

Camera array image rectification and calibration for stereoscopic and autostereoscopic displays

Vincent Nozick

▶ **To cite this version:**

Vincent Nozick. Camera array image rectification and calibration for stereoscopic and autostereoscopic displays. Annals of Telecommunications - annales des télécommunications, 2013, 68 (11), pp.581-596. hal-00908586

HAL Id: hal-00908586
https://hal.science/hal-00908586

Submitted on 25 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Camera array image rectification and calibration for stereoscopic and autostereoscopic displays

Vincent Nozick

Received: date / Accepted: date

Abstract This paper presents an image rectification method for an arbitrary number of views with aligned camera center. This paper also describes how to extend this method to easily perform a robust camera calibration. These two techniques can be used for stereoscopic rendering to enhance the perception comfort or for depth from stereo. In this paper, we first expose why epipolar geometry is not suited to solve this problem. Second, we propose a non linear method that includes all the images in the rectification process. Then, we detail how to extract the rectification parameters to provide a quasi-Euclidean camera calibration. Our method only requires point correspondences between the views and can handle images with different resolutions. The tests show that it is robust to noise and and to sparse point correspondences among the views.

Keywords image rectification · stereoscopic displays · camera array · camera array calibration

1 Introduction

In recent years, stereoscopic technologies have been subject to an impressive growth and became incontrovertible in the movie maker industry. More recently, this technology has advanced from stereoscopic to autostereoscopic displays, involving more than two views and hence more than two cameras. The use of these multiview devices emphasizes technical issues in term of video stream synchronization, colorimetric correction, data compression, camera calibration or geometrical issues. This paper deals with the two last problems. More specifically, this paper presents an extension of an image rectification method [26] to perform a camera calibration for an arbitrary number of views with aligned camera center. This technique can be used for stereoscopic rendering to enhance the perception comfort or for depth from stereo. In this paper, we first expose why epipolar geometry is not suited to solve this problem. We propose a non linear method that includes all the images in the rectification process. Our method only requires point correspondences between the views and can handle images with different resolutions. Then, we present a method to perform a quasi-Euclidean calibration of the cameras from the parameters computed during the rectification process. The tests show that the method is robust to noise and to sparse point correspondences among the views.

Stereo image rectification consists in the transformation of two images of the same scene such that for each pixel correspondence between the two views, the two pixels are aligned along the image horizontal axis. The main purpose of image rectification in computer vision is to optimize depth from stereo methods both in computational time and in robustness. A less known application concerns stereoscopic image visualization where the vertical parallax between any point correspondences should be minimized. The large majority of the proposed methods are dedicated to 2-views systems. In this paper, we present an image rectification method for more than two views, as depicted in Figure 1. Rectifying more than two views implies some constraints on the cameras' devices, especially the fact that the cameras' center should be aligned. Multiple view image rectification is essential for autostereoscopic camera setup and can have some applications for depth from stereo methods.

Gaspard Monge Institute, UMR 8049
Paris-Est Marne-la-Vallee University, France
E-mail: vincent.nozick@univ-mlv.fr



Fig. 1 Multiple image rectification : (Up) height input images. (Down) the height rectified images.

A basic approach consists of pairwise image rectifications. In this paper, we propose an alternative method that ensures a computation providing a globally optimized solution using simultaneously all the views rather than pair by pair. Our method can handle images with different resolution, orientation and focal length. Then, we show how the computed orientation and focal length parameters are used to calibrate all the cameras.

2 Image Rectification

Image rectification methods have been known for long by photogrammetrists [30] and have been improved later by software approaches, like with Ayache and Hansen [5]. Most of these early methods involved the camera to be calibrated, i.e. to know the camera projection matrices. Then, this constraint has been released with methods taking advantage of epipolar geometry to align the two images. The main issue of the image rectification process is that the rectified pair is not unique, as show in Figure 2. Most of existing methods deal with how to find an image rectification that minimizes the image distortion or that preserves some predefined image properties.

Robert et al. [28] attempt to reduce the amount of distortion by finding the rectification transform that is closest to preserving orthogonality about the image centers. However, orthogonality is not an adequate criterion since even an Euclidean image rectification can involve a loss of orthogonality. Correcting this non-orthogonality might decrease the Euclidean property of the rectified images. Hartley [14] proposes a linear method to minimize the horizontal parallax among the point correspondences used for the rectification. Loop and Zhang [19] decompose the rectification process into affine and projective components. Al-Zahrni et al. [1] propose a method that prevent the rectification process from a distortion on a selected common plane specified from 3 point correspondences on the two images. Gluckman et al. [12] propose a method to find the transformation that best preserves the sampling of the original stereo pair, i.e. each pixel in the unrectified image should map to a single pixel in the rectified image. Mallon and Whelan [22] optimize each transformation in order to minimize perspective distortions, such that the rectified images look like the original images as closely as possible. Monasse et al. [24] perform an



Fig. 2 Image rectification solution is not unique. First row: two images to rectify. Second row: a possible image rectification. Third row: an horizontal scale preserves the rectificatio properties. Fourth row: a skew also preserves the rectificatio properties.

image rectification by three successive rotations. Isgrò and Trucco [16] do not explicitly compute epipolar geometry but generate a rectifying pair of homographies that are conform to the fundamental matrix form of a rectified image pair. Finally, Pollefeys et al. [27] propose a rectification method based on a reprojection onto a cylindrical surface instead of a plane in order to reach an optimal pixel distribution on the rectified images to avoid any pixel loss.

All these methods minimize an image distortion and thus are well suited for depth from stereo methods but not for stereoscopic rendering since the generated images are not consistent with a camera transformation that would result in a natural correction. In other words, there is no guarantee to obtain a pair of rectified images that corresponds or is close to an Euclidean camera setup. Moreover, most of these methods are based on epipolar geometry and hence cannot be directly extended to handle more than two views, as we will show in section 5.2.

3 Rectifying more than two views

3.1 Image rectification for stereoscopic rendering

In recent years, stereoscopic systems advanced from stereoscopic to autostereoscopic displays. These devices can provide more than two views simultaneously, usually around ten views, and the users do not need to wear any specific glasses. To provide a comfortable stereoscopic rendering, a stereoscopic image should avoid vertical parallax between correspondence points. This problem has been deeply studied (see Allison [2]) and can be corrected with an adequate image rectification if the cameras' center are aligned. This latter constraint makes possible to rectify an image with its right and left neighbors simultaneously. Finally, to reach orthostereoscopic rendering [18], i.e. the scene is perceived with its real geometric proportions, the image rectification should be Euclidean (i.e. metric). A quasi-Euclidean rectification where the respective camera focal lengths are coherent but only known up to a common scale factor, provides a quasi-orthostereoscopic rendering that is also acceptable.

Zhou and Li [37] propose an image rectification dedicated to stereoscopic rendering based on epipolar geometry that provides good results, but is adapted only for two views. Fusiello and al. [11] and later Fusiello et al. [10] propose a non-linear method to rotate and zoom a pair of projective camera such that they fit to a pair of rectified camera, according to the epipolar constraints. This approach is specially well suited for stereoscopic rendering since this method provides quasi-Euclidean image rectification. However, this method is hardly to be extended to multiple images rectification since it is based on an implicit epipolar relation between the two cameras. Our method is mainly related to this approach, especially concerning the quasi-Euclidean rectification.

3.2 Rectifying more than two views

Ayache and Hansen [5], Sun [33] and also An et al. [3] present some methods to perform an image rectification over three views. They combine a horizontal image rectification between a central image and a left image, and a vertical image rectification between the central image and a bottom image. This approach is designed to extend depth from stereo methods to three views, however this technique cannot be used for three aligned cameras.

Kang et al. [17] present an image rectification from multiple calibrated images. They adapt the images orientation and focal such that the cameras share a common image plane. Boutarel and Nozick [7] present a GPU image rectification that can support multiple images. This method requires the camera to be calibrated and performs a projection on a common image plane, followed by a back projection on a set of ideal rectified cameras. This method can handle slight misalignment of the camera center if the common image plane is chosen carefully since it minimizes the error around this common projection plane. The main drawback of these latter methods is the need of calibrated cameras when all the other methods can deal only with point correspondences.

3.3 Camera array calibration

The camera calibration process for a camera array can be a laborious task. The cameras can be calibrated one by one, using a 3d pattern and the "gold standard algorithm" detailed in [15].

A much easier way is to use a 2d pattern with the method presented by Zhang [36], like in the reconfigurable camera array described in [35]. Lucchese [21] presents an extension of Zhang method designed for camera arrays. The main drawback of these methods is that an operator has to go on the scene to move the pattern such it is captured by all the cameras with various orientations.

A simpler method, based on point correspondences, consists in making a rough camera calibration and then perform a non-linear refinement. The estimated camera projection matrices are usually computed with the

5-points algorithm [15,25]. Then, the final camera calibration is performed with a bundle adjustment on all the data [15,34]. If the process is successful, the camera calibration is accurate. The drawbacks of this method are first to require the cameras internal parameters to be known for the 5-points algorithm and second, to be very sensitive to the initial camera calibration. A moderately accurate initial calibration may lead to completely false result.

4 Notations

In this paper, we will follow the same notation as in [15]. We indicate column vectors of $\mathbb{P}^2$ in bold lower-case letters such as $\mathbf{x} = (x, y, w)^\top$ and column vectors of $\mathbb{P}^3$ in bold upper case letters such as $\mathbf{X}$. Row vectors are transposed column vectors, such as $\mathbf{x}^\top$. The y coordinate of a vector is denoted by $(\cdot)_y$. Matrices are denoted by upper case letters, such as $\mathtt{H}$. A set of points is denoted by an italic upper case letter like U and its cardinality by $|U|$.

5 Epipolar geometry and more than two views

Epipolar geometry [15] is an usual support for stereo image rectification methods. Thus, it can also appear to be a good solution to rectify multiple images.

5.1 Epipolar geometry

The epipolar geometry describes the relations that exist between two images and can be described by the following equation:

$$\mathbf{x}'^\top \mathtt{F} \mathbf{x} = 0$$

where $\mathbf{x}$ and $\mathbf{x}'$ are homogeneous representation of corresponding points and the fundamental matrix $\mathtt{F}$ is the algebraic representation of epipolar geometry. An epipole is the projection in one view of the camera center of the other view. Numerically, the epipoles $\mathbf{e}$ and $\mathbf{e}'$ are the right and left null space of $\mathtt{F}$ and can be computed by solving the homogeneous equations:

$$\mathtt{F}\mathbf{e} = \mathbf{0} \qquad \mathtt{F}^\top \mathbf{e}' = \mathbf{0} \tag{1}$$

5.2 More than two views

Rectifying more than two views involves that the camera centers are aligned. Indeed, camera rectification process consists in the image transformation such that the rectified images share the same image plane. Moreover, the horizontal axis of the rectified images should be parallel to the camera baseline such the epipoles are matched to infinity in the horizontal direction. To satisfy this constraint for each view, the camera centers should be aligned. Figure 3 presents a example where the cameras are not aligned, background and forground can not be rectied simultaneously.



Fig. 3 Three unaligned cameras. The left and rigth images are rectified together. The middle image's background is also rectified with these two images. Scaling the middle image would align the foreground but misalign the background (see the lights around the head).

If we consider more than two views where the camera centers are perfectly aligned, an extension of epipolar geometry is conceivable according to the fact that an image has a unique epipole whatever the other view. Indeed, if we consider N aligned cameras, the projection of the i^{th} camera's center on the j^{th} image is equivalent to the intersection of the line passing throw all the camera center and the j^{th} image plane. Since this epipole $\mathbf{e}_{ij}$ is constant $\forall i \in \{1 \cdots N\}_{i \neq j}$, then we note $\mathbf{e}_{ij} = \mathbf{e}_j$ and the equation (1) leads to $\mathtt{F}_{ij}\mathbf{e}_j = \mathbf{0}$, where $\mathtt{F}_{ij}$ is the fundamental matrix between the image i and j. This equation can be extended in a least square form:

$$\begin{bmatrix} \mathtt{F}_{1j} \\ \mathtt{F}_{2j} \\ \vdots \\ \mathtt{F}_{Nj} \end{bmatrix} \mathbf{e}_j = \mathbf{0}$$

Unfortunately, this approach is numerically unstable unless the cameras' center are perfectly aligned. Due to practical constraints, this condition is usually not

perfectly satisfied. The alternative that consists of computing the average epipole between an image and all the others is also unstable since a misalignment of a camera center has strong consequences on the computed average epipole. Imagine the situation where an epipole is at infinity with one view and on a finite point with another. Computing the average position would have no meaning.

As a consequence, epipolar geometry appears to be incompatible with more than two views, neither by an overall process nor by pairwise computation.

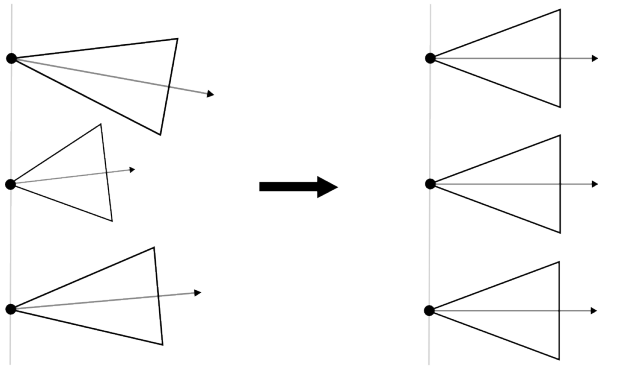


Fig. 4 Image rectification for each camera consists at most in a rotation and a change of focal length.

6 Outline of our method

Image rectification process can be considered as a transformation on the images such that they share a common image plane. Let's consider a set of N cameras with projection matrix $\mathtt{P}_i = \mathtt{K}_i \left[\mathtt{R}_i | -\mathtt{R}_i \mathbf{c}_i\right]$. We want to find the corrected cameras $\mathtt{P'}_i = \mathtt{K'}_i \left[\mathtt{R'}_i | -\mathtt{R'}_i \mathbf{c'}_i\right]$ such that all the cameras' focal plane become coplanar. This transformation can be expressed by a rotation around the optical center and an update of the focal length. Without loss of generality, the camera coordinate system can be associated to the first corrected camera coordinate system. Hence, the projection matrices of the corrected cameras can be expressed as:

$$\mathtt{P'}_i = \mathtt{K'}_i \left[\mathtt{Id} | -\mathbf{c'}_i\right]$$

where the camera center are:

$$\mathbf{c'}_i = \begin{pmatrix} x_i \\ 0 \\ 0 \end{pmatrix}$$

During the image rectification process, we do not have to consider the cameras' position. Thus, the relation between the initial and the corrected projection matrix can be defined as an homography $\mathtt{H}_i$:

$$\mathtt{H}_i = \mathtt{P'}_i \mathtt{P}_i^{-1} = \mathtt{K'}_i \mathtt{Id}_i \left(\mathtt{K}_i \mathtt{R}_i\right)^{-1} = \mathtt{K'}_i \mathtt{R}_i^{-1} \mathtt{K}_i^{-1} \tag{2}$$

Let $\widehat{\mathtt{R}}_i = \mathtt{R}_i^{-1}$ be the rotation applied to $\mathtt{P}_i$ to have the same the orientation as $\mathtt{P'}_i$. The equation (2) can be rewritten as follows:

$$\mathtt{H}_i = \mathtt{K'}_i \widehat{\mathtt{R}}_i \mathtt{K}_i^{-1} \tag{3}$$

As depicted in Figure 4, the rectification problem becomes how to find $\widehat{\mathtt{R}}_i$ and $\mathtt{K'}_i$ such that $\mathtt{H}_i$ rectifies the images. Given some point correspondences between each view, we want to find $\mathtt{H}_i$ such that the transformed point correspondences are horizontally aligned.

Once the rotations matrices $\widehat{\mathtt{R}}_i$ are retrieved, the only remaining information required to perform the camera calibration is the camera position. Assuming that the cameras are aligned, the camera order and position can be extracted from the point correspondences, as presented in section 9. The resulting camera calibration is quasi-Euclidean, meaning that the focal length are coherent among all cameras, but defined up to a common scale factor.

This method is not related to epipolar geometry and hence can be extended for an arbitrary number of views. Moreover, it does not involve any prior knowledge about the cameras' projection matrices and requires only point correspondences between the views. Furthermore, contrary to epipolar geometry based methods, the minimal number of point correspondences is not 8 but 4 (i.e. the minimal point correspondence to define an homography). Finally, this method is well suited for stereoscopic rendering since the operations guaranty a quasi-Euclidean rectification.

7 Multiple view image rectification

Consider a set of point correspondences $\{\mathbf{x}_{U^k}\}$ where U^k denotes the set of cameras involved by the k^{th} correspondence and where $\forall k,\ |U^k| \geq 2$ (i.e. each correspondence relates at least two views).

Let $\mathtt{K'}_i$ and $\widehat{\mathtt{R}}_i$ the rectifying internal parameters and rotation for each camera, then the rectified point correspondences should be horizontally aligned:

$$\left(\mathtt{H}_i \mathbf{x}_i^k\right)_y = y^k, \forall i \in U^k$$

where y^k represents the vertical coordinate of the rectified point k on each view. In practice, the y^k can be set as the average y-coordinate of the k^{th} rectified points since the rectified point correspondences should have the same y-coordinate:

$$y^k = \frac{\sum_{i \in U^k} \left(\mathtt{H}_i \mathbf{x}_i^k\right)_y}{|U^k|}$$

Hence, the homographies $\mathtt{H}_i$ should satisfy:

$$(\mathtt{H}_i \mathbf{x}_i^k)_y - y^k = 0 \tag{4}$$

We propose to find $\mathtt{H}_i$ with a non-linear process where each $\mathtt{K'}_i$ and $\widehat{\mathtt{R}}_i$ are optimized to satisfy equation (4) by minimizing the residual error e over all rectified point correspondences:

$$e = \sum_k \frac{\sum_{i \in U^k} |(\mathtt{H}_i \mathbf{x}_i^k)_y - y^k|}{|U^k|} \tag{5}$$

The error computation of each correspondence k is normalized by $|U^k|$ since the correspondence points may not involve all the views. We perform a bundle adjustment on $\mathtt{K'}_i$ and $\widehat{\mathtt{R}}_i$ from each view using Levenberg-Marquartd method.

To simplify the computation process, we reduce the number of parameters of $\mathtt{K'}_i$ by specifying a zero-skew, a unit aspect ratio and a principal point centered on the image. The only remaining parameter is the focal length. Hence, the cameras' internal parameter matrix is defined as follows:

$$\mathtt{K}_i = \begin{bmatrix} f_i & 0 & w_i/2 \\ 0 & f_i & h_i/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where w_i and h_i are respectively the width and height image resolution of the i^{th} view. An usual initial value for the focal length f_i is $f_i^0 = \sqrt{w_i^2 + h_i^2}$. As suggested in [23] and [10], it is numerically recommended to represent the focal length by a variable ranging in $[-1, 1]$ using zero as an initial value for the non-linear computation. It is common to expect f to have a value roughly ranging in $[f_i^0/3, f_i^0 \times 3]$, thus f_i is represented by a value $\alpha_i = \log_3(f_i/f_i^0)$. Hence, starting the non-linear process with the default focal length induces an initial value of α_i set to zero. Then, during the non-linear process, the current focal length is computed by:

$$f_i = f_i^0 3^{\alpha_i}$$

Each rotation matrix $\widehat{\mathtt{R}}_i$ is defined by three Euler angles θ_i^x, θ_i^y and θ_i^z such that $\widehat{\mathtt{R}}_i = \mathtt{R}_{\theta_i^z} \mathtt{R}_{\theta_i^y} \mathtt{R}_{\theta_i^x}$. For each view, the unknowns are the focal length represented by α_i and the three Euler angles θ_i^x, θ_i^y and θ_i^z which make a total of four unknowns per camera.

All the variables $(\theta_i^x, \theta_i^y, \theta_i^z, \alpha_i)$ are set to zero at the beginning of the procedure, meaning no orientation change on the cameras and the default focal length. To avoid a free rotation of all the images around the cameras' base-line, we select a reference camera and force its angle θ_r^x to be zero during the minimization process. We also let the focal length f_r of this camera constant for the same reasons.

As specified in equation (5), the point correspondences do not have to cover all the views. The only constraint is that each view should be directly or indirectly linked to all the others. Finally, our method can handle multiple view image rectification but is also very well suited for a two images rectification.

8 Different image resolution

In our method presented in section 7, we consider that all the input images have the same resolution. If it is not the case, the method should be adapted to overcome some computational errors. More specifically, a change in the focal length may lead to a misalignment for two images with different resolutions if the origin of the pixel coordinates is not centered. Indeed, the focal length variations provide centered transformations, but the image coordinates are not subject to this scale, as described in equation (3). Figure 5 shows that even with an adequate update of the focal length, the images are not aligned. This misalignment may lead to a rotation along the image baseline that will distort the image and degrade the image rectification. This problem does not occur if the pixel coordinates are centered, as depicted in Figure 6.

Therefore, our method should be modified such that the data is centered. Let $\mathtt{H}_i^c$ be the homography that centers the pixels of the i^{st} image, defined as:

$$\mathtt{H}_i^c = \begin{bmatrix} 1 & 0 & -w_i/2 \\ 0 & 1 & -h_i/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Equation (3) becomes:

$$\mathtt{H}_i = \mathtt{H}_{out}^{c-1} \mathtt{K'}_i \widehat{\mathtt{R}}_i \mathtt{K}_i^{-1} \mathtt{H}_i^c \tag{8}$$

Where the matrix $\mathtt{H}_{out}^c$ is a centering homography corresponding to the output image format that specifies a common pixel coordinates origin for all the images. In practice, $\mathtt{H}_{out}^c$ can be the centering homography of the smallest image. Moreover, the intrinsic parameters matrices should not include the origin of the image, since the data is already centered. These matrices become:

$$\mathtt{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Finally, the non-linear process can find the best parameters to rectify the images. It is not required for the input views to have neither the same image resolution nor the same focal length. The bundle adjustment process is summarized by the minimization function described in Algorithm 1.

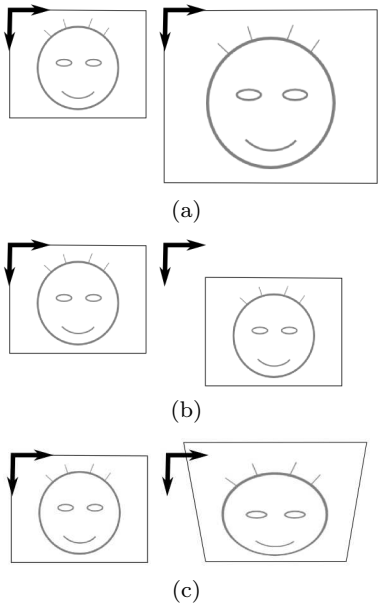


Fig. 5 (a) Input images with different resolution. (b) The minimization process will include a centered variation of the focal length of the second image, i.e. a scale of the bigger image performed in the principal point coordinate system. The scale coeficeint is adequate but the fact that it is not performed on the image coordinate system generates a misalignment. (c) This misalignment can be reduced by a combining a rotation around the camera baseline. However, this image rectification is far from optimal.

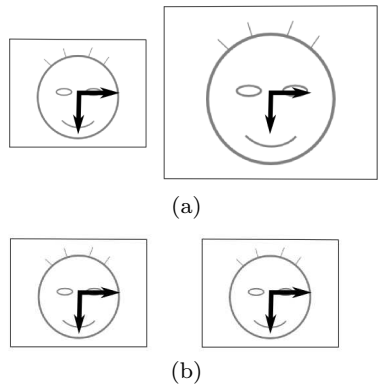


Fig. 6 (a) Input images with different resolution. (b) A centered variation of the focal length of the second image, where the image coordinate system is defined as the principal point. The sacle does not introduces any misalignment, as in Figure 5.

If all the input images have the same resolution, Algorithm 1 can be simplified by replacing the equation of line 3 with equation (6) and the line 7 with equation (3). However using Algorithm 1 provides very good results whatever the image resolution and does not alter the numerical performance even if the images have the same resolution.

Algorithm 1: minimization function

input : a set of candidates $\{(\theta_i^x, \theta_i^y, \theta_i^z, \alpha_i)\}$
a set of constant $\{(\mathtt{K}_i, \mathtt{H}_i^c)\}$ and $\{\mathbf{x}_{U^k}\}$
output: a set of $\mathtt{H}_i$ and the alignment error

get back the data
1 **foreach** camera i **do**
2 $f_i = 3^{\alpha_i}\mathtt{K}_i(0,0)$
3 $\mathtt{K}'_i = \mathrm{diag}(f_i, f_i, 1)$
4 $\widehat{\mathtt{R}}_i = \mathrm{eulerAngles}(\theta_i^x, \theta_i^y, \theta_i^z)$
5 **end**

compute the homographies
6 **foreach** camera i **do**
7 $\mathtt{H}_i = \mathtt{H}_{out}^{c\,-1}\mathtt{K'}_i\widehat{\mathtt{R}}_i\mathtt{K}_i^{-1}\mathtt{H}_i^c$
8 **end**

compute the average vertical coordinates
9 **foreach** correspondence $\mathbf{x}_U^k$ **do**
10 $y^k = 0$
11 **foreach** $i \in U^k$ **do**
12 $y^k += (\mathtt{H}_i\mathbf{x}_i^k)_y$
13 **end**
14 $y^k = y^k/|\,U^k|$
15 **end**

compute the global error
16 errorTotal $= 0$
17 **foreach** correspondence $\mathbf{x}_U^k$ **do**
18 error $= 0$
19 **foreach** $i \in U^k$ **do**
20 error $+= |(\mathtt{H}_i\mathbf{x}_i^k)_y - y^k|$
21 **end**
22 errorTotal $+=$ error$/|\,U^k|$
23 **end**

24 **return** errorTotal and $\{\mathtt{H}_i\}$

9 Cameras identification

In practical situations, it is usual to have a setup where the cameras are not identified, meaning that the order of the cameras is unknown. This is typically the case with usb cameras where the camera identifier depends on the usb port detection order. The geometrical camera order can be retrieved using point correspondences between the images.

9.1 Camera coordinate system

Assuming that the cameras are rectified with Algorithm 1, the camera position can be computed from the corresponding points. Indeed, if we consider the point correspondences in the rectified camera coordinate system, the point correspondences disparity between views is proportional to the horizontal position of the camera. Note that this assertion is true only in the case of rectified camera.

Algorithm 2: ordering the images

```
input  : a set of k point correspondences {F_i}
         (possibly incomplete) between n images.
output: an ordering array a
a = 0_n
foreach image I_i do
    foreach image I_{j>i} do
        left = 0
        right = 0
        foreach correspondence F_k do
            if F_k(i) and F_k(j) exists then
                d = F_k(i) - F_k(j)
                don't consider equality
                if d < 0 then right++
                if d > 0 then left++
            end
        end
        if left > right then a_i++
        else a_j++
    end
end
return a
```

Since the rectified images have the same resolution, the common image system coordinate can be defined by any image point. A direct solution is to select the top left corner of the image, with coordinates $(0,0)^\top$. Another easy solution is to choose the corrected principal point of the rectified images defined by the parameters $(w_i/2, h_i/2)^\top$ used on the centering homography $\mathtt{H}^c_{out}$ presented in Equation (7).

9.2 Ordering the cameras

As mention above, the rectification process involves that the disparity (in $x-$axis) of the point correspondences between views is proportional to the horizontal position of the camera. The method we propose consists in counting for each pair of image $\{I_i, I_j\}_{i\neq j}$ the number of point correspondences where the disparity from I_i to I_j is positive and respectively negative. These numbers will lead to a geometrical ordering of the two images. Extending the process to all pair of images will sort all the images. This method is described in Algorithm 2, where an element $\mathbf{a}_i$ indicates the index of the i^{st} camera.

Finally, in the case of aligned cameras with a non-uniform repartition, i.e. all the camera are not equidistant to their left and right neighbors, it is possible to extract the $x-$position of each camera from the disparity of the point correspondences. Indeed, equidistant cameras would involve a constant disparity between two pairs of consecutive cameras for a selected correspondence.

The computation of the camera $x-$position starts with the definition of a unit metric. It is standard to define the distance between the first and second camera as $\delta x_{0,1} = 1$. Then, the distance $\delta x_{0,i}$ between the first camera and the i^{st} camera is computed from the average of the disparity ratio among all correspondences k:

$$\delta x_{0,i} = \text{average}\left(\frac{d_k(0,i)}{d_k(0,1)}\right)_k$$

or recursively with:

$$\begin{cases} \delta x_{0,1} = 1 \\ \delta x_{0,i+1} = \delta x_{0,i} + \delta x_{i-1,i}.\text{average}\left(\dfrac{d_k(i,i+1)}{d_k(i-1,i)}\right)_k \end{cases}$$

Thus, each camera center can be estimated with:

$$\mathbf{c}_i = (\delta x_{0,i}, 0, 0)^\top \tag{9}$$

9.3 Point correspondence null space

We propose another approach to sort and find the position of the cameras using a Singular Value Decomposition (SVD). This approach is quite elegant but unfortunately not robust to false positive or missing point correspondences. Given a matrix of disparities between an arbitrary reference view and the other images, the centered and normalized disparities form a principal axis that can be extracted from a Principal Component Analysis or a SVD. This method is summarized on algorithm 3. The resulting vector $\mathbf{a}$ represents the sorted indexes of the cameras.

10 Cameras calibration

10.1 Rectified camera calibration

The rectified camera have the form:

$$\mathtt{P'}_i = \hat{\mathtt{K}}_i \left[\mathtt{Id}\right| - \mathbf{c}_i]$$

where the camera position $\mathbf{c}_i$ is defined by Equation (9):

$$\mathbf{c}_i = \begin{pmatrix} \delta x_{0,i} \\ 0 \\ 0 \end{pmatrix}$$

The internal parameter matrix $\hat{\mathtt{K}}_i$ does not corresponds exactly to the matrix $\mathtt{K'}_i$. Indeed, at the beginning of the minimization process, it is assumed that all the input cameras have a default focal length. For example, if all cameras have the same image resolution, they will have an identical default internal parameter matrix, when in

Algorithm 3: ordering the images with a SVD

input : a set of k point correspondences $\{F_i\}$ (possibly incomplete) between n images.
output: an ordering array $\mathbf{a}$

compute the disparity matrix
$\mathtt{A} = \mathtt{O}_{kn}$
foreach correspondence F_k **do**
  foreach image I_i **do**
    $\mathtt{A}_{ki} = F_k(i) - F_k(0)$
  end
end

normalize the data
foreach correspondence F_k **do**
  center and normalize the set $\{\mathtt{A}_{ki}\}_{i=0\ldots n-1}$
end

compute the right null space of A
$\mathtt{UDV}^\top = \mathrm{SVD}(\mathtt{A})$
$\mathbf{a}$ = last column of $\mathtt{V}$

normalize the data
$\mathbf{a}_m = \min(\mathbf{a})$
$\mathbf{a}_M = \max(\mathbf{a})$
foreach image i **do**
  $\mathbf{a}_i = (n-1) \times \dfrac{\mathbf{a}_i - \mathbf{a}_m}{\mathbf{a}_M - \mathbf{a}_m}$
end

return a

reality, their focal length may differ. The rectification process will find a scale on this internal parameter such the corrected images share a common image plane parallel to the camera baseline. Hence this process involves that the rectified camera have actually the same focal length and internal parameter matrix $\hat{\mathtt{K}}$. More precisely, the focal length scale factor computed during the minimization process is related to the initial camera parameter but not to the corrected one. Figure 4 describes this process. Thus, the internal parameters of the rectified camera has the following form:

$$\hat{\mathtt{K}} = \begin{bmatrix} f_r & 0 & w_{out}/2 \\ 0 & f_r & h_{out}/2 \\ 0 & 0 & 1 \end{bmatrix}$$

where f_r is the focal length of the reference camera and (w_i, h_i) defined by Equation (7) and the output image format.

The resulting camera calibration is called *quasi Euclidean*, in opposition to Euclidean (metric) or perspective calibration. Quasi-Euclidean calibration means that the focal length f_r of the cameras is defined up to a scale factor common for all the cameras. In our case, finding an adequate focal length f_r makes the calibration Euclidean for all the cameras, providing metric reconstruction (for computer vision applications) or an orthostereoscopic rendering (for stereoscopic applications). Choosing a "standard" focal length will provide quite correct 3d reconstructions or stereoscopic rendering. This is absolutely not the case of a perspective reconstruction where the results have nearly no chance to be acceptable.

10.2 Initial camera calibration

According to section 6, the initial estimation of the original camera projection matrices are given by:

$$\mathtt{P}_i = \mathtt{K}_i \left[\mathtt{R}_i \middle| -\mathtt{R}_i \mathbf{c}_i\right]$$

with

$$\mathtt{R}_i^{-1} = \widehat{\mathtt{R}}_i$$

and where $\mathbf{c}_i = (\delta x_{0,i}, 0, 0)^\top$ is the same camera center computed for $\mathtt{P}'_i$ in Equation (9).

As specified in section 10.1, the initial value for the focal length f_i^0 is a default value that is scaled during the minimization process, to converge to a final value $\hat{f}_i$. This process looks like how to transform similar focal lengths to find the corrected ones when in reality it is how to transform different focal lengths to make them identical. Hence, the initial camera focal length f_i can be found with the inverse transformation, that is:

$$f_i = f_i^0 \times \frac{f_i^0}{\hat{f}_i}$$

So finally:

$$\mathtt{K}_i = \begin{bmatrix} f_i & 0 & w_i/2 \\ 0 & f_i & h_i/2 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathtt{P}_i = \mathtt{K}_i \left[\widehat{\mathtt{R}}_i^{-1} \middle| -\widehat{\mathtt{R}}_i^{-1} \mathbf{c}_i\right] \tag{10}$$

Again, the calibration is quasi-Euclidean.

The resulting internal parameters have the principal point centred in the image. In practice, this approximation is acceptable for a large part of the camera systems. However, if more accuracy is required, this constraint can be relaxed using a bundle adjustment.

11 Results

We implemented our method in C++ using *Eigen* mathematic library [13] and our implementation of Levenberg-Marquardt method implemented from [15].

11.1 Input data

We tested out method on synthetic data and also on real data where the point correspondences were selected manually. Point correspondences detection is not the topic of this paper, there exist various automatic methods to find such correspondences that usually first find interest points [20,6], then match them from an image to the other and finally detect outlilers [9]. For multiple images, there exists dedicated methods like Avidan et al. [4] or Shafique and Shah [29], that can find robust point correspondences among several images.

The input images may also require a radial distortion correction. Indeed, radial distortion affects each part of the image differently and generates vertical disparity only in some parts of the image, leading to a less accurate image rectification. As all image rectification methods mentioned in this paper, we consider that the radial distortion has already been corrected on the input images as a preprocessing step. The radial distortion can be corrected using point correspondences with the method presented by Stein [31] or using the plumb line constraint with a non linear approach like with Devernay and Faugeras [8] or with a linear method as presented by Strand [32].

Finally, the tests of image rectification with more than two views are performed on images with camera centers aligned. A strong misalignment of the camera centers would lead to a rectification that may include strong vertical disparity between some corresponding pixels. Moreover, if the point correspondences are selected such they correspond to the same depth in the scene, the rectification process will succeed for this specific part part of the scene and may fail for the rest of the image, as shown in Figure 3 where the point correspondences where selected on the background of the scene. A misalignment of the cameras becomes a problem when the distance between a camera and the common camera baseline is not negligible compared to the distance between the cameras and the nearest objects of the scene. In any case, our method will minimize the global vertical disparity and rectify the images as much as it is possible.

11.2 Rectification tests for two views

We first tested our method on image pairs in order to compare our results with some state of the art methods, namely Hartley [14] since it is one of the most popular image rectification method and with Fusiello et al. [10] since this method is known to be one of the most effective in term of image preservation. Indeed, as presented in Section 2, most of the existing image rectification methods directly minimize the image distortion but are not consistent with a camera transformation. These methods are usually linear and thus fast, but the image distortion constraints may involve some stereoscopic visual artifacts that does not appear with non-linear methods consistent with the pinhole camera model.

The tests were performed on our implementation of [14] and on the author implementation of [10]. We selected some point correspondences manually and then computed the per-pixel error after rectification. The results are presented in Table 1. Our method performs every time better that [14] and is approximatively equivalent to [10], i.e. sometimes a bit better and sometimes a bit worse. Figure 8 depicts some results with both better and worse vertical disparity minimization compared to [10].

Table 1 Average disparity in pixel computed on images pairs using different methods.

Image pair	Hartley [14]	Fusiello [10]	our method
Birthday Cake	0.643632	0.629794	0.597752
Cluny church	0.434034	0.329592	0.324025
Carcassonne	0.648803	0.548186	0.584983
Beijing lion	1.19991	0.428227	0.577816

Finally, our method can compute an image rectification with at least 4 point correspondences where all the other methods require at least 8 point correspondences to compute a fundamental matrix. Figure 7 shows a result of our method with 4 point correspondences.

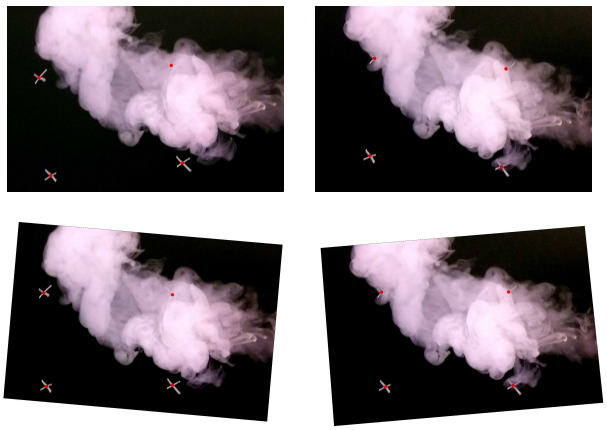


Fig. 7 Image rectification with only 4 point correspondences. In this kind of scene with smoke or fire, it is complicated to find more point correspondences.



Fig. 8 Left: Carcassonne images. Right: birthday cake. From top to bottom: input images, Hartley's method [14], Fusiello et al. [10] and our method.

11.3 Rectification tests for more than two views

We first tested our method on synthetic data consisting in a set of aligned projective cameras with controlled focal length, image resolution and orientation. A bunch of 3D random points are projected on each camera to get point correspondences. We performed this procedure on five (800×600) virtual cameras and 50 point correspondences with various camera settings:

1. identical cameras.
2. different orientations, same focal length.
3. same orientation, different focal lengths.
4. different orientations and focal length.

where "different" means a 10% variation between each view on the concerned parameter. In this situation, even if we have the ground truth camera calibration parameters, there is no quantitative measure of the distortion that reflects the quality of the image rectification. Therefore we compute the average of the y-axis disparity between the average y^k and the rectified points among each correspondence k. The procedure is repeated with an isotropic Gaussian noise on the correspondence points with various amplitudes. These results are summarized in Table 2 and show that the proposed method is robust to Gaussian noise. The tests also show that the method is a bit more efficient to correct orientation than focal length.

Table 2 Average disparity (in pixel) on the y-axis between 50 point correspondences with five (800×600) images applied on various data sets defined in section 11.3. The data is corrupted by an isotropic Gaussian noise with standard deviation $\sigma = 0.4$ under various amplitudes.

data set	zero-noise	Gaussian noise ampl.: 2 pixels	Gaussian noise ampl.: 5 pixels
1	0.00	0.54	1.36
2	0.06	0.55	1.36
3	0.13	0.57	1.33
4	0.11	0.56	1.37

The second part of our tests concerns the point correspondences density. We removed some point correspondences from the full data set such that some correspondences do not relate all the views. The results are presented in Table 3 and clearly show that our method still provides very good results, even when a large part of the point correspondences are missing.

Table 3 Average disparity (in pixel) **with sparse point correspondences** on the y-axis. The tests are performed with different correspondences density by randomly removing some correspondences. The full data contains 50 point correspondences relating the five (800×600) images. These tests are applied on various data sets defined in section 11.3.

data set	full data	90% data	60% data	40% data
1	0.00	0.00	0.00	0.00
2	0.06	0.11	1.01	2.00
3	0.13	0.07	0.07	0.07
4	0.11	0.04	0.06	1.16

Next, we tested our method on real data where we selected the point correspondences manually. Figure 9 shows a set of images that could be encountered for a stereoscopic setup, i.e. with only few correction to do to get rectified images. Figure 10 presents a set of images with converging optical axis. A zoom of the vertical disparity of the corrected images is shown in Figure 12. Finally, Figure 11 depicts a sample with strong orientation differences between the input images. The results shows that our method performs very well even with initial views far from their rectified form.

These tests show that our method succeed to perform a quasi-Euclidean image rectification and hence can be exploited for a quasi-orthostereoscopic rendering. The computation of the matrices $\mathtt{H}_i$ should be performed only once at the beginning of the video acquisition process and the video stream rectification can be computed on-line on the GPU as shown in [7].



Fig. 12 Zoom of Figure 10. Vertical disparity of some point correspondences before and after image rectification (top of the black tea packet, top and bottom frame corners of the wooden box).

11.4 Cameras ordering tests

The image ordering method summarized on Algorithm 2 has been tested on synthetic data. The tests focus on point correspondence accuracy effects and on missing data robustness. We first generated a set of rectified image point correspondences where the $x-$coordinates were subject to a Gaussian noise. Our program generated 50 point correspondences among height 400 pixels wide images. Then, the images were randomly shuffled. We automatically launched a test loop and counted the number successful retrieval of the original order. As shown in Table 4, our method is robust to point corre-



Fig. 9 Multiple image rectification : (Up) six input images. (Down) the six rectified images.

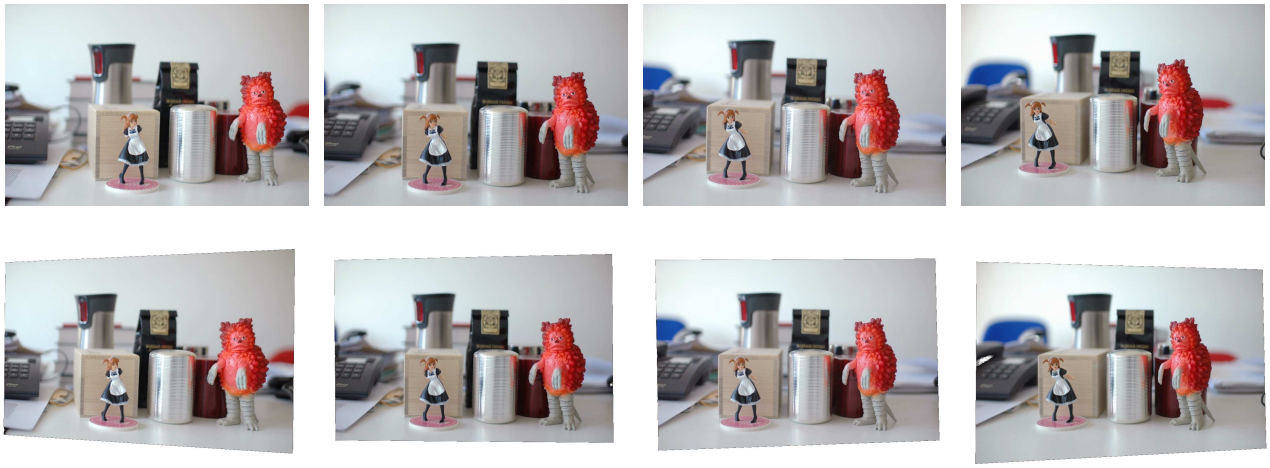


Fig. 10 Multiple image rectification : (Up) four input images. (Down) the four rectified images.

spondence inaccuracy. Indeed, the method is still robust to a noise of 10% of the image width (i.e. ±40 pixels) when a standard noise is usually expected to be 1% of the image width.

Table 4 Robustness of the ordering method (Algorithm 2) against noise on the point correspondence $x-$position.

Gaussian noise	success rate
zero-noise	100%
$\sigma = 10\%$ image width	100%
$\sigma = 25\%$ image width	64%

Then, we tested the robustness of the ordering method (Algorithm 2) against missing point correspondences between views. Our program generated 50 point correspondences among 8 views. The images were randomly shuffled and some randomly selected points are removed. We automatically launched a test loop and counted the number successful retrieval of the original order. As show in Table 5, our method is very robust to missing point correspondences. The method is still robust with 50% of the data missing.

Table 5 Robustness of the ordering method (Algorithm 2) against missing point correspondence between views.

missing data	success rate
50%	100%
70%	98%
90%	82%



Fig. 11 Multiple image rectification : (Up) four input images. (Down) the four rectified images.

11.5 Calibration tests

We tested our camera calibration method on real data. We present two sets of images and their respective camera representations. The first data set consists in 8 converging images shown in Figure 1. The corresponding camera reconstruction is presented in Figure 14. A simpler form of the same data set is presented in Figure 10 where only four of the eight images appear. Its corresponding camera reconstruction is shown on Figure 13. Finally, we present a data set with four nearly parallel cameras and the associated camera reconstructions depicted in Figure 15.

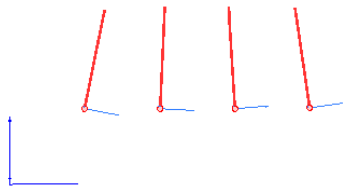


Fig. 13 Camera calibration of the set of images of Figure 10.

Moreover, we tested the calibration accuracy by measuring the reprojection error of 3D triangulated points from point correspondences. We also applied a bundle adjustment on our calibrated data to measure the reprojection error difference with an optimal calibration setup. Again, we used Levenberg-Marquardt method described in [15] to implement our bundle adjustment.

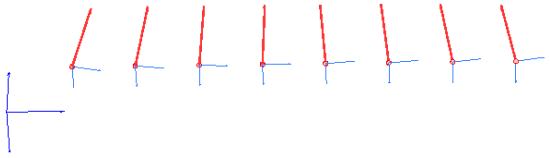


Fig. 14 Camera calibration of the set of images of Figure 1.

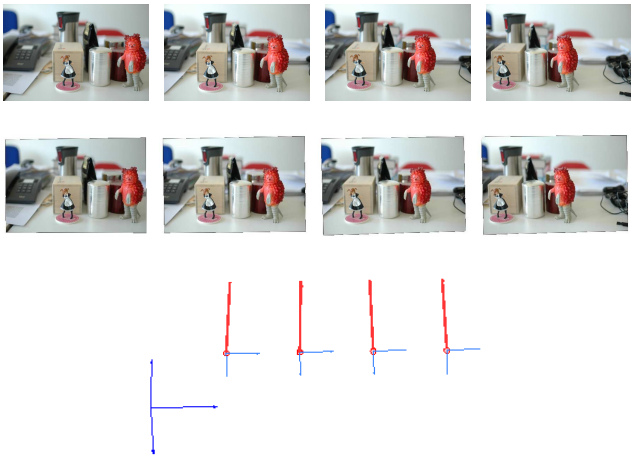


Fig. 15 Multiple image rectification: parallel setup. (Up) four input images. (Middle) the four rectified images, (Down) the calibrated cameras.

Our test first sorts the cameras and estimates the camera position with Algorithm 2. Then, the cameras are calibrated with Algorithm 1 and Equation (10). We perform a linear triangulation on the initial point correspondences to measure the reprojection error obtained

with our method. Then, we apply a bundle adjustment on the data to measure how far is our method from an optimal solution. Finally, we directly apply a bundle adjustment from the ordered camera, without our calibration method, i.e. with a good approximation of the camera position, but with camera principal ray parallel and default internal parameters. Table 6 presents the results of the test for two data sets.

Table 6 Average reprojection error per point correspondences. The tests concern the two data sets from Fig 10 (1000×664) and Fig 11 (900×600). In any case, the camera's center are estimated with Algorithm 2. The average reprojection error is computed on data computed with our method, with our method followed by a Bundle Adjustment (BA), or directly with a Bundle Adjustment on the sorted cameras.

data set	→ sort → our method	→ sort → our method → BA	→ sort → BA
Fig 10	15 pixels	0.6 pixels	4.1 pixels
Fig 11	45 pixels	3 pixels	no conv.

The results show that the proposed method is a good first estimation of the camera projection matrix. Moreover, our method is also a good initial solution for a bundle adjustment usually very sensitive to initial conditions. Table 6 shows that the default aligned camera projection matrices are not suited to be used directly for a bundle adjustment. Indeed, the bundle adjustment does not converge on the data set of Fig 11 and does not provide the optimal solution on data set from Fig 10 since one of the camera did not perfectly converge.

12 Conclusion

This paper presents an image rectification and calibration method that can handle more than two views. Our method does not involve any prior knowledge about the camera projection matrices and requires only point correspondences between the views. Moreover, the point correspondences between the views do not have to concern all the images simultaneously. Finally, the method supports input images with different image resolutions.

This method is well suited to remove vertical parallax for stereoscopic rendering without any damage on the perspective perception since the operations performed during the rectification process guaranty a quasi-Euclidean rectification. To our knowledge, this is the only method that provides quasi-orthostereoscopic images without full camera calibration. The camera sorting and calibration process is a crucial tool for many stereoscopic applications.

Our method has been validated by several tests concerning both the robustness with inaccurate point correspondence and sparse point correspondences over all views.

References

1. Al-Zahrani, A., Ipson, S.S., Haigh, J.G.B.: Applications of a direct algorithm for the rectification of uncalibrated images. Inf. Sci. Inf. Comput. Sci. **160**, 53–71 (2004). DOI 10.1016/j.ins.2003.07.012
2. Allison, R.: An analysis of the influence of vertical disparities arising in toed-in stereoscopic cameras. Journal of Imaging Science and Technology **51**, 317–327 (2007)
3. An, L., Jia, Y., Wang, J., Zhang, X., Li, M.: An efficient rectification method for trinocular stereovision. In: Proceedings of the 17th International Conference on Pattern Recognition, *ICPR '04*, vol. 4, pp. 56–59. IEEE Computer Society, Washington, DC, USA (2004)
4. Avidan, S., Moses, Y., Moses, Y.: Centralized and distributed multi-view correspondence. International Journal of Computer Vision **71**(1), 49–69 (2007)
5. Ayache, N., Hansen, C.: Rectification of images for binocular and trinocular stereovision. In: 9th International Conference on Pattern Recognition, *ICPR '88*, vol. 1, pp. 11–16. IEEE Computer Society, Rome , Italy (1988)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Journal of Computer Vision and Image Understanding **110**(3), 346–359 (2008)
7. Boutarel, F., Nozick, V.: Epipolar rectification for autostereoscopic camera setup. In: EAM Mechatronics 2010, pp. 133–136. Yokohama, Japan (2010)
8. Devernay, F., Faugeras, O.: Straight lines have to be straight. In: Machine Vision and Applications, vol. 13, pp. 14–24 (2001)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
10. Fusiello, A., Irsara, L.: Quasi-euclidean uncalibrated epipolar rectification. In: Proceedings of the 19th International Conference on Pattern Recognition (ICPR), pp. 1–4. Tampa, Florida, USA (2008). DOI 10.1109/ICPR.2008.4761561
11. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications **12**(1), 16–22 (1999)
12. Gluckman, J., Nayar, S.K.: Rectifying transformations that minimize resampling effects. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1**, 111 (2001)
13. Guennebaud, G., Jacob, B., et al.: Eigen v3. http://eigen.tuxfamily.org (2010)
14. Hartley, R.I.: Theory and practice of projective rectification. International Journal of Computer Vision **35**(2), 115–127 (1999)
15. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press, ISBN: 0521540518 (2004)
16. Isgrò, F., Trucco, E.: Projective rectification without epipolar geometry. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1**, 94–99 (1999)

17. Kang, Y., Lee, C., Ho, Y.: An efficient rectification algorithm for multi-view images in parallel camera array. In: 3DTV08, pp. 61–64 (2008)
18. Lambooij, M.T.M., IJsselsteijn, W.A., Heynderickx, I.: Visual discomfort in stereoscopic displays: a review. Stereoscopic Displays and Virtual Reality Systems XIV **6490**(1) (2007)
19. Loop, C., Zhang, Z.: Computing rectifying homographies for stereo vision. Computer Vision and Pattern Recognition **1**, 1125 (1999)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
21. Lucchese, L.: Geometric calibration of digital cameras through multi-view rectification. Image Vision Comput. **23**(5), 517–539 (2005)
22. Mallon, J., Whelan, P.F.: Projective rectification from the fundamental matrix. Image Vision Comput. **23**, 643–650 (2005)
23. Medioni, G., Kang, S.B.: Emerging Topics in Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA (2004)
24. Monasse, P., Morel, J.M., Tang, Z.: Three-step image rectification. In: Proceedings of BMVC, pp. 89.1–10 (2010)
25. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–777 (2004)
26. Nozick, V.: Multiple view image rectification. In: proc. of IEEE-ISAS 2011, International Symposium on Access Spaces, pp. 277–282. Yokohama, Japan (2011). E-ISBN: 978-1-4577-0715-5, print ISBN: 978-1-4577-0716-2, Digital Object Identifier: 10.1109/ISAS.2011.5960962
27. Pollefeys, M., Koch, R., Gool, L.V.: A simple and efficient rectification method for general motion. IEEE International Conference on Computer Vision **1**, 496 (1999)
28. Robert, L., Zeller, C., Faugeras, O., Hébert, M.: Applications of non-metric vision to some visually guided robotics tasks. Tech. Rep. RR-2584, INRIA (1995)
29. Shafique, K., Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 51–65 (2003)
30. Slama, C.: Manual of Photogrammetry, 4th edn. American Society of Photogrammetry (1980)
31. Stein, G.: Lens distortion calibration using point correspondences. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **0**, 602 (1997)
32. Strand, R., Hayman, E.: Correcting radial distortion by circle fitting. In: Proc. BMVC, pp. 9.1–9.10 (2005). Doi:10.5244/C.19.9
33. Sun, C.: Uncalibrated three-view image rectification. Image and Vision Computing **21**, 259–269 (2003)
34. Triggs, B., McLauchlan, P., Hartley, R.I., Fitzgibbon, A.: Bundle adjustment – a modern synthesis. Vision Algorithms: Theory and Practice pp. 153–177 (2000)
35. Zhang, C., Chen, T.: A self-reconfigurable camera array. In: ACM SIGGRAPH 2004 Sketches, SIGGRAPH '04, pp. 151–162 (2004)
36. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
37. Zhou, J., Li, B.: Image rectification for stereoscopic visualization. JOSA A **25**(11), 2721–2733 (2008)