Hybrid HFR Depth: Fusing Depth and Color Cameras for High Speed, Low Latency Depth Camera Interactions

Jiajun Lu[1,2], Hrvoje Benko[1], Andrew D. Wilson[1]

[1]Microsoft Research
Redmond, WA, USA
benko@microsoft.com,
awilson@microsoft.com

[2]University of Illinois
Urbana, IL, USA
jlu23@illinois.edu

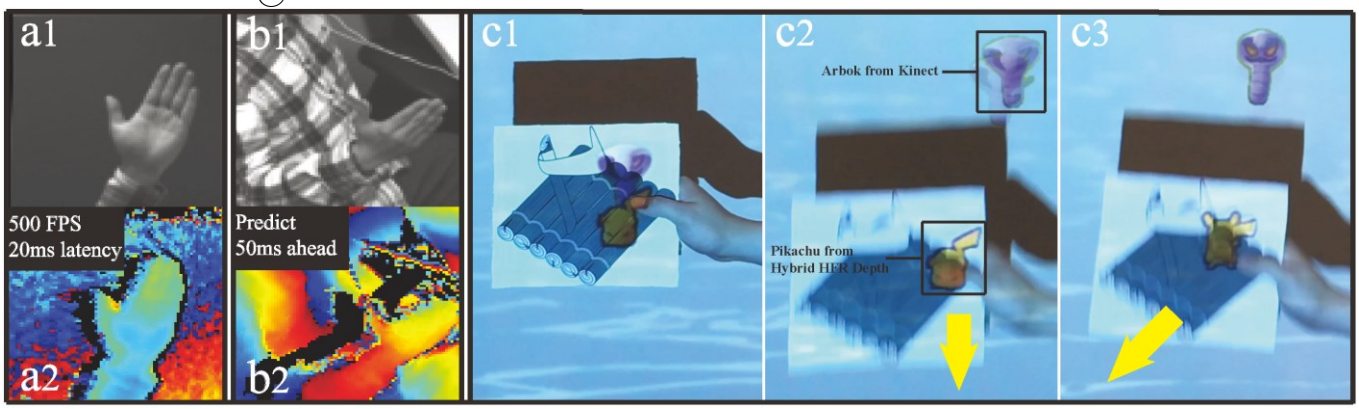

Figure 1. Hybrid HFR Depth is a high frame rate, low latency, configurable depth camera solution built from a Kinect and a color camera. Depth output is 500 frames per second (maximum) and 20 milliseconds latency in (a1-2), and depth is predicted 50ms into the future in (b1-2). Our high frame rate, low latency depth solution supports applications sensitive to lag and framerate: (c1-3) illustrates interactive projection mapping, where Arbok and Pikachu are projected onto the raft using either standard Kinect depth stream (Arbok) or Hybrid HFR Depth stream (Pikachu) to track the projection surface. (c1) Arbok and Pikachu are projected on the raft when hand is still. (c2) (c3) Arbok is off the raft and Pikachu is on the raft when hand is moving.

ABSTRACT

The low frame rate and high latency of consumer depth cameras limits their use in interactive applications. We propose combining the Kinect depth camera with an ordinary color camera to synthesize a high frame rate and low latency depth image. We exploit common CMOS camera region of interest (ROI) functionality to obtain a high frame rate image over a small ROI. Motion in the ROI is computed by a fast optical flow implementation. The resulting flow field is used to extrapolate Kinect depth images to achieve high frame rate and low latency depth, and optionally predict depth to further reduce latency. Our "Hybrid HFR Depth" prototype generates useful depth images at maximum 500Hz with minimum 20ms latency. We demonstrate Hybrid HFR Depth in tracking fast moving objects, handwriting in the air, and projecting onto moving hands. Based on commonly available cameras and image processing implementations, Hybrid HFR Depth may be useful to HCI practitioners seeking to create fast, fluid depth camera-based interactions.

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025478

Author Keywords

depth camera; Kinect; latency; frame rate; configurable

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): Input devices and strategies (e.g., mouse, touchscreen)

INTRODUCTION

Consumer depth cameras such as Microsoft Kinect have been useful in various HCI applications. But the frame rate (30Hz) and high latency (at 60-80 milliseconds) of such cameras can limit their use in interactive applications sensitive to lag and framerate. Low frame rates can complicate tracking, particularly for fast moving, flexible objects like hands. High latency during interaction can confuse or even nauseate users, particularly in AR and VR applications, or when overlaying graphics on physical objects that move unpredictably.

In the absence of commercially available high frame, low latency depth sensors, we explore solutions involving commodity depth sensors and commonly available color cameras. We demonstrate a *hybrid high frame rate depth* (Hybrid HFR Depth) camera solution, which registers a Kinect v2 to a Point Grey Grasshopper 3 camera, a commercially available configurable, high frame rate, high

resolution and low latency color camera. With the two cameras aligned, we calculate optical flow over a small region of interest (ROI) of the color camera. The frame rate and latency of CMOS cameras such as the Point Grey camera is related to the size of the ROI, with small ROIs obtaining high frame rate (500Hz) and low latency. We apply the resulting flow field to warp the depth samples in the image plane. Our experiments demonstrate that the errors in measurement resulting from our approach are small. Our approach also allows a small amount of prediction to further reduce latency.

We demonstrate three applications of Hybrid HFR Depth. Our applications are chosen to expose the benefits of high frame rate, and low or negative latency. We show Hybrid HFR Depth tracking a ping pong ball with dense, accurate samples of depth. We use Hybrid HFR Depth to track a fingertip, producing an accurate, dense trace useful for rendering and gesture recognition. Finally, we show that Hybrid HFR Depth is well-suited to interactive projection mapping applications. High latency in a projection mapping system can cause projected graphics to slide or shift on fast moving objects. Hybrid HFR Depth's high frame rate and latency reduction allows projection mapping onto a moving hand.

By utilizing inexpensive, commonly available cameras, and common computer vision image processing routines (OpenCV) we hope Hybrid HFR Depth enables HCI practitioners to create fast, fluid depth camera-based interactions.

RELATED WORK

There are a variety of previous works that seek to increase the frame rate and reduce the latency of depth cameras. They include a range of approaches including simple prediction using a dynamical model, modifications of existing depth image computation algorithms possibly involving exotic hardware, and highly specialized hardware.

Prediction by Dynamical Models

Dynamical models allow a degree of prediction in software, thereby reducing latency. Xia et al. [19] sought to find a camera and software-based approach to reduce the latency of touchscreen interaction. Based on collected training data, Xia et al. estimated touch down locations and triggered device interactions in advance. Knibbe et al. [8] used a Kalman Filter to predict the motion of a ball in flight. Kitani et al. [7] placed a camera inside a ball and used image processing to determine its speed of rotation, triggering the camera at precise moments to capture the scene below. At low frame rates, such dynamical models must be very accurate to make helpful predictions, limiting the utility of the approach.

Specialized Depth Processing

Other related works improve performance of depth cameras by exploiting low-level properties of how depth is computed. Such approaches tend to rely on exotic components, modifications of existing components, and intimate knowledge of how depth is computed on today's depth cameras. The focus of these works is on improving frame rate; improvements in latency are not discussed.

Schmidt et al. [15] presented a method to implicitly calibrate multi-tap 3D Time of Flight sensors, increasing the frame rate by a factor of two. Their prototype is a proof-of-concept written in Matlab.

Stuhmer et al. demonstrated that a modified Kinect v2 operating at 300Hz can track a fast moving ping pong ball accurately [17]. They modified the Kinect depth sensor to capture raw infrared images, and performed model-based tracking against these raw captures. The method requires models for both object and motion, and so works only for rigid simple shapes in simple motion. In contrast, our method obtains high frame rate and low latency depth map for a region of interest under general motion.

Fanello et al. [3] combined an exotic high-frame rate full-frame camera with a Kinect v1 structured light projector to obtain depth frames at 375Hz. They contribute a machine learning approach to optimize the Kinect v1 computation of disparity to keep up with the high frame rate camera.

Increasing Frame Rate by Specialized Hardware

Specialized hardware may be used to reduce latency and increase frame rate for a particular application. Papadakis et al. [14] minimized latency in head-tracking immersive simulations by reducing buffering latency in their display hardware, achieving a 50% reduction in overall system latency. Lumospheres [9] presented a hardware optimization approach to accurately project on balls under projectile motion. Okumura et al. [13] introduced a low latency camera with a series of saccade mirrors for ball tracking technology. Their system is purely vision based (no depth), so it can only track visually salient objects.

Customized hardware generates impressive results, but the expense of such approaches put them out of reach of typical HCI practitioners. In contrast, our method uses off-the-shelf, affordable hardware.

Hybrid Cameras

Lu et al. [11] demonstrated improving the spatial resolution of the depth stream by calibrate the Kinect with a color camera. They used an edge map and optimization-based interpolation and optical flow to aggressively upsample depth. Their work demonstrated that depth is not hard to interpolate accurately; that current optical flow estimates are accurate enough to support depth interpolations; and registering a depth sensor with color camera can produce an improved depth stream.

In contrast to previous work, our approach shows that color camera can empower depth camera in all aspects. Our approach combines off-the-shelf hardware by vision algorithms to reduce latency and increase frame rate. It is a compromise between exotic and expensive customized

hardware approaches and limited software-based approaches.

OVERVIEW AND CONTRIBUTIONS

Our Hybrid HFR Depth approach combines an off-the-shelf Xbox Kinect v2 and Point Grey Grasshopper 3 camera (see Figure 2(a)), and uses computer vision algorithms to create configurable high frame rate and low latency depth images. Our Grasshopper 3 GS3-U3-51S5M-C Mono supports configuration of region of interest, frame rate, resolution, shutter speed and so on. It features a Sony IMX250 CMOS, 2/3" imager, running at 75Hz at full 2448x2048 resolution, and at 500Hz with a 256×256 region of interest (ROI).

Table 1 compares Hybrid HFR Depth with the individual sensors used in our system. As the table shows, Hybrid HFR Depth inherits characteristics of both Point Grey camera and Kinect v2.

	Kinect v2	Point Grey	Hybrid HFR
Frame Rate	30Hz	75-500Hz	75-500Hz
Latency	60-93ms	10ms	22ms
Full frame	512×424	2448×2048	410×340
Min ROI	21×21	126×126	21×21
Max ROI	170×170	1020×1020	170×170

Table 1. Specifications of Hybrid HFR Depth and the two sensors used. Hybrid HFR Depth full frame dimensions are less than that of Kinect due to Kinect having a slightly wider field of view than Point Grey camera.

A main contribution of this paper is to use off-the-shelf cameras to create a practical, affordable and easily accessible high frame rate, low latency depth image solution. Compared to customized hardware [9, 13, 14] and software-only approaches [3, 7, 8, 15, 17, 19], it is hybrid in nature. We evaluate high frame rate depth qualitatively and quantitatively and demonstrate the usefulness of our Hybrid HFR Depth in three application scenarios.

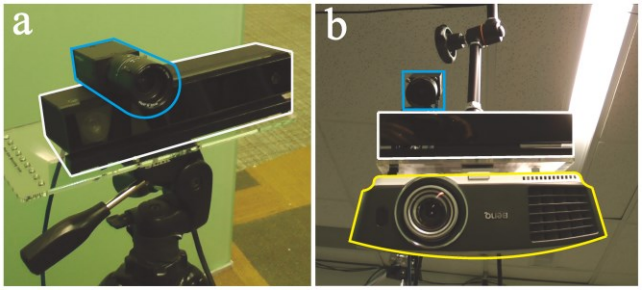

Figure 2. (a) Our Hybrid HFR Depth uses off-the-shelf Point Grey camera and Kinect v2. (b) Our projection mapping system adds an off-the-shelf projector.

METHOD

Hybrid HFR Depth combines Kinect v2 and Point Grey RGB camera by vision algorithms. We first briefly introduce the pipeline and then include details of each part. First, we rigidly mount the two sensors together and calibrate them (Figure 2a). The Point Grey camera has about 6 times higher pixel count in each dimension than Kinect depth frames, so we down sample the color frames and align them with the depth frames. Second, we estimate the delay of each sensor to accurately align the two data streams in time to perform our vision algorithms. Third, we use GPU implementation of the Brox optical flow [2] to estimate the flow from Point Grey color images. This flow field is used to warp the Kinect v2 depth data in XY direction. Because the depth samples not only move in XY direction, but also move in Z direction (as depth value changes), we use linear extrapolation to predict movement in the Z direction. We can further reduce the latency due to the Point Grey camera and image processing. Because we have reliable high frame rate depth stream, we find a simple acceleration model works well to reduce or eliminate latency or predict future depth frames.

Sensor Calibration

We find that a simple, fast calibration modeling only lens distortion and affine transformation works well. First, we use OpenCV's camera calibration routines to estimate the lens distortion of Kinect IR camera and Point Grey color camera. Then, we use RANSAC to estimate the affine transformation matching corresponding points in color images and IR images.

Alignment in Time

Kinect v2 and Point Grey cameras have very different latencies. To align the sensors in time to perform vision algorithms and understand the latency of the system, we measure the latency of each sensor. Point Grey camera latency is measured by recording the time between sending an image request and receiving an image, around 10ms (varies slightly according to the image size). Kinect's latency is more difficult to measure. Instead we measure its latency relative to the Point Grey camera. We swing a scissor in circles like a clock (scissor is the hour hand of clock), and continuously take images with both Point Grey camera and Kinect. For each Kinect image, we find the image from Point Grey camera that best matches the clock (scissor in same rotation angle). Because all images have time stamps, it's easy to find that the relative latency is 50ms. Since the Point Grey camera has absolute latency of 10ms, the actual latency of each Kinect frame is about 60 milliseconds. Kinect generates depth frames every 33 milliseconds (30Hz), so the actual latency is between 60 and 93ms. With sensor latencies and time stamps for both data streams, we can accurately align Kinect depth frame and Point Grey color frame. We align depth frame D_k at time k to color frame at time $t = k - 50ms$.

Our Hybrid HFR Depth obtains a minimum 20ms latency, which is the sum of Point Grey camera latency and image processing time. Our depth stream can be as fast as 500Hz, so the actual minimum latency, without further prediction, is 20+2 milliseconds.

Region of Interest (ROI)

A common CMOS camera can run very fast (high frame rate and low latency) by reducing the region of interest (ROI). We exploit this feature by calculating our fast depth stream on a moving ROI, focusing on the object of interest. This strategy greatly increases effective frame rate while reducing computation cost. We crop the Kinect depth image to the corresponding ROI in software, and configure the Point Grey ROI using the camera's API. Determining the ROI in an application is generally easy, because we need only low frame rate Kinect depth to determine the position of fast moving objects of interest. For example, if we want to focus on the right hand, we can use the Kinect depth to find the center of the right hand at 30Hz and update the Point Grey ROI continuously to track the center of the hand. If the full size depth stream is needed, we can copy the calculated ROI depth back to the full size Kinect depth image. This full frame obtains high framerate and low latency inside the ROI and usual Kinect performance elsewhere.

Generating High Frame Rate Depth Images

High frame rate depth images are computed by warping the latest Kinect depth image by dense optical flow of high frame rate Point Grey images. From color images I_t and I_{t+1} we obtain the flow field $F_{t,t+1}$ and define an image warping function $I_{t+1} \approx warp_{F_{t,t+1}}(I_t)$. A high frame rate depth image H_{t+1} is obtained by warping the latest Kinect depth image D_k by the flow field: $H_{t+1} = warp_{F_{t,t+1}}(D_k)$. Subsequent frames $H_{t+i}, (i > 1)$ are not computed by warping the previous frame H_{t+i-1} but by updating the previous flow field so that it models all motion since the latest depth image. For $i > 1$:

$$F_{t,t+i} = warp_{F_{t,t+i-1}}(F_{t,t+i-1}) + F_{t+i-1,t+i}$$

$$H_{t+i} = warp_{F_{t,t+i}}(D_k)$$

where $F_{t+i-1,t+i}$ is the output of optical flow computation on the two latest color frames.

We use OpenCV's GPU implementation of Brox's algorithm [2]. However, even with GPU acceleration, the flow calculation is slow (maximum 150Hz) compared to our Point Grey camera (maximum 500Hz). We find that the relation between flow calculation time and image size is not linear: the flow calculation time increases much more slowly than image size. We can use batch processing to compute flow over multiple pairs of images to speed up computation at a small cost in latency (refer to Figure 4, increasing batch size by one will increase minimum 2ms latency). For example, with a batch size of n, we concatenate n Point Grey images, run optical flow on this large image, and then split the result into n small flow images. This approach allows Hybrid HFR Depth to run as fast as the Point Grey camera.

The warping process described above operates over the spatial domain of the image but does not model changes in the depth values themselves (e.g.; an object moves closer or further away from the camera). To calculate accurate depth values for H we linearly extrapolate the Z component of Kinect depth images. We find the most recent two Kinect depth images D_{k-1} and D_k and their corresponding color images I_{t_1} and I_{t_2}. Assuming that change in depth is constant over a small period, we update the Z component of H_t as:

$$Z_t = warp_{F_{t_2,t}}(D_k + (D_k - warp_{F_{t_1,t_2}}(D_{k-1}))\frac{t - t_2}{t_2 - t_1})$$

This relationship is depicted in Figure 3.

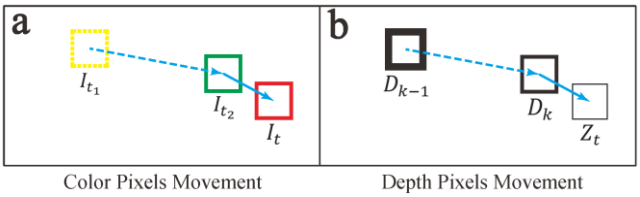

Figure 3. Generating high frame rate depth images. (a) Optical flow is used to estimate motion in the color images from I_{t_1} and I_{t_2}, which correspond to Kinect depth images D_{k-1} and D_k. High frame rate depth image H_t is generated by warping D_k by the flow I_{t_2} to I_t. (b) depth values Z are determined by linear extrapolation of the change in depth from D_{k-1} to D_k, following the motion in the color image.

Prediction

Because they are densely sampled in time, the high frame rate depth frames can be used to predict future frames using a second order model of motion. We estimate accurate velocity and acceleration with 30 samples, and assume the acceleration is constant during prediction. With the flow estimation and the acceleration model, we can predict future flows and future depth frames. We can use this approach to reduce latency, possibly to zero or even negative values. Aggressive prediction will naturally affect depth image accuracy, especially around discontinuities in the depth image. Applications that are less sensitive to inaccuracies around object boundaries in the depth image (e.g., object tracking) may benefit from significant prediction.

CONFIGURATION DETAILS

In this section, we consider various implementation details of the Hybrid HFR Depth approach. These can be configured to suit a given application.

Hybrid HFR Depth specification

There are four quantities that relate to the performance of the proposed technique: ROI, batch size, frame rate, and latency. *ROI* can be configured from 40 pixels in each dimension, enough to capture the whole hand, to 170 pixels in each dimension, enough to capture the whole upper body. As noted above, our implementation may compute optical flow over several images. *Batch size* indicates the number of images included in this computation. Changing batch size trades off latency and framerate. The relation between optical flow calculation time and batch size is not linear. For example, when the ROI is 50 pixels in each dimension and batch size is 1, the frame rate is around 100Hz. If batch size is 8, the frame rate is around 400Hz. In general, the *frame rate* varies from around 100Hz to 500Hz, depending on ROI

and batch size, and can be configured to target a given application. *Latency* in our system is the sum of Point Grey camera latency and processing latency (greater with larger batch sizes). When prediction is used, latency can be reduced to zero or even become negative. We illustrate the relationship between frame rate and latency without prediction under different ROIs and batch sizes in Figure 4.

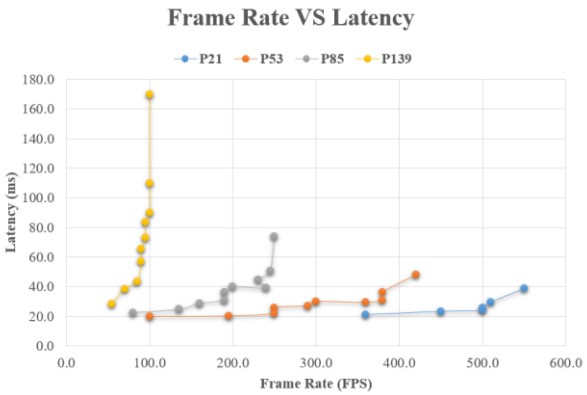

Figure 4. Batch processing optical flow calculations impacts frame rate and latency: when batch size increases, both frame rate and latency increase at a rate that depends on the size of the ROI. We illustrate the empirical relationship between frame rate and latency for four ROI sizes (P21, e.g., denotes an ROI 21 pixels in each direction). The first point along each curve indicates a batch size of one (no batching). Each point thereafter indicates a batch size of one more frame. In general, larger ROI size causes greater latency, higher frame rates correspond to greater latency (due to batch processing), and latency increases slowly with batch size, at first.

We give the configurations of three example applications we present later in Table 2. The basic principle is to tune the most critical aspect of configuration for a given application. In hand writing, for example, frame rate is the most important factor. In tracking, both higher frame rate and lower latency is desirable. In projection mapping, frame rate similar to projector refresh rate (60Hz) is adequate, while aggressive prediction is needed.

	ROI Size	Batch Size	Latency	Frame Rate
Writing	85×85	6	40ms	200
Tracking	75×75	4	30ms	200
Mapping	106×106	1	22 (-70)ms	70

Table 2. Hybrid HFR Depth configurations for the three example applications presented in this paper. With projection mapping, if no prediction is used, the latency is 22 milliseconds. If prediction is used, the latency is -70 milliseconds.

Other Configuration Options

Increased Field of View: Fixing the ROI while down sampling the input depth image sacrifices spatial resolution but significantly increases the effective field of view without impacting computation time. For example, resizing the depth image to 25% increases the field of view fourfold. This can be useful for applications that require larger field of view.

Increased Accuracy: Optical flow calculations are subpixel in nature, but its accuracy is related to the resolution of the input images. Higher resolution images yield more accurate optical flow. In order to increase accuracy, we can use color images with resolution four times that of the depth image to more accurately estimate optical flow, and then down sample the flow. This approach will generate higher accuracy high frame rate depth at the cost of lower frame rate and higher latency.

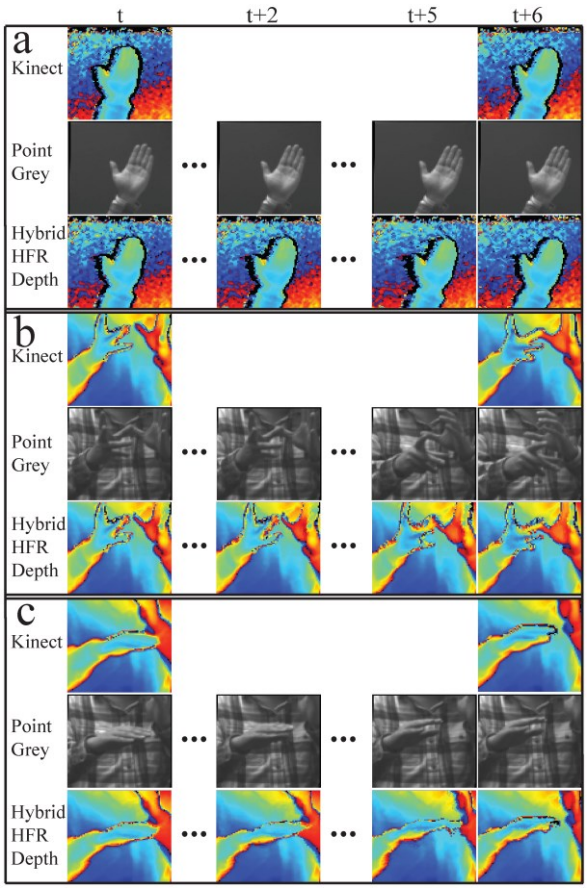

Figure 5. Example sequences of hand movement illustrate the quality of high frame rate depth images. Hybrid HFR Depth images have higher frame rate, lower latency (hands move ahead of Kinect depth) and quality similar to the input Kinect depth images.

EVALUATION

Figure 5 illustrates the quality of the high frame rate depth image on examples of hand motion. The accompanying video gives further examples and gives a better sense of the quality of the generated sequences. Depth direction hand motion example is not visually obvious, so we only include a quantitative example in Table 3.

Quantitative Evaluation

To evaluate the quality of the Hybrid HFR Depth images we would like to compare the generated 300Hz high frame rate

depth image to an equivalently 300Hz high frame rate ground truth depth image. Because we have no such 300Hz high frame rate ground truth available, we instead use the 30Hz Kinect depth image stream as ground truth, and take every tenth Kinect depth image as 3Hz input. This essentially slows down the sequence by a factor of 10. During usage, we speed up both depth streams 10 times to simulate 300Hz ground truth and 30Hz input to the Hybrid HFR Depth algorithm (see Figure 6). In recording test sequences, we may take some care in producing motions that are approximately slowed down by a factor of 10.

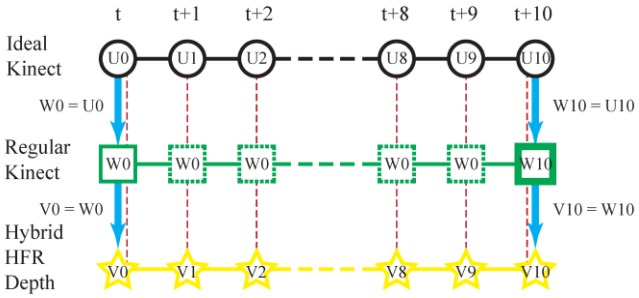

Figure 6. Illustration for quantitative evaluation. "Ideal Kinect" is simulated by capturing slow motion with Kinect (30Hz). "Regular Kinect" frames are meant to simulate the standard Kinect stream before the 10x slow simulation and are held for 10 frames. "Hybrid HFR Depth" takes as input every tenth Kinect frame. Symbols inside sample point (circle, rectangle and star) means the depth frames. Red dash line indicates comparisons reported in the text.

Dataset

We collected seven sequences of hand motions: three simple hand motions S1-3 (similar to first row of Figure 5), three more complex hand motions C4-6 (similar to second row of Figure 5) and one hand motion only in depth direction D7 (scene is similar to first row of Figure 5 and hand moves in depth). In each sequence, the subject moved their hands in approximately one tenth normal speed. We save only frames that have both Kinect depth and Point Grey images, so we get a sequence of normal hand moving speed, and each Point Grey color image has a corresponding depth images. When these sequences are processed, we feed every Point Grey image and one out of ten "Ideal Kinect" depth images into the system, to simulate the different frame rates of two sensors. During evaluation, "Ideal Kinect" depth images are used as ground truth.

Metrics

To compare the generated depth image against "Ideal Kinect" ground truth, we calculate average absolute per pixel difference across the image, ignoring pixels that have no value (black pixels in our figures).

Results

Pixel noise around discontinuities in depth can be large. For example, Figure 7(f) and (g) illustrate two successive Kinect frames when the subject is not moving. Figure 7(h) depicts the absolute value. In this example the average absolute per pixel difference is 15mm. Table 3 shows the error for the seven test sequences. In this table, "Regular Kinect" refers to simply using the "Regular Kinect" stream described earlier as the high frame rate depth image. This serves as a simple baseline measure of performance if one were to naively upsample the Kinect depth image. To understand the influence of boundary pixels in our evaluation metrics, we threshold the absolute per pixel error to maximum 500mm, and have errors of 46.9, 38.3, 39.2mm for D7 in Table 3.

As expected, the "Hybrid HFR" and "Hybrid HFR w/ prediction" errors are substantially reduced from that of naive upsampling. "Hybrid HFR w/ prediction" refers to the high frame rate depth image generation with prediction to reduce latency to zero. This results in a slight increase in error compared to the regular "Hybrid HFR" which is synchronized with the Point Grey camera (no prediction).

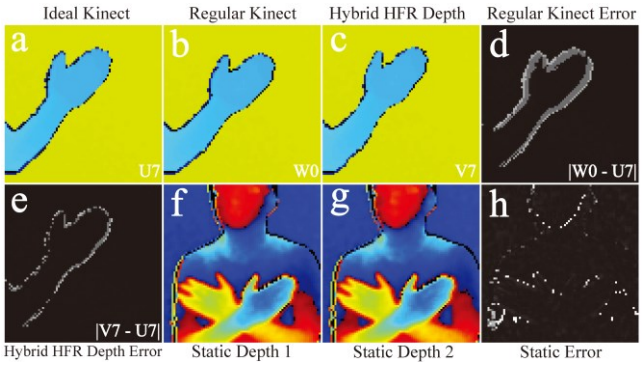

Figure 7. Depth errors are large around depth boundaries. (a-e) corresponds to Figure 5a: (a) is ground truth depth U7 of the example from "Ideal Kinect", (b) is the corresponding "Regular Kinect" depth W0, seven frames before the ground truth image, (c) is Hybrid HFR Depth image V7, (d) shows the per pixel error for "Regular Kinect", compared to ground truth, (e) is the same for Hybrid HFR Depth. We evaluate the Kinect sensor noise in (f-h). (f) is a Kinect depth frame at time t, (g) is a Kinect depth frame at time t+1, (h) is the absolute difference between the two depth frames.

Sequence	"Regular Kinect"	Hybrid HFR	Hybrid HFR w/ prediction
S1	42.1mm	28.9mm	29.1mm
S2	65.8	35.2	37.4
S3	72.1	43.4	44.2
C4	65.3	45.9	47.2
C5	47.5	40.1	40.2
C6	65.2	50.6	52.0
D7	99.7	78.6	79.4

Table 3. Average per pixel error in millimeters between calculated depth image and ground truth depth image. S1-S3 are simple sequences, C4-C6 are more complex sequences, D7 includes motion in depth. "Regular Kinect" is described in Figure 6. "Hybrid HFR" refers to our high frame rate depth image with no prediction (minimum latency 20ms), while "Hybrid HFR w/ prediction" includes prediction to reduce latency to zero.

APPLICATIONS

Hybrid HFR Depth benefits a variety of applications that work better with high frame rate and/or low latency sensing. We show three important and representative tasks to demonstrate the usefulness of our system. First, we demonstrate that our low latency and high frame rate depth stream can be used to track fast moving objects. Then, we demonstrate that the high frame rate depth stream can be used in small gesture control and hand writing interfaces. Finally, we demonstrate using prediction to reduce latency in interactive projection mapping.

Object Tracking

Tracking moving objects is a fundamental task in computer vision. Kinect can be very useful in object tracking, particularly when the object moves against a known background depth. However, the low frame rate of Kinect can make tracking fast moving objects accurately more difficult. Our Hybrid HFR Depth can provide robust high frame rate and no latency object tracking, even for fast moving objects.

To demonstrate Hybrid HFR Depth in an object tracking, we give a simple but informative example of tracking a bouncing ping pong ball, which moves relatively quickly and changes direction quickly when it bounces. Tracking the ping pong ball throughout the bouncing motion is relatively easy with the high frame rate and low latency depth map. We record the ping pong ball bouncing events with Kinect, Hybrid HFR Depth and Point Grey cameras. From Figure 8, we find that Hybrid HFR Depth produces high quality depth images with high frame rate and low latency.

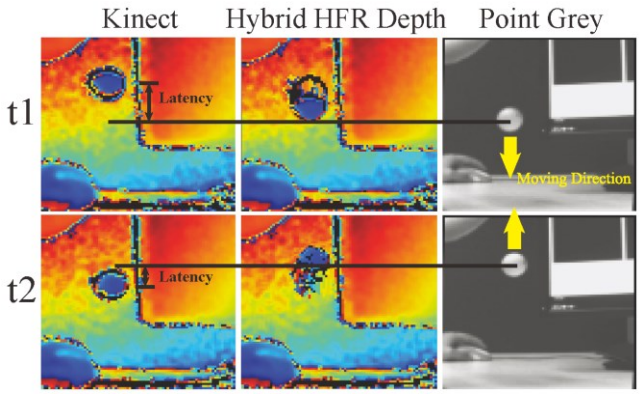

Figure 8. A ping pong ball is tracked with both Kinect v2 and Hybrid HFR Depth. The Hybrid HFR Depth image exhibits lower latency than Kinect. The high frame rate depth image is far ahead of the Kinect image but is slightly behind the Point Grey image (due to computation time).

Small Gesture Input

The Kinect sensor has been popularly applied to gesture recognition and control interfaces. Many existing systems either require the users to perform large scale hand gestures or require the users to be near to the controller. Our system enables the capture of small gestures imaged from a long distance (1.5m to 4m). By accurately detecting fast fingertip gesturing and writing, we can give commands to the remote computer or write on the screen.

High frame rate is particularly important for hand tracking systems, because fingers are flexible and fast, making them especially hard to track. The best open source hand tracker (by Oikonomidis et al.; [12]) still suffers from periods of lost tracking due to high computation cost and low frame rate. Sharp et al. introduced a new pipeline that estimates hand pose per frame [16]. However, tracking is performed at low frame rate.

We can find the fingertip in each depth frame to obtain a trace of the fingertip. It often suffices to find the closest point in the region of the hand. A high quality trace can be difficult to obtain when the frame rate is low (i.e., the fingertip moves a lot between two frames) and the depth noise is high (i.e.; closest point is not always the fingertip). To combat noise, we can smooth the fingertip trace using a Kalman filter. However, such smoothing alters the shape of the trace in undesirable ways. Figure 9 shows a number of gestures as recorded by Hybrid HFR depth, and the regular Kinect image with and without smooth. The traces show that users can reliably draw simple gestures with small finger motions in the air at a distance of 1.5m to 4m. With a gesture recognizer such as the "$1 gesture recognizer" [18], traces can be converted to corresponding commands for a remote interface. This could allow for a user to control their Xbox with small finger movement while lying on the sofa, for example.

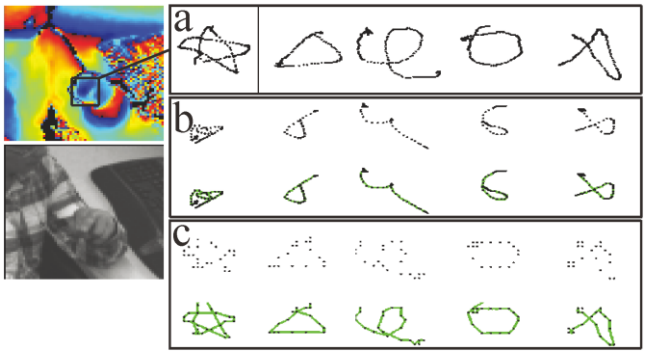

Figure 9. The user may draw simple shapes in the air to control a remote interface. (a) shows fingertip trace captured by Hybrid HFR Depth, with Kalman Filter applied. (b) is the same motion captured by Kinect, with Kalman Filter applied. In this case the Kalman filter alters the shape of the input gesture because of aliasing. (c) is the trace of Kinect skeleton hand tip without a Kalman Filter. Clearly high frame rate and subpixel accuracy improves writing in the air.

Besides drawing simple shapes, users may write words in about 2 meters away. This is a challenging task because only high frame rate and accurate depth stream can preserve all the sharp curves, small circles and other details during fast finger movement. Figure 10 demonstrates writing a phrase, using the same three techniques as above. The algorithm for Hybrid HFR Depth and Kinect depth hand writing are identical. When the finger moves slowly, the differences

between these methods are small. However, when the finger moves quickly we find that Hybrid HFR Depth tracks the fingertip more reliably, and the resulting trace looks better.

Figure 10. We demonstrate the advantages of hand writing with Hybrid HFR Depth. (a)(b)(c) are generated from the same motion sequence. The task is challenging because writing words involves many sharp turns and curves, requiring high frame rate and high accuracy. (b) illustrates using the regular Kinect depth stream (closest point), and (c) illustrates the trace of the Kinect skeleton hand tip. (a) illustrates Hybrid HFR Depth trace, which is of noticeably higher quality due to greater sampling frequency.

Interactive Projection Mapping

Projector-camera systems afford various interaction possibilities, combining both natural and mixed-reality 3D interaction. Jones et al. [4] introduced a system to augment the area surrounding a television with projected visualizations to enhance traditional gaming experiences. RoomAlive [5] transformed a room into an immersive, augmented entertainment experience through the usage of video projectors. Later, Benko et al. [1] proposed a spatial augmented reality system that uses dynamic projection mapping to support interaction with 3D virtual objects. However, in current projection mapping systems, accumulated latency from the depth sensor, image processing, graphics rendering and projection introduces errors in alignment in dynamic scenes. Projected graphics can seem to slip from its expected position, adversely impacting on the immersive experience. Recently, Knibbe et al. [8] and Koike et al [9] used software-based prediction to reduce the system latency. However, they only work with rigid objects in projectile motion. Our Hybrid HFR Depth can be used to track arbitrary motions and objects.

A custom hardware approach such as that of Koike et al [9] can reduce latency, typically with some expense and complexity. Using fast touch sensors rather than cameras, Jota et al. [6] describe a custom hardware-based projection-based drawing system that demonstrates that users are sensitive to surprisingly small amounts of latency (approximately 1ms) in direct manipulation settings such as that of interactive projection mapping.

To make projection mapping work reliably for moving objects with non-customized hardware, we can use our high frame rate and no latency Hybrid HFR Depth stream. While systems such as [8] must predict up to 110ms + 33ms to eliminate constant latency and latency due to camera frame rate, our system can reduce the time to 70 + 4ms (10ms Point Grey camera latency, 10ms optical flow calculation latency and 50ms projection mapping system latency), making prediction easier. Moreover, the higher frame rate can result in higher quality prediction to further reduce latency.

Calibration

We use RoomAlive toolkit [5] to calibrate our projector-camera setup (see Figure 2(b)). By applying extrinsic and intrinsic camera parameters, a point in depth image space can be transformed to projector space. A low or zero latency depth image can be used to render virtual objects into a similarly low-latency projection.

Combating Latency

To obtain the best interactive experience with our projection mapping system, we aim to completely eliminate latency, so that graphics projected onto moving objects appear to stay on the object. Operating system, rendering and projector latencies combine to an approximate total of 50ms. Given the 20ms minimum latency of the Hybrid HFR Depth approach, we must apply a prediction of approximately 70ms.

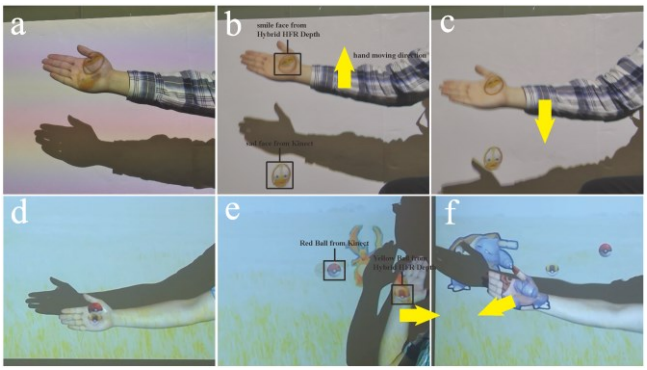

Figure 11. We compare dynamic projection mapping results with Kinect v2 and our Hybrid HFR Depth. We use Kinect v2 depth to project a sad face or a red ball onto the hand, and use Hybrid HFR Depth to project a happy face or a yellow ball onto the hand. In (a) and (d), hand is still, in others, hand is moving in the direction of the yellow arrow. In (f), hand is moving very fast, and yellow ball is off the hand, but still ahead of the red ball. In general, the Hybrid HFR Depth based methods stick graphics to moving hand better.

Figure 11 shows projection mapping results with Hybrid HFR Depth and Kinect v2. To numerically evaluate how our system works, we count the number of frames in which the projected graphics is fully on the hand, fully off the hand, or partially on the hand, as in [8]. We count these frames manually by analyzing videos of the system in operation, and compare the performance of the standard Kinect depth

stream with Hybrid HFR Depth. Table 4 shows these counts as a percentage of the test sequences for three speeds of motion.

	Kinect			Hybrid HFR Depth		
	All on	Partially off	Fully off	All on	Partially off	Fully off
Slow	18.6%	23.2%	58.2%	71.1%	28%	0.9%
Medium	13.7%	20.4%	65.9%	63.3%	35.6%	1.1%
Fast	9.5%	16.7%	73.8%	44.2%	44.8%	11%

Table 4. Percentage of test frames for which projected graphics is fully on, partially off, or fully off the user's hand. We manually count the percentage from three 900 frame sequences of slow, medium and fast motions.

LIMITATIONS

Our Hybrid HFR Depth system has a number of limitations. First, the system requires good, stable lighting. Dim or very strong lighting, or rapidly changing lighting will affect the quality of the optical flow-based motion estimation and therefore impact Hybrid HFR Depth quality. The high frame rate of the Point Grey camera limits the effective exposure time and light gathering capability of the camera; for example, 500Hz capture leaves only 2ms to collect light. In our projection mapping example, rapidly changing graphics might significantly affect the result quality. In this case it may be possible to use infrared imaging, though some care must be taken to avoid the narrow-band illumination of the Kinect's time-of-flight imager.

Second, optical flow has other limitations that can impact Hybrid HFR Depth quality. For example, optical flow can fail when there is a lack of texture, repeating textures, or strong reflections in the scene.

Third, OpenCV's optical flow implementationis still computationally expensive on today's hardware. Our current Hybrid HFR Depth prototype optionally employs batch processing of optical flow, potentially creating an undesirable tradeoff between latency, frame rate and ROI size (see Figure 4). Recently, Kroeger et al. [10] demonstrated a fast optical flow algorithm, which runs at 300-600Hz on a single CPU core with high resolution images. This may remove the need for batch processing, help obtain greater than1000Hz frame rate through multi-core scheduling, decrease depth latency and enlarge ROI.

Fourth, the specification of the Point Grey camera used in our prototype is another limitation. With Hybrid HFR Depth, we can reach the maximum frame rate of the Point Grey camera in most ROI size by manipulating batch size. A better color camera might allow a higher frame rate.

Finally, latency-sensitive applications such as projection mapping may require as much as 70ms of prediction. Our prediction model uses a simple acceleration model and is optionally further smoothed by a Kalman filter. Prediction is typically adequate because our inputs have higher frame rate and less time needs to be predicted. However, during sharp turns and accelerations, predictions are likely to be inaccurate. We still observe that our predictions will return to correct values faster than Kinect depth. More sophisticated data-driven approach to prediction is likely to improve the results.

FUTURE WORK

Hybrid HFR Depth could be improved in a number of ways to make it more powerful and useful.

One interesting direction of future work is to further exploit the complementary nature of the two cameras by creating real time super resolution depth images. This would help in applications where Kinect struggles to deliver adequate resolution. For example, when at a distance of 4m, an adult hand may be 15 pixels wide, making it difficult to resolve individual fingers. Meanwhile, our prototype's color image has about six times more pixels in each dimension. Combining depth information from Kinect with the color information from Point Grey camera, we can potentially generate a depth map six times that of Kinect.

While exploiting faster and more exotic cameras may enable higher frame rates, lower latency and a larger ROI, we also see the value in limiting our implementation to cameras that use inexpensive, commodity imagers, such as common web cameras. Employing an inexpensive common camera might broaden the appeal of the Hybrid HFR Depth approach among practitioners, much as the original Kinect did for interactive applications. CMOS imagers continue to rapidly advance in technology, with many smartphones able to record 240Hz videos.

Regarding applications, Hybrid HFR Depth might be particularly useful in Augmented Reality and Virtual Reality systems, where higher frame rate and lower latency are important factors in rendering stable, accurately aligned graphical augmentation, and in improving user comfort.

CONCLUSION

This paper proposes an approach to combine off-the-shelf hardware to create a high frame rate, low latency configurable depth stream. Hybrid HFR Depth is more powerful than a purely software-based approach, and less demanding than a customized hardware approach. We presented detailed specifications of Hybrid HFR Depth to demonstrate the capability and flexibility of the system needed to address a variety of applications. Our evaluation of Hybrid HFR Depth shows that the quality of the generated depth image is good enough to benefit three demonstrated interactive applications.

REFERENCES

1. Hrvoje Benko, Andrew D. Wilson, and Federico Zannier. 2014. Dyadic projected spatial augmented reality. In Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 645-655. ACM, 2014. DOI: http://doi.org/10.1145/2642918.2647402

2. Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. 2004. High accuracy optical flow estimation based on a theory for warping. In European conference on computer vision, pp. 25-36. DOI: http://doi.org/10.1007/978-3-540-24673-2_3
3. Sean Ryan Fanello, Christoph Rhemann, Vladimir Tankovich, A. Kowdle, S. Orts Escolano, D. Kim, and S. Izadi. 2016. Hyperdepth: Learning depth from structured light without matching. In CVPR, vol. 2, p. 7. 2016.
4. Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2013. IllumiRoom: peripheral projected illusions for interactive experiences. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 869-878. DOI: http://doi.org/10.1145/2470654.2466112
5. Brett R. Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 637-644. DOI: http://doi.org/10.1145/2642918.2647383
6. Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 2291-2300. DOI: http://dx.doi.org/10.1145/2470654.2481317
7. Kris Kitani, Kodai Horita, and Hideki Koike. 2012. BallCam!: dynamic view synthesis from spinning cameras. In Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology, pp. 87-88. DOI: http://doi.org/10.1145/2380296.2380335
8. Jarrod Knibbe, Hrvoje Benko, and Andrew D. Wilson. 2015. Juggling the Effects of Latency: Motion Prediction Approaches to Reducing Latency in Dynamic Projector-Camera Systems. Microsoft Research Technical Report MSR-TR-2015--35. DOI: http://dx.doi.org/10.1145/2815585.2815735
9. Hideki Koike, and Hiroaki Yamaguchi. 2015. LumoSpheres: real-time tracking of flying objects and image projection for a volumetric display. In Proceedings of the 6th Augmented Human International Conference, pp. 93-96. DOI: http://doi.org/10.1145/2735711.2735824
10. Till Kroeger, Radu Timofte, Dengxin Dai and Luc Van Gool. 2016. Fast Optical Flow using Dense Inverse Search. In Proceedings of the European Conference on Computer Vision (ECCV). DOI: http://doi.org/10.1007/978-3-319-46493-0_29
11. Jiajun Lu, and David Forsyth. 2015. Sparse Depth Super Resolution. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2245-2253. DOI: http://doi.org/10.1109/CVPR.2015.7298837
12. Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In BMVC, vol. 1, no. 2, p. 3. DOI: http://dx.doi.org/10.5244/C.25.101
13. Kohei Okumura, Hiromasa Oku, and Masatoshi Ishikawa. 2011. High-speed gaze controller for millisecond-order pan/tilt camera. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 6186-6191. DOI: http://doi.org/10.1109/ICRA.2011.5980080
14. Giorgos Papadakis, Katerina Mania, and Eftichios Koutroulis. 2011. A system to measure, control and minimize end-to-end head tracking latency in immersive simulations. In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 581-584. DOI: http://doi.org/10.1145/2087756.2087869
15. Mirko Schmidt, Klaus Zimmermann, and Bernd Jähne. 2011. High frame rate for 3D Time-of-Flight cameras by dynamic sensor calibration. In ICCP 2011, pp. 1-8. DOI: http://doi.org/10.1109/ICCPHOT.2011.5753121
16. Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann et al. 2015. Accurate, robust, and flexible real-time hand tracking. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3633-3642. DOI: http://doi.org/10.1145/2702123.2702179
17. Jan Stuhmer, Sebastian Nowozin, Andrew Fitzgibbon, Richard Szeliski, Travis Perry, Sunil Acharya, Daniel Cremers, and Jamie Shotton. 2015. Model-Based Tracking at 300Hz using Raw Time-of-Flight Observations. In ICCV, pp. 3577-3585. DOI: http://doi.org/10.1109/ICCV.2015.408
18. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proceedings of the 20th annual ACM symposium on User interface software and technology, pp. 159-168. DOI: http://doi.org/10.1145/1294211.1294238
19. Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency tapping: using hover information to predict touch locations and eliminate touchdown latency. In Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 205-214. DOI: http://doi.org/10.1145/2642918.2647348