iGRaND: An Invariant frame for RGBD Sensor Feature Detection and Descriptor Extraction with Applications

Andrew R. Willis[a] and Kevin M. Brink[b]

[a]University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223
[b]Air Force Research Laboratory, Munitions Directorate, Eglin AFB, FL, 32542

ABSTRACT

This article describes a new 3D RGBD image feature, referred to as iGRaND, for use in real-time systems that use these sensors for tracking, motion capture, or robotic vision applications. iGRaND features use a novel local reference frame derived from the image gradient and depth normal (hence iGRaND) that is invariant to scale and viewpoint for Lambertian surfaces. Using this reference frame, Euclidean invariant feature components are computed at keypoints which fuse local geometric shape information with surface appearance information. The performance of the feature for real-time odometry is analyzed and its computational complexity and accuracy is compared with leading alternative 3D features.

1. INTRODUCTION

Recent introduction of consumer-grade 3D imaging devices is transforming how we use imaging sensors. These sensors provide a data stream that includes both visual, i.e., RGB color, images and depth images of the viewed scene. When merged, each image RGB+Depth, or RGBD, image provides appearance and position information for surfaces within the range of the sensor.

Direct availability of depth information avoids the computationally expensive and error-prone process of estimating the depth from collections of color images in time, or via multi-camera capture. As such, RGBD cameras have gained popularity in robotic applications for perception, navigation and mapping.[1–3]

Despite the benefits afforded by directly sensing the scene depth, new challenges arise. When operated at full resolution, the Primesense RGBD sensor provides a data stream containing RGB color images and depth images at a resolution of 640x480 pixels at a rate of 30Hz. Microsoft devices, e.g., the Kinect Xbox360, encode depth with an unsigned 2-byte value at each pixel indicating the scene depth in mm. Asus devices, e.g., the XTion Pro Live, encode depth with a floating point (4-byte) value at each pixel indicating the scene depth in m. The data rate of these devices given their distinct encoding for the depth is 44MB/sec for the Kinect Xbox 360 and 62MB/sec for the Asus XTion Pro Live. Processing images in real-time given this high data-rate challenges current desktop computers. As such, special performance considerations are required when designing algorithms that will run in real-time or near real-time on the limited computing resources of an embedded computer.

Existing approaches for 3D point cloud and RGBD image feature extraction often manifest as a generalization of a previously existing feature detection approach originally used to solve the same problem in 2D images. Unfortunately, RGBD image data exhibits a number of phenomena that makes processing these images distinct from processing 2D color images. The four most significant phenomena that prevent direct generalization of 2D image algorithms to RGBD depth images are listed below:

1. Invalid, i.e., out-of-range, values occur when the scene depth is either too short or too far for sensor to measure. When this occurs, the pixel value is assigned a special out-of-range value. The Microsoft Kinect Xbox 360 out-of-range value is 0 and the Asus XTion sensor assigns out-of-range pixels to Not-a-Number (NaN) values as defined by the IEEE 854 floating point number standard.

Further author information: (Send correspondence to A. Willis)
A. Willis: E-mail: arwillis@uncc.edu, Telephone: 1 704 687 8420

2. Constant sample spacing in the image can correspond to highly-irregular spatial samplings of the 3D surface. This is especially true when the view direction is nearly tangential to the measured surface. This warps the geometry of the depth image such that planar 3D surfaces manifest as quadric surfaces in the depth image. The curvature of these quadric surfaces in the depth image (x, y) directions depends on the relative orientation of the surface and the view direction and these curvatures are zero when the view direction is perpendicular to the surface (parallel to the surface normal).

3. Measurement noise for RGBD (X, Y, Z) measurements are a non-isotropic and non-linear (quadratic[4]) function of depth. This imposes a unique noise model on each pixel in the depth image that depends on both the specific (x, y) image location and the image depth, $Z = D(x, y)$, measured at that location. Further, the standard noise model[4] is valid for smoothly varying surface regions and has been shown to be significantly inaccurate at locations of depth discontinuities.[5]

4. RGBD sensors have systematic depth errors at the image corners. Researchers have estimated these errors to be approximately ~1cm. at a distance of 1m. and they increase rapidly (as a quadratic function) of the measured depth. While calibration of the RGBD camera can correct most of this distortion (reducing it to $\pm 2mm.$ at 6m), users must perform the RGBD camera calibration procedure and RGBD processing algorithms must accommodate these calibration parameters for them to have impact.[6]

These difficulties are faced by any algorithm that seeks to process RGBD image data and preclude useful outcomes from classical approaches such as linear filtering which work best when their underlying assumptions: linearity, stationary (or wide-sense stationary) noise, and uniform sampling are satisfied. In fact, typical RGBD image data simultaneously violates all of these assumptions![4–6] Hence, additional processing logic must exist in RGBD algorithms to cope with these phenomena.

The contribution of this article is to describe a stable Euclidean invariant coordinate system for those pixels in the RGBD image where the visual image gradient is well-defined, i.e., at edge pixels of the color image. We refer to this coordinate system as an iGRaND coordinate system since its two mutually perpendicular axes are determined by the local image gradient (**iGR**) and the normal to the depth surface (**aND**). This concept is then abbreviated to the acronym: **i**mage **GR**adient **a**nd **N**ormal **D**epth; **iGRaND**. Note that iGRaND refers is a local Euclidean invariant coordinate system for RGBD data (or any RGB/intensity-attributed surface data) and, as such, it is not an RGBD image feature or descriptor. Yet, as part of the contribution of this article we define a new feature detector and descriptor based on the iGRaND frame to demonstrate that reliable detectors and descriptors can be defined using the iGRaND frame whose performance is competitive with current state-of-the-art. In fact, as our results show, our chosen feature detector and descriptor outperforms many leading descriptors particularly when the viewed scene includes rich geometric structure. This circumstance is common in man-made and cluttered indoor scenes which makes this performance boost important for applications in these environments, e.g., robotic navigation.

2. PRIOR WORK

Feature detection and descriptor extraction in images is a classical topic in the computer vision and pattern recognition literature. Feature detection algorithms process image measurement data and provide as output a collection of (x, y) pixel locations seen as distinct using some distinctiveness metric. As described in 7 , there are several criteria by which the feature points are selected which include: (1) repeatability, (2) distinctiveness, (3) locality, (4) quantity, (5) accuracy and (6) computational efficiency. Feature descriptors seek to obtain a vector of values at each feature location which can be easily and reliably compared against feature descriptors from a second image to identify corresponding (x, y) feature locations in the image pair. Algorithms for feature detection and descriptor computation is a topic of such importance that it is not uncommon for book chapters[8] or entire books[9] that discuss the latest algorithms available for solving this problem.

Research on 3D features was limited prior to the availability of inexpensive depth capture devices. Algorithms developed during this time were typically designed to extract feature points from LiDAR point-cloud data which is typically a collection of 3D range measurements collected on a discretized grid imposed on the sphere.[10,11] For some sensors, e.g., the Faro/SICK LiDAR sensors, range information is complemented with a surface reflectance measurement. However, due to the low resolution and significant sensor noise inherent in these measurements, most algorithms discard this information and work directly on the point cloud data.

Research on RGBD feature detection and descriptor extraction has generally followed a trend of extended existing approaches for 2D image processing to include depth information. This includes extensions of the Harris Corner detector[12] to 3D,[13, 14] extension of the the SUSAN corner detector[15] to 3D,[16] extension of the SIFT detector[17] to 3D,[18, 19] extension of the SURF detector[20] to 3D[21] and extension of the Histogram of Gradients (HOG) detector[22] to 3D.[23] Other detectors use intrinsic, i.e., Euclidean invariant, properties of the 3D geometric data to compute geometric features such as the surface curvature,[24] spin images,[25] Point Feature Histograms (PFH)[26] and Intrinsic Shape Signatures.[21] There are several surveys which describe these methods and their performance.[27–29]

A related approach for RGBD data is the Normal Aligned Radial Features (NARF) proposed in 30. This approach prescribes an invariant coordinate system for depth data very similar to the coordinate system derived from the second fundamental form, i.e., the coordinate system defined by the surface principal curvatures at the point. NARF adds robust processing to the standard curvature estimation procedure which includes smoothing, non-maximum suppression and a search procedure to reliably extract a principal (maximum) direction of curvature.

As evident from this discussion, there is a wealth of approaches available for surface feature detection. Unfortunately, these methods either carry a large computational burden or focus uniquely on the geometric information in the surface data. Those having high computational complexity cannot be applied in real-time scenarios and those that omit consideration of the intensity image information discard information that can be highly discriminative and, in doing so, their recognition performance suffers. The proposed approach has low computational complexity and includes functions that consider *both intensity and geometric shape for both detection and description* of feature locations.

Our feature detection and descriptor extraction algorithm is most closely related to the Oriented FAST and Rotated BRIEF (ORB) detector and descriptor algorithm described in 31. Here, the authors note a shortcoming in the BRIEF descriptor whose recognition performance degrades significantly under image rotation. Similar to our detector, the authors ORB detection algorithm merge the FAST corner detector[32] with an orientation detection step that assigned each corner a orientation using the intensity centroid method from 33 described in §4.2. ORB descriptor extraction uses a modified form of the BRIEF descriptor,[31] construct via a sequence of binary tests evaluated in the vicinity of the detected feature points. The modification is designed as rBRIEF as it uses the corner orientation to rotate the BRIEF test patterns into a common, rotation invariant, coordinate system. They show that orienting the BRIEF test patterns prior to computing the brief descriptor adds significant recognition performance to the extracted descriptor, especially under image rotation.

While the 2D ORB approach[31] is similar to our approach, our feature detection and descriptor extraction algorithm are for "intensity+depth" (RGBD) images. As such, the coordinate frame is a 3D iGRaND frame derived from depth-and-intensity values rather than a 2D intensity-only frame and the rBRIEF descriptor extracted at a corner location and includes binary tests on both the 3D surface geometry and intensity rather than only the 2D intensity image. These tests are important in practice as shown in our results section. Specifically, confusing visual patterns, e.g., regular patterns like a checkerboard, are likely to cause poor recognition performance for intensity-only approaches as the descriptors at the chessboard corners will have similar values. For our approach these patterns are better distinguished when they occur in locations of geometric variation. Further, when the problematic visual patterns lie on a flat surface, they will be de-emphasized by our detector which seeks locations that have corner-like behavior in *both* the depth and intensity images. In these circumstances, the iGRaND algorithm reverts to an ORB-style detector.

3. DERIVATION OF THE IGRAND FRAME

The definition of the iGRaND-frame is obtained by appending an intensity term to the second fundamental form, (II-form), of a generic geometric surface.[34] The appended intensity term is derived from the RGBD intensity image, $I(x, y)$, that is co-located/registered with the depth image, $D(x, y)$. As we will show, the appended II-form allows for definition of extrinsic invariant iGRaND coordinate frames. However, since the second fundamental form without the appended term is also an extrinsic invariant we must justify the benefits provided by appending the intensity-term to the classical II-form.

In it's classical formulation,[34] the second fundamental form specifies a unique 3D coordinate frame, referred to as the II-frame, without need for an additional intensity term. To form the II-frame, one takes the three frame axes as the surface normal and the directions of maximum / minimum surface curvature.[24, 35] As mentioned previously, these axes are extrinsic invariants and the resulting 3D coordinate frame therefore is also an extrinsic invariant. *Unfortunately, these axes are well-defined/unique only when the surface has two distinct curvatures.* And, even when the principal curvatures and their associated directions are available, the uncertainty in assignment of the signs to eigenvalues and their associated eigenvectors necessary to estimate the frame may generate reflected coordinate axes. To summarize, the II-frame is not a feasible option as an extrinsic invariant RGBD frame for the following reasons:

1. Stability of the II-frame[34] depends on stable estimates of surface curvature, a quantity typically avoided when processing real-world data as it is generally computationally expensive to compute and it is notorious for being corrupted by noise which is amplified by applying multiple differential operators to the measured position data.

2. The II-frame fails to provide a unique frame for planar, spherical and nearly-planar / nearly-spherical surfaces which commonly occur in real world scenes; especially for man-made and indoor scenes where RGBD sensors are typically applied.[34] Further, when the II-frame does exist, the sign of the axes that span the tangent plane are not unique resulting in potential reflection of the tangent plane axes.

For these reasons, purely geometric parameterizations are not desirable when seeking to find invariant coordinate frames for generic RGBD image data.

To cope with these shortcomings, our generalization adds a fourth term to the second fundamental form for the geometric surface whose value is directly computed from the intensity image; $i = I(x, y)$. The contribution of this term adds a fourth dimension to the second fundamental form. The amended formulation for the second fundamental form affords a new coordinate frame, the iGRaND-frame, to be defined as the direction of the surface normal, $\mathbf{n}$, the direction of the image gradient in the surface tangent plane, $\mathbf{u}$, and the cross product of these two vectors, $\mathbf{v} = \mathbf{n} \times \mathbf{u}$. The extended version of the second fundamental form is non-degenerate at surface points where either (a) the surface has two distinct curvatures or (b) the image gradient is not zero. *The immediate benefit of the expanded form is that unique extrinsic invariant frames can be formed at planar and spherical surface points where the intensity is not constant.* Further, computation of the frame is a function of only the first order derivatives of the depth image (to obtain $\mathbf{n}$) and first order derivatives of the gray scale image (to obtain $\mathbf{u}$). This promises to make these frames more robust to measurement noise than frames derived from second-order derivatives, e.g., the eigenvectors of the Hessian matrix. For RGBD images, recovery of an iGRaND coordinate frame from RGBD data requires only that the surface patch around the chosen point is visible, i.e., is measured, and that the intensity image has a non-zero gradient value at the chosen point.

4. COMPUTATION OF THE IGRAND FRAME

Our derivation of the iGRaND-frame operates in two distinct domains: (1) the 3D (X, Y, Z) coordinates derived from the depth image and (2) the (x, y, i) coordinates of the intensity image. Please note that our notation uses upper-case letters, e.g., X, to denote 3D coordinates (in $m.$) and lower-case letters, e.g., x, to denote image-space coordinates (in pixels). Also, bold face variables, e.g., $\mathbf{n}$, denote vector valued variables.

Both the depth image and the intensity image are modeled as a Monge patch surface. In both cases we seek to estimate the gradient of the surface, i.e., $\nabla S(X, Y, Z)$ and $\nabla I(x, y)$. For Monge patch surfaces, the normal to the surface is a direct function of the surface gradient. For the intensity image, the outward pointing surface normal, i.e., the normal pointing towards the viewer, is given by equation (1). Note that we assume a standard optical/camera coordinate system whose z-axis is taken as the camera's optical axis.

$$\mathbf{n}_I = \frac{\begin{pmatrix} I_x, & I_y, & -1 \end{pmatrix}^t}{\sqrt{I_x^2 + I_y^2 + 1}} \tag{1}$$

For the depth image, the 3D surface normal pointing toward the camera/viewer is defined similarly from the 3D (X, Y, Z) surface coordinates as given by equation (2). Note that we have adopted the standard coordinate system for the camera where the depth, Z, increases as we move forward on the optical axis.

$$\mathbf{n}_S = \frac{\begin{pmatrix} D_X, & D_Y, & -1 \end{pmatrix}^t}{\sqrt{D_X^2 + D_Y^2 + 1}} \tag{2}$$

While equations (1) and (2) appear similar, they are not. To compute $\mathbf{n}_S$ the 3D coordinates (X, Y) of equation (2) must be reconstructed from the depth image using the RGBD 3D point reconstruction equations (see § 4.1 Point Cloud Reconstruction).

The iGRaND frame is defined from the 3D surface normal, $\mathbf{n} = \mathbf{n}_S$, and the projection of the vector $\mathbf{n}_I$ into the 3D surface tangent plane, $\mathbf{u} = \mathbf{n}_I - (\mathbf{n}_I \cdot \mathbf{n}_S)\mathbf{n}_S$. The third axis, $\mathbf{v}$, is taken as the cross product of these axes; $\mathbf{v} = \mathbf{n} \times \mathbf{u}$. For performance, our implementation avoids the normalization constants as prescribed by the divisors in equations (1) and (2). Rather, we store the iGRaND frame as a tuple of 4 values (D_X, D_Y, I_x, I_y). This representation of the iGRaND frame preserves the absolute magnitude of the gradient which is important for corner detection as they are the elements of the second moment matrices for the intensity and depth images as shown in equation (7).

4.1 Point Cloud Reconstruction

Measured 3D (X, Y, Z) positions of sensed surfaces can be directly computed from the intrinsic RGBD camera parameters and the measured depth image values. The Z coordinate is directly taken as the depth value and the (X, Y) coordinates are computed using the pinhole camera model. In a typical pinhole camera model 3D (X, Y, Z) points are projected to (x, y) image locations, e.g., for the image columns the x image coordinate is $x = f_x \frac{X}{Z} + c_x - \delta_x$. However, for a depth image, this equation is re-organized to "back-project" the depth into the 3D scene and recover the 3D (X, Y) coordinates as shown by equation (3)

$$\begin{array}{rcl} X & = & (x + \delta_x - c_x)Z/f_x \\ Y & = & (y + \delta_y - c_y)Z/f_y \\ Z & = & Z \end{array} \tag{3}$$

where Z denotes the sensed depth at image position $D(x, y)$, (f_x, f_y) denotes the camera focal length (in pixels), (c_x, c_x) denotes the pixel coordinate of the image center, i.e., the principal point, and (δ_x, δ_y) denote adjustments of the projected pixel coordinate to correct for camera lens distortion.

4.2 Estimating the Image Gradient

For over 20 years researchers have used the image gradient as an indicator of semantic or structural information in an image.[12, 15, 33] While our iGRaND frame is stable at all locations where the image gradient magnitude is large, adjacent points lying on the same image edge often have similar descriptors. As such, detection of these points often gives rise to incorrect matches in typical use cases where descriptors are matched to solve the image correspondence problem. Hence, as with most detectors, we seek to find locations in the image where the image gradient is multi-valued, i.e., corner locations. These locations are known to be promising locations for descriptor computation and tend to produce distinctive descriptors whose match performance outperforms other location types.[7, 33]

Numerous approaches have been proposed to detect points lying at corner locations. Different approaches for gradient estimation at image corner locations attempt to cope with two difficulties: (1) noise in the image and (2) the gradient operator (a vector of partial differentials) is not well-defined at discontinuities, e.g., corner locations. Many approaches are based on computation of the structural tensor of the image data (also referred to as the second moment matrix) as shown in equation (7). As others have in the past, we adopt the intensity centroid method[33] to estimate the orientation of a corner as shown in equation (4).

$$m_{pq} = \sum_{x,y \in W} x^p y^q I(x,y), \quad \theta = atan2(m_{10}, m_{01}) \tag{4}$$

As shown in 33, the intensity centroid provides corner orientation estimates similar to computing the average of the gradient in x and y but has been found to be more stable. Other leading methods have also adopted this approach to provide orientation for rotation sensitive descriptors. For example, the ORB algorithm[36] uses the orientation provided by the intensity centroid to compute the rotated-BRIEF (rBRIEF) descriptor. Similar to,[36] we compute the intensity centroid using a circular window to make the intensity centroid operator invariant to image rotation.

4.3 Estimating the Depth Gradient

As mentioned in the introduction, computation of the 3D gradient of the depth surface must be handled differently than doing a similar operation on the intensity image. Here, we must reconstruct the 3D (X, Y, Z) values of the surface using the RGBD reconstruction equations (3). Let $X(x, y)$ and $Y(x, y)$ denote images that hold the 3D (X, Y) coordinates reconstructed from the x and y coordinates of equations (3). The 3D surface gradient of the non-uniformly spaced 3D samples can then be computed to second-order accuracy using the central difference operator on the reconstructed 3D coordinate values as shown in equation (5)

$$D_X = \frac{dD(x,y)}{dX(x,y)} = \frac{D(x+1,y) - D(x-1,y)}{X(x+1,y) - X(x-1,y)} \tag{5}$$

$$D_Y = \frac{dD(x,y)}{dY(x,y)} = \frac{D(x,y+1) - D(x,y-1)}{Y(x,y+1) - Y(x,y-1)}$$

To obtain a robust estimate of the local surface gradient, we apply the Sobel filter (a 3x3 Gaussian averaging kernel) to the central difference operators of equation (5) to produce $\widehat{D}_X(x, y)$ and $\widehat{D}_Y(x, y)$; our estimate of the Monge patch surface X and Y partial derivatives, as shown in equation (6).

$$\begin{aligned} \widehat{D}_X(x,y) &= \tfrac{1}{4}\left(D_X(x,y+1) + 2D_X(x,y) + D_X(x,y-1)\right) \\ \widehat{D}_Y(x,y) &= \tfrac{1}{4}\left(D_Y(x+1,y) + 2D_Y(x,y) + D_Y(x-1,y)\right) \end{aligned} \tag{6}$$

The resulting gradient values $\widehat{D}_X$ and $\widehat{D}_Y$ are taken as our estimate for the local surface normal to the depth surface.

5. AN IGRAND FEATURE DETECTOR

Feature detection algorithms process image measurement data and provide as output a collection of (x, y) pixel locations. As described in 7, there are several criteria by which the feature points are selected which include: (1) repeatability, (2) distinctiveness, (3) locality, (4) quantity, (5) accuracy and (6) computational efficiency. For our iGRaND detector, we seek to detect (x, y) image positions of corresponding 3D surface points and seek to reliably extract the same subset of 3D surface points from the scene despite variations in the image scale, illumination and viewpoint.

Our approach for detection seeks achieve these goals by marking locations in RGBD data that have corner-like behavior in *both* the depth image and the intensity image. As a result, the detection approach is sensitive to depth and intensity variations and will tend to prefer marking (x, y) locations where the 3D surface geometry and intensity gradient exhibit corner-like behavior. Typical real-world data often contains a small number of features that exhibit depth and intensity variation simultaneously. Since the iGRaND frame requires a non-zero image gradient, it is sufficient to find only corner locations in the intensity image and then sort these corners by a score that places the subset of image corners that also have depth variation at the top of the list. Using this approach our feature detection algorithm is sensitive to both depth and intensity and, when there is little or no geometric structure in the scene, it defines iGRaND frames at corner locations in the intensity image.

Our corner detection algorithm uses the FAST algorithm[32] to quickly locate corner positions in the intensity image. Then, at each detected corner location, we compute the iGRaND frame parameters (I_x, I_y, D_X, D_Y) at that location and attribute the detected feature with a salience score α as defined in equation (8)

$$\mathbf{S} = \sum_{x=-m}^{m} \sum_{y=-m}^{m} \nabla\mathbf{I}\nabla\mathbf{I}^t + \lambda\nabla\mathbf{D}\nabla\mathbf{D}^t = \sum_{x=-m}^{m} \sum_{y=-m}^{m} \begin{bmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & I_z^2 \end{bmatrix} + \lambda \begin{bmatrix} D_X^2 & D_XD_Y & D_XD_Z \\ D_XD_Y & D_Y^2 & D_YD_Z \\ D_XD_Z & D_YD_Z & D_Z^2 \end{bmatrix} \quad (7)$$

$$\alpha = det\left(\mathbf{S}\right) - \kappa trace^2(\mathbf{S}) \quad (8)$$

where the factor λ is a relative weight of an image intensity difference and a depth image difference and κ the a sensitivity parameter from the Harris corner detection[12] and is typically assigned a value between 0.04-0.15 (our experimental results use $\kappa = 0.04$). While different choices of λ are possible, our experiments seek to normalize the values by their uncertainty. To do so, we estimate the uncertainty of an intensity value as approximately 30% of its range, e.g., 75 for an 8-bit intensity, and the error in a depth value to be approximately 2cm (at a nominal range of 3.5m.). This specifies our experimental value for $\lambda = \frac{75^2}{.02^2}$ which conceptually attributes a geometric variation of 2cm. equivalent to an intensity variation of 75 intensity levels. The magnitude of this scale factor is necessary to make the geometric shape factor significantly in the feature score, α. Note that, for our data sets, over-estimation of the intensity image noise did not adversely impact feature selection as our indoor scenes exhibits corners that often had large intensity gradients (nearing 200 intensity levels). In these circumstances, one must emphasize/bias geometric variations in preference to intensity variations in order to ensure geometric shape variations are a factor in feature selection.

Our implementation of this algorithm uses OpenCV and incorporates a multi-scale detection component which is often desirable for robust recognition. Multi-scale detection uses image pyramids to detect features at different scales by applying the algorithm to resized versions of the recorded intensity image. Our OpenCV implementation takes as input the number of desired features to detect, N, the number of scales, L, to consider and the scaling factor, γ, that determines the size reduction between sequential scales, e.g., $\gamma = 2$ indicates the image at level $l + 1$ has half the size of the image at level l. Given these input parameters, the steps of our detection algorithm are listed below:

1. If necessary, the RGB (color) image is converted to a gray scale image.

2. An image pyramids consisting of L levels is constructed by resizing the recorded gray scale image, I, and depth image, D.

3. For each pyramid level, the FAST algorithm[32] is applied and provides as output a set of candidate (x, y) feature/corner locations in the intensity image, $C(L)$. Each candidate corner location is then assigned a scale-dependent score $\alpha(L)$ as described in equation (8).

4. The final set of features is computed by sorting the set $C(L)$ by the feature score, $\alpha(L)$, and returning the first $(\frac{N}{L})$ occurrences of a corner for each level.

The algorithm described in steps (1-4) above provide as output a collection of N feature points whose salience is a representative sampling of the depth-sensitive iGRaND feature detections across all computed image scales.

6. AN IGRAND FEATURE DESCRIPTOR BASED ON ORB

Feature descriptors seek to obtain a vector of values at each feature location which can be easily and reliably compared against feature descriptors from a second image to identify corresponding (x, y) feature locations in the image pair. Since our feature detector is intended to locate similar 3D (X, Y, Z) surface points, the feature descriptor we seek must provide extrinsic invariant measurements from the image data and simultaneously be distinct enough from other descriptors to prevent the occurrence of incorrect descriptor correspondences matches.

Our approach extracts extrinsic invariant depth features using the detected iGRaND frames to create binary descriptors that capture *both* intensity and depth variations in the RGBD data at each detected feature location. This is accomplished by augmenting the ORB/oriented-BRIEF binary descriptors[36] with bits that result from evaluating a sequence of Euclidean-invariant depth tests, taken in the iGRaND frame, at the feature location.

When applied, the BRIEF binary features simply test the relative values of intensities at locations in the vicinity of the feature point. In 31, the authors explore several rationales for choosing the locations points for each test. They found that choice of locations having a random distribution provide the best recognition results. While the authors suggest using a Gaussian distribution test point selection, uniform distributions for the test point locations and a uniformly sampled polar distribution for test point locations provide nearly the same recognition rate.[31] The rBRIEF test at the image location pair (x_0, y_0) and (x_1, y_1) returns a value of one or zero depending on which location has higher intensity as shown in equation (9).

$$\tau_I(x_0, y_0, x_1, y_1) = \begin{cases} 0 \quad if \quad I(x_0, y_0) < I(x_1, y_1) \\ 1 \qquad\qquad\qquad otherwise \end{cases} \tag{9}$$

Our extension to the rBRIEF descriptor uses the same test pattern rationale and a similar test. Yet, instead of using the intensity values, we use height of the test point locations when measured in the local iGRaND frame. Given a detected corner location at position (x, y), let $\mathbf{o}$ denote the 3D position of the iGRaND frame generated by reconstructing the 3D position (X, Y, Z) from the depth image value at $D(x, y)$ and $\mathbf{n}$ denote the iGRaND z-axis at $\mathbf{o}$ as defined in § 4.3. Note we assume that the intensity and depth images are registered, i.e., the intensity observed at $I(x, y)$ is measured from the (X, Y, Z) surface point given by applying the reconstruction equations (3) to the depth $D(x, y)$. Similar to equation (9) our descriptor bits are computed by testing the Z-coordinates of random locations in the vicinity of the corner location in the iGRaND frame. Equation (10) shows the form of our binary depth test function when evaluated at the test point locations (x_0, y_0) and (x_1, y_1) in the depth image.

$$\tau_D(x_0, y_0, x_1, y_1) = \begin{cases} 0 \quad if \quad (Z(x_0, y_0) - \mathbf{o}) \cdot \mathbf{n} < (Z(x_1, y_1) - \mathbf{o}) \cdot \mathbf{n} \\ 1 \qquad\qquad\qquad\qquad otherwise \end{cases} \tag{10}$$

The complete descriptor is formed by concatenating results of a sequence of intensity and depth tests into a single binary value as shown in equation (11).

$$f_{n_d}(x, y) = \sum_{1 \leq i \leq n_A} 2^{i-1} \tau_I(x_0, y_0, x_1, y_1) + \sum_{n_A + 1 \leq i \leq n_d} 2^{i-1} \tau_D(x_0, y_0, x_1, y_1) \tag{11}$$

The resulting binary feature vector, $f_{n_d}(x, y)$, is composed from the result of n_d binary tests. Our depth-sensitive version of this binary descriptor includes n_A bits of binary information resulting from the image intensity tests of equation (9) and $n_d - n_A$ bits of binary intensity resulting from the depth image tests of equation (10). The ratio of intensity tests to depth tests is controlled by the algorithm parameter $\gamma = \frac{n_A}{n_d}$. Four our experimental results we use four intensity image tests and four depth image tests that are concatenated into an 8-bit binary descriptor ($\gamma = 0.5$).

Typical comparison approaches for binary feature vectors use Hamming distance as a metric which counts the number of distinct bits in the two binary feature vectors. By storing all of the binary tests as bits in one or more bytes the Hamming distance can be quickly computed by performing a logical XOR between the corresponding binary descriptors and counting the number of ones in the resulting value.

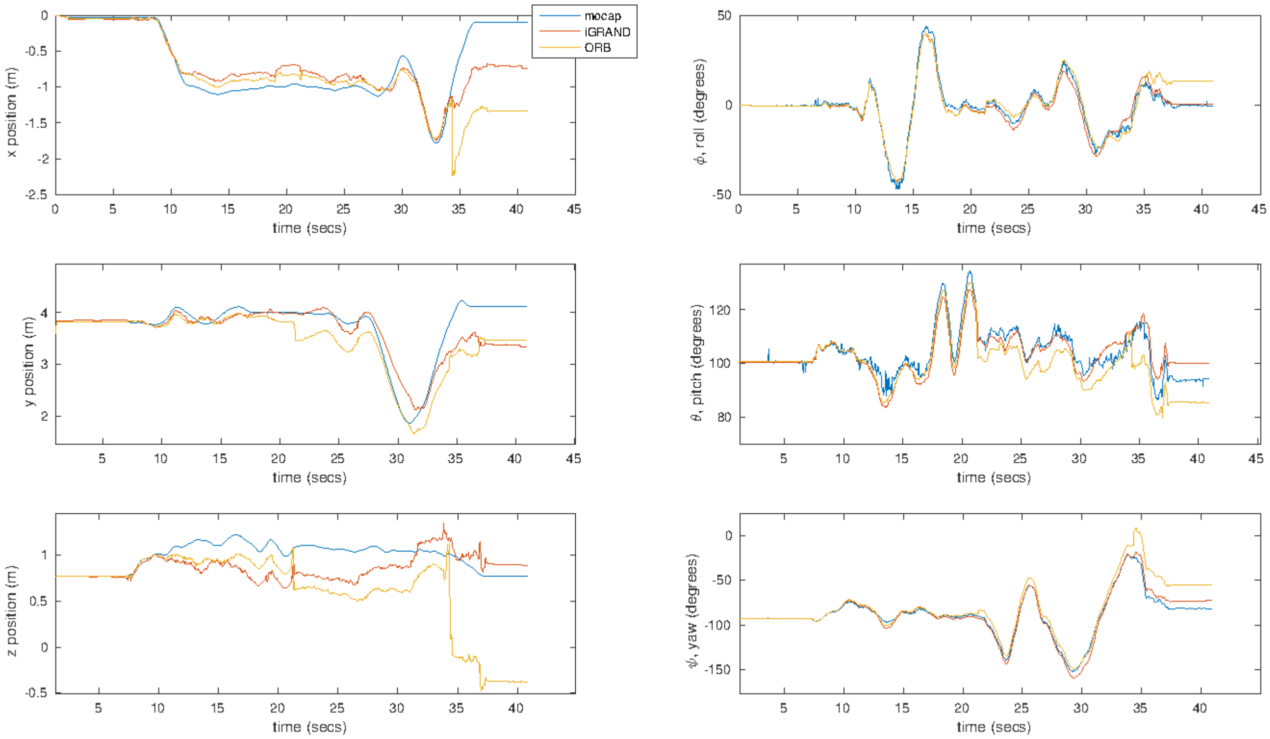


Figure 1: The estimated 6DoF (X, Y, Z) position (left column) and $(roll, pitch, yaw)$ orientation (right column) of the RGBD camera are shown when estimated using the iGRaND (red) and ORB (yellow) algorithms. Note that iGRaND significantly outperforms the ORB feature as it tracks the (ground-truth/motion capture) pose (blue) more accurately.

7. RESULTS

Our evaluation of the proposed iGRaND-based detector and descriptor applies these algorithms for the purpose of computing real-time odometry. In this case, the odometry algorithm uses the feature detector and descriptor algorithms to identify distinctive locations in the scene and extract Euclidean invariant values at these locations that can be matched when the same scene point is viewed from a different viewpoint. Distinctive locations in the RGBD image are provided by a feature detection algorithm, e.g., the algorithm of §5. The feature descriptor algorithm, e.g., the algorithm of §6, extracts a Euclidean invariant value, i.e., descriptor, at each detected image location.

Our visual odometry algorithm[37] matches feature descriptors computed at detected feature locations in sequential frames of RGBD image data. The odometry algorithm aligns 3D (X, Y, Z) surface locations from the RGBD depth images at matched descriptor locations to estimate the motion of the camera during the time that passed in-between the two recorded images. As such, odometry values are estimates for the vehicle velocity, i.e., they indicate how the position and orientation of the vehicle changed between the two recorded RGBD images.

The accuracy of the visual odometry estimate depends upon the measured scene content and how effectively the feature detection and feature descriptor algorithms perform for that content. The most important aspect of the feature detector is its ability to reliably detect the same 3D distinctive scene locations despite changes in the environment such as the viewpoint or scene illumination. The most important aspect of the feature descriptor is to facility identification of correct correspondences between features in successive image frames. Accomplishing this goal requires computation of extrinsic invariant values at detected feature locations that are distinct from other descriptors so that correct correspondences can be computed.

Since the visual odometry algorithm accuracy is directly tied to performance of the feature detector and descriptor extractor algorithms we use the overall accuracy of the visual odometry as an indication of the

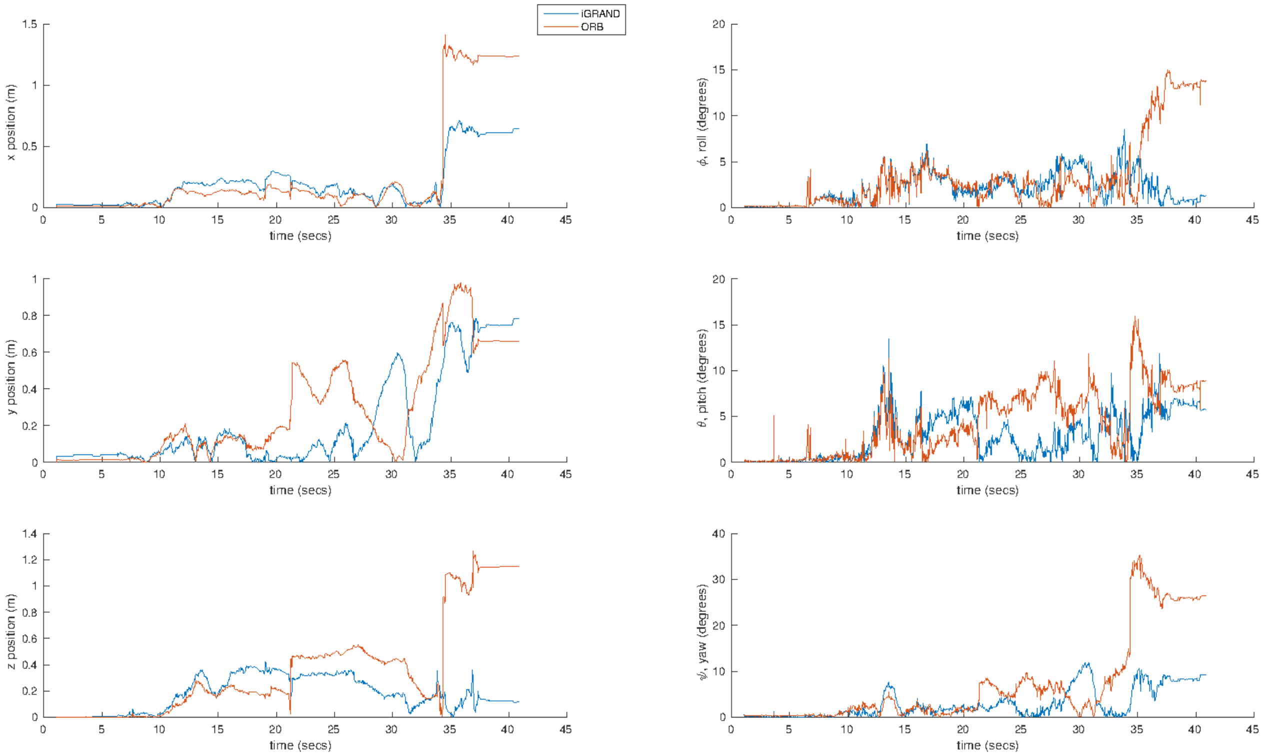
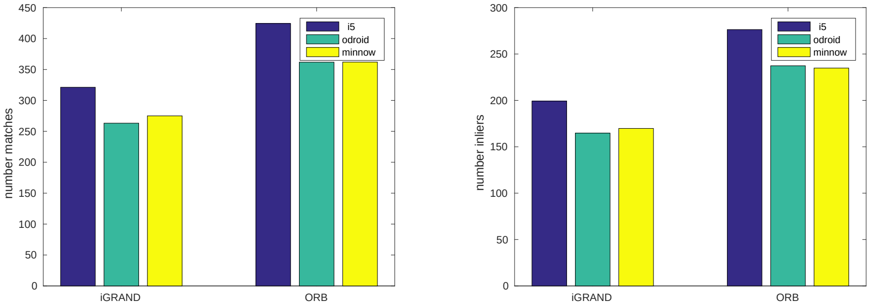


Figure 2: The 6DoF absolute trajectory error, i.e., the absolute difference between the motion capture and the estimated pose is shown when using the iGRaND (blue) and ORB (red) algorithms. Note that iGRaND significantly outperforms ORB in all channels apart from the Y position and *yaw* angle where the two algorithms have similar error relative to ground-truth.

performance of *both* the proposed detector and descriptor algorithms. Performance is measured by feeding a recorded data stream of real-time RGBD sensor data to the iGRaND algorithms and it's closest relative, the ORB feature. We feel this is a good metric for comparison because the ORB feature is closely related in its technical definition, has similar computational cost, and is a state-of-the-art feature. Our application-driven evaluation is distinct from other approaches in the literature which apply detection and descriptor algorithms on simulated or tightly-controlled image data. While this approach has advantages, our interest is to apply these algorithms for real-time robotic navigation and we feel that the presentation of the results in this way are more relevant to this application and also provide an indication of performance for other computer vision applications that use the same underlying feature detection and descriptor extraction pipeline, e.g., 3D mapping, 3D scanning, SLAM, etc.

Our experimental setup uses the RGBD odometry algorithm[37] implemented in the Robot Operating System (ROS)[38] to compute visual odometry using an XTion Pro Live RGBD camera at full-frame (640x480) resolution and framerate (30Hz). The frame-to-frame odometry performance is tracked by calibrating the pose of the RGBD camera to a motion capture system (Optitrack). The motion capture data is used as ground truth for computing odometry error and averages $\sim \pm 0.5$mm positional and $\sim \pm 1$ degree of angular error. Motion capture data captured simultaneously with the RGBD sensor data and captures the ground truth camera pose at a rate of 100 Hz. In our experiments, we initialize the RGBD camera pose to coincide with the measured pose as given by averaging five seconds (500 samples) of motion capture data while the RGBD camera is stationary. After initialization, the pose of the RGBD camera is measured by the motion capture system independent from the pose obtained via the time integration of the frame-to-frame odometry estimates.

Results are generated from a 40 second hand-held flight of the RGBD camera in the motion-tracked space. During this time, the sensor views an indoor environment (the laboratory) which includes office furniture, boxes,



(a) (b)

Figure 3: (a,b) show the correspondence computation performance on three different computers using the same recorded sensor data. Both ORB and iGRaND detect nearly 1000 features in each image. (a) shows that the ORB algorithm averages approximately 10% (100) more matches than iGRaND. (b) shows that this margin is nearly the same in terms of the number of inliers used to estimate odometry. Since Figure 2 shows smaller trajectory errors for iGRaND. We believe that even though there are less matches from the iGRaND algorithm, the matches generated for iGRaND features are more reliable than ORB features.

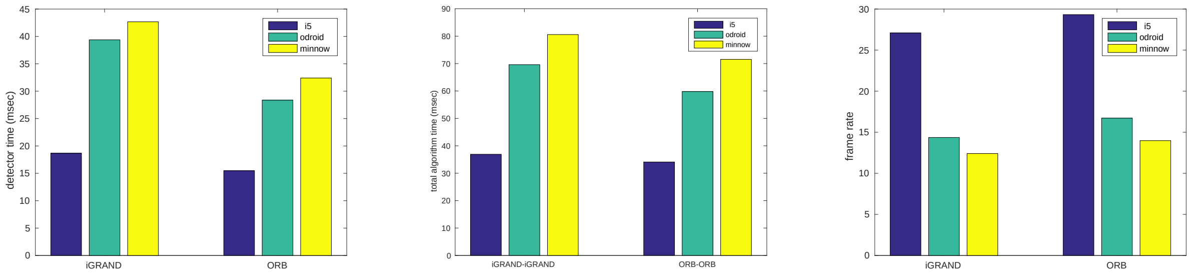
chairs and a camera calibration pattern. Hand-held flight provides opportunities to stress the odometry algorithms with steep pitch, roll and yawing motions which would be difficult to track and difficult to control in practice within this environment. The full 6DoF range of motions were generated in the experiment and includes motions having simultaneous (X, Y, Z) and (roll, pitch, yaw) variations at positional velocities approaching 1m/s and angular velocities approaching 100 degrees/s. The experiment also includes medium (∼2m. range) and long (∼5m. range) depth images and images of highly distinct surfaces, e.g., shipping boxes and textured floor tiles as well as visually confusing surfaces, e.g., a calibration pattern consisting of regularly spaced black squares on a white background. All of these contexts promise to provide opportunities to observe and characterize the odometry parameter estimate accuracy under a wide range of real-world indoor scene variations.

Figure 1 shows the pose, i.e., the estimated position and orientation, of the RGBD camera during the experiment. The results indicate that for our experimental data, the iGRaND features outperformed their closest relative, ORB, by a significant margin.

Figure 2 shows the absolute trajectory error, i.e., the magnitude of the error in the estimated position and orientation relative to ground truth, for each algorithm during the experiment. Again, the results indicate that the depth-sensitive points found and matched using the iGRaND frame track the camera position and orientation more accurately than their closest relative, the ORB feature. While iGRaND outperforms ORB, there are still errors and certain scenes generate larger odometry errors than others. To address this, the algorithm[37] also produces a 6DoF covariance which is sensitive to the estimate stability and allows odometry estimates to be dynamically weighted in an associated navigation system.

Figure 3(a,b) show the match performance, i.e., the number of corresponding descriptors found in each image pair, for the experimental data when processed on three different CPU platforms: (1) a desktop Intel i5 (2.67GHz), (2) a Arm7 8-core Odroid XU4, and (3) a Minnowboard-MAX. For both the iGRaND and ORB algorithms the total number of detected features was limited to 1000 locations in each image. Figure 3(a) shows that during the vehicle motion (time interval 5s-35s) both algorithms were able to match approximately 30% or 300 of the 1000 detected features with the ORB algorithm typically finding about 10% more correspondences the iGRaND algorithm. As shown in figure 3(b), approximately 10% of these matches were rejected by RANSAC leaving about 20% of the original 1000 matches as inliers that ultimately determine the value of the odometry estimate. Despite the fact that there are less matched iGRaND features, we believe that the correspondences found by iGRaND were more reliable as evidenced by the smaller absolute trajectory errors shown in Figure 2.

Figure 4(a-c) show the computational cost analysis for the iGRaND and ORB algorithms on the same computing platforms of Figure 3. Figure 4(a) shows that the detector for ORB requires approximately 27%



(a) (b) (c)

Figure 4: (a-c) compare the ORB and iGRaND feature computational cost on three different computers. (a) shows the source of the biggest increase in cost which is due to computation of the depth structural tensor of equation (7). (b,c) show that, despite the increased computational cost, the iGRaND detector is a real-time detector which can run at rates of over 25Hz on an Intel i5 processor at full frame (640x480) resolution.

less time than the iGRaND algorithm. We attribute this difference to the additional computation required to construct the depth image pyramid and to compute the depth structure tensor from §5. Despite this additional computation the iGRaND is still quite fast and, as shown in figures 4(b,c) the total time to run the algorithm is only 15% slower than the ORB detector. This difference in time is smaller than we expected and we attribute the unexpected time gain for the iGRaND algorithm to a reduced computational burden in the correspondence computation phase which, as noted in figure 3, requires comparison of 10% less descriptors.

Figure 5(a-d) show representative images from the experimental data. Detected iGRaND features are shown in green (left column) and ORB features for the same images are shown in red (right column). Note that when the scene structure is rich, the two detector return similar feature locations (a,b). Yet, as shown in (c,d) the iGRaND feature will prefer to select areas of geometric variation, e.g., the desk and chair on the left, in preference to features on planar regions, e.g., the checker board pattern on the right.

8. ACKNOWLEDGMENT

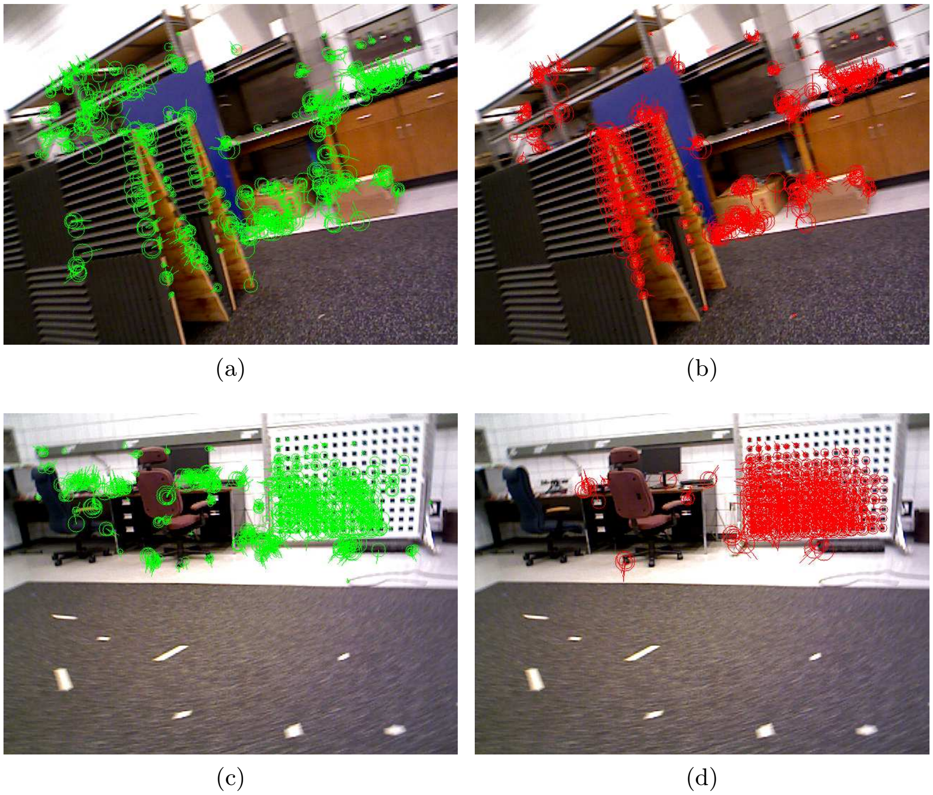
This research is sponsored by an AFRL/National Research Council fellowship and results are made possible by resources made available from AFRL's Autonomous Vehicles Laboratory at the University of Florida Research Engineering Education Facility (REEF) in Shalimar, FL. Inspiration for this work and the iGRaND acronym were driven by discussions with Grant D. Willis and Andrew W. Willis.

9. CONCLUSION

The key contribution of this work is to propose a extrinsic invariant coordinate frame, referred to as the iGRaND frame, and to detail the implementation of a feature detector and descriptor that uses this frame to define a hybrid "visual-and-geometric" feature which we demonstrate to have desirable properties for RGBD sensor based visual odometry and its downstream applications such as pose estimation, robotic navigation and mapping from RGBD image data. The novel aspects of this detect/descriptor pair is that it simultaneously leverages shape information provided by sensed depth images and appearance information provided by sensed intensity images. Feature choice and algorithm design is driven by our target application which seeks to process RGBD data in real time on a UAV platform, e.g., quadcopters. This application limits the size, weight and power consumption of the computational system that can be applied and, by extension, this places restrictions on the computational complexity and cost that our RGBD feature algorithm can have.

REFERENCES

1. A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 1524–1531, May 2014.



(a) (b)

(c) (d)

Figure 5: (a,b) and (c,d) compare feature detection in selected images from the recorded data. Our feature visualization shows feature locations (as circles) coded by scale (larger circle indicates larger scale) and orientiation (shown as a line emanating out of the circle). (a,c) show detections (in green) generated by the depth-sensitive iGRaND detector. (b,d) show detections (in red) generated by the ORB detector. Note that the iGRaND feature prefers designating feature locations at locations have rapid depth and intensity changes (note the features on the chair in (c) not present in ORB detections of (d)). Selection of features at geometrically rich locations tends to improve accuracy and reliability in estimated odometry.

2. R. Valenti, I. Dryanovski, C. Jaramillo, D. Perea Strom, and J. Xiao, “Autonomous quadrotor flight using onboard rgb-d visual odometry,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 5233–5238, May 2014.
3. C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for rgb-d cameras,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 2100–2106, Nov. 2013.
4. K. Khoshelham and S. O. Elberink, “Accuracy and resolution of kinect depth data for indoor mapping applications,” *Sensors* **12**(2), p. 1437, 2012.
5. I. Dryanovski, R. Valenti, and J. Xiao, “Fast visual odometry and mapping from rgb-d data,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2305–2310, May 2013.
6. E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, “Assessment and calibration of a rgb-d camera (kinect v2 sensor) towards a potential use for close-range 3d modeling,” *Remote Sensing* **7**(10), p. 13070, 2015.
7. T. Tuytelaars and K. Mikolajczyk, *Local Invariant Feature Detectors: A Survey*, Now Publishers Inc., Hanover, MA, USA, 2008.
8. R. Szeliski, *Computer Vision: Algorithms and Applications*, ch. Feature detection and matching, pp. 181–234. Springer London, London, 2011.
9. M. Nixon and A. S. Aguado, *Feature Extraction & Image Processing, Second Edition*, Academic Press, 2nd ed., 2008.
10. M. Pauly, R. Keiser, and M. Gross, “Multi-scale feature extraction on point-sampled surfaces,” *Computer Graphics Forum* **22**(3), pp. 281–289, 2003.
11. Y. Li and E. Olson, “Structure tensors for general purpose lidar feature extraction,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
12. C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, pp. 147–151, 1988.
13. I. Sipiran and B. Bustos, “A robust 3d interest points detector based on harris operator,” in *Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval, 3DOR ’10*, pp. 7–14, Eurographics Association, (Aire-la-Ville, Switzerland, Switzerland), 2010.
14. I. Sipiran and B. Bustos, “Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes,” *Vis. Comput.* **27**, pp. 963–976, Nov. 2011.
15. S. M. Smith and J. M. Brady, “Susan&mdash;a new approach to low level image processing,” *Int. J. Comput. Vision* **23**, pp. 45–78, May 1997.
16. H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang, “Integral invariants for robust geometry processing,” *Comput. Aided Geom. Des.* **26**, pp. 37–60, Jan. 2009.
17. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision* **60**, pp. 91–110, Nov. 2004.
18. C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, “Feature detection on 3d face surfaces for pose normalisation and recognition,” in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pp. 1–6, Sept 2010.
19. A. Flint, A. Dick, and A. v. d. Hengel, “Thrift: Local 3d structure recognition,” in *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, pp. 182–188, Dec 2007.
20. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.* **110**, pp. 346–359, June 2008.
21. Y. Zhong, “Intrinsic shape signatures: A shape descriptor for 3d object recognition,” in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 689–696, Sept 2009.
22. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01, CVPR ’05*, pp. 886–893, IEEE Computer Society, (Washington, DC, USA), 2005.
23. H. Skibbe, M. Reisert, and H. Burkhardt, *Pattern Recognition: 33rd DAGM Symposium, Frankfurt/Main, Germany, August 31 – September 2, 2011. Proceedings*, ch. SHOG - Spherical HOG Descriptors for Rotation Invariant 3D Object Detection, pp. 142–151. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

24. J. Li and H. Fan, "Curvature-direction measures for 3d feature detection," *Science China Information Sciences* **56**(9), pp. 1–9, 2013.
25. A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, pp. 433–449, May 1999.
26. R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, *ICRA'09*, pp. 1848–1853, IEEE Press, (Piscataway, NJ, USA), 2009.
27. S. Filipe and L. A. Alexandre, "A comparative evaluation of 3d keypoint detectors in a RGB-D object dataset," in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 1, Lisbon, Portugal, 5-8 January, 2014*, pp. 476–483, 2014.
28. A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *Int. J. Comput. Vision* **89**, pp. 348–361, Sept. 2010.
29. S. Salti, F. Tombari, and L. D. Stefano, "A performance evaluation of 3d keypoint detectors," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pp. 236–243, May 2011.
30. B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Narf: 3d range image features for object recognition," in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, (Taipei, Taiwan), October 8, 2010 2010.
31. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, *ECCV'10*, pp. 778–792, Springer-Verlag, (Berlin, Heidelberg), 2010.
32. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, *ECCV'06*, pp. 430–443, Springer-Verlag, (Berlin, Heidelberg), 2006.
33. P. L. Rosin, "Measuring corner properties," *Comput. Vis. Image Underst.* **73**, pp. 291–307, Feb. 1999.
34. H. Guggenheimer, *Differential Geometry*, McGraw-Hill series in higher mathematics, McGraw-Hill, 1963.
35. F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, pp. 789–805, Aug. 1992.
36. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the 2011 International Conference on Computer Vision*, *ICCV '11*, pp. 2564–2571, IEEE Computer Society, (Washington, DC, USA), 2011.
37. A. R. Willis and K. M. Brink, "Real-Time RGBD Odometry for Fused-State Navigation Systems," in *IEEE/ION Position Location and Navigation Symposium*, (Savannah, Georgia), April, 14-16 2013.
38. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.