EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Efficient registering of color and range images

Juan D Gomez*, Guido Bologna and Thierry Pun

Abstract

We present a simple yet highly efficient method to register range and color images. This method does not rely upon calibration parameters nor does it use visual features analysis. Our assumption is that if the transformation that registers the images is a mathematical function, we can approximate with little number of samples. To this end, thin-plate spline-based interpolations are used in this paper. Therefore, the registration of one point in our method takes only the solving of a nonlinear function. Drastically enhanced performances in the computational processing are attained under this condition. In fact, we show that ultimately our computational algorithm is independent of the complexity of the mathematical model underlying it. Finally, this paper reports on the results of experiments conducted with various range camera models that endorse the proposed method. Eventually, three key features can be derived from our method: practicality, accuracy, and wide applicability.

Keywords: Time-of-flight; Registration; Color images; Range images; Spline

Introduction

With the increasing use of 3D entertainment and multipurpose representation of virtual environments, range cameras continue to gain in popularity as prices are getting lower. While generally promising, there are shortcomings to the use of these sensors, which need to be resolved. Particularly, these cameras lack for color and some do not even provide a gray level or intensity image. This fact dramatically diminishes their scope for expansion into computer vision where image intensity is essential.

The advent of Microsoft Kinect (a cost-efficient solution; Microsoft Corporation, Redmond, WA, USA) partly alleviated this shortcoming by embedding a depth-color camera pair in one sensor (Figure 1). Unfortunately, Kinect's internal color camera often lags behind the needs for quality in mainstream applications. In such a context, the use of an external high-definition (HD) color camera began to draw the attention of scientists working on 3D imaging. In general, coupling range and HD color cameras benefit a broad range of applications in which neither alone would suffice.

Although a number of interesting ideas emerged from this problem, when it comes to couple two camera systems, image registration is perhaps the most affordable approach. Yet classic registration methods yield no suitable results in this particular case. Much is known about intensity images registration; however, there are still many open questions about registering an intensity image and a surface that lacks color and geometric features. In this spirit, the work here introduced presents a general method to register range and red, green, and blue (RGB) digital images. Unlike any other, our approach needs neither calibration of the cameras nor estimation of visual image descriptors.

Further, this paper reports evidence to endorse three features of our method: practicality, accuracy, and wide applicability. Overall, we show that it is a nonlinear function that underlies the registration of depth-color image pairs. In our method, samples of this function are manually taken to construct an interpolated surface model. We use this model to find corresponding shifts that are needed to register points of one image into the other. Also in the general case, nonlinear models are required because images are often corrupted by distortion. Hence, we study as well how to cope with the computation of nonlinear models with no loss of efficiency. In this way, we attain fairly important performance gains in computational terms. In fact, our method might be regarded as an approach to correct distortion in range images, an issue that remains challenging. Finally, when the need of accuracy permits it, our method could use a linear model that yields acceptable results in undistorted cameras such as Kinect.

* Correspondence: juan.gomez@unige.ch
Department of Computer Science, CUI, University of Geneva, Geneva, Switzerland

Figure 1 Leftmost column shows two mounted systems made up by depth sensors (Kinect on top, SwissRanger at the bottom) and a HD web camera. In the middle column, the re-sized depth maps and the color images (webcam) have been merged. Finally, in the rightmost column, we repeat the merging right after the depth maps have been registered into the color images using our algorithm. A twofold aim may be targeted: the addition of depth in a HD cam or the improvement in color resolution of Kinect. Moreover, images at the bottom of this figure may attest that our algorithm is valid for complex scenes which exhibit flexural geometry.

We therefore annex to this paper relevant information in this regard (Appendix).

This article is organized as follows. Section 'Background' describes our approach to the problem of registering images from two close-positioned cameras. A step-by-step description of the method proposed in this paper is given in Section 'A new approach'. Section 'The shifting basis function' deals with the mathematical deduction of the basis function underlying our method for range-color image registration. Finally, Section 'Algorithmic performance, experiments, and comparisons' presents an overview of practical sides of our approach, namely, algorithmic performance, distortion issues, experiments, and comparisons. Relevant discussion concludes this article with Section 'Conclusions'.

State of the art

The registration of RGB and range images of a same scene aims at matching color images and surfaces which lack color [1]. This problem remains largely unexplored in computer vision. Nonetheless, its applicability is fairly well defined. As examples it is worth to mention the following: 3D-laser extrinsic parameters estimation [2], color improvement in depth-color camera pairs [3], and joint-depth and color calibration [4]. Particularly, extrinsic calibration of colorless ToF (time-of-flight) cameras or 3D lasers is a relentless challenge that is usually approached in more refined ways: Hirzinger et al. [5] describe a multi-spline model that requires a robotic arm to know the exact pose of the intended sensor. Zhu et al. [6] describe a high-cost algorithm for fusing stereo-based depth and ToF cameras via triangulation. Unfortunately, a method that is easy-to-use, accurate, and applicable to a wide range of sensor has largely been missing.

An assessment of the general problem of image registration might be useful. In general, a vast range of techniques exists in the literature. Yet, more needs to be done to progress toward general solutions, if any. In 2003 Zitová and Flusser [7] published a complete review of the classic and recent image registration methods. Following them, Deshmukh et al. widened the spectrum of solutions by including updated advances in a more recent review in 2011 [8]. In all these works the image registration problem is defined as the matching of images of a scene taken from different sources, viewpoints and/or times. Yet, the former condition (inter-source) is limited to the variability of RGB sources only. Therefore, registration methods such as the one proposed by Yang [9] using artificial neural networks, or others that use belief propagation strategies as is the case of Sun et al. in [10], are likely to fail. This is mostly the case because they rely on the matching of color-based visual features common (or mappable) in both images.

In mainstream applications of computer vision, depth and color together as complementary cues about the scene are highly desirable [4,11-14]. Yet, while low resolution of the ToF camera is enough to segment depth-based areas, higher-resolution RGB camera allows for accurate image processing. In this spirit, Huhle et al. [1] present a novel registration method that combines geometry and color information in order to couple a PMD (photonic mixer device) camera with an external color camera. The alignment carried out in this work is based on the normal distributions transform [15] of the range images and a scale invariant feature transform [16] feature detector applied to the high-resolution color images. Thus, the authors claim to combine the robustness of the globally applicable feature-based approach and the precise local fitting via NDT. More recently, in 2011, Van den

Bergh and Van Gool [4] combined a digital camera and SwissRange ToF sensor using a regular calibration method for stereo systems [17]. The key idea of this approach was treating the output of the range sensor as though it was a RGB image.

In [18] the authors conduct a comparative study of some of the most important depth-and-color calibration algorithms. This work includes implementations as well as performance comparisons in both real-world experiments and simulation. Two algorithms stand out in this study: first, Zhang's method [19] that presents a maximum likelihood solution. This method uses checkboards and relies on co-planar assumptions. Also, manual correspondences need to be specified to improve calibration accuracy. Second is Herrera's method [20], in which authors claim to achieve features such as accuracy, practicality, and applicability. The method requires planar surface to be imaged from various poses and presents a new depth distortion model for the depth sensor. One more method called DCCT (depth-camera calibration toolbox) is studied in [18], though the article about this method is to date unpublished. Also, the authors in [19] have not shared their code so we only use [20] for comparisons later in this work.

Finally, as discussed by Han et al. in [7], the 'parallax' is perhaps the most challenging problem when it comes to image registration. Algorithms suffer from this problem by virtue of the assumption that the scene can be regarded as approximately planar. This is of course not satisfied by large depth variation in the images with raised objects [21]. Paulson et al. [22] presented in 2011 an outstanding idea to cope with the parallax problem by leveraging approximated depth information. Basically, their idea was to recover the depth in the image region with high-rise objects to build accurate transform function for image registration. The drawbacks from which this method suffers are fourfold: motion camera parameters are vital, significant manual work is needed, inaccurate approximations based on heuristics are very likely, and no real time. It is worth noticing that by feeding from a depth source (ToF sensor), the parallax phenomenon is no longer an issue in our method.

Background

An image is in theory an infinite assemblage of successive planes that eventually makes up the depth effect. Thus, in stereo-vision systems (stereo images captured by a camera rig), depth is discretized into many parallel planes (see Figure 2). The shift required to attain an exact overlap of two parallel planes is well known as the disparity [23]. Disparity is usually computed as a shift to the right of a point when viewed in the left plane (distance between blue 'left' and red 'right' points in Figure 2). Also in Figure 2, we can see that each pair of parallel planes presents a different amount of disparity (e.g., parallel planes captured at d_1 and d_2). Furthermore, the following observations may be made on the same figure:

a. Only the x-axis is prone to have disparity.
b. The disparity decreases as the distance of the planes (d_i) augments.
c. Disparity is constant for all the points into parallel planes.

It is worth noticing that (a) will be met provided that the stereo images are first rectified [6]. In other words, both images are rotated (until their epipolar lines get aligned [23]) to allow for disparities in only the horizontal

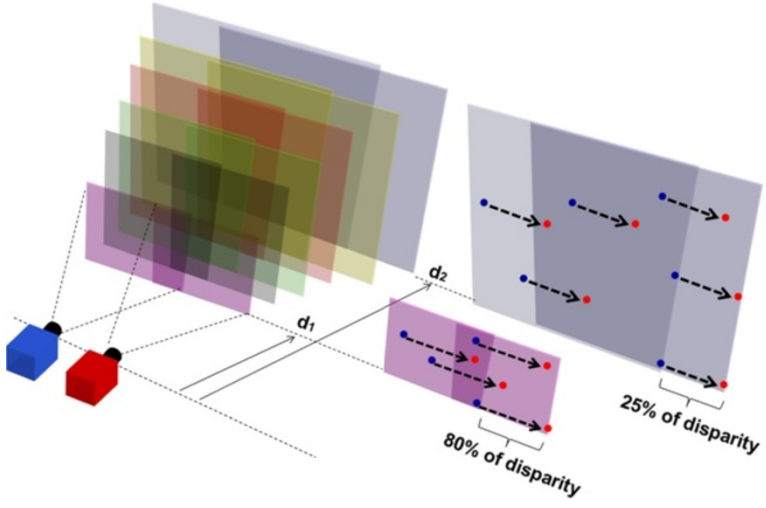

Figure 2 A camera rig. Parallel planes of the images increasingly overlap each other with distance. Two parallel planes need a constant shift (disparity) to fully overlap (matched). Under ideal conditions, this disparity must be constant and decreases with depth.

direction (i.e., there is no disparity in the y coordinates). Regarding (b), while the relation there pointed might be a common-place observation, the rate at which it is given will be further studied in this section (i.e., 'Disparity vs. depth'). In fact, it will be shown that this relation is nonlinear. Finally for (c), we want to stress that this is very much expected in an ideal system and this is the reason why we hypothesize about it. Nonetheless, this might not be the case in manually assembled camera rig studied in this paper. Our method, however, performs efficiently in any case.

Ultimately, the actual registration of two images demands a functional description of the displacements (disparities) between parallel planes across the depth. Thus, objects lying on an image plane (left) shall be accurately shifted to their counterparts on the parallel image plane (right). *Our idea is to sample as many pairs of points as possible* (*blue-red pairs in Figure 2*) *in as many parallel planes as possible, too*. Thus, we can interpolate the function that describes twofold information: firstly, the variation of the disparity between parallel planes, if any; secondly, the variation of the disparities with depth, which is expected to be nonlinear. Before we go any further with this idea, however, some important aspects need to be studied in order to endorse the assumptions made so far. This will be of help later in formulating our algorithm.

Disparity vs. depth

In Figure 3, an upper view of the standard stereo configuration with rectified images is presented. When aiming at recovering the position of P (a point in the space) from its projections p and p', we need to consider similar triangles (ΔPO_RO_T and $\Delta Ppp'$):

$$\frac{B}{Z}=\frac{(B+x_T)-x_R}{Z-f}\Rightarrow Z=\frac{Bf}{x_R-x_T}=\frac{Bf}{d}\ \Rightarrow Z(d)=\frac{Bf}{d} \quad (1)$$

where $x_R - x_T$ is the disparity (d), Z is the depth of P, and B represents the distance between the two cameras. The fixate location (f) is known as the distance in which the planes of projections are fixed.

In order to substitute one of the cameras (either O_R or O_T) by a depth sensor in Figure 3, few considerations are only needed: The range map is to be regarded as a regular image within which disparities with its colored peer may be encountered. Also, Z turns into a known variable accessible from the range map itself. This being so, (1) still holds when a color camera is replaced and should describe the relation 'disparity (d) vs. depth (Z)' as a nonlinear hyperbolic function. Finally, Figure 4a,b depicts the process for manually assessing of disparity over a point within a depth map and its peer in a color image given a specific depth distance.

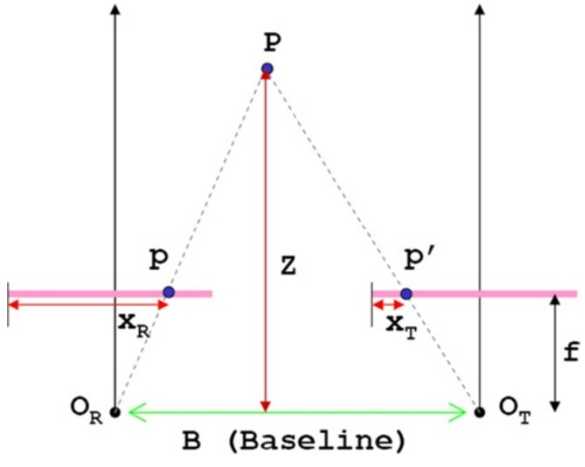

Figure 3 Aerial view of Figure 2, also known as standard stereo configuration. P is a point in the 3D space whose depth (Z) may be recovered using p and p' (its projections into the focal planes placed at f). B is the distance between O_R and O_T (the cameras). As long as the system has been rectified, the disparity may be assessed by subtracting x_R and x_T (the x coordinate values for p and p').

A new approach

Derived from the previous section, we aim here at aligning images from both sources. To do so, a spatial relation between coordinate systems will be set up. This relation in turn is described by a 2D vector flow which the function basis (expected linear by far) needs to be calculated only once. After images have been aligned, color and depth can be merged into one four-dimensional image [24]. Our method aimed at approximating this spatial relation using planar regressions is described as follows:

1. To sample as many planes as possible within the range of depth, several objects are placed at different distances in front of the cameras.
2. To capture nearly the same scene with the two cameras (color and range camera), two images (I_c and I_d) are taken as synchronized as possible (Figure 5).
3. Sufficient landmarks are selected in I_c along with their peers in I_d. For each landmark, threefold information is assessed:
   a) The x and y coordinates of the landmark in I_c, namely (x_c, y_c)
   b) The x and y coordinates of the landmark in I_d, namely (x_d, y_d)
   c) The depth of the landmark, namely D.

   Note that D is accessible likewise from I_d and pinpoints the distance plane on which the landmark was observed. Thus, $\Delta = (x_d, y_d) - (x_c, y_c)$ is but an

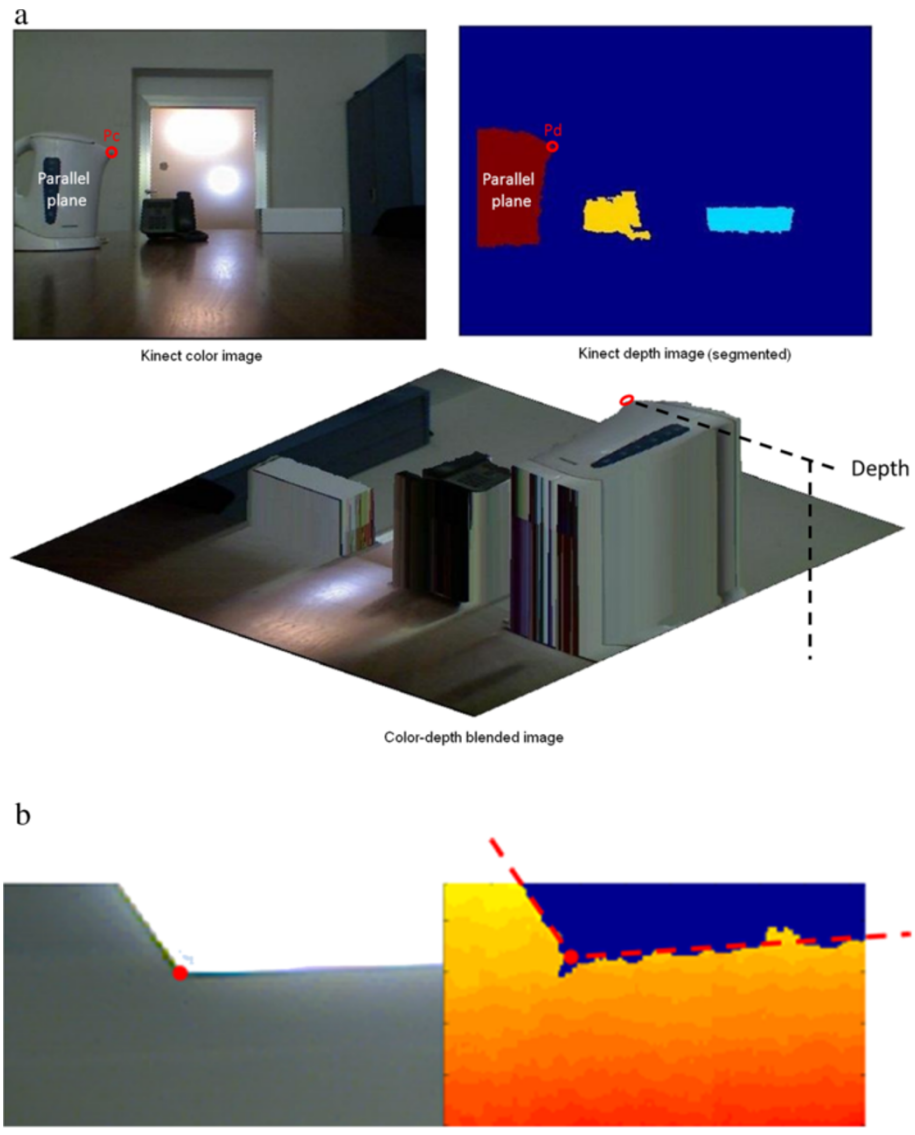

Figure 4 Manual assessment of disparity over a point its peer in a color image. (a) Pc represents a point given in a color image while Pd represents its counterpart in the range image. By measuring the distance (in pixels) between Pc and Pd, a sample of the disparity between the parallel planes (given at depth) can be assessed. Notice that for this figure, color and depth images were already calibrated (depth map was modified) and merged on the bottom image, therefore, the disparity was corrected to zero. **(b)** A point (red) manually marked within a zoomed area in both a color image (right) and a depth map (left). While manual markup to the right is regarded as an easy task, manual markup to the left is not. Automatic algorithms fail to detect this point due to boundaries' discontinuities caused, in turn, by physical issues of range measurement hardware. The dashed lines represent the human-eye intervention needed to approximate the boundaries and calculate the point precisely.

example of the shifting of the images at distance D and not elsewhere. In general, each landmark provides evidence of the offset of the images at a specific distance. In practice, taking as many distinct landmarks as possible for a given distance D is advisable at all (as many distances as possible). As noted in the previous section, the shifting Δ behaves linearly at D (i.e., disparity varies linearly into parallel planes).

4. Now the landmarks are used as a set of samples on which a global shifting function (Δ) can be interpolated. Eventually, this function can be regarded as a 2D vector flow describing the offset of the images. Hence, one function per coordinate is

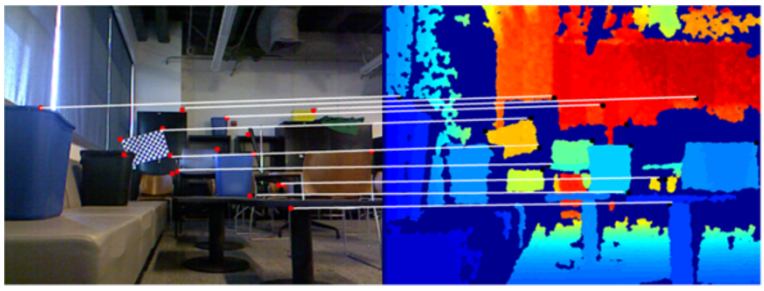

Figure 5 Synchronized images of the two cameras. I_c (left image) and I_d (right image). Red (I_c) and black (I_d) pairs of dots are landmarks manually selected. White lines coupling some of them make this figure more understandable. This amount of landmarks is quite enough for our method to work fairly well.

finally estimated, and Δ can be reformulated as follows:

$$\Delta = \left(\Delta x(x_d, D), \Delta y(y_d, D)\right) \tag{2}$$

The resulting function **Δ** is now vector-valued: it maps each point (x_d, y_d) in I_d to its shifted homolog (x_c, y_c) in I_c so that $x_d + \Delta x = x_c$ and $y_d + \Delta y = y_c$ for any given D. Yet, only few samples of this function are still known. Next section deals with the estimation of the model that best fits these samples. Also, this model will let us interpolate the function in its entirety.

The shifting basis function Δ

As studied in Section 'Disparity vs. depth', since the n data points (landmarks) do not show a linear distribution; therefore, we must use nonlinear fitting models to approximate Δ. Also, image distortion makes apparent the need of nonlinear models for Δ. This is because cameras suffering from distortion are known to wrap the image with nonlinear aspect [3]. Although in this paper we use an adaptable class of splines [25], there is no constraint in this regard. The thin-plate smoothing spline f used in this work to approximate $\Delta\mu$ ($\mu = \{x \mid y\}$) given a set of n data points or landmarks $\left(x_d^j,\ y_d^j, D^j\right) \cup \left(x_c^j,\ y_c^j, D^j\right)$, $\forall j \in \{1, \ldots, n\}$ can be regarded as a unique minimizer of the weighted sum:

$$\kappa E(f) + (1-p)R(f), \tag{3}$$

with $E(f) = \sum_j \left|\Delta\mu^j - f\left(x_d^j,\ y_d^j, D^j\right)\right|^2$ as the error measure, and $R(f) = \int(|\partial_1\partial_1 f|^2 + |\partial_2\partial_2 f|^2)$ the roughness measure. Here, the integral is taken over all of $\mathbf{R}^2$, $|z|^2$ denotes the sum of squares of all the entries of z, and $\partial_i f$ denotes the partial derivative of f with respect to its ith argument. The smoothing parameter κ in (3) is derived from preprocessing of the set of data.

Let now f be the shifting function [also known as Δ in (2)] so that f maps $(x_d, y_d) \rightarrow (x_c, y_c)$ for a given D. The general equation for f is given as follows:

$$f(x_d, y_d, D) = a_1 + a_x x_d + a_y y_d + a_D D + \sum_{i=1}^{n} w_i U\left(\left|\left(x_d^i,\ y_d^i, D^i\right) - (x_d, y_d, D)\right|\right). \tag{4}$$

Similar to (2), f may be also expressed in terms of its components as

$$f = \left(f_x(x_d, y_d, D), f_y(x_d, y_d, D)\right);$$

therefore, the shift of the x and y coordinates is described independently [i.e., $f_x = \Delta x$ and $f_x = \Delta y$ in (2)]. In Equation 4, n is the number of samples (landmarks) we shall use to interpolate f. Whereas a_1, a_x, a_y, a_D, and all w_i are the unknown coefficients we need to calculate. As for U, this is a special function underlying the thin-spline [25] defined as $U(x,y) = U(r) = r^2\log(r^2)$, with r being the distance $\sqrt{x^2 + y^2}$ from the Cartesian origin. Now, for the calculation of the unknown coefficients in Equation 4, we need to consider $r_{j,i} = \left|\left(x_d^i,\ y_d^i, D^i\right) - \left(x_d^j,\ y_d^j, D^j\right)\right|$, $\forall j$, and $\forall i \in \{1, \ldots, n\}$. Therefore,

$$K = \begin{bmatrix} 0 & U(r_{1,2}) & \cdots & U(r_{1,n}) \\ U(r_{2,1}) & 0 & \ldots & U(r_{2,n}) \\ \vdots & \ldots & \ddots & \vdots \\ U(r_{n,1}) & U(r_{n,2}) & \cdots & 0 \end{bmatrix}, n \times n; \tag{5}$$

$$P = \begin{bmatrix} 1 & x_d^1 & y_d^1 & D^1 \\ 1 & x_d^2 & y_d^2 & D^2 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_d^n & y_d^n & D^n \end{bmatrix}, n \times 4; \quad V = \begin{bmatrix} x_c^1 & x_c^2 & \ldots & x_c^n \\ y_c^1 & y_c^2 & \ldots & y_c^n \\ D^1 & D^2 & \ldots & D^n \end{bmatrix}, 3 \times n, \tag{6}$$

and,

$$L = \begin{bmatrix} K & P \\ P^T & \mathbf{0} \end{bmatrix}, \tag{7}$$

where $^{\mathrm{T}}$ is the matrix transpose operator and $\mathbf{0}$ is a 4×4 matrix of zeros. Then let $\mathbf{Y} = (\mathbf{V} \mid 0000)^{\mathrm{T}}$ be a vector of length $n + 4$. Finally, define $W = (w_1, w_2, \ldots, w_n)$ and the coefficients a_1, a_x, a_y, a_D by the equation:

$$L^{-1}Y = \left(W \mid a_1\ a_x\ a_y\ \ a_D\right)^{\mathrm{T}}, \tag{8}$$

the solution of $L^{-1}Y$ gives all the necessary information to construct f. Note that this is a matrix of size of $2 \times (n + 4)$, with each row providing w_i, a_1, a_x, a_y, and a_D for both, f_x and f_y.

Algorithmic performance, experiments, and comparisons

In this section, our algorithm for color-range calibration is outlined. Further, its computational performance is assessed too. Eventual concerns regarding nonlinearity along with efficient solutions are introduced and treated here in Subsections 'Practical test' and 'Efficient use of splines'. Finally, comparisons with related methods are conducted in this section. It is worth noticing though that our method is proposed as a general framework to couple any depth-color camera pair. We have limited the comparisons in the Section 'Evaluation of the method' to a specific case where our algorithm may be specifically applied as well i.e., internal Kinect calibration. The approaches whose efficiency is compared to that of our method in Subsection 'Evaluation of the method' are threefold:

A-1. Calibration of Kinect (mapping of depth data onto the RGB images) using typical checkerboard-based stereo calibration [4,26] i.e., assuming the range camera as digital
A-2. Calibration of Kinect using the drivers provided by manufacturer (PrimeSense, Tel Aviv, Israel)
A-3. Herrera's method [20] that uses a new depth distortion model to calibrate depth and color sensors

Algorithm

i. Construct matrices **K**, **P**, and **V**, using Equations 5 and 6.
ii. Construct matrix **L**, using **K**, **P** and $\mathbf{P}^{\mathrm{T}}$ as described in Equation 7.
iii. Let $(\mathrm{V} \mid 0000)^{\mathrm{T}}$ be the vector termed Y.
iv. Solve $\mathbf{L}^{-1}\mathbf{Y} = (W \mid a_1\ a_x\ a_y\ a_D)^{\mathrm{T}}$ to obtain a_1, a_x, a_y, a_D, and all w_i in Equation 4.
v. Get I_{c} and I_{d} from corresponding sensors.
vi. Find Δx for each $(x_{\mathrm{d}}, y_{\mathrm{d}}, D)$ solving f_x described in Equation 4.
vii. Find Δy for each $(x_{\mathrm{d}}, y_{\mathrm{d}}, D)$ solving f_y described in Equation 4.
viii. Shift all $(x_{\mathrm{d}}, y_{\mathrm{d}})$ towards $(x_{\mathrm{c}}, y_{\mathrm{c}})$.

Note that steps (i) to (iv) are related to the calculation of the spline model and are performed offline only once. Furthermore, if the system is ever decoupled, no recalculation of these steps is needed when recoupling. One can do the readjustment of the cameras by hand until acceptable matching of the images is attained. Also, we can see that the calculation of these offline steps is not computationally heavy. Typically, $\mathbf{L}^{-1}\mathbf{Y} = (W \mid a_1\ a_x\ a_y)^{\mathrm{T}}$ is solved as a system of linear equations of the kind of $\mathbf{A}^*\mathbf{x} = \mathbf{B}$. The computation of this linear system requires around $O(d^3 + d^2 n)$ computations (where n is the number vectors or landmarks and d indicates their dimension). Theoretically, in our method only 3 three-dimensional landmarks are needed (three points are enough to calculate a plane). In practice, however, the typical number of landmarks is approximately 20.

On the other hand, steps (v) to (viii) make up the workflow to be performed online. Particularly, we are concerned with steps (vi) to (viii) which are actually in the core of our computational approach. Having a_1, a_x, a_y, a_D, and w_i as constant data resulting from the offline phase, the solving of spline equation [steps (vi) and (vii)] requires moderate number of elemental operations [27]. This is perhaps the densest part of our algorithm. In next section, however, we will see how this part in this paper is further lowered in computational terms (*efficient use of splines*). Finally, step (viii), in turn, is but a constant array assignation. Overall, the complexity of our online algorithm is linear with the size of the images N [i.e., $O(N)$]. For images as I_{c} and I_{d} that usually do not exceed the order of megabytes [28], the complexity is noticeably low.

Practical test

Here below results of experiments conducted to endorse our model will be presented. Three camera systems were mounted as shown in Figure 6 (RGB-Kinect, RGB-CamCube, RGB-SR4000). Six different scenes (as described in Section 'A new approach' 1, 2) were captured as follow: first two using RGB-Kinect and, two more using RGB-CamCube (PMD Technologies Gmbh, Siegen, Germany) and the last two using RGB-SR4000. Following, corresponding shifting functions Δx and Δy were estimated for each scene. Each scene's landmarks (Figure 5) provided, then, two data sets to be interpolated. A total of twelve data sets were collected and twelve interpolations performed in this test. Nonlinear fitting models (thin-plate splines) were used; also all these models

Figure 6 Three mounted systems. From left to right: (RGB-Kinect) HD webcam-KinectMicrosoft, (RGB-CamCube) HDWebcam-PMDCamCube, (RGB-SR4000) HDWebcam-SwissrangeSR4000.

were cross-validated. Figure 7 shows the mean error for these interpolations.

Figure 7 reveals that in the general case [as expected from (1)], nonlinear fitting models are suitable enough to interpolate the Δx and Δy functions. In average, an error of just 2.3 pixels affected these interpolations. This error was totally expected as a consequence of the manual markup to which this method is subjected. In the first four cases (RGB-Kinect), the interpolating splines presented small roughness parameter ($\kappa \approx 0$, Equation 3). In other words, the interpolating surfaces were nearly flat (planes). In these cases and of course, depending on the needs of accuracy, we observed that Δx and Δy could be acceptably fitted by planes. This has to do with the range of depth in which the cameras are intended to be registered. When it comes to short ranges, it was a commonplace observation in our workflow that (1) could be tolerably approximated by linear surfaces (Appendix). Kinect, for instance, has a limited range rather small in comparison to other depth sensors, likewise, whether or not the images are undistorted matters as well. It is well known that Kinect also meets this feature since its images are internally preprocessed. This, of course, was not the case in the last eight interpolations were κ was rather large, making the nonlinearity apparent.

Efficient use of splines

As observed in (1), planar regressions would fail to keep accurate fitting of the functions Δx and Δy. Also according to Figure 7, splines happen to be a very precise method to model the nonlinearity inherent to the problem. In general, ToF cameras also suffer from distortions both on the measured depth and on the ray direction [3]. This makes even more apparent the need of nonlinear models in this problem. Although Kinect is not an exception, this sensor is calibrated during manufacturing. The calibration parameters come internally stored and are used by the official drivers. This might partially explain why linear models could be moderately acceptable in this case. Yet in the general case (including Kinect), the use of splines yields optimal results, which is otherwise unachievable. Unfortunately, the use of nonlinear spline models [steps (vi) and

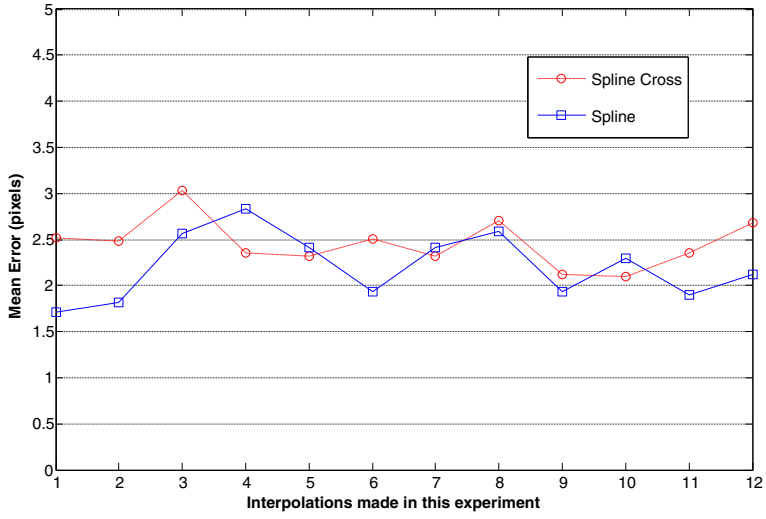

Figure 7 Validation: *X*-axis represents the twelve interpolations made in this experiment. First four interpolations belong to the scenes imaged by RGB-Kinect. The eight remaining belong to RGB-CamCube, and RGB-SR4000, respectively. *Y*-axis represents the mean error of the interpolations using linear models (planes), nonlinear models (splines), four cross-validated linear models, and finally, four cross-validated nonlinear models. Finally, the error marked by the crossed models is the mean error of the four validations.

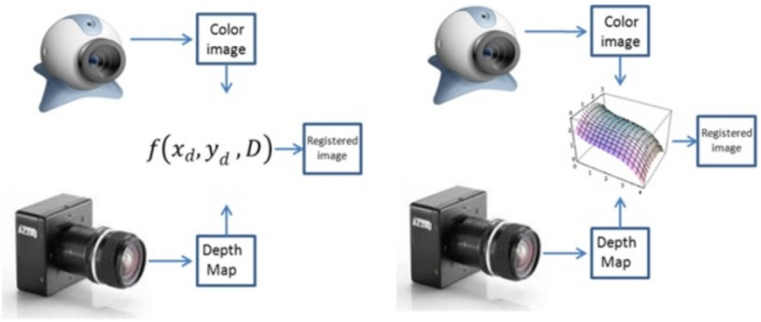

Figure 8 Simplification of the steps in the algorithm. (left) The analytical function of a spline is used to assess the shifting of the depth image in order to match the color image. This assessment has to be done for every depth-color pair. In other words, Equation 4 needs to be mathematically solved using the parameters found in steps (i) to (iv). (right) Using the parameters found in steps (i) to (iv), the spline surfaces of Equation 4 are built and stored in memory. Therefore, Equation 4 needs no longer to be solved for every depth-color pair. Instead, the surfaces are evaluated at needing.

(vii) of Subsection 'Algorithm'] could raise concerns regarding computational performance. This fact makes a lineal model more desirable for the proposed methods. Nevertheless, we will see that the use of a nonlinear model does not affect at all the actual efficiency of the proposed method.

To achieve the results shown in Figure 7, thin-plate smoothing splines [25] (described in Section 'The shifting basis function Δ) have been used to fit the surface underlying the data. The determination of the smoothing spline, however, involves heavily mathematical steps, such as the solution of linear systems. The solving thus usually takes a long time into our online routine [steps (vi) and (vii) of Section 'Algorithm']. In principality, this fact is drastically detrimental to our algorithm. To cope with this drawback, the regression model is no longer solved every time into the online workflow. Instead, we evaluate (offline) the splines so as to build their surfaces which are finally stored in memory. Therefore, we no longer have to analytically solve f_x and f_y described in Equation 4 to find Δx and Δy. In lieu of this, we pick values from the surfaces saved in memory. Thus, the online process becomes independent of the mathematical model (either linear or nonlinear), as we simply access the memory to read intended values. Eventually, steps (vi) and (vii) are lowered elementally, which enhance even more the performance of our method. Figure 8 roughly summarizes this idea.

Evaluation of the method

Using Kinect, three patterns which the edges are known to be lines are imaged from multiple views. A set of 20 pairs (depth and color) of raw images is gathered in the end. A manual segmented version of all the color images serves as ground truth. Three areas are segmented from each image i.e., the three patterns. Moreover, we register the 20 pairs of images by shifting the depth maps using A-1, A-2, A-3, and, the method being described in this paper. These shifted maps are automatically segmented

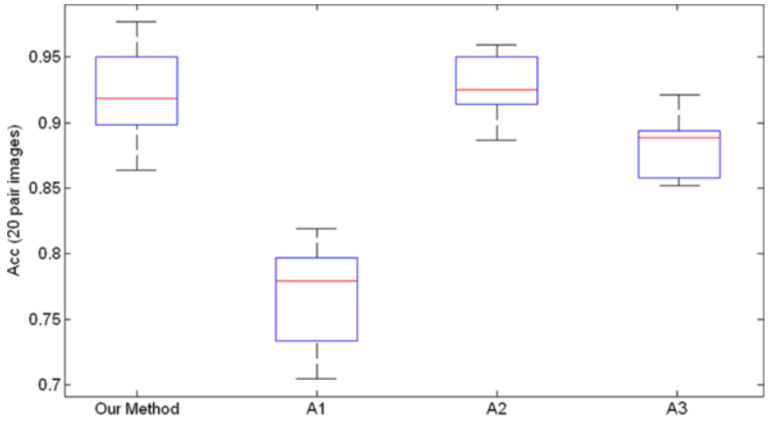

Figure 9 Calibration of internal depth-color camera pair using three different methods. Calibration of internal depth-color camera pair of Kinect (a specific case of registration) using three different methods. The accuracy Acc of our method for general registration of any depth-color camera pair is almost as accurate as that of the Kinect manufacturer.

Table 1 Computational performance

	Our method	A1	A2	A3
Time (s)	0.021	0.036	0.027	0.033
Potential fps	46	27	37	30

Our method is fairly efficient in computational terms.

in three areas as well. Finally, we compare the common areas between these segmented maps (sm) and those of the ground truth (gt). Common areas must overlap exactly each other under the assumption of perfect registration. Thus, for each pair of overlapped areas $\left(a_i^{\mathrm{gt}}, a_i^{\mathrm{sm}}\right)$ we assess its intersection $\left(a_i^{\mathrm{gt}} \cap a_i^{\mathrm{sm}}\right)$, $\forall i \in \{1, 2, 3\}$.

An indicator of the accuracy (***Acc***) of certain method to register a depth-color pair of images is given by $\frac{1}{6}\sum_{i=1}^{4} \mathrm{c} \left| \frac{\boldsymbol{a}_i^{\mathrm{gt}} \cap \boldsymbol{a}_i^{\mathrm{sm}}}{\boldsymbol{a}_i^{\mathrm{sm}}} + \frac{\boldsymbol{a}_i^{\mathrm{gt}} \cap \boldsymbol{a}_i^{\mathrm{sm}}}{\boldsymbol{a}_i^{\mathrm{gt}}} \right|$. Notice that ***Acc*** is expected to be 1 for images successfully registered and below in other cases. We also measure the time elapsed during the registration of two images using the four methods. Figure 9 and Table 1 summarize the results of this section.

An average *Acc* equal to 1 was not expected for any of the methods. This is mostly the case because segmented areas in range images are known to present highly noisy edges (Figure 10, bottom row). Thus, flawless intersection with the areas in the ground truth (Figure 10, middle row) is unlikely. As a consequence, the accuracy of the manufacturer (A2) can be regarded as a baseline. By showing no substantial difference with this baseline, our method roughly reaches the maximum expectation of accuracy in this experiment. Moreover, having a standard deviation slightly smaller, results obtained with A2 may be regarded as a more consistent. Nonetheless, this very fact allows our method to achieve better accuracy than A2 in some cases (not outliers). As for A3, much better accuracy than A1 was noticeably reached, although the method certainly failed to surpass the threshold of 90% accuracy. This leads our method to a slightly better performance with nearly 92%. It is worth noticing that A1, A2, and A3 are methods that require extrinsic and intrinsic parameters of both cameras. Hence, use of extensive calibration techniques with checkboards and heavily manual work is unavoidable. The efficiency of our method suppresses these procedures, as well as maintains an average accuracy for otherwise unreachable.

Finally, A2 is inextensible to the general problem of color and range images registration. This calibration is conducted during manufacturing and internally stored into the official drivers. Therefore, coupling the Kinect range sensor with an external color camera using A2 turns out to be of no use. On the other hand, A1 method does apply to the general problem. There is no apparent reason, however, to expect better accuracy by varying either of the cameras. The problem here relies on the treatment of noisy range images (with not even visual features) as highly defined color images. With regard to Table 1, it is worth stressing that both our method and A1 were implemented in Matlab, whereas, method A2 is an internal routine of the Kinect driver written is C++. Therefore, drastically better performance is expected for a binary compiled version of our algorithm.

Conclusions

In this work, a method for registration of color and range images was presented. We showed that the proposed method is practical, accurate, and widely applicable. Firstly, the problem of registering two close-positioned cameras and the underlying nonlinearity was studied in this paper. Following, an itemized outline of our method was given as well as a mathematical description of the nonlinear basis function. Furthermore, we presented a computational approach that preserves efficiency and accuracy regardless of

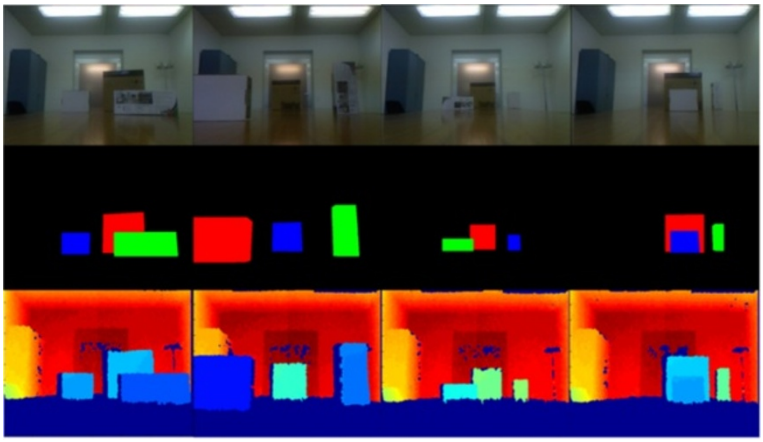

Figure 10 Some randomly selected images of this test. First row, color images. Second row, manual segmented images (ground truth). Third row, raw depth images.

the underlying mathematical model. This leads our method to be useful in correcting range camera distortion, a problem that remains challenging.

Overall, our method may be suitable in several applications: be it for Kinect color improvement, coupling of range and color cameras, or the calibration of the depth-color camera pair of Kinect. As for the latter, we presented an evaluation of our method that revealed as much accuracy as that of the manufacturer when matching the internal depth-color image pairs. Besides, our computational performance was fairly better compared to those of manufacture and other method aiming the same goal. However, our method is not yet fully automatic due to user markup required in the initial calculation of the shifting function. This is perhaps the major limitation of the work here presented. Therefore, future works will be focused on this line of research. We would like to explore automatic markup based on saliency points or corner detection. The problem to be tackled is that automatic algorithms will fail to detect, for instance, corners in range images due to boundaries' discontinuities caused by physical issues of range measurement hardware.

Unlike the others, the approach here proposed does not require intrinsic calibration of neither of the cameras. Also, the simplicity of its implementation leading to fairly important performance gains relies on the novel approach based on splines here presented. This method was motivated by the fact that cutting-edge depth-color cameras present low-quality RGB images. Even worse, some ToF sensors (e.g., CamCube) provide not even an intensity image, which dramatically diminishes their scope for expansion into computer vision. Additionally, this camera, among others, presents distortion as significant as to make calibration a must in the workflow. Typical checkboard-based calibration, however, is of no use when not even gray-level information is provided. Eventually, this problem may be solved if after registering the color and the depth images, the distortion is corrected in the former and then extrapolated to the latter.

Appendix

In this paper, we compared our spline-based approach with planar fitting models (linear approach). Although the general case is ruled by (1), in exceptional cases, planes yield acceptable interpolations (Section 'Practical test'). Therefore, in this section we will show how those planes were built to interpolate both, Δx and Δy. In this view, the problem can reduce to a linear regression in a three-dimensional space.

Let μ be either of the variables x, y so that $\Delta\mu$ denotes either of the functions Δx, Δy. Given a set χ of n data points (landmarks) $(\mu_{\mathrm{d}}^{(1)}, D^{(1)}, \Delta\mu^{(1)})$, $(\mu_{\mathrm{d}}^{(2)}, D^{(2)}, \Delta\mu^{(2)})$,..., $(\mu_{\mathrm{d}}^{(n)}, D^{(n)}, \Delta\mu^{(n)})$. We want to find the equation of the plane that best fits our set. A vector version of the equation of this plane can be formulated as follows:

$$(\mu_{\mathrm{d}}, D, \Delta\mu)^{\wedge} a - b = 0,$$

where a is a normal vector of the plane and b is a vector that results from the product of a and the mean of the set of data points (i.e. $b = a^{\wedge} \frac{1}{n} \sum_{i=1}^{n} (\mu_{\mathrm{d}}^{i}, D^{i}, \Delta_{\mu}{}^{i})$. Therefore, a happens to be the only variable unknown.

Principal components analysis can be used to calculate a linear regression that minimizes the perpendicular distances from the data to the fitted model [29]. In other words, given the three data vectors μ_{d}, D, and $\Delta\mu$, one can fit a plane that minimizes the perpendicular distances from each of the points $(\mu_{\mathrm{d}}^{(i)}, D^{(i)}, \Delta\mu^{(i)})$ to the plane, $\forall i \in \{1, \ldots, n\}$. In short, the first two principal components of χ define the plane; the third is orthogonal to them and defines the normal vector of the plane [30], namely a.

Competing interests

The authors declare that they have no competing interests.

Received: 8 November 2012 Accepted: 10 July 2013
Published: 19 July 2013

References

1. B Huhle, P Jenke, W Strasser, On-the-fly scene acquisition with a handy multi-sensor system. Int. J. Intelligent Syst. Tech. Appl. **5**(3), 255–263 (2008)
2. D Scaramuzza, A Harati, R Siegwart, Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes, in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2007* (San Diego, California, 2007), pp. 4164–4169
3. D Herrera, J Kannala, J Heikkila, Joint depth and color camera calibration with distortion correction. IEEE Trans Pattern Anal Mach Intell **34**(10), 2058–2064 (2012)
4. M Van den Bergh, L Van Gool, Combining RGB and ToF Cameras for Real-time 3D Hand Gesture Interaction, in *IEEE Workshop on Applications of Computer Vision (WACV)* (Kona, Hawaii, 2011), pp. 66–72
5. S Fuchs, G Hirzinger, Extrinsic and depth calibration of ToF cameras, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)* (Anchorage, Alaska, 2008), pp. 1–6
6. J Zhu, L Wang, R Yang, J Davis, Fusion of time-of-flight depth and stereo for high accuracy depth maps, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)* (Anchorage, Alaska, 2008), pp. 11–18
7. B Zitová, J Flusser, Image, Registration methods: a survey. Elsevier Image and Vision Computing **21**(11), 977–1000 (2003)
8. M Deshmukh, U Bhosle, A survey of image registration. Int. J. Image Process. (IJIP) **5**(3), 245–269 (1992)
9. SH Yang, Neural network based stereo matching algorithm utilizing vertical disparity, in *Proceedings of the Annual Conference on IEEE Industrial Electronics Society (IECON)* (Glendale, AZ, 2010), pp. 1155–1160
10. J Sun, N Zheng, H Shum, Stereo matching using belief propagation. IEEE Trans Pattern Anal Mach Intell **25**(7), 787–800 (2002)
11. B Han, C Paulson, J Wang, D Wu, Depth-based image registration, in *Proceedings of the Algorithms for Synthetic Aperture Radar Imagery* (Honolulu, 2010), pp. 542–562
12. A Staranowicz, F Morbidi, G-L Mariottini, Depth-camera calibration toolbox (dcct): accurate, robust, and practical calibration of depth cameras, in *Proceedings of the British Machine Vision Conference (BMVC)* (Leeds, 2012)
13. D Tilak, Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace. Appl Ergon **43**(4), 645–649 (2011)
14. M Andersen, T Jensen, P Lisouski, A Mortensen, M Hansen, T Gregersen, P Ahrendt, *Kinect depth sensor evaluation for computer vision applications* (Technical Report (Aarhus University, Department of Engineering), 2012)

15. P Biber, W Straßer, The normal distributions transform: a new approach to laser scan matching, in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)* (, Las Vegas, 2003)
16. DG Lowe, Distinctive image features from scale-invariant keypoints. J Comput Vision **60**(2), 91–110 (2004)
17. H Golub, L Van, *Matrix Computations*, 3rd edn. (John Hopkins University, Baltimore, 1996)
18. A Staranowicz, G-L Mariottini, A comparative study of calibration methods for Kinect-style cameras, in *Proceedings of the International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)* (Crete, Greece, 2012)
19. C Zhang, Z Zhang, Calibration between depth and color sensors for commodity depth cameras, in *Proceedings of the Workshop on Hot Topics in 3D (ICME)* (Barcelona, Spain, 2011)
20. D Herrera, J Kannala, J Heikkila, Accurate and practical calibration of a depth and color camera pair, in *Proceedings of the 14th International Conference on Computer Analytical Images (CAIP)* (Sevilla, 2011)
21. D Crispell, J Mundy, G Taubin, Parallax-Free Registration of Aerial Video, in *Proceedings of the British Machine Vision Conference (BMVC)* (Leeds, 2008), pp. 245–250
22. B Han, C Paulson, D Wu, Depth based image registration via 3D geometric segmentation. J. Visual Communication and Image Representation (JVCIR) **22**(5), 421–431 (2012)
23. D Marr, T Poggio, A computational theory of human stereo vision. Proc R Soc London, Ser B **204**, 301–328 (1979)
24. J Gomez, G Bologna, T Pun, A virtual ceiling mounted depth-camera using orthographic Kinect, in *Proceedings of the International Conference on Computer Vision (ICCV)* (Barcelona, 2011), pp. 50–55
25. FL Bookstein, Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans Pattern Anal Mach Intell **11**(6), 567–585 (1989)
26. J-Y Bouguet, *Matlab Camera Calibration Toolbox*, 2010. http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed 2 March 2012
27. Z Guo, Reduced complexity Schnorr-Euchner decoding algorithms for MIMO systems. IEEE Communication Lett. **8**(5), 286–288 (2004)
28. J Anderson, S Mohan, Sequential coding algorithms: a survey and cost analysis. IEEE Trans Commun **32**(6), 169–176 (1994)
29. B Moore, Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans Autom Control **26**(1), 17–32 (1981)
30. E Oja, Neural networks, principal components, and subspaces. Int J Neural Syst **1**(1), 48–71 (1989)

doi:10.1186/1687-5281-2013-41
Cite this article as: Gomez *et al.*: Efficient registering of color and range images. *EURASIP Journal on Image and Video Processing* 2013 **2013**:41.