W., 1998. Mustererkennung 1998: 20. DAGM-Symposium. Stuttgart, 29. September - 01. Oktober 1998 (Informatik aktuell) (German Edition). Informatik aktuell 1998, Springer Berlin Heidelberg, chapter Circular Coded Target and Its Application to Optical 3D-Measurement Techniques, pp. 245–252.
- Atcheson, B., Heide, F. and Heidrich, W., 2010. Caltag: High precision fiducial markers for camera calibration. In: Vision, Modeling, and Visualization (2010), The Eurographics Association, pp. 41–48.
- Brown, D. C., 1971. Close-range camera calibration. Photogrammetric Engineering 37(8), pp. 855–866.
- Fiala, M., 2004. ARTag Revision 1. A Fiducial Marker System Using Digital Techniques. NRC/ERB-1117. November 24, 2004. 46 Pages. NRC 47419.

Fraser, C. S., 1997. Innovations in automation for vision metrology systems. The Photogrammetric Record 15(90), pp. 901–911.

Fraser, C. S. and Edmundson, K. L., 2000. Design and Implementation of a Computational Processing System for Off-Line Digital Close-Range Photogrammetry. ISPRS Journal of Photogrammetry and Remote Sensing 55(2), pp. 94–104.

Hartley, R. and Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Cambridge University Press.

Horn, B. K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A: Optics, Image Science, and Vision 4(4), pp. 629–642.

Kraus, K., 2007. Photogrammetry: Geometry from Images and Laser Scans (de Gruyter Textbook). Walter de Gruyter, Berlin.

Kraus, K., Jansa, J. and Kager, H., 1997. Photogrammetry, Vol.2, Advanced Methods and Applications. Dümmlers Verlag, Bonn.

Luhmann, T., 2000. Nahbereichsphotogrammetrie,. Herbert Wichmann Verlag, Berlin.

Rahmann, S. and Burkhardt, H., n.d. Praktikumsversuch Kamerakalibrierung. [http://lmb.informatik.uni-freiburg.de/lectures/praktika\\_brox/bvpraktikum/BVAnl\\_kam\\_kalib.pdf](http://lmb.informatik.uni-freiburg.de/lectures/praktika_brox/bvpraktikum/BVAnl_kam_kalib.pdf).

Spratling, B. B. and Mortari, D., 2009. A survey on star identification algorithms. Algorithms 2, pp. 93–107.

Zeng, Z. and Wang, X., 1992. A general solution of a closed-form space resection. Photogrammetric Engineering & Remote Sensing 58(3), pp. 327–338.