# Real-Time Image Registration of RGB Webcams and Colorless 3D Time-of-Flight Cameras

Juan D. Gomez, Guido Bologna, and Thierry Pun

University of Geneva, Computer Science Department, Switzerland
{juan.gomez,guido.bologna,thierry.pun}@unige.ch

**Abstract.** In line with the boom of 3D movies and cutting edge technologies, range cameras are increasingly common. Among others, time-of-flight (TOF) cameras give it the ability to capture three-dimensional images that reveal object's distances. A shortcoming of these sensors however, lies in that the majority does not provide color information (not even gray). Therefore they are useless in computer vision applications for which color is crucial. The PMD [vision] ® CamCube 3.0 is one example of an expensive colorless TOF camera. In this work, we attempt the addition of color to this camera by means of inexpensive resources. A regular webcam is stuck on top of the CamCube and its color images are registered into the TOF distance maps. To get this done, we developed an algorithm to enable real-time registration based solely on depth. Thus, this algorithm requires neither intrinsic parameters nor mono-calibration of none of the cameras. We finally show a tracking application in which a stumble is foretold if an object approaches following a threatening trajectory.

## 1 Introduction

After long waiting, the first operational model of 3D-cameras based on the time-of-flight principle became available on the market. The colorless nature of these cameras however, has prevented the computer-vision community from using this technology. This is because while depth is very often an important feature, color (at least gray) is made always essential in computer vision. Indeed, cost-efficient solutions have come recently to alleviate this drawback. The Microsoft Kinect 3D sensor is an example of cheap TOF cameras that provide full color. Notwithstanding, this fun-only camera is indoor operational only. Besides, the accuracy of its distance images is found drastically diminished. Therefore, TOF 3D-cameras such as the PMD CamCube, continue to be highly desirable for solving unconstrained problems in natural environments.

This work introduces a method for the addition of color to the PMD CamCube camera. Furthermore, this technique can be extended to any another TOF sensor. Given that prices of these cameras still fall far from affordable, we use inexpensive resources to achieve this purpose. A regular webcam is only needed to turn the PMD CamCube operational in applications where the treatment of color and depth together is a must. We implement an online algorithm that matches color images from the webcam with distance images acquired with the TOF sensor in parallel. This algorithm estimates a shift function that in turn describes the misalignment of both

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 589–592, 2012.

**Fig. 1.** Apparatuses coupled in this work: The TOF sensor (PMD[vision] ® CamCube 3.0) and the web-cam (HD Webcam C525 Logitech)

coordinates systems. Therefore, we are able to remove the spatial variations between images caused during acquisition to build a 4-dimensions image (red, green, blue, depth).

Finally in this paper, we present an application aimed to identify which would be an obstacle only if its trajectory points straight to the camera after detection. For this application we combine depth-based segmentation with in-color tracking methods. Currently, this is an alerting method used to prevent users from stumbling in a navigation aid for visually impaired persons [1].

## 2 Image Registration

Here we aim at removing variations between images from both sources (TOF sensor and webcam) due to the differences in acquisition (points of view) which cause the images to be misaligned (their coordinates systems do not match). To register images, a spatial transformation is found which will remove these variations. This transformation has to be calculated once and needs no re-calculation unless the cameras have been decoupled. Once the images are aligned, color and time-of-flight-based distances can be merged into one 4-dimentions image. Our method aimed to approximate such a transformation, falls back on the seminal idea presented in [3]. The authors state that depth measured in stereo vision systems (two cameras closely embed) is discretized into parallel planes (one for each disparity value). Therefore, we assume that while shifts of coordinates between images (where all the depth planes are captured) cannot be described with a single number, shifts between local planes coordinates can certainly be. Our method that calculates all this numbers using planer regressions is described as follows:

- Several objects at different distances are placed in front of the cameras along the TOF range.
- A shot of the scene covering all the objects is captured using both cameras at the same time (one color and one distance/depth image).
- As many landmarks as possible are selected in the color image, as well as their peers in the counterpart distance image. Each landmark then, provides the $x$ and $y$

coordinates of a point into the color image, into the distance image, and also its own depth $D$ (obtained as well from the distance image).

- Based on this set of landmarks (samples from now on) we can estimate a relation that describes the shift of the coordinates in function of depth ($D$). We calculate a function per coordinate, this linear function estimates the shift traced from a coordinate into the depth image $(x_d, y_d)$ to its shifted peer into the color image $(x_c, y_c)$: $ShiftX(x_d, D)$ and $ShiftY(y_d, D)$. So that, $ShiftX + x_d = x_c$ and $ShiftY + y_d = y_c$ for a given $D$.

Once we retrieve the parameters of the two shift functions by interpolation of the samples, we go though every pixel in the distance image and recover its shifted position in the color image (Fig. 2). A one-to-one correspondence between images is then applied. We apply this correspondence using an efficient programming method based on matrixes calculation. Thus, we are able to attain performances up to 0.004 seconds for each image pair. Same method has to be repeated over each pair of images as long as a video is being shot.

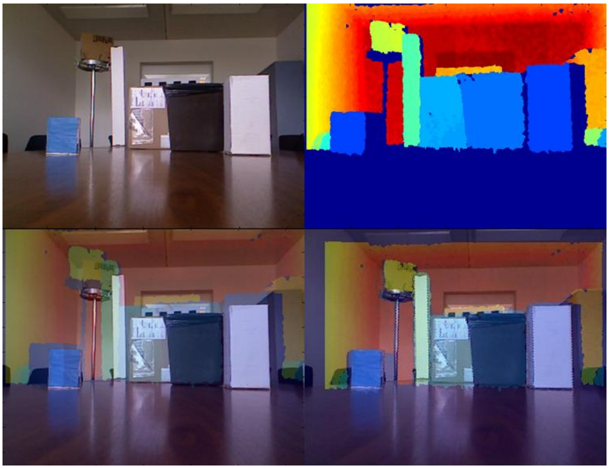

**Fig. 2.** (top-left) Webcam-provided color image. **(top-right)** TOF-provided distance image (resized to fit top-left size). **(bottom-left)** A blended image showing the registration of (top-left) into (top-right). **(bottom-right)** A blended image showing the registration of (top-left) into the rectified version of (top-right) using our algorithm.

## 3 Application to Obstacle Analysis in Aided Navigation

There are many applications that merely use spatial awareness to meet a goal (e.g., automotive sensors, shapes analysis, robotic). Another gamut of applications however, cannot dispense with color to work properly. For instance, in this application we aim at detecting an approaching obstacle, yet we would like also to trace his trajectory so as to avoid a potential stumble. While this first task relies solely on the TOF attributes, the latter certainly needs of tracking and/or object detection algorithms. Typically, this sort of algorithms falls back on color to achieve the purpose. In this work, we detect a potential obstacle and estimate how likely that is going to lead to a stumble, as follows:

- A riskiness plane ***P*** with thickness ***t*** is fixed at certain depth ***d***.
- We constantly scan ***P*** to detect whether or not an object appears on it (i.e. a cluster of pixels with distance ***d*** and greater that a tolerance in the TOF image).
- If an object happened to be detected, we retrieve 15 color images from the webcam back in time (time in which the object traveled from deeper planes).
- We use a short-term tracker (e.g. Lukas-Kanade [2]) to backtrack the object over this set of frames starting from within a bounding box fixed on the position of the TOF-based detection.
- Once the path of the object has been traced, we estimate its trajectory from ***P*** up to the camera plane. If a stumble with the camera is likely, we activate an alarm that evidences the hazard of the situation. We do not launch any warning otherwise.

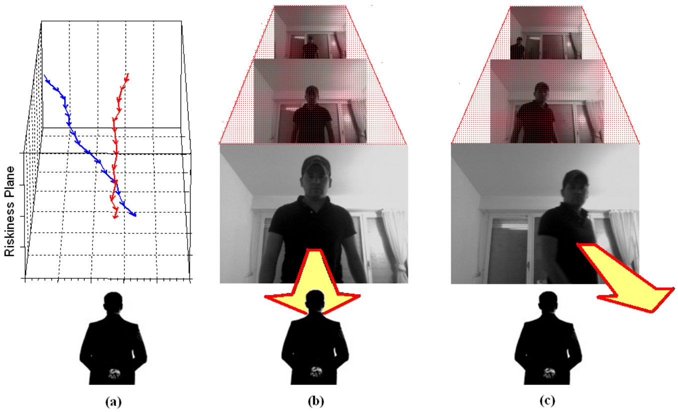


**Fig. 3.** (a) The resultant path that an object traces up to the riskiness plane, from deeper planes. This path has been backtracked using a short-term tracker algorithm. Red vectors represent an object approaching directly over the user (b). Blue vectors belong to an object with diagonal trajectory (c). In turn, (b) and (c) are illustrative representations that explain both situations by showing some samples of the backtracked sequences. For (b) a stumble is expected whereas, for (c) no stumble is likely. Therefore, only (b) produces a warning alarm.

## References

1. Bologna, G., Deville, B., Gomez, J., Pun, T.: Toward local and global perception modules for vision substitution. Neurocomputing 74(8), 1182–1190 (2010)
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. DARPA, Image Understanding Workshop (1981)
3. De-Maeztu, L., Mattoccia, S., Villanueva, A., Cabeza, R.: Linear stereo matching. In: Proc. ICCV, International Conference on Computer Vision (2011)