Accurate and Computationally-inexpensive Recovery of Ego-Motion using Optical Flow and Range Flow with Extended Temporal Support

Graeme A. Jones
dircweb.kingston.ac.uk/graeme/

Digital Imaging Research Centre
School of Computing and Information
Systems, Kingston University,
Penrhyn Road, Kingston upon Thames,
UK, KT1 2EE

Abstract

The *ego-motion* of a moving RGB-Z sensor is computed using combined *range flow* and *optical flow* pixel constraints. The emphasis of the work is on computationally cheap yet accurate estimation of inter-frame translation and rotation parameters using a linear *small rotation* model. To ensure accurate inter-frame motion estimates, an iterative form of the estimator is developed which repeatedly warps the target frame and measures its misalignment with the current frame. As these motion estimates are integrated temporally, to minimise drift in pose over time, additional temporal constraint is provided through the use of *anchor frames*. The algorithm is evaluated on the recently published TUB RGB-D Benchmark which also includes a set of standard metrics. Results presented suggest that performance is commensurate with alternative methodologies such as SLAM but at a fraction of the computational cost.

1 Introduction

Recovering the ego-motion of a moving camera within a static scene supports many applications in robotics and computer vision. The presented work is motivated by *pre-vis* applications in the film industry; specifically the ability to render digital assets into the scene during production in real-time. A low-cost commodity depth camera can be easily mounted on and calibrated to a high quality production cameras and used to extract changes in sensor pose from the induced motion of the rigid scene. This work explores the effectiveness of the computationally efficient *range flow* technique to generate this real time pose information directly from the depth stream of a Kinect sensor.

A number of challenges within the approach are addressed. First, an iterative version of the *small rotations* motion estimator is developed to ensure the most accurate inter-frame estimates. Second, the substantial issue of *drift* is addressed - the accumulated error between true and estimated sensor pose as motion estimates are temporally integrated. *Anchor frames* which enjoy significant overlap with subsequent frames are stored and used to provide additional temporal range flow constraint within the estimation process. Where there are loops in the data sequence, it is advantageous to select anchors from previously seen parts of the scene. Finally, in some scene configurations, there is insufficient constraint from the depth images. We exploit the availability of registered intensity images to further constrain the sensor motion using the *optical flow* framework.

The evaluation of new algorithms which exploit the depth modality require depth datasets with associated ground truth. Our work is validated in section 5 using the recently published TUB RGB-D Benchmark. This resource contains extensive sets of depth and video sequences of varying lengths and differing levels of challenge such as large visual velocities or large dominant planar surfaces. The resource also provides an evaluation framework of metrics as well as downloadable evaluation scripts and an online evaluation tool.

2 Related Work

There have been a variety of approaches developed to recover the motion of an image sensor moving within a scene. These include the use of corresponding depth features [15], the *iterated closest point* algorithm (ICP) [4, 12], *simultaneous location and mapping* (SLAM) using tracked 2D features [6, 14] and *range flow* [10, 11]. Impressive real-time results have been achieved particularly by SLAM which typically employs extended Kalman tracking of persistent 2D features to recover necessarily both a 3D 'map' of the position of these features and the location of the camera within the scene. As is the case in this work, an important and potentially restrictive assumption is that the scene is rigid without independent moving scene elements. Viable depth sensors have been available for over two decades whether using laser range finders or stereo vision systems. It was quickly realized that the *optical flow* mechanism could be easily extended to depth images [11, 16]. The recent availability of commodity frame-rate depth sensors has given rise to renewed interest in recovering motion from depth data[9, 12].

Analogous to optical flow, *range flow* is a per-pixel constraint on the 3D displacement of an imaged 3D point given its local spatio-temporal depth derivatives. These must be combined across a region or an image to provide sufficient constraint to extract 3D motion. Where the goal is to recover a 3D displacement field given a possibly moving scene with independently moving objects (or even a single non-rigid object), this per-pixel constraint can be embedded in a global energy functional which penalises pixel motions which do not satisfy the local range flow constraints and which are not locally smooth [3, 9] - an approach essentially equivalent to regularisation in optical flow[2]. Alternatively if some motion model is used which is valid for the rigid scene (excluding any independently moving foreground objects), then a large number of pixel constraints can be used to over constrain very few motion parameters. In the case of optical flow, such models include affine motion, zoom and the *small rotation* approximation of the 3D rotation and translation of the camera[1].

The benefits of combining the optical flow constraint with range flow have already been recognised[3, 10, 13]. Barron and Spies embed these as error terms in a global energy functional with additional local first-order smoothness constraints to extract the 3D displacement field[3]. Quiroga *et al* generate 3D translation fields for image patches using a template matching approach[13], while Haville *et al* recover the 3D pose changes of pre-segmented image regions using an affine projection assumption[10].

Attention is also drawn to the relevant optical flow literature which informs the optimization approach adopted here. In particular, the iterative robust regression approaches of Black and Anandan[5] and Giaccone and Jones[8].

3 Deriving constraint from the depth and intensity images

3.1 The Motion and Projection Models

Scene displacement is induced by the movement of the camera whose rotation ω and translation $\mathbf{t}$ motion parameters can be modelled using the useful linear *small rotation* formulation when the movement between consecutive frames is relatively small. Here the 3D displacement $\Delta\mathbf{X} = (\Delta X, \Delta Y, \Delta Z)^T$ of a 3D scene point $\mathbf{X} = (X,Y,Z)^T$ is given by

$$\Delta\mathbf{X} = \omega \times \mathbf{X} + \mathbf{t} = \begin{bmatrix} Z\omega_y - Y\omega_z + t_x \\ X\omega_z - Z\omega_x + t_y \\ Y\omega_x - X\omega_y + t_z \end{bmatrix} = M(\mathbf{X})\mathbf{a} \tag{1}$$

where $\mathbf{a}$ is a concatenation of the motion parameters $\mathbf{a} = (\omega, \mathbf{t})^T$, and

$$M(\mathbf{X}) = \begin{bmatrix} 0 & Z & -Y & 1 & 0 & 0 \\ -Z & 0 & X & 0 & 1 & 0 \\ Y & -X & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Any small 2D image pixel displacement $\Delta\mathbf{x}$ can be related directly to the 3D displacement $\Delta\mathbf{X}$ of the point $\mathbf{X}$ which gave rise to it by differentiating the perspective projection equations $x = x_0 + Xf_x/Z$ and $y = y_0 + Yf_y/Z$ with respect to the components of the 3D point $\mathbf{X}$ (where f_x and f_y are the normalised focal lengths and $(x_0, y_0)^T$ is the centre of the image).

$$\frac{\partial\mathbf{x}}{\partial\mathbf{X}} = P(\mathbf{X}) = \begin{bmatrix} f_x/Z & 0 & -Xf_x/Z^2 \\ 0 & f_y/Z & -Yf_y/Z^2 \end{bmatrix} \tag{3}$$

Thus equations 1 and 3 linearly relate the 2D displacement $\Delta\mathbf{x}(\mathbf{x},\mathbf{a})$ at pixel $\mathbf{x}$ to the 3D motion $\mathbf{a}$ which gave rise to the displacement as

$$\Delta\mathbf{x}(\mathbf{x},\mathbf{a}) \approx P(\mathbf{X})M(\mathbf{X})\mathbf{a} \tag{4}$$

3.2 The Range Flow Constraint

Analogous to the *constant brightness equation* used to generate the *optical flow* constraint, the following depth constraint relates how a 3D point is captured in temporally separated depth images. A 3D point $\mathbf{X}$ (measured in the depth camera's coordinate system) is captured at pixel position $\mathbf{x} = (x,y)^T$ in the depth map Z_t. This point undergoes a 3D motion $\Delta\mathbf{X}$ which results, first, in an image motion $\Delta\mathbf{x}$ between frames t and τ, and second, in a change of the depth ΔZ of the 3D point captured at this new image location $\mathbf{x} + \Delta\mathbf{x}$. Thus the *range flow* constraint is formulated as

$$Z_\tau(\mathbf{x} + \Delta\mathbf{x}) = Z_t(\mathbf{x}) + \Delta Z \tag{5}$$

In practice an accurate estimator needs to iteratively refine estimates of the underlying 3D motion $\mathbf{a}$ which gave rise to the displacements $\Delta\mathbf{X}$ and $\Delta\mathbf{x}$. To achieve this, the above depth constraint must be reformulated as follows

$$Z_\tau(\mathbf{x} + \Delta\mathbf{x}(\mathbf{x}, \mathbf{a} + \Delta\mathbf{a})) = Z_t(\mathbf{x}) + \Delta Z(\mathbf{x}, \mathbf{a} + \Delta\mathbf{a}) \tag{6}$$

where $\Delta\mathbf{a}$ is an update of the current motion estimate $\mathbf{a}$, and the image $Z_\tau(\mathbf{x} + \Delta\mathbf{x}(\mathbf{x}, \cdot))$ is a version of $Z_\tau(\mathbf{x})$ warped by the displacement field $\Delta\mathbf{x}(\mathbf{x}, \cdot)$.

Using equation 4, the term $\Delta\mathbf{x}(\mathbf{x},\mathbf{a}+\Delta\mathbf{a})$ from the left hand side of equation 6 can be rewritten as $\Delta\mathbf{x}(\mathbf{x},\mathbf{a}+\Delta\mathbf{a}) = \Delta\mathbf{x}(\mathbf{x},\mathbf{a}) + P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}$. Similarly, using equation 1, the term $\Delta Z(\mathbf{x},\mathbf{a}+\Delta\mathbf{a})$ from the right hand side can be rewritten as $\Delta Z(\mathbf{x},\mathbf{a}) + M_3(\mathbf{X})\Delta\mathbf{a}$ where $M_3(\mathbf{X})$ is the third row of equation 2. Thus equation 6 can be rewritten as

$$Z_\tau\Big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a}) + P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}\Big) = Z_t(\mathbf{x}) + \Delta Z(\mathbf{x},\mathbf{a}) + M_3(\mathbf{X})\Delta\mathbf{a} \tag{7}$$

A first-order Taylor expansion of the left hand side of the above generates

$$Z_\tau\Big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a}) + P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}\Big) \approx Z_\tau\big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\big) + \nabla Z_\tau\big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\big)P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}$$

which allows the constraint equation to be rewritten to relate the motion update $\Delta\mathbf{a}$ linearly to (i) the gradient of the warped depth image ∇Z_τ, (ii) the temporal depth difference, and (iii) the current estimate of the change in depth. Combining the above two equations gives

$$\left\{\nabla Z_\tau\big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\big)P(\mathbf{X})M(\mathbf{X}) - M_3(\mathbf{X})\right\}\Delta\mathbf{a} = \Big\{Z_t(\mathbf{x}) - Z_\tau\big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\big)\Big\} + \Delta Z(\mathbf{x},\mathbf{a}) \tag{8}$$

Computing the image gradients $\nabla Z_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)$ of the warped image is problematic for two reasons. First, these gradients would require recomputing with each iteration. Second, the gradient calculation enhances the noise introduced by any interpolation process used in the warping. However as $\mathbf{a}+\Delta\mathbf{a} \rightarrow \mathbf{a}^*$, the true motion, $Z_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right) \rightarrow Z_t\left(\mathbf{x}\right)$. Therefore, we exploit our third approximation $\nabla Z_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right) \approx \nabla Z_t\left(\mathbf{x}\right)$ to generate the final constraint equation

$$\Phi(\mathbf{X})\Delta\mathbf{a} = \Big\{Z_t(\mathbf{x}) - Z_\tau\big(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\big)\Big\} + \Delta Z(\mathbf{x},\mathbf{a}) \tag{9}$$

where

$$\Phi(\mathbf{X}) = \left\{\nabla Z_t\left(\mathbf{x}\right), -1\right\}P(\mathbf{X})M(\mathbf{X}) = \begin{bmatrix} -Y - Z_y f_y - Z_x XY f_x/Z^2 - Z_y Y^2 f_y/Z^2 \\ X + Z_x f_x + Z_x X^2 f_x/Z^2 + Z_y XY f_y/Z^2 \\ -Z_x Y f_x/Z + Z_y X f_y/Z \\ Z_x f_x/Z \\ Z_y f_y/Z \\ -1 - Z_x X f_x/Z^2 - Z_y Y f_y/Z^2 \end{bmatrix}^T$$

and $\nabla Z_t(\mathbf{x}) = (Z_x, Z_y)$ are the spatial derivatives of $Z_t(\mathbf{x})$.

3.3 The Optical Flow Constraint

The *constant brightness equation* relates how the luminance at a 3D scene point is captured in temporally separated intensity images. A 3D point $\mathbf{X}$ is imaged at pixel position $\mathbf{x} = (x,y)^T$ in the intensity map I_t. This point undergoes a 3D motion $\Delta\mathbf{X}$ which results in an image motion $\Delta\mathbf{x}$ between frames t and τ and reprojects with the same intensity at the new image location $\mathbf{x}+\Delta\mathbf{x}$. Thus the *optical flow* constraint begins with the formulation

$$I_\tau(\mathbf{x}+\Delta\mathbf{x}) = I_t(\mathbf{x}) \tag{10}$$

As before, iterative refinement of motion estimates require a formulation in terms of an update $\Delta\mathbf{a}$ to the current motion estimate $\mathbf{a}$ i.e.

$$I_\tau(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a}+\Delta\mathbf{a})) = I_t(\mathbf{x}) \tag{11}$$

A first-order Taylor expansion of the left hand side of the above generates

$$I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})+P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}\right)\approx I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)+\nabla I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}$$

which allows the constraint equation to be rewritten in the classical optical flow formulation relating the motion update $\Delta\mathbf{a}$ linearly to the gradient of the warped intensity image ∇I_τ and the temporal depth difference as follows.

$$\nabla I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)P(\mathbf{X})M(\mathbf{X})\Delta\mathbf{a}=I_t(\mathbf{x})-I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right) \tag{12}$$

As before, rather than computing the intensity gradients $\nabla I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)$ of the warped image we again exploit the observation that as $\mathbf{a}+\Delta\mathbf{a}\rightarrow\mathbf{a}^*$, the true motion, $I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)\rightarrow I_t\left(\mathbf{x}\right)$ to generate the final intensity constraint equation

$$\Psi(\mathbf{X})\Delta\mathbf{a}=I_t(\mathbf{x})-I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right) \tag{13}$$

where

$$\Psi(\mathbf{X})=\nabla I_t\left(\mathbf{x}\right)P(\mathbf{X})M(\mathbf{X})=\begin{bmatrix}-I_yf_y-I_xXYf_x/Z^2-I_yY^2f_y/Z^2\\ I_xf_x+I_xX^2f_x/Z^2+I_yXYf_y/Z^2\\ -I_xYf_x/Z+I_yXf_y/Z\\ I_xf_x/Z\\ I_yf_y/Z\\ -I_xXf_x/Z^2-I_yYf_y/Z^2\end{bmatrix}^T$$

and $\nabla I_t(\mathbf{x})=(I_x,I_y)$ are the spatial derivatives of $I_t(\mathbf{x})$.

3.4 A Least Squares Estimator

The goal is to derive an estimator for recovering the rotational and translational motion of a rigid moving scene between successive frames. This parameter estimation problem will be posed as the optimisation of an error functional where the range flow and optical flow constraints are expressed as the error terms $e_Z(\mathbf{x},\Delta\mathbf{a})$ and $e_I(\mathbf{x},\Delta\mathbf{a})$ respectively *i.e.*

$$e_Z(\mathbf{x},\Delta\mathbf{a})=\Phi(\mathbf{x})\Delta\mathbf{a}-\Gamma(\mathbf{x},\mathbf{a}),\quad e_I(\mathbf{x},\Delta\mathbf{a})=\Psi(\mathbf{x})\Delta\mathbf{a}-\Lambda(\mathbf{x},\mathbf{a}) \tag{14}$$

where $\Gamma(\mathbf{x},\mathbf{a})=Z_t\left(\mathbf{x}\right)-Z_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)+\Delta Z(\mathbf{x},\mathbf{a})$ and $\Lambda(\mathbf{x},\mathbf{a})=I_t\left(\mathbf{x}\right)-I_\tau\left(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})\right)$. Combining such error terms from multiple pixels across the depth image $p\in\mathcal{I}$ generates the following pseudo-inverse estimator

$$\Delta\mathbf{a}=\Big[\sum_{p\in\mathcal{I}}\Phi(\mathbf{x}_p)^T\Phi(\mathbf{x}_p)+\lambda_I\Psi(\mathbf{x}_p)^T\Psi(\mathbf{x}_p)\Big]^{-1}\sum_{p\in\mathcal{I}}\Phi(\mathbf{x}_p)^T\Gamma(\mathbf{x}_p,\mathbf{a})+\lambda_I\Psi(\mathbf{x}_p)^T\Lambda(\mathbf{x}_p,\mathbf{a}) \tag{15}$$

The weight λ_I controls the relative influence of the two types of constraint. Least squares estimators are particularly sensitive to *outliers*. In this application there are two principle sources of such outliers. First, any independently moving object whose motion does not conform to the rigid scene assumption. Second, pixels whose current and warped depth or intensity data belong to different surfaces. For this reason, pixels where the difference in depth between current and warped images exceeds 50mm are excluded from the estimator. Similarly pixels whose intensity difference exceeds 33 greylevels are also excluded.

4 Minimising drift using Anchor Frames

Simply integrating between-frame motion estimates over time will inevitably result in *drift* *i.e.* the accumulated error between true and estimated sensor pose. To illustrate this, Figure 1 compares over time the three estimated translation components of the camera position with the ground truth for the *Freiburg 1 Room* sequence from the publicly available RGB-D SLAM Dataset and Benchmark[17]. The error plot (Euclidean distance between estimated and ground truth position) clearly increases over time reaching a discrepancy of over 75cm.

To minimise this drift, additional temporal constraint can be included. Specifically we introduce the concept of an *anchor frame*. In addition to recovering a parameter update for the motion between the current frame and the previous frame, the same update is also constrained by the motion between the current frame and its anchor. A depth frame is an anchor to all subsequent frames with which it retains a significant degree of overlap. Once the amount of overlap falls below a threshold, the last frame is promoted as the next anchor. Updates to the motion are now computed from two sources of range flow constraint

$$\begin{aligned} e_Z(\mathbf{x},\Delta\mathbf{a}) &= \Phi(\mathbf{x})\Delta\mathbf{a} - \Gamma(\mathbf{x},\mathbf{a}_{t,t-1},Z_t,Z_{t-1}) \\ e_{\mathcal{A}}(\mathbf{x},\Delta\mathbf{a}) &= \Phi(\mathbf{x})\Delta\mathbf{a} - \Gamma(\mathbf{x},\mathbf{a}_{t,\mathcal{A}_t},Z_t,Z_{\mathcal{A}_t}) \end{aligned} \tag{16}$$

where $\Gamma(\mathbf{x},\mathbf{a},Z_t,Z_\tau) = Z_t(\mathbf{x}) - Z_\tau(\mathbf{x}+\Delta\mathbf{x}(\mathbf{x},\mathbf{a})) + \Delta Z(\mathbf{x},\mathbf{a})$, the parameters $\mathbf{a}_{t,\tau}$ refer to the motion from frame t to frame τ, and the index $\mathcal{A}_t$ refers to the anchor frame of the current frame at time t. To exploit this additional constraint, the estimator of equation 15 must be modified as follows

$$\begin{aligned} \Delta\mathbf{a} = \Big[\sum_{p\in\mathcal{I}} \lambda_Z\Phi(\mathbf{x}_p)^T\Phi(\mathbf{x}_p) + \lambda_{\mathcal{A}}\Phi(\mathbf{x}_p)^T\Phi(\mathbf{x}_p) + \lambda_I\Psi(\mathbf{x}_p)^T\Psi(\mathbf{x}_p)\Big]^{-1} \\ \sum_{p\in\mathcal{I}}\Big[\lambda_Z\Phi(\mathbf{x}_p)^T\Gamma(\mathbf{x}_p,\mathbf{a}_{t,t-1},Z_t,Z_{t-1}) + \lambda_{\mathcal{A}}\Phi(\mathbf{x}_p)^T\Gamma(\mathbf{x}_p,\mathbf{a}_{t,\mathcal{A}_t},Z_t,Z_{\mathcal{A}_t}) \\ + \lambda_I\Psi(\mathbf{x}_p)^T\Lambda(\mathbf{x}_p,\mathbf{a}_{t,t-1},I_t,I_{t-1})\Big] \end{aligned} \tag{17}$$

where the positive weights λ_Z, $\lambda_{\mathcal{A}}$ and λ_I (such that $\lambda_Z + \lambda_{\mathcal{A}} + \lambda_I = 1$) control the relative influence of the depth, anchor and intensity constraints.

When there are loops in the sequence, it would be advantageous to select anchors from previously seen data rather than using the last frame. Such constraint from early frames makes a significant impact on the degree of drift. To this end, a list of all anchor frames is maintained. When a new anchor is required, this list is searched for the earliest anchor overlapping the current frame. However, a consequence of this approach, is the linear growth in storage requirements for these anchors and in the computational cost of searching through these anchors as the length of the video sequence grows.

Whereas the initial estimate of the motion between a frame and its predecessor is $\mathbf{a}_{t,t-1}^{(0)} = \mathbf{0}$, the initial estimate $\mathbf{a}_{t,\mathcal{A}_t}^{(0)}$ of the motion between frame t and its anchor is computed from the pose of the camera when the anchor frame was selected and the pose of the camera at the previous frame. In fact, on the basis that the last motion estimate is a good approximation of the current motion, in both cases, these initial estimates are first updated by $\mathbf{a}_{t-1,t-2}^{(\infty)}$.

For this new estimator, Figure 2 compares the three estimated translation components of the camera position with the ground truth for the *Freiburg 1 Room* sequence. In comparison to Figure 1, the error plot clearly shows the impact of exploiting additional temporal constraint from the anchor frames.

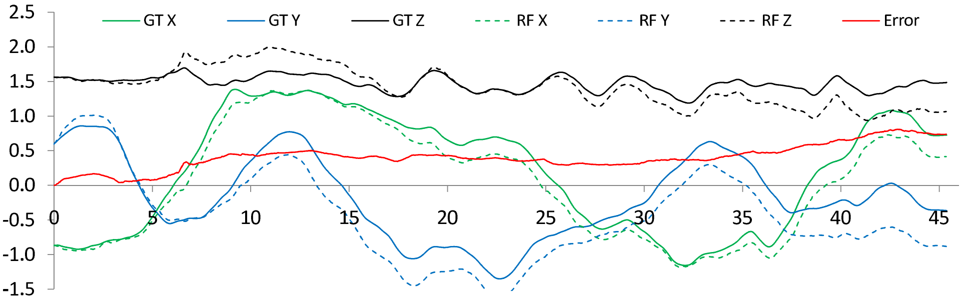

Figure 1: Pose Accuracy: No drift control (solid *ground truth*, dotted *estimated*)

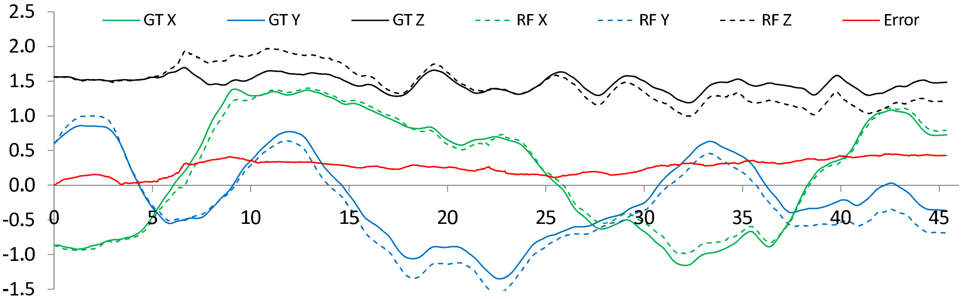

Figure 2: Pose Accuracy: Drift control using *anchor* frames (solid *ground truth*, dotted *estimated*)

5 Evaluation

The recently published TUM RGB-D Benchmark [17] is used to evaluate the these motion estimators. This resource provides Kinect depth and registered RGB sequences with synchronized ground truth of the sensor pose for extensive set of sequences of varying lengths and differing levels of challenge such as large visual velocities or large dominant planar surfaces. As in the case of Endres *et al* (2012)[7], our study uses nine Freiburg1 (FR1) sequences in which the Kinect sensor is moved within a typical indoor environment. The resource also provides an evaluation tool that computes the root mean square error (RSME) between an estimated trajectory and the associated ground truth once these have been aligned. Specifically we use the translation and rotation RMSE measures, T-RMSE and R-RMSE respectively, and add the maximum value of the *absolute trajectory error* (MATE) which identifies the maximum sensor positional error anywhere along the estimated trajectory.

5.1 Algorithm Parameters

Optical flow and range flow constraint is calculated from a subset of the image pixels; specifically every 14 horizontal and vertical pixels. Greylevel and depth spatial gradients are computed by plane fitting to a 9×9 neighbourhood around each subsampled pixel. Warping uses a simple *nearest neighbour* interpolation scheme. The maximum number of iterations per pixel is 50. However more typical values are 12-15. New anchors are selected once the degree of overlap between a current depth frame and its current anchor is less than 80%.

5.2 Combining Different Sources of Constraint

The first experiment shows the impact of augmenting the basic range flow algorithm with additional sources of constraint. Tables 1 and 2 present the T-RMSE, R-RMSE and MATE metrics for three different versions of the algorithm based on equation 17:
1. the basic range flow estimator ($\lambda_Z = 1,\ \lambda_{\mathcal{A}} = 0,\ \lambda_I = 0$) in columns 3-5 of Table 1;
2. an estimator using depth and *anchors* ($\lambda_Z = 0.33,\ \lambda_{\mathcal{A}} = 0.67,\ \lambda_I = 0$) in columns 6-8;
3. an estimator using depth, anchors and intensity ($\lambda_Z = 0.25,\ \lambda_{\mathcal{A}} = 0.5,\ \lambda_I = 0.25$) in columns 3-5 of Table 2.

Sequence	Length	**Depth Constraint Only**			**Depth and Anchor Constraint**		
	(frames)	T-RMSE	R-RMSE	MATE	T-RMSE	R-RMSE	MATE
FR1 xyz	798	4.1 cm	3.1°	7.9 cm	3.2 cm	2.0°	6.6 cm
FR1 room	1360	7.4 cm	3.8°	35 cm	6.5 cm	3.2°	40 cm
FR1 rpy	722	7.9 cm	5.1°	23 cm	7.2 cm	4.9°	21 cm
FR1 360	755	10.6 cm	7.5°	24 cm	9.3 cm	6.7°	40 cm
FR1 teddy	1418	10.5 cm	5.5°	32 cm	9.7 cm	4.2°	34 cm
FR1 desk2	639	9.8 cm	6.4°	44 cm	12.6 cm	7.3°	41 cm
FR1 plant	1139	18.2 cm	5.8°	98 cm	13.7 cm	4.2°	44 cm
FR1 desk	595	Failed at frame 300			Failed at frame 300		
FR1 floor	1245	Failed			Failed at frame 0		

Table 1: Measuring the Impact of Sources of Constraint

Sequence	Length	**Depth, Anchor and Intensity**			**RGB-D SLAM**[7]		
	(frames)	T-RMSE	R-RMSE	MATE	T-RMSE	R-RMSE	MATE
FR1 xyz	798	3.3 cm	2.5°	6.5 cm	2.1 cm	0.9°	-
FR1 room	1360	8.5 cm	3.1°	44 cm	21.9 cm	9.0°	-
FR1 rpy	722	7.5 cm	4.9°	23 cm	4.2 cm	2.5°	-
FR1 360	755	10.3 cm	6.7°	40 cm	10.3 cm	3.4°	-
FR1 teddy	1418	10.4 cm	4.8°	34 cm	13.8 cm	4.8°	-
FR1 desk2	639	12.6 cm	7.3°	41 cm	10.2 cm	3.8°	-
FR1 plant	1139	13.9 cm	4.2°	40 cm	14.2 cm	6.3°	-
FR1 desk	595	23.7 cm	12.5°	83 cm	4.9 cm	2.4°	-
FR1 floor	1245	8.2 cm	2.8°	52 cm	5.5 cm	2.4°	-

Table 2: Comparing the Range Flow Estimator with D-SLAM[7]

While there is considerable variation in performance between sequences, it is clear from a comparison of corresponding metrics that the use of *anchors* significantly reduces drift. Empirically it has been found that a ratio $\lambda_{\mathcal{A}}/\lambda_Z \approx 2$ leads to optimal performance. Figures 1 and 2 provide a visual demonstration of the benefit of this additional temporal support.

Despite the improvement, the method fails entirely on the *desk* and *floor* sequences. Analysis of the frames at which this failure occurs points to a degeneracy in the structure of 3D data: specifically, the scene data is effectively planar. However, when constraint from the intensity image is included (columns 3-5 of Table 2), plausible estimates of the sensor motion continue to be generated. Overall, however, inclusion of constraint based on intensity reduces the accuracy of the motion estimator. Empirically it has been determined that a weight $\lambda_I = 0.25$ generates an optimal balance of increased robustness versus accuracy.

5.3 Comparison with RGB-D SLAM

To provide a comparison of our approach with a state of the art technique, we present the equivalent performance metrics of the RGB-D SLAM method of Endres *et al*[7]. In this approach SIFT intensity features are extracted and matched between frames. Using the depth data associated with these features, the 3D inter-frame motion is recovered using RANSAC. A further optimisation across all previous frames pose is used to generate global consistent pose interpretation. The algorithm is run on the same nine TUM RGB-D Benchmark sequences. Their results are reproduced in columns 6 and 7 of Table 2. A comparison with the corresponding metrics for the *Range Flow* approach (columns 3 and 4 of Table 2) for each of the sequences suggests that their performances are commensurate.

5.4 Execution Times

Table 3 gives a breakdown of the typical processing times for each stage of the algorithm. The actual time is largely determined by the number of iterations before convergence. Reading depth frames from the disk takes a substantial amount of time ($\approx$ 43 msec) as compared to the 3 msec required to read a depth and intensity frame from the Kinect device. As a consequence, the algorithm does generate real-time pose estimates when directly connected to a Kinect device. This C++ coded algorithm is compiled in Release Mode in Visual Studio and run on a Samsung P460 Notebook i.e. a Intel Core2 Duo T6400 @ 2.00GHz processor.

Stage	**Time** (msecs)
Reading images	42.8
Anchor Selection	3.0
Masks and gradients	13.3
Errors and motion	12.4
Warping images	8.6
Total	80.1

Table 3: Algorithm Times

6 Discussion

A real-time range flow-based estimator has been developed and evaluated on the TUB RGB-D Benchmark. The estimator recovers the translation and rotation components of a sensors motion and integrates these temporally. To minimise drift in the pose, additional temporal constraint is provided through the use of anchor frames. Compared to traditional SLAM approaches, range flow estimators enjoy significant advantages. No computationally expensive recovery of complex image features is required, Moreover no tracking of these features is required. Range flow and optical flow constraints are computed relatively cheaply using simple smoothed gradients and temporal differences on simple rectangular grids. A notable failure mode of the basic approach arises where there is insufficient constraint to perform the matrix inversion in equation 17. The integration of optical flow constraints robustly addressed this problem in a very straight forward manner.

Using the TUM RGB-D Benchmark, pose accuracy can reasonably be judged as commensurate with SLAM approaches (as represented by the RGB-D SLAM system) but available at a fraction of the computational cost. Indeed a real-time implementation is running on an old Samsung P460 Notebook. Given the low computational cost of generating reasonably accurate pose estimates, the presented approach could usefully bootstrap more computationally expensive techniques.

References

[1] P. Anandan, J.R. Bergen, K.J. Hanna, and R. Hingorani. *“Hierarchical Model-Based Motion Estimation"*. Kluwer Academic Publishers, Boston, 1993.

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M.J. Black, and R. Szeliski. “A Database and Evaluation Methodology for Optical Flow". *International Journal of Computer Vision*, 92(1): 1–31, March 2011.

[3] John L. Barron and Hagen Spies. “The Fusion of Image and Range Flow”). In *Proceedings of the 10th International Workshop on Theoretical Foundations of Computer Vision: Multi-Image Analysis*, pages 171–189, London, UK, UK, 2001.

[4] P.J. Besl and N.D. McKay. “A Method for Registration of 3-D Shapes”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239âĂŞ256, 1992.

[5] M. J. Black and P. Anandan. “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields". *CVGIP: Image Understand.*, 63(1):75–104, 1996.

[6] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. “MonoSLAM: Real-time Single Camera SLAM”. *IEEE Transactions Pattern Analysis Machine Intelligence*, 29(6):1052–1067, 2007.

[7] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. “An Evaluation of the RGB-D SLAM System". In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, St. Paul, MA, USA, May 2012.

[8] P. Giaccone and G.A. Jones. “Spatio-Temporal Approaches to Computation of Optical Flow". In *British Machine Vision Conference*, pages 420–429, Colchester, Sept. 1997.

[9] J.-M. Gottfried, J. Fehr, and C.S. Garbe. “Computing Range Flow from Multi-modal Kinect Data”. In *Advances in Visual Computing*, volume 6938 of *Lecture Notes in Computer Science*, pages 758–767. Springer, 2011.

[10] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. “3D Pose Tracking with Linear Depth and Brightness Constraints”. In *International Conference on Computer Vision*, pages 206–213, 1999.

[11] B.K.P. Horn and J. Harris. “Rigid Body Motion from Range Image Sequences”. *CVGIP: Image Understanding*, 3(1):1–13, January 1991.

[12] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. “Kinectfusion: Real-time dense surface mapping and tracking”. In *IEEE Intern. Symp. on Mixed and Augmented Reality*, 2011.

[13] J. Quiroga, F. Devernay, and J. Crowley. “Scene Flow by Tracking in Intensity and Depth Data”. In *Computer Vision and Pattern Recognition Workshops*, pages 50–57, Providence, June 2012.

[14] G. Klein S.A. Holmes and D.W. Murray. “An O(Nˆ2) Square Root Unscented Kalman filter for Visual Simultaneous Localization and Mapping”. *IEEE Transactions Pattern Analysis Machine Intelligence*, 31(7):1251–1253, 2009.

[15] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. “A Review of Recent Range Image Registration Methods with Accuracy Evaluation". *Image and Vision Computing*, 25(5):578–596, May 2007.

[16] Hagen Spies, Bernd Jahne, and John L. Barron. “Range Flow Estimation”. *Computer Vision and Image Understanding*, 85:209–231, 2002.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. “A Benchmark for the Evaluation of RGB-D SLAM Systems". In *Int. Conf. on Intelligent Robot Systems*, October 2012.