Computing Range Flow from Multi-modal *Kinect* Data

Jens-Malte Gottfried[1,3], Janis Fehr[1,3], and Christoph S. Garbe[2,3]

[1] Heidelberg Collaboratory for Image Processing (HCI), University of Heidelberg
[2] Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg
[3] Intel Visual Computing Institute (IVCI), Saarland University, Saarbrücken

Abstract. In this paper, we present a framework for range flow estimation from *Microsoft's* multi-modal imaging device *Kinect*. We address all essential stages of the flow computation process, starting from the calibration of the *Kinect*, over the alignment of the range and color channels, to the introduction of a novel multi-modal range flow algorithm which is robust against typical (technology dependent) range estimation artifacts.

1 Introduction

Recently, *Microsoft's* novel low cost, multi-modal depth imaging device *Kinect* has drawn a lot of attention in the computer vision and related communities (such as robotics). In the rather short time span of its availability, *Kinect* already triggered numerous computer vision applications, mostly in the area of human-computer-interaction.

In this paper, we present a framework for the estimation of range flow fields from multi-modal (depth + color) video sequences captured by *Kinect*. The computation of optical flow [2] from 2D image sequences or range flow [11] from depth image sequences plays an important role in the middle layer of a wide range of computer vision algorithms, such as object tracking, camera motion estimation or gesture recognition. Flow estimation has been investigated for a long time and many sophisticated algorithms (especially for optical flow) have been introduced so far [2].

However, due to the *Kinect* technology, standard range flow algorithms cannot simply be used "out of the box" – which makes the computation of range flow fields from this device a non-trivial task. The main difficulties result from the facts that *Kinect* provides only uncalibrated data where color and depth channels (which are recorded by different cameras) are not aligned, that the depth channel may contain large areas of invalid values and that edges in the depth-map are not always stable.

The **main contribution** of this paper is twofold: First, we introduce a novel channel alignment algorithm which largely reduces image areas without valid measurements compared to previous methods. Secondly, we extend existing range flow approaches to cope with invalid and unstable depth estimates. The proposed methods are intended to be applied to data captured with the *Kinect*

G. Bebis et al. (Eds.): ISVC 2011, Part I, LNCS 6938, pp. 758–767, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Fig. 1. *Kinect* device and its raw data output. **Top:** Kinect device with projector (A), color cam (B) and IR cam (C); color and depth channel overlayed. **Bottom** (left to right): raw color image; pseudo-color coding of the 11bit depth image provided by the on chip depth estimation; raw IR image showing the projected point pattern.

device, but should work in any multi-modal setting where different cameras are used to capture the projected pattern and the color image.

Related Work. To best of our knowledge, there has not been any publication on range flow estimation from *Kinect* data. However, there have been several approaches to solve some of the algorithmic steps in our method independently. We use the open source driver [6] to access and capture *Kinect* data. The calibration and alignment of color and depth channels has been addressed by [4]. The drawback of this method is, that the resulting data set still contains large areas of invalid values which results in poor flow estimation results (see section 5). The proposed method for the computation of the actual range flow is based on ideas introduced by [11], combined with ideas from [12].

The remainder of the paper is organized as follows: Section 2 provides a discussion of *Kinect's* depth imaging technology and points out the main problems that have to be solved in order to use the raw data for range flow estimation. Section 3 introduces the calibration process and our novel algorithm for the alignment of depth and color channels, before the actual flow algorithm is discussed in section 4. Finally, results and experimental evaluations are presented in section 5.

2 A Brief Introduction to the *Kinect* Imaging Hardware

Kinect's depth imaging device is based on a *structured-illumination* approach [3]. The range information is estimated from the distortion of a projected point pattern which is captured with a camera placed at a certain baseline distance from the projector.

Figure 1 shows the hardware setup: *Kinect* uses an IR laser diode to project a fixed point pattern which is invisible to the human eye. In combination with an

IR camera, the estimation of the depth-map is computed directly in hardware at approximately 30 frames per second at VGA resolution[1]. The depth resolution is approximately 1cm at 2m optimal operation range [6].

The main advantage of the *Kinect* concept is, that it allows a more or less dense depth estimation of a scene even if it contains objects with little or no texture. Additionally, a second camera captures VGA color images from a third position.

2.1 Limitations of *Kinect* Data

The main problem, at least from a computer vision perspective, is the computation in hardware which is not user accessible nor can it be circumvented altogether. Hence, the entire process is a "black-box", with very little publicly known details on the obviously extensive post-processing. It most likely includes some sort of edge preserving smoothing and up-scaling of the depth-map. This has significant impact on the noise characteristics and correlations.

The provided depth estimation is more or less dense, but there is a systematic problem common to all structured-light approaches which use a camera offset: there are regions where the projected pattern is shadowed by foreground objects – making it impossible to estimate the depth at these positions (see fig. 1, depth and IR image). Another problem is that depth values tend to be unstable and inaccurate at object boundaries. This is caused by the fact that the dense depth-map is interpolated from discrete values measured at the positions of projected point patterns.

Finally, since there are two different cameras capturing color and depth images, these images are not necessarily aligned to each other (fig. 1 top right). In addition, the two cameras have slightly different focal length and the optical axes are not perfectly parallel.

3 *Kinect* Calibration and Data Alignment

Since we are proposing a multi-modal flow algorithm (see section 4), it is essential that the color and depth image information at a certain image location belong to the same object point. In this section, we introduce a novel alignment algorithm which is especially suitable for *Kinect* data.

Our approach is based on previous methods by [4], but performs a more complex inverse mapping from the depth image onto the color image – whereas [4] uses the straight forward mapping of the color image onto the depth image. The advantage of our method is, that we are still able to compute at least the xy-flow for areas with invalid depth values, whereas all information is lost in such areas if one applies the original alignment approach. Figure 2 shows examples for both approaches.

[1] Note that VGA resolution is probably a result of the extensive on-chip post-processing. The true hardware resolution is bounded by the number of projected points, which is much lower than VGA.

3.1 Camera Calibration

In a first step, we perform a stereo-calibration of the cameras. Since camera calibration is a common task, we will not discuss this part in detail. We simply use a standard checker-board target with good IR reflection properties and extra illumination in the IR spectrum and apply a standard stereo-calibration procedure as provided in [7].

3.2 Data Alignment

The actual data alignment algorithm is based on the assumption that the raw depth values provided by *Kinect* are linearly correlated with the point-wise disparity d between pixels in the color image and their corresponding raw depth values z (just like one would expect in a standard stereo setting). We use a PCA over the positions of the checker-board corners from the calibration process to obtain this linear map $d(z) = a \cdot z + b$.

The original approach in [4] showed, that it is easy to warp the color image $I(x, y)$ such that it is aligned with the depth image $Z(x, y)$ by use of the disparity field $D(x, y) = d(Z(x, y))$:

$$\tilde{I}(x, y) = I(x + D(x, y), y) \tag{1}$$

where $\tilde{I}(x, y)$ is the warped color image. Since $x+d$ may be fractional pixels, one has to interpolate the integer pixel values of I. For regions, where $Z(x, y)$ has no valid depth value, no value for $\tilde{I}$ can be computed. Such regions are visible e.g. at the shadow of the hand shown in fig. 1. They have to be marked, e.g. setting the color to black. Hence, the color image information in these regions is completely lost. Therefore, we propose not to modify the color image, but to invert the mapping and align the depth image to the color image. In order to do this, we have to compute the *inverse disparity field* $D^*(x, y)$ such that

$$D(x, y) = -D^*(x + D(x, y), y) \tag{2}$$

This problem is similar to the computation of the inverse optical flow $\mathbf{h}^*(\mathbf{x})$ as proposed in [9, appendix A]. Here, we consider a special case of this approach because the disparity is known to be limited to the x-direction only (i.e. the y-component of the flow $\mathbf{h}$ vanishes). Using this simplification, the ideas of [9] may be reformulated (for the i-th pixel in x-direction, i.e. at position (x_i, y)) as

$$D^*(x_i, y) = -\frac{\sum\limits_j D(x_j, y)\ p(x_i, x_j + D(x_j, y))}{\sum\limits_j p(x_i, x_j + D(x_j, y))} \tag{3}$$

where the weighting function p is computed using

$$p(x, x') = \max\{0, r - |x - x'|\}. \tag{4}$$

The radius r specifies the region of influence of each pixel. Invalid depth values $D(x_j, y)$ have to be excluded from summation. For some target positions, the

denominator can become very small (i.e. if no depth values would be warped to this position). In this case we mark $D^*(x_i, y)$ as invalid as well. Using D^* we compute the alignment of the depth map $\tilde{Z}(x, y)$ to $I(x, y)$ in a similar way as before in (1):

$$\tilde{Z}(x, y) = Z(x + D^*(x, y), y) \tag{5}$$

Figure 2 shows a qualitative comparison of the results of our proposed method and the pure stereo calibration approach. Our calibration/alignment software will be published as open source together with this paper (link: [1]).

Another approach to align the data could be a reprojection of the 3D data points given from the IR camera using the projection matrix known from the calibration step. The problem here is that the raw depth values (as given by the device) are not the real z-coordinate values but proportional to the point pattern disparity. Several different methods are proposed for computing z-values from this raw depth. Our proposed method uses the fact that the raw depth values are directly proportional to the pixel shift between the images and hence avoids the problems of computing accurate pixel z-coorinates.

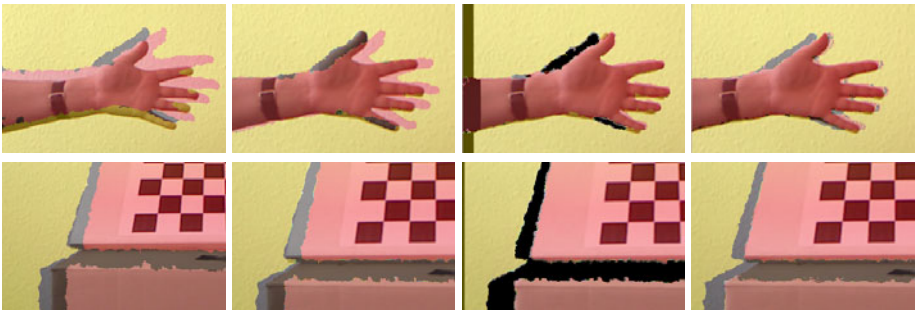

Fig. 2. Comparison of calibration results. In gray overlay regions, the depth values are invalid. In black regions, there are neither valid color nor depth values.
Columns (left to right): uncalibrated; pure stereo calibration (epipolar); aligned warping color image with D; aligned warping depth image with D^*

4 Range Flow

Range flow (syn.: *Scene flow* [14]) is the established term for the 2.5D extension of *optical flow*, describing the local 3D motion in an image sequence. Mathematically, the range flow field is a 3D vector field $\mathbf{h}_R$ which is defined on a 2D image plane, i.e.

$$\mathbf{h}_R : \mathbb{R}^2 \mapsto \mathbb{R}^3 \qquad \mathbf{h}_R(x, y) = (u, v, w)^T \qquad \mathbf{h}(x, y) = (u, v)^T, \tag{6}$$

where the first two components (u, v) are identical to the motion description in standard 2D optical flow $\mathbf{h}$ and w encodes the motion in the depth direction.

Numerous algorithms have been proposed in literature to solve the standard *optical flow* (u, v) (e.g. see [2]). For the sake of simplicity, we base our proposed method on a refined standard method for global *optical flow* [12] (which it self is a reinterpretation of the classical flow paper by Horn and Schunk [5]). It should be noted that our proposed algorithm (see section 4.1) should also work with most other global methods (like [8] or [13]). However, since we focus on the range extension for *Kinect* data, we are keen to keep the 2D terms as simple as possible. Moreover, most of the "more advanced" techniques focus on increased sub-pixel accuracy, which is not very likely to be a useful property given the accuracy of real world *Kinect* data.

The method by [12] applies a pixel-wise brightness constancy assumption called *optical flow constraint* (OFC/BCC) that may be formulated as

$$\nabla I \cdot (u, v)^T + I_t = 0 \Leftrightarrow (I_x, I_y, 0, I_t) \cdot (u, v, w, 1)^T = 0 \tag{7}$$

where I is the 2D image data (here: color image converted to gray-scale) and the indices denote derivation with respect to the specified variable.

As proposed by [11], a similar term may be formulated for the depth data Z, adding the motion in depth direction w:

$$\nabla Z \cdot (u, v)^T + w + Z_t = 0 \Leftrightarrow (Z_x, Z_y, 1, Z_t) \cdot (u, v, w, 1)^T = 0 \tag{8}$$

This equation is called *range flow motion constraint* (RFMC). As one can see, the 2D terms in eqns. (7) and the depth term in (8) are based on the same principle. Using this fact, range flow estimation using color (converted to gray-scale) images and depth data may be performed using any optical flow estimation algorithm with an additional data term incorporating eqn. (8).

4.1 Robust Flow Estimation

The multi-modal data which is computed from our data alignment algorithm (see section 3.2) has a dense color channel. Still, there can be invalid values in the depth channel and artifacts at object borders. Both artifacts do not remain constant in time, resulting in estimated depth changes even if there is no motion.

Therefore it is essential to exclude these regions for a robust flow computation. Regions with invalid depth values may be recognized by simply thresholding the depth channel (in the raw depth output, these pixels are marked by the hardware with a depth integer value of 2047 =`0x7FF`, i.e. the largest 11 bit integer value). Object borders may be recognized by thresholding the edge strength $(Z_x^2 + Z_y^2)$. If at least one of these threshold conditions is met, pixels are excluded from the RFMC data term (8), i.e. only the color image data is used for computation at this position.

Since the linear filters used to compute the derivatives of the depth image usually have a width of 3 pixels (Sobel or Scharr filters [10]), this exclusion region has to be extended, e.g. using the morphologic dilation operator. A radius of 2 pixels showed to be sufficient. In the excluded regions, the value of w is interpolated from the valid neighbours by regularization of the range flow field.

Strong regularization leads to smooth flow fields but also causes blur effects at motion edges. Since motion edges often correspond to edges in the depth image, estimation results can be improved further by using the exclusion mask. Using strong regularization in valid and weak regularization in excluded regions yields much sharper motion edges and separation between fore- and background motion. This adaptive regularization is another benefit of using the depth image information.

4.2 Algorithm Summary

The final implementation of our approach is realized in a standard pyramid scheme with two levels of iteration. The outer iteration implements the multi-scale image pyramid. The inner iteration (line 10 of algo. 1) recomputes the flow increments on the given input image pairs (I_1, I_2, Z_1, Z_2), where the second has been warped with the flow $\mathbf{h}_R^0$ computed during the previous iteration.

This is done by minimizing the following energy functional:

$$\iint \Big[(I_x u + I_y v + I_t)^2 + \lambda_Z(x,y)(Z_x u + Z_y v + w + Z_t)^2 + \lambda_R(x,y)(|\nabla u|^2 + |\nabla v|^2 + |\nabla w|^2) \Big] \mathrm{d}x\, \mathrm{d}y \quad (9)$$

where

$$\lambda_Z(x,y) = \begin{cases} c_z > 0 & \text{where } M = 0 \\ 0 & \text{else} \end{cases} \qquad \lambda_R(x,y) = \begin{cases} c_{R1} > 0 & \text{where } M = 0 \\ c_{R2} > 0 & \text{else} \end{cases} \quad (10)$$

with $c_{R1} \gg c_{R2}$.

Using weak regularization in invalid regions (where $M = 1$) may be counter-intuitive. As stated above, the mask M is also used as edge detection so weak regularisation in invalid regions leads to sharper motion borders on edges.

5 Evaluation

Unfortunately, there is no publicly available data base with ground truth range flow fields for *Kinect* data, as this is the case for 2D optical flow (e.g. as provided by [2]). Hence, no generally agreed and sufficiantly accurate methodology for quantitavely analyzing our algorithm is available. Therefore we give a qualitative discussion of our results only.

5.1 Results

Qualitative results. of our range flow estimation are shown in fig. 3. A moving hand sequence has been recorded. The rows in this figure represent different

Algorithm 1. Final *Kinect* Range Flow Algorithm

for all multi-modal image pairs I_1, I_2, Z_1, Z_2 **do**
  SCALE I_1, I_2, Z_1, Z_2 and $\mathbf{h}_R^0$ accordingly
  for all flow iterations **do**
    WARP I_2, Z_2 with $\mathbf{h}_R^0 \rightarrow \tilde{I}_2, \tilde{Z}_2$
    if Z_1 or $\tilde{Z}_2$ invalid **then**
      SET exclusion mask $M = 0$
    else
      SET exclusion mask $M = 1$
    end if
    COMPUTE FLOW $\mathbf{h}_R$ using OFC (7) and RFMC (8) data terms with strong regularization where $M = 1$, OFC only with weak regularization where $M = 0$.
    APPLY MEDIAN FILTER on $\mathbf{h}_R$ to suppress outliers
    SET $\mathbf{h}_R \rightarrow \mathbf{h}_R^0$
  end for
end for

algorithm configurations, i.e. combinations of optional usage of the region validity masks and usage of the forward or inverse disparity (D or D^*) for warping.

The first column shows the first frame of the image pair as used for range flow estimation. In the second column, the first two components of the range flow result $\mathbf{h}_R$, i.e., the standard optical flow $\mathbf{h}$ are visualized. The distance between the arrows is 16 px, an arrow legnth of 5 px correnspponds to a flow magnitude of 1. For better visualization of the flow contours, a hsv representation using flow angle as hue and flow magnitude as saturation has been drawn in the background. The third column shows the depth dimension of $\mathbf{h}_R$, i.e. w. The last column shows the raw depth data with exclusion mask. Since the depth value of invalid depth pixels is `0x7FF`, these pixels appear bright. Note that the exclusion mask also consists of edges in the depth image, the contours of the hand are reproduced well.

Since strong regularization is applied within connected valid depth regions, the flow result in the first two rows is smooth over the hand area as well as in the background. The hand contours are reproduced well, only low blurring effects at the borders are visible. Without usage of the masks, outliers as well in the (u, v) as in the depth channel w show up as visible in the last two rows.

We did not yet tune the algorithms to be higly parallelized or using GPU computing, so the runtime is currently about half a minute per frame pair. Significant speed improvements are expected from such an optimized implementation.

For testing, we used a very simple sequence that simulates the targeting application area of gesture recognition. Future research should focus on generating more realisting test sequences with given ground truth such that quantitative analysis becomes possible.

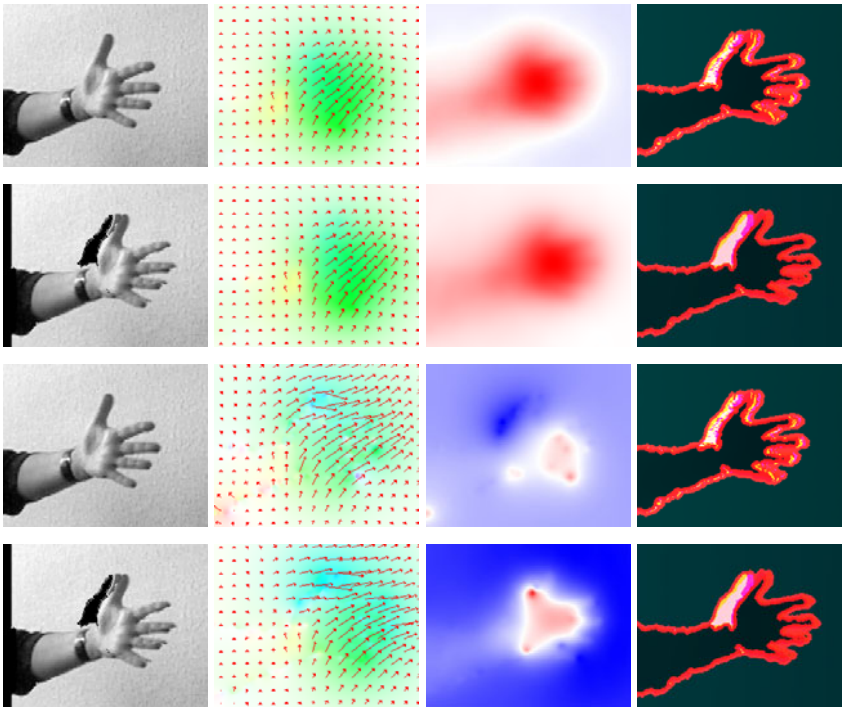

Fig. 3. Range flow estimation result of an image pair of a moving hand sequence.
Rows (top to bottom): warp depth with D^*, using masks; warping color with D, using masks; warp depth with D^*, no masks; warping color with D, no masks; cf. algo. 1
Columns (left to right): gray image; optical flow $\mathbf{h}$ (hsv visualization with quiver); range flow depth component w (blue: positive, red: negative values); depth image with exclusion mask (red/bright=excluded)
The exclusion mask is shown even if it has not been used to compute the flow.

6 Conclusions

In this paper, we have presented a novel framework for robust range flow estimation from multi-modal *Kinect* data. Our calibration and alignment algorithm with the back-ward mapping scheme provides a general and robust solution to register the color and depth channels provided by the hardware, which can also be applied to a wide range of other applications. The presented range flow estimation provides stable flow fields and is able to cope with the systematic errors induced by the hardware setting. Hence, our framework provides a useful middle-layer for the further development of high level algorithms for the *Kinect*.

Supplementary Material

A website containing the captured *Kinect* data with different alignment strategies and the flow results discussed in fig. 3 (links to full video sequences) are available at `http://hci.iwr.uni-heidelberg.de/Staff/jgottfri/papers/flowKinect.php`, also presenting further experiments using more sequences (targeting gesture recognition) and flow algorithms.

Acknowledgements. The authors acknowledge financial support from the Intel Visual Computing Institute (IVCI, Saarbrücken) and from the "Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences" (DFG GSC 220).

References

1. Web page with used data and experiments of this paper, `http://hci.iwr.uni-heidelberg.de/Staff/jgottfri/papers/flowKinect.php`
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV, pp. 1–8. IEEE, Los Alamitos (2007), `http://vision.middlebury.edu/flow`
3. Besl, P.J.: Active, optical range imaging sensors. Machine vision and applications 1(2), 127–152 (1988)
4. Burrus, N.: Kinect calibration - calibrating the depth and color camera, `http://nicolas.burrus.name/index.php/Research/KinectCalibration`
5. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. 17(1-3), 185–203 (1981)
6. Martin, H.: Openkinect project - drivers and libraries for the xbox kinect device, `http://openkinect.org`
7. Opencv (open source computer vision) - a library of programming functions for real time computer vision, `http://opencv.willowgarage.com`
8. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision 67(2), 141–158 (2006)
9. Salgado, A., Sánchez, J.: A temporal regularizer for large optical flow estimation. In: ICIP, pp. 1233–1236. IEEE, Los Alamitos (2006)
10. Scharr, H.: Optimale Operatoren in der digitalen Bildverarbeitung. Ph.D. thesis, Universität Heidelberg (2000)
11. Spies, H., Jähne, B., Barron, J.L.: Range flow estimation. Computer Vision and Image Understanding 85(3), 209–231 (2002)
12. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR, pp. 2432–2439. IEEE, Los Alamitos (2010)
13. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning optical flow. In: Forsyth, D.A., Torr, P.H.S., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
14. Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. In: ICCV, pp. 722–729 (1999)