Learning to be a Depth Camera
for Close-Range Human Capture and Interaction

Sean Ryan Fanello[1,2] Cem Keskin[1] Shahram Izadi[1] Pushmeet Kohli[1] David Kim[1] David Sweeney[1]
Antonio Criminisi[1] Jamie Shotton[1] Sing Bing Kang[1] Tim Paek[1]

[1]Microsoft Research [2]iCub Facility - Istituto Italiano di Tecnologia

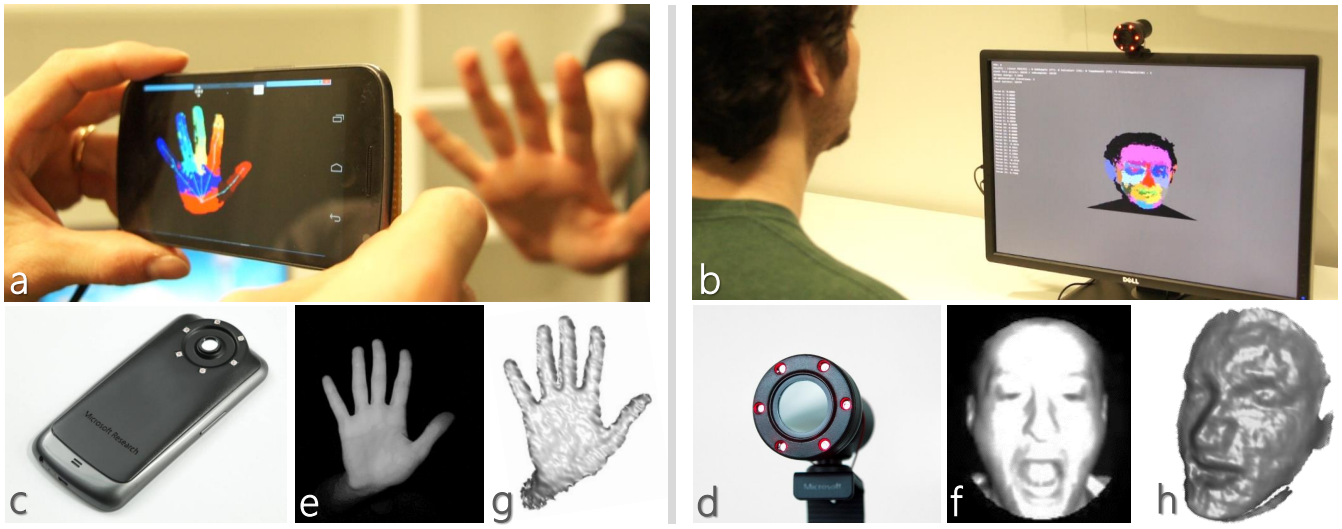

Figure 1: **(a, b)** *Our approach turns any 2D camera into a cheap depth sensor for close-range human capture and 3D interaction scenarios.* **(c, d)** *Simple hardware modifications allow active illuminated near infrared images to be captured from the camera.* **(e, f)** *This is used as input into our machine learning algorithm for depth estimation.* **(g, h)** *Our algorithm outputs dense metric depth maps of hands or faces in real-time.*

Abstract

We present a machine learning technique for estimating absolute, per-pixel depth using any conventional monocular 2D camera, with minor hardware modifications. Our approach targets close-range human capture and interaction where dense 3D estimation of hands and faces is desired. We use hybrid classification-regression forests to *learn* how to map from near infrared intensity images to *absolute*, metric depth in real-time. We demonstrate a variety of human-computer interaction and capture scenarios. Experiments show an accuracy that outperforms a conventional light fall-off baseline, and is comparable to high-quality consumer depth cameras, but with a dramatically reduced cost, power consumption, and form-factor.

CR Categories: I.3.7 [Computer Graphics]: Digitization and Image Capture—Applications I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Range Data

Keywords: learning, depth camera, acquisition, interaction

Links: DL PDF

ACM Reference Format
Fanello, S., Keskin, C., Izadi, S., Kohli, P., Kim, D., Sweeney, D., Criminisi, A., Shotton, J., Kang, S., Paek, T. 2014. Learning to be a Depth Camera for Close-Range Human Capture and Interaction. ACM Trans. Graph. 33, 4, Article 86 (July 2014), 11 pages. DOI = 10.1145/2601097.2601223 http://doi.acm.org/10.1145/2601097.2601223.

0730-0301/14/07-ART86 $15.00.
DOI: http://dx.doi.org/10.1145/2601097.2601223

1 Introduction

While range sensing technologies have existed for a long time, consumer depth cameras such as the Microsoft Kinect have begun to make real-time depth acquisition a commodity. This in turn has opened-up many exciting new applications for gaming, 3D scanning and fabrication, natural user interfaces, augmented reality, and robotics. One important domain where depth cameras have had clear impact is in human-computer interaction. In particular, the ability to reason about the 3D geometry of the scene makes the sensing of whole bodies, hands, and faces more tractable than with regular cameras, allowing these modalities to be leveraged for high degree-of-freedom (DoF) input.

Whilst depth cameras are becoming more of a commodity, they have yet to (and arguably will never) surpass the ubiquity of regular 2D cameras, which are now used in the majority of our mobile devices and desktop computers. More widespread adoption of depth cameras is limited by considerations including power, cost, and form-factor. Sensor miniaturization is therefore a key recent focus, as demonstrated by Intel[1], Primesense[2], PMD[3] and Pelican Imaging[4], and exemplified by Google's Project Tango[5]. However, the need for custom sensors, high-power illumination, complex electronics, and other physical constraints (e.g. a baseline between the illumination and sensor) will often limit scenarios of use, particularly when compared to regular cameras. Even if these issues are to be addressed, there remains many legacy devices which only contain 2D cameras.

[1] http://us.creative.com/p/web-cameras/creative-senz3d
[2] http://www.primesense.com/solutions/3d-sensor/
[3] https://www.cayim.com/
[4] http://www.pelicanimaging.com/
[5] http://www.google.com/atap/projecttango/

In this paper, we describe a very cost-effective depth sensing system which, for specific acquisition and interaction scenarios, can turn a regular 2D monocular camera into a depth sensor. Specifically, we devise an algorithm that learns the correlation between pixel intensities and absolute depth measurements. The algorithm is implemented on conventional color or monochrome cameras. The only hardware modifications required are: i) the removal of any near infrared (NIR) cut filter (typically used in regular RGB sensors), and ii) the addition of an bandpass filter and low-cost/power LEDs (both operating in a narrow NIR range).

There has been much progress in the fields of lighting-based geometry estimation and shape-from-shading ([Horn 1975; Zhang et al. 1999]). However, the problem is fundamentally ill-posed due to the unknown varying surface geometry and reflectance, and traditional approaches often resort to *explicit* assumptions, such as careful camera and illuminant calibration, known surface reflectance, or geometry (e.g., [Vogel et al. 2009]). Even so, achieving precise dense depth measurements remains challenging.

We instead take a data-driven *machine learning* approach, for the specific scenario of capturing the geometry of hands and faces. We propose to use hybrid classification-regression forests to learn a direct mapping from NIR intensity images to absolute, metric depth, for these specific scenarios. The forest automatically learns to associate a depth value with a pixel, based on its intensity in the NIR range, *and* the intensities of its neighboring pixels. The use of a learned model of spatial context is key to produce naturally smooth depth images which preserve transitions at occlusion boundaries.

We train our system using either synthetically rendered intensity images and associated ground truth depth maps; or a calibrated physical rig, where depth maps (captured using a high quality depth camera) are registered with intensity images acquired from our modified camera. The data (and thus the learned model) *implicitly* encodes information about the subject's surface geometry and reflectance, the camera intrinsics, vignetting effects, and the active and ambient illuminants. Our forest models are learned using a small set of subjects and camera devices. However, even with limited training data captured at low-effort, we demonstrate that our models are able to generalize well, even across subjects and devices.

Our use of decision forests yields an efficient algorithm which runs in real-time on portable devices. We validate our algorithm for human-computer interaction applications, and compare quantitatively with both existing high-quality consumer depth cameras and standard light fall-off techniques. Our algorithm enables us to turn practically any camera into a real-time depth camera for close-range human capture and interaction scenarios, without high power consumption, bulk, and expense. Whilst not a general purpose depth camera, it has the potential to enable a wide variety of new applications for mobile and desktop 3D interaction and acquisition.

In summary, our paper makes the following contributions:

- We demonstrate a new technique for turning a cheap color or monochrome camera into a depth sensor, for close-range human capture and interaction. Our hope is to allow practitioners to more rapidly prototype depth-based applications in a variety of new contexts.
- We present two practical hardware designs for depth sensing: (i) a modified web camera for desktop depth sensing, and (ii) a modified cellphone camera for mobile applications. We demonstrate efficient and accurate hand and face tracking in both scenarios.
- We propose specializations of existing *multi-layered* decision forests algorithms for the task of depth prediction which can achieve 100Hz performance on commodity hardware.
- We present experimental and real-world results that illustrate depth estimation accuracies for our specific scenarios that are comparable to state of the art consumer depth cameras.

2 Related Work

Many different depth sensing approaches have been proposed over the last few decades. Here we briefly cover relevant techniques; see [Besl 1988; Batlle et al. 1998; Zhang et al. 1999; Scharstein and Szeliski 2002; Blais 2004; Lanman and Taubin 2009] for detailed reviews.

Depth from Passive Stereo: Given images captured from two or more displaced RGB cameras, stereo methods identify points that are projections of the same 3D scene point. The point depth is related to the displacements of its image projections. Many algorithms have been proposed, including real-time methods [Scharstein and Szeliski 2002; Brown et al. 2003]. The biggest limitation of such approaches is that textureless regions yield inherent depth ambiguities. This results in incorrect depth estimates, or the need for expensive regularization or post-processing.

Structured Light and Active Stereo: The problem of textureless regions can be mitigated with the use of structured illumination (see e.g. [Besl 1988; Batlle et al. 1998; Blais 2004; Zhang 2010]). Many example of coded patterns exists, ranging from dynamic temporal sequences to single fixed patterns. Systems either employ two cameras plus illumination source, or use a single camera and calibrated projector to perform 3D triangulation. Even in these systems problems remain when estimating depths at object boundaries, where large depth discontinuities lead to outliers, holes, and edge fattening [Scharstein and Szeliski 2002; Brown et al. 2003]. Additionally all setups require a distance between projector and camera, and require a high-quality and costly/complex illumination source.

The Primesense (Kinect) camera projects a pseudo-random dot pattern and uses a displaced custom NIR sensor to triangulate depth. It demonstrates reasonably small form-factor and reduced cost. However, a high quality single-mode laser, diffractive optical element (DOE), high-resolution (1280x960) NIR camera, thermoelectrical cooling, and baseline between emitter and sensor are required. Our method uses any cheap modified 2D camera and simple LED-based illumination, without the need for a baseline. This allows legacy devices to be turned into depth cameras, but only for specific human-interaction scenarios.

Time of Flight: Many other depth sensing techniques exist beyond triangulation-based methods. Time-of-flight (ToF) cameras either modulate NIR lasers/LEDs and look at the shift in phase of the return signal (often referred to as *continuous-mode* devices), or use high frequency (physical or electronic) shutters in front of the image sensor to gate the returning pulses of light according to its time of arrival (often referred to as *shuttered* or *gated* devices). Almost all devices, including the recent Xbox One sensor[6] work on the continuous-mode principle. Shuttered devices are rarer, but include the legacy ZCam from 3DV[7]. Whilst the principle of ToF is used in precise measurement for expensive laser range finders (which give a single range measurement at a time), full frame ToF cameras typically suffer from high noise, including depth jitter and mixed pixels [Remondino and Stoppa 2013], and require costly custom sensors and electronics.

Geometry Estimation from Intensity Images: In terms of hardware, a cheaper method to acquire the 3D shape of an object is to use shape-from-shading (SFS) where the naturally occurring intensity

[6] http://www.xbox.com/en-GB/xbox-one/innovation
[7] http://en.wikipedia.org/wiki/ZCam

patterns across an image are used to extract the 3D geometry from a single image [Horn 1975; Zhang et al. 1999]. The mathematics of SFS is well-understood particularly when surface reflectance and light source position is known. [Prados and Faugeras 2005] reconstruct various objects including faces, using a light source at the center of the camera. [Ahmed and Farag 2007] demonstrate geometry estimation for non-Lambertian surfaces and varying illumination conditions. [Visentini-Scarzanella et al. 2012] exploit an off-axis light source, specularities and SFS for metric reconstruction.

Whilst the physics of SFS is well known, the problem is inherently ill-posed, and achieving compelling results requires strong scene and lighting assumptions and computationally complex algorithms. As such, real-time performance has rarely been demonstrated. This has led to work on *photometric stereo* where *multiple* images of a scene are captured under different controlled illumination to compute geometry. Photometric stereo has demonstrated extremely compelling results including reconstruction of surfaces with complex reflectance properties [Mulligan and Brolly 2004; Hernández et al. 2008; Ghosh et al. 2011; Tunwattanapong et al. 2013], as well as mobile uses[8]. However, these approaches can require complex lighting setups, a large baseline between light sources and camera, and/or sequential capture with changing illumination direction (although colored lights can allow for single frame multi-channel capture [Hernández et al. 2008]). These constraints make real-time dynamic scene capture on self-contained mobile devices challenging.

Related systems exploit the *inverse square law* to estimate depth from light fall-off. [Liao et al. 2007; Gurbuz 2009; Liu et al. 2011] capture images of the scene, with a fixed camera and light sources at varying distances, and measure depth from intensity differences. The Cyclops camera from Dinast[9] and the Digits system [Kim et al. 2012] uses a simpler light fall-off approximation for estimating coarse, *relative* depth per frame. Light fall-off measurements are influenced by surface albedo and geometry, ambient light and object inter-reflections, all of which lead to low-quality depth estimation. Often careful calibration of light source and camera is required, and/or multiple captures of the scene are needed under varying illumination; again making interactive mobile scenarios challenging.

Learning-based and Statistical Methods: Given these challenges, more data-driven approaches to solving the SFS problem have been proposed. [Barron and Malik 2013] jointly solve for reflectance, shape and illumination, based on priors derived statistically from images. Our approach does not impose strong priors on shape recovery. [Khan et al. 2009] learn weighting parameters for complex SFS models to aid facial reconstruction. [Wei and Hirzinger 1996; Ben-Arie and Nandy 1998; Jiang et al. 2003] use deep neural networks to learn aspects of the physical model for SFS, demonstrating moderate results for very constrained scenes.

Other learning-based approaches operate on single color images of outdoor environments; for example, [Hoiem et al. 2005] extracts regions based on identity ('sky', 'ground', 'vertical'), and estimates depth coarsely. The technique of [Saxena et al. 2009; Karsch et al. 2012] relies on training depth cues and spatial relationships based on ground-truth image-depth pairs, but again only derives coarse depth. Whilst these approaches are closest to our work, none of this past work has demonstrated real-time performance, or the ability to smoothly infer absolute pixel-wise depth for video sequences. Real-time 2D-to-3D conversion techniques do exist, but they tend to rely on fragile video cues [Ideses et al. 2007]. Other approaches use learning for material and BRDF estimation [Hertzmann and Seitz 2005; Rother et al. 2011; Vineet et al. 2013], but do not directly address real-time depth prediction given intensity images.

[8] http://www.trimensional.com

[9] http://www.dinast.com

Other related approaches fit face and human body models to single images using statistical shape models e.g., [Guan et al. 2009; Blanz and Vetter 1999; Wang and Yang 2010]. [Smith and Hancock 2008] use a statistically derived SFS model specifically for facial reconstruction. In our work, we do not rely on specific global geometric models or shape priors for depth recovery.

Near-Infrared Imaging: Our approach can be thought of as an extension to SFS, where the problem is made tractable by focusing on dense and accurate depth prediction of human hands and faces (which is particularly important for interactive scenarios) and using controlled near-infared (NIR) based illumination with a simple camera modification.

Prior work and amateur photographers have exploited the fact that digital camera sensors are inherently sensitive to the NIR part of the electromagnetic spectrum (typically up to 1100nm) [Fredembach and Susstrunk 2008; Krishnan and Fergus 2009]. To prevent the NIR contamination of images, an IR cut filter (hot mirror) is often placed in front of the sensor, typically blocking wavelengths above 700nm. Prior work has explored ways in which this NIR signal can be used for photo enhancement and video denoising by removing this filter [Fredembach and Susstrunk 2008; Krishnan and Fergus 2009] and even using active illumination [Krishnan and Fergus 2009].

In our work we explore how this NIR signal and controlled illumination can be used for depth sensing. The use of controlled NIR lighting is critical for three main reasons. First, for our scenarios of sensing hands and faces, it allows us to approximate human skin as Lambertian. This is both due to the light scattering properties of skin under NIR [Simpson et al. 1998], and the smaller angles of incidence between the surface and light source [Marschner et al. 1999]. Second, different human skin tones have been shown to have similar reflectance properties under NIR [Simpson et al. 1998], making our approach more robust than visible light. Finally, NIR provides less intrusive sensing than using the visible spectrum.

Our system leads to a compact form-factor, low power and cost compared to existing active ToF or triangulation-based sensors. Our approach to the SFS problem is also fully data-driven, requiring no explicit calibration of camera or illumination, and predicating absolute depth in real-time without the need of multiple scene captures or complex lighting.

3 Hardware Setup

As illustrated in Figures 1, 2 and 3, our hardware setup consists of a regular commodity camera with minor modifications. First, we remove the IR cut filter typically present, permitting sensitivity to the spectrum range of ∼400-1100nm. Next, an IR bandpass filter operating at 850nm (±10nm) is used to limit all other wavelengths. This makes the camera sensitive only to this specific NIR range. Finally, we add diffuse LED illumination emitting at this spectral range. To ensure uniform lighting and limit shadowing, we build a ring of six NIR LEDs around the camera, with a minimal baseline. This setup is extremely cheap compared with stereo, structured light, or ToF. It also enables a very small form-factor that could easily be embedded into a modern smartphone.

We experiment with two instantiations of the above. The first adapts a Microsoft LifeCam (see Figure 2), and the second adapts a Smartphone Galaxy Nexus (see Figure 3 and related article[10]). Our diffuse IR LEDs operate at 850nm. The LEDs only consume a small amount of power (average power is 35mW) and can of course be switched off when not in use. A typical LED has a certain beam spread with luminous intensity attenuating away from the main axis. The LEDs

[10] http://www.eigenimaging.com/DIY/NexusDYI

Figure 2: *A standard Microsoft LifeCam web camera (top left) is modified to support depth sensing (bottom row). A bandpass filter operating at 850nm (+/-10nm) is added (a), once the front casing is opened and the IR cut filter is removed (b). A new 3D printed case (c) and ring of NIR LEDs (d) are additionally added.*

Figure 3: *A modified smartphone with additional NIR LED ring, bandpass filter, and custom 3D printed casing.*

we use in our system have a beam angle of ± 75 degree with 80% percent attenuation at the periphery. We reduce the unevenness of illumination by using a ring of LEDs.

The camera images are downsampled by a factor of three to 640x480 for our implementation (both devices support full HD capture). Downsampling can aid performance, and mitigate issues of defocus blur. In both our hardware implementations we measured the defocus blur extent to be $\sim$2 pixels. In addition we prefilter with a Gaussian filter, prior to subsampling, which substantially removes the effect of the different gains of the Bayer pattern of RGB filters in the IR spectrum (the signals in the visible range are blocked by the band-pass filter). Note also that chromatic aberrations are removed, because the color wavelengths are cut off by the IR bandpass filter.

4 Depth Prediction

This section details our depth prediction algorithm that learns to map a given a pixel $\mathbf{x}$ in the NIR image I to an absolute depth value. We model this continuous mapping $y(\mathbf{x}|I)$ with a multi-layered decision forest, following the formulation described in [Keskin et al. 2012]. This method attempts to simplify the problem by dividing it into sub-problems in the first layer, and then applies models trained for these sub-problems in the second layer to solve the main problem efficiently. For the first layer we employ a classification forest, and for the second layer we use regression forests (see [Amit and Geman 1997; Breiman 2001; Criminisi and Shotton 2013]).

For our task, the problem can be significantly simplified by restricting the depths of the objects to a certain range (primarily because we cannot use depth invariant features [Shotton et al. 2011], since depth is unknown). For such a constrained set, an *expert* forest can be trained to regress continuous and absolute depth values more efficiently. Thus, our first layer learns to infer a coarsely quantized depth range for each pixel, and optionally pools these predictions across all pixels to obtain a more reliable distribution over these depth ranges. The second layer then applies one or more expert regressors trained specifically on the inferred depth ranges. We aggregate these results to obtain a final estimation for the absolute depth y of the pixel $\mathbf{x}$. The resulting multi-layered forest is illustrated in Figure 4.

Multi-layered forests are discriminative models that learn a mapping conditioned on observations, *implicitly* capturing variation that exist in the training set. Therefore, the forests do not need to explicitly model scene illumination, surface geometry and reflectance, or complex inter-reflections. Later we demonstrate that spatially smooth, absolute metric depth images can be obtained from these NIR images for specific interaction scenarios, without the need to explicitly know these factors that are required by traditional SFS methods.

The next section presents details of our multi-layer decision forest-based depth estimation method.

4.1 Multi-layered Forest Architecture

Coarse, Discrete Depth Classification: Given an input pixel $\mathbf{x}$ and the infrared image I, the classification forest at the first layer infers a probability distribution $p(c|\mathbf{x}, I)$ over coarsely quantized depth ranges indicated by c, where $c \in \{1, \ldots, C\}$. The forest learns to map the pixel and its spatial context into one of the depth bins for each pixel. The experts at the next layer can be chosen based on this local estimate of c (denoted 'local expert network', or LEN), or alternatively the individual local posteriors can be aggregated (and averaged) over all the pixels to form the more robust estimate $p(c|I)$ (denoted 'global expert network', or GEN) [Keskin et al. 2012]. We compare these two techniques in detail later, but we found the aggregated posterior $p(c|I)$ to be more robust for our scenarios.

Examples of the inferred quantized depth predictions can be seen in Figure 5 for the specific case of $C = 4$. Evidently, the per-pixel predictions are somewhat noisy. However, using GEN, we aggregate across the entire image to predict a single quantized depth value, and this redundancy has proved to improve accuracy.

Fine, Continuous Depth Regression: As described above, each pixel or image is assigned a set of posterior probabilities over the depth bins in the first layer. In the second layer, we evaluate all the expert regression forests to form a set of absolute depth estimates. The final output depth y is a weighted sum over the estimates y_c of the experts, where the weights w_c are the posterior probabilities estimated by the first layer

$$y(\mathbf{x}|I) = \sum_{c=1}^{C} w_c y_c(\mathbf{x}|I) \; . \tag{1}$$

Here, w_c can either be the local posterior $p(c|\mathbf{x}, I)$ in the case of LEN, or the aggregated posterior $p(c|I)$ in the case of GEN. GEN is slightly more costly than LEN because of the extra pooling step after the first layer, but is more robust. Note that it is possible to threshold the posteriors to select a subset of the experts (or choose k experts where $k < C$) instead of triggering all the experts.

Multi-layered forests carry two main benefits over conventional single-layered forests. First, the multi-layered model can infer potentially useful intermediate variables that simplify the primary task, which in turn increases the accuracy of the model. Second, multi-layered forests have a reduced memory footprint than training deeper single-layered forests (as trees grow exponentially with depth). The

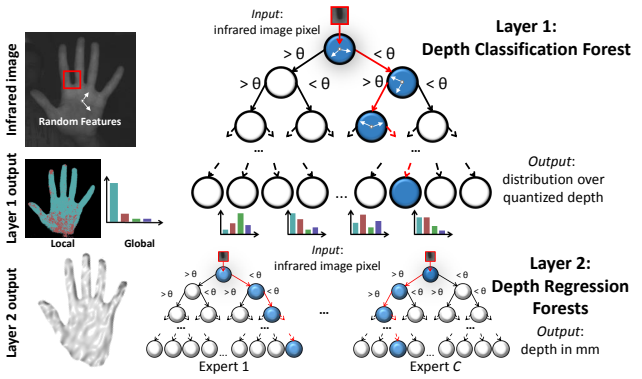

Figure 4: *Multi-layer decision forest for depth prediction. The first layer coarsely quantizes the output space of depth values into discrete bins, and uses a classification forest to predict a distribution over these bins for each pixel in the image. These per-pixel results form the 'local' predictions, and a distribution aggregated over all image pixels forms a 'global' prediction. The second layer then uses 'expert' regression forests, each of which is trained only on data within a particular bin. The regression forests are applied at each pixel and output an absolute, continuous depth value.*

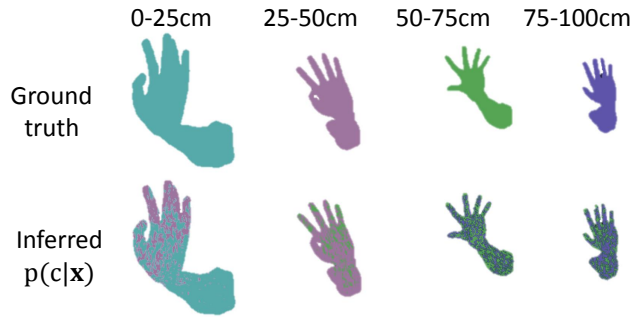

Figure 5: *Examples of first layer depth classification results with $C = 4$ bins. Each pixel predicts a quantized depth value. Almost all confusion is between neighboring bins, which are harder to distinguish. Note that errors in this layer can be corrected at the aggregation step after evaluating the experts (see text).*

multi-layered forest achieves the same task as the single-layered forest with $C + 1$ forests instead of one. However, because the task is simplified for the experts, they are typically much shallower than a single-layered forest that has the same accuracy. This reduction in model complexity usually more than makes up for the linear increase in the number of trees, provided that the inferred intermediate labels are useful for the task. For instance, a classification tree of depth 22 and $C = 4$ experts of depth 20 have the same size as a single tree of depth 24, but a single-layered forest may need to reach depth 28 to have similar accuracy, which is 16 times larger. This gain in memory is crucial for target mobile hardware with limited resources.

Furthermore, the ability of the global weighting to aggregate depths across all image pixels has the potential to make the final prediction more robust than possible with a single layer. This is borne out in experiments shown later in the paper.

Forest Predictions: The predictions $p(c|\mathbf{x}, I)$ from layer one and $y_c(\mathbf{x}|I)$ from layer two are made in a similar fashion using a decision forest. Each forest is an ensemble of multiple trees. At test time, a pixel $\mathbf{x}$ is passed into the root node. At each split (non-leaf) node j, a split function $f(\mathbf{x}; \theta_j) \in \{\mathrm{L}, \mathrm{R}\}$ is evaluated. This computes a binary decision based on some function of the image surrounding pixel $\mathbf{x}$ that depends on learned parameters θ_j. Depending on this decision, the pixel passes either to the left or right child, and the next split function is evaluated. When a leaf is reached in layer one, the stored distribution over quantized depth is output. The individual tree distributions are averaged together to form the forest output.

The distribution over depth values y at leaf nodes in layer two is multi-modal, and simply outputting the mean can lead to poor performance [Girshick et al. 2011]. We thus instead store a small set $\{\hat{y}_c^1(\mathbf{x}), \hat{y}_c^2(\mathbf{x}), \ldots\}$ of multi-modal predictions about possible values of the depth for this pixel. A median filter is then applied over these predictions within a small patch around pixel $\mathbf{x}$ across all trees in the forest, resulting in the final per-pixel prediction $y_c(\mathbf{x})$ that is then locally or globally weighted as described above.

Visual Features: Each split node contains a set of learned parameters $\theta = (\mathbf{u}, \mathbf{v}, \tau)$, where $(\mathbf{u}, \mathbf{v})$ are 2D pixel offsets and τ represents a threshold value. The split function f is evaluated at pixel $\mathbf{x}$ as

$$f(\mathbf{x}; \theta) = \begin{cases} \mathrm{L} & \text{if } \phi(\mathbf{x}; \mathbf{u}, \mathbf{v}) < \tau \\ \mathrm{R} & \text{otherwise} \end{cases} \quad (2)$$

$$\phi(\mathbf{x}; \mathbf{u}, \mathbf{v}) = I(\mathbf{x} + \mathbf{u}) - I(\mathbf{x} + \mathbf{v}) \quad (3)$$

where I is the input NIR image. This kind of pixel difference test is commonly used with decision forest classifiers due to its efficiency and discriminative power. The features also give additive illumination invariance which can help provide generalization across ambient illumination or penumbra. The relative offsets $\mathbf{u}$ and $\mathbf{v}$ can be quite large (up to ± 128 pixels in a 640x480 image) and allow the forests to learn about the spatial context in the image [Shotton et al. 2006]. We also investigated using only a 'unary' feature (with $\phi(\mathbf{x}; \mathbf{u}, \mathbf{v}) = \mathbf{I}(\mathbf{x} + \mathbf{u})$) but this did not appear to improve results.

Note the forest predictions are extremely efficient as only a small set of these simple feature tests are performed for each pixel. Furthermore, the pixels and trees can be processed in parallel. As shown later, this results in an efficient depth estimation algorithm, allowing real-time performance even on mobile hardware.

4.2 Training

Each tree in the forest is trained independently on a random subset S of the training data. For our application, set S contains training examples $(\mathbf{x}, y, c)$ where $\mathbf{x}$ identifies a pixel within a particular training image, y is the pixel's ground truth depth label, and c is the pixel's quantized depth label. Starting at the root, a set of candidate split function parameters θ are proposed at random. For each candidate, S is partitioned into left $S_\mathrm{L}(\theta)$ and right $S_\mathrm{R}(\theta)$ child sets, according to (2). The objective function below is evaluated given each of these partitions, and the candidate θ that maximizes the objective is chosen. Training then continues greedily down the tree, recursively partitioning the original set of training pixels into successively smaller subsets. Training stops when a node reaches a maximum depth, contains too few examples, or has too low an entropy or differential entropy (as defined below).

Each internal node is trained by selecting the parameters θ which maximize the information gain objective

$$Q(\theta) = E(S) - \sum_{d \in \{\mathrm{L}, \mathrm{R}\}} \frac{|S_d(\theta)|}{|S|} E(S_d(\theta)) \,. \quad (4)$$

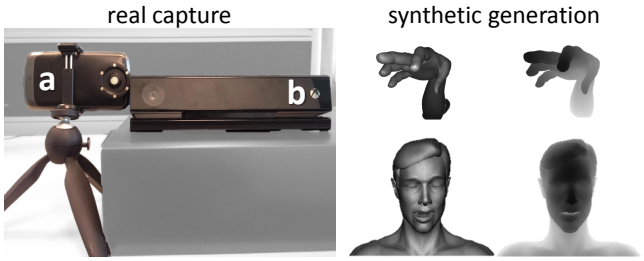

Figure 6: *Left: Real data is captured using a calibrated depth camera (b) registered to the NIR camera (a). Right: Synthetically generated infrared-depth image pairs of hands and faces.*

For the first layer (classification), $E(S)$ is the Shannon entropy of the (discrete) empirical distribution $p(c|S)$ of the quantized depth labels c in S:

$$E(S) = -\sum_{c=1}^{C} p(c|S) \log p(c|S), \text{with} \quad (5)$$

$$p(c|S) = \frac{1}{|S|} \sum_{(\cdot,\cdot,c') \in S} [c = c'] \, . \quad (6)$$

For the second layer (regression), $E(S)$ is instead the differential entropy of the empirical continuous density $p(y|S)$, where we model $p(y|S)$ as a one-dimensional Gaussian. Computing the mean μ_S and variance σ_S^2 in the standard way from the samples $(\cdot, y, \cdot) \in S$, the continuous entropy reduces to

$$E(S) = \log(\sigma_S) \, . \quad (7)$$

As mentioned above, at each leaf of the second layer regression forest, we store a small set of modes of the training density over depths y. This is obtained using mean shift mode detection [Comaniciu and Meer 2002]. For training data we investigate the use of both real and synthetic data.

Real Data: To acquire data from a *real* physical setup, we calibrate a depth sensor and register this to our NIR camera as depicted in Figure 6 (left). We first calibrate the intrinsic parameters of both cameras, and then the extrinsics using the method of [Zhang 2000]. We sequentially capture NIR-depth image pairs, by first capturing a single NIR image from our camera (with LEDs turned on); and then capturing a 'ground truth' depth map using the calibrated depth sensor. Using a signal generator, the illumination of each device is turned on sequentially to avoid cross-talk, and the user moves slowly to avoid motion artifacts. We pre-process the NIR images by applying a fixed intensity threshold to segment the hand or face from the background. This removes the need to train with varied backgrounds, reduces the compute load at test time, and works well in practice modulo extreme ambient illumination.

The validity of our approach depends critically on its ability to generalize from training data to previously unseen test data. We are thus particularly careful to ensure that training sequences are not used as test data. Cross-subject experiments later illustrate the generalization we achieve. In our current implementation, a total of $\sim$100K images are captured across a variety of different genders, age groups and skin tones.

Synthetic Data: The use of real ground-truth depth data can be further extended with the use of synthetic training data. Given the camera intrinsic parameters and the known intensity and angular range of the LEDs, it is possible to render realistic-looking infrared images with corresponding ground-truth depth. This can be done using a variety of 3D rendering software. In this work work we use Poser to generate around 100K hand and face (infrared, depth) image pairs (see Figure 6 right) that are uniformly distributed over a depth range of 20cm to 1m. To generate hand images, we use a 26-DoF articulated hand model attached to a forearm. Highly realistic renders can be obtained by posing this hand model with a series of angles applied to each joint, provided that the angles are randomly sampled with feasible kinematic constraints. Additionally, the hand and the fingers can be scaled to add shape variation to the data which helps with generalization to new subjects. We generate many variations of common hand poses such as pinching, pointing and grasping (and many more), performed at different ranges and X-Y positions. Likewise for faces we employ a realistic model with 100+ blendshapes that model face geometry and expressions. We randomize the weights of these blendshapes and apply a global transformation to the face to produce realistic renders.

In our renderer we model effects such as the illumination fall off from the LEDs, shot noise (added as Poisson noise), vignetting effects, as well as subsurface scattering for skin. Of the renders, we generate a hold-out set of 15K images for testing, and the remaining 100K images are used for training. Using synthetic images also lets us generate other labels (such as part colors) associated with depth pixels and images, which in turn provides training data for advanced applications such as hand or face pose tracking (as shown later).

5 Experiments

This section validates our approach experimentally. We present detailed qualitative and quantitative results for our method.

For our initial experiments, our model is trained using *real* data. Specifically, we use the calibrated setup in Figure 6 (left). The modified Microsoft LifeCam device is used for training unless otherwise stated. We use 20K training examples per subject (10K for hands and 10K for faces). A total of 5 subjects are captured, producing a dataset of 100K NIR/depth map pairs. For testing, we capture an additional 10K NIR/depth map pairs across the 5 subjects (1K images of hands and 1K of faces per subject).

Two main baselines are used for comparisons. The first uses depth maps captured from the commercially-available Xbox One ToF camera (denoted XBOXONE). The second (denoted INVERSESQUARE) computes depth maps using a SFS approach. A standard Lambertian reflectance model [Zhang et al. 1999] is combined with the inverse square law, and the intensity of each pixel $\mathbf{x}$ is modeled as

$$I(\mathbf{x}) = \frac{A\rho(\mathbf{x})\mathbf{n}(\mathbf{x}) \cdot \mathbf{s}}{D(\mathbf{x})^2} \quad (8)$$

where A is the intensity of the light source, ρ is the spatially-varying reflection coefficient that depends on the surface material, $\mathbf{n}$ the spatially-varying surface normal vector, $\mathbf{s}$ is the light source direction (assumed to be at infinity), and $D(\mathbf{x})$ is the depth of each pixel. The reflectances ρ, surface normals $\mathbf{n}$, and depths D are unknown.

By making some simplifying assumptions, we obtain an approximate depth estimate as a baseline. Assuming NIR illumination with known strength A that dominates ambient lighting, and a known, roughly constant reflectance ρ (reasonable for skin under NIR), we can obtain depth from intensity as:

$$D(\mathbf{x}) = \sqrt{\frac{A\rho \, \mathbf{n}(\mathbf{x})}{I(\mathbf{x})}} \, . \quad (9)$$

Note that to make this baseline more precise we compute surface normals using the real depth map obtained from the Xbox One.

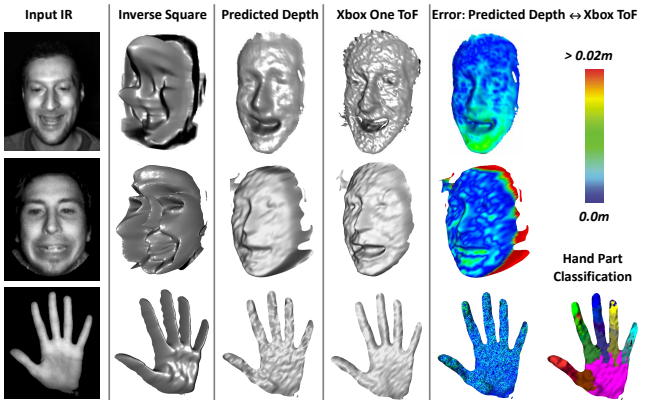

Figure 7: *Qualitative real data results for a trained face (top row), untrained face (middle row) and untrained hand (bottom row). Input NIR images (first column),* INVERSESQUARE *prediction (second column), our method (third column) and* XBOXONE *(fourth column). Distance error between* XBOXONE *and our method (fifth column). Hand part classification on our depth prediction (bottom right).*

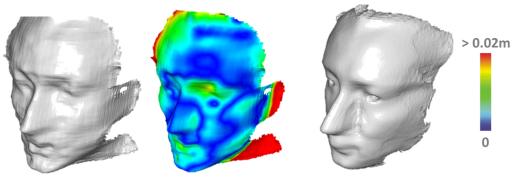

Figure 8: *Error (middle) from a 3D fused model created using our depth estimation method (left) and* XBOXONE *baseline (right).*

5.1 Qualitative Results

Qualitative comparisons are shown in Figure 7. The first column shows the input NIR images captured with our modified hardware. The second column shows the INVERSESQUARE prediction. Notice how with this SFS approximation there are major distortions in the resulting depth. Our depth prediction is shown in the third column, and XBOXONE depth maps in the fourth. The fifth column gives the distance error between the depth maps of our method and XBOXONE. The average error across all test images is ∼0.01m.

Note the first row shows results for an explicitly *trained* face i.e. we include data from this subject in the training set. The middle row shows the results for an *untrained* face i.e. we train on the four other subjects only, excluding the test subject. The final row shows results for an untrained hand, with part classification results using our depth prediction shown far right.

In Figure 8 (left) we show results after fusing multiple depth maps (in this case 1000 frames) to generate a 3D model of a user's face, using the KinectFusion system [Newcombe et al. 2011]. This shows errors less than 0.5cm (Figure 8 (middle)) when compared to a scan using the XBOXONE baseline (Figure 8 (right)). Note that this is based on an untrained subject.

5.2 Generalization Across Different Subjects

We next evaluate generalization across different subjects, for our multi-layered forest, with the leave-one-subject-out technique, i.e. training on 4 subjects, testing on a new one. Here both training and test are based on the previous real dataset. Therefore for a single subject, 80K training examples are used from all other subjects, and testing is then performed on the subject's 2K test examples.

Each column in Table 1 lists the results on a specific subject's dataset using leave-one-subject-out. The first row lists the accuracy we get from storing the average depth values at the leaves instead of the cluster modes, and the second row shows the error rates after storing two cluster modes in the leaves (see 'Forest Predictions' in Section 4.1).

	Subj. A	Subj. B	Subj. C	Subj. D	Subj. E
Mean	24.75	26.75	25.12	31.86	25.8
Mean shift	18.3	22.5	20.1	24.2	20.75

Table 1: *Leave-one-subject-out test results on each subject, in millimeters. The two rows correspond to storing the average value vs. the two primary cluster modes at the leaves in the forest.*

Note that the generalization to Subject D is slightly less accurate than the other subjects. This is attributed to much darker skin color Subject D had compared to the others. However, the net effect is only a small increase in error, and of course, for best accuracy one could train specifically for this skin color. In practice, given the use of NIR light, which has a similar response to skin irrespective of tone [Simpson et al. 1998], we found that skin color does not significantly impact the performance of the system.

5.3 Generalization Across Different Cameras

We also investigate cross-device generalization by training on a certain NIR camera and testing on a different one. Specifically, we test the modified Microsoft LifeCam (denoted WEBCAMERA) and Galaxy Nexus (denoted SMARTPHONE) described previously. For each device, we again capture real data with our Xbox One calibration rig (100K training examples, plus a different 15k for test, across 5 subjects, for each device). We adapt the learned model from one camera to another by simply re-scaling the feature offsets $\mathbf{u}$ and $\mathbf{v}$ by the relative change in focal lengths, and empirically derive a global intensity scaling and shift.

Results of the experiment are reported in Table 2. Unsurprisingly, the lowest error is obtained when trained and tested on the same sensor (just over 10mm), however the system exhibits good generalization capabilities across multiple devices. With the maximum error being below 25mm. Additionally we train on 100K synthetic images (50K hands and 50K faces) and test on both these devices (denoted SYNTHETIC). As shown, our error is below 23mm.

		Tested on	
		WEBCAMERA	SMARTPHONE
Trained on	WEBCAMERA	10.2	21.3
	SMARTPHONE	24.6	13.4
	SYNTHETIC	23.0	22.2

Table 2: *Depth prediction error obtained in the cross-device experiments. Models are trained on one device and tested on another. We report the mean errors in millimeters.*

5.4 Testing Vignetting and Illumination Effects

A major issue of SFS techniques is vignetting, where there is an undesired reduction of intensity at the image periphery. This can be caused by a combination of camera lens limitations, non-uniform LED illumination and the effects of a bandpass filter. Because our model is trained with real-world examples, these vignetting issues are represented in the training data. To measure how well our model copes with these lens, filter and illumination issues, we carry out a further experiment. Again the same real dataset is used

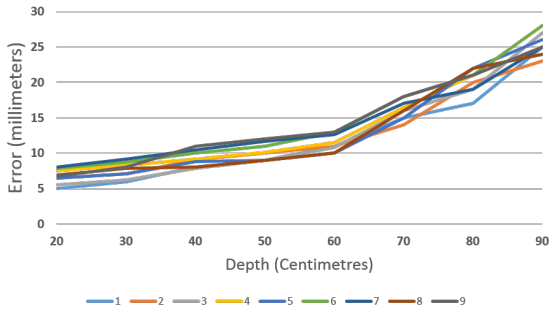

Figure 9: *Error across nine uniformly spaced regions in the camera image, as a function of depth.*

for training (based on the LifeCam), but for test a subject's hand is moved uniformly through the camera's view frustum at varying depths (100k test samples are collected).

The image plane is subdivided into nine uniformly spaced rectangular regions (numbered in scanline order, from 1 to 9 with 5 being the central region). Each predicted depth map is binned into one of these regions (based on the highest number of overlapping pixels), and averaged according to quantized depth. We then measure the error in each bin across varying depths (from 20-90cm at 10cm increments). Figure 9 shows that there is no substantial difference between errors reported in each bin (e.g. at 20cm error ranges from 5-7mm, and at 90cm this is 23-27mm). As expected, error increases (quadratically) with depth. However, no significant differences in error is found between bins as depth increases. This shows our model is able to *implicitly* learn to cope with vignetting and non-uniform illumination, which often need explicit calibration in SFS systems.

5.5 Model Comparisons and Parameter Selection

In this section, we evaluate the benefits of our multi-layered model, compared to a single layered baseline. We then empirically select optimal parameters for our model. In this section, all training and test is conducted using synthetic data (100K for training and 15K test, each split evenly across hands and faces).

First we compare a single forest trained to directly regress the depth from NIR images, with the proposed two-layered, classification-regression approach. Figure 10 shows the depth prediction error as a function of tree depth. A forest containing a single tree is used for both direct regression and the multi-layered method. The two-layered approach achieves an average test error of 35mm, and the curve levels off around about depth 25. The direct regression method instead reaches a minimum error of $\sim$50mm, and requires a considerably deeper (and thus less efficient) tree.

The classification layer aims to assign the correct quantized depth label to each pixel in the input image. The parameters that affect this accuracy are the number of quantized bins C, the number of trees T and the tree depth d. Via cross-validation, the optimal values are estimated as $T = 3$ and $d = 25$. As the value of C has a large effect on the accuracy of the multi-layered pipeline, its value is estimated by considering the final accuracy instead. The optimal C is found to be 4 for our experiments as shown in Figure 11(b).

We also quantify the effect of using GEN vs. LEN (as discussed in Section 4.1). Figure 11(a) compares these two weighting schemes. In our experiments LEN achieved an overall average error of 27mm, whereas the pooling step in GEN proved to be more robust and achieved an average error of 22mm. In these experiments, we limited the number of experts k to two. Given the improved accuracy, GEN was used for all other experiments in this paper.

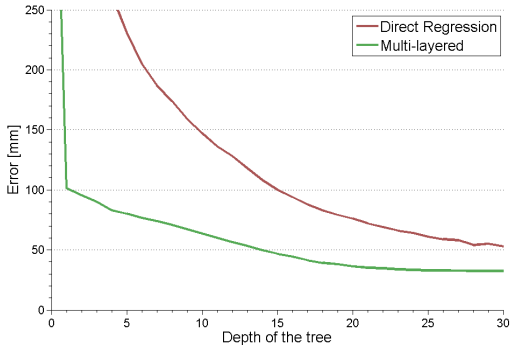

Figure 10: *Depth prediction error (in mm) as a function of tree depth for a single-layer direct regression baseline, and our proposed multi-layered classification-regression forest.*

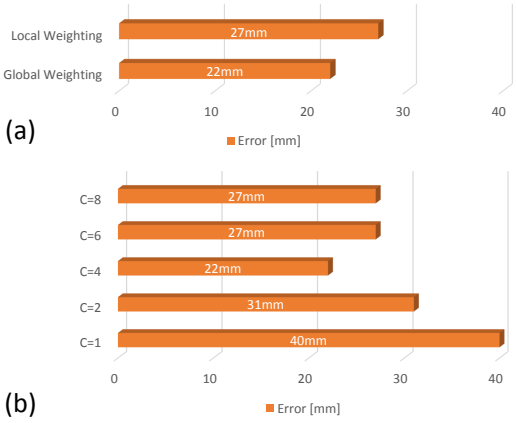

Figure 11: *(a) Comparison of GEN vs. LEN. (b) Error in second-layer depth prediction with respect to the number of depth bins (denoted C) in the first layer.*

5.6 Computational Analysis

The run-time complexity of the algorithm depends linearly on the size of the image I, as each foreground pixel is evaluated independently. For each pixel, the cost of evaluating the classification forest is $O(T_c D_c)$, where T_c is the classification forest size and D_c is the depth of the trees in the forest. Likewise, the time complexity of the second layer is $O(kT_r D_r)$, where k is the number of experts triggered. The method can easily be parallelized over all foreground pixels. The average runtime per forest is around 2-4 ms on high end PCs. In practice, we can obtain an average frame rate of 100 fps for the multi-layered architecture and around 250 fps for a single forest of depth 28. Further speed improvements are possible with GPU implementations. Given the availability of faster frame rate 2D cameras, it would therefore be possible to build a depth sensor with a much higher frame rate than is possible in the current generation of depth cameras.

6 Interacting with our Predicted Depth

So far we have shown how accurate depth maps can be predicted from NIR images using our method. In this section, we further analyze the quality of our depth estimation, specifically in the context of hand pose estimation. This is an important area of research with applications ranging from gaming to touch-less user interfaces. Here we focus on hand part labeling ([Shotton et al. 2011; Keskin et al. 2012]), and assess the need for depth prediction in this context.

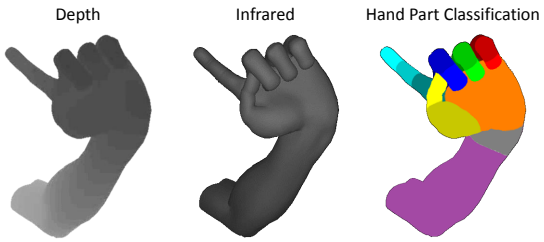

Figure 12: *Hand part classification training data. From left to right: Ground truth depth, associated IR image, and hand parts.*

To limit biases introduced from real data, in this section we again use synthetic data for training and test. An example of the dataset used is shown in Figure 12. We compare the accuracy of part classification using our depth predication (denoted MULTILAYERED) to four other modalities. The first uses ground truth depth maps for hand part classification (denoted DEPTH). The second (denoted INFRARED) uses the synthetic intensity image to perform hand part classification, i.e. without any depth prediction. The third (denoted INVERSEDEPTH) again uses the SFS depth approximation introduced previously. We also evaluate against a single-layered regression forest (denoted DIRECTREGRESSION).

Hand part classification results: We divide the arm-hand into 17 classes, according to the hand anatomy (see Figure 12 right). The training data is composed of pairs (x, y), where x is one of the modalities and y is the output space of the class labels. We train a multi-class classifier for each of the considered modalities as described in Section 4.1. Here the synthetic intensity images are used to train the MULTILAYERED, DIRECTREGRESSION and INVERSESQUARE modalities.

We use 6000 images for the training set and test on a different 2500 set. Parameters are selected via cross-validation. We use *depth probes* as features for computing the split function in the depth-based conditions [Shotton et al. 2011], and intensity-based features in the INFRARED case. We compute the average pixel accuracy for each class, showing the mean accuracy across the 17 classes.

In Figure 13, we show quantitative results. On the training data, the mean accuracy of INFRARED and INVERSESQUARE conditions is 10-15% below the DEPTH and MULTILAYERED conditions. On test data, DEPTH obtains the highest accuracy (87.3%), while MULTILAYERED follows closely with 81.0%. DIRECTREGRESSION achieves a lower 78.7%, but is still about 10% higher than INFRARED and INVERSESQUARE baselines (68.4% and 69.5% respectively). Perhaps the most interesting finding is that the two stage approach of first predicting depth and then classifying hand parts is more precise that directly classifying the intensity image. The primary reason is that by first predicting depth we are able to use depth invariant features to improve the hand part classification accuracy.

Generalization of our technique: To qualify the generalization capabilities that our technique is able to achieve, we conducted a further transfer learning experiment. In transfer learning, the goal is to acquire knowledge from one problem and try to apply it to a different but related task. In our setting, we train the hand part classifiers using synthetic depth maps, and then use the depth predicted by MULTILAYERED, DIRECTREGRESSION and INVERSESQUARE as a comparison. Results are shown in Figure 14. Our approach achieves the remarkable accuracy of 76.7%, whereas the INVERSESQUARE condition reaches just 10.4%. Furthermore we highlight the improvements obtained by the multi-layered system: the DIRECT REGRESSION approach obtains a lower average accuracy of

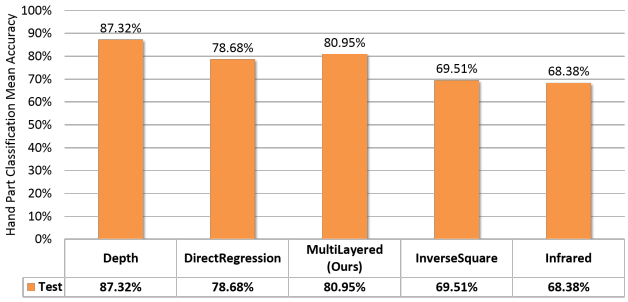

Figure 13: *Quantitative results for hand part classification where each condition is trained and tested on its own modality, i.e. trained on depth and tested on depth. See text for details.*

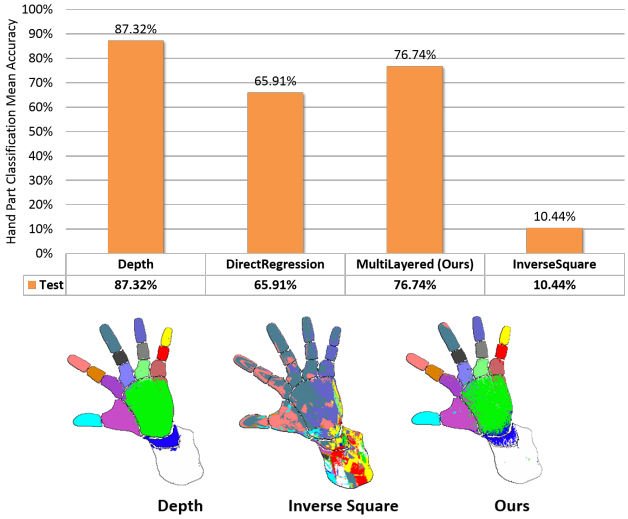

Figure 14: *Transfer learning hand part classification results. Here models trained on ground truth depth are tested on the other modalities. Bottom row: qualitative examples of transfer learning. Here* INVERSESQUARE *produces very poor results, our predicted depth produces results close to the real depth. This proves the generalization capabilities of our method. See text for details.*

65.9%. These results also suggest that our technique can be used in pre-existing applications that use depth maps, avoiding costly re-training phases.

6.1 Example applications

Some example applications are shown in Figure 15, as well as the supplementary video. These prove the effectiveness of the proposed method for gesture recognition, real-time face part recognition and finally hand tracking on mobile phones.

We designed an interface that allows a user to drive Windows 8 applications with hand gestures in real-time. The module recognizes when the hand is open, closed and pointing with a (depth-based) forest classifier. Then, the hand part classification pipeline is used to infer skeleton joint locations, which are used used to track finger trajectories, enabling capabilities such as drawing. Notably, this application, and associated model were trained from Kinect (version 1) depth images. However, as shown previously, these learned models can be transferred to our depth prediction system, producing compelling results.

The hand tracking pipeline extracts useful information from part labels. Inspired from this, we implemented an analogous system that

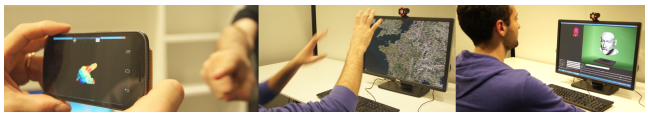

Figure 15: *Some example applications created using our method.*

estimates part labels for the face, which are then used to track certain landmarks and expressions. Here, we train a facial part classifier and an expression regressor on synthetic face images, and then replace the depth data with our method at test-time. See Figures 1 and 15 and the supplementary video for examples.

Finally, to demonstrate the capabilities of our method, as well as the simplicity and applicability of the hardware customization step, we designed a mobile phone application that first predicts depth, then performs hand part classification, followed by simple model fitting. This allows fully articulated hand tracking on the mobile phone, in a form-factor that is prohibitive for current generations of depth cameras. Examples are given in Figures 1 and 15 and the supplementary video.

6.2 Limitations

Whilst we have demonstrated the utility of our approach, there are clearly limitations. Firstly we train for uniform surface albedo (in this case skin) which limits our depth estimation to human faces or hands, or any other specific object. Our system fails to predict depth for surfaces with varying reflectance properties. Another issue of our approach is the sensitivity to ambient IR (a problem with other depth camera techniques as well). The narrow bandpass filter helps alleviates some of these issues. A further possibility here is to subtract ambient illumination, by turning the illuminant on and off at alternate frames, and subtracting background ambient IR. This extension would require lower-level access to the sync signal from the camera, and can suffer from motion artifacts, although with a high-enough frame rate this can be alleviated. Alternatively, it may be possible to include sufficiently large ambient illumination variations in the training data (especially when using synthetic data) that the system learns some level of invariance to ambient IR illumination.

Other camera specific effects such as vignetting can also be an issue, particularly when training our system for general cross device usage. Note however when training on real data on a single device, our proposed model implicitly learns to account for vignetting effects without the need to explicitly formulate them. The forest learns a mapping conditioned on these vignetting effects. Given sufficient training data, we found the system to be remarkably insensitive to absolute spatial location in the view frustum. However, this type of per-device training can be costly or impractical in certain scenarios.

Another limitation is that the camera modification limits the ability to capture visible light images. One approach to enable both visible and IR imaging is to replace the standard RGB Bayer pattern with an RGBI pattern. Camera manufacturers such as OmniVision and Aptina now produce such cameras. This would still be significantly lower cost and power than a full depth camera, though would require the use of a custom sensor.

7 Conclusion

In this paper, we proposed and demonstrated a low-cost technique to turn any 2D camera into a real-time depth sensor with only simple and cheap modifications. Diffuse NIR LEDs illuminate objects near the camera, and capture the reflected light with the help of an added band pass filter. The actual depth calculation is done by a machine learning algorithm, and can learn to map a pixel and its context to an absolute, metric depth value. As this is a data driven, discriminative machine learning method, it learns to capture any variation that exists in the dataset, such as changes in shape, geometry, skin color, ambient illumination, complex inter-object reflections and even vignetting effects, without the need to explicitly formulate them. To capture this much information via simple rules encoded in the decision forests, we employed a multi-layered forest that simplifies this problem in the first layer by predicting coarse quantized depth ranges for the object.

We demonstrated the efficiency of this method through qualitative and quantitative experiments. In particular we showed comparisons with other modalities for a range of applications, cross-subject and cross-device generalization capabilities, as well as the high quality inferred depth for hand and face tracking, and 3D reconstruction. It should be noted that the method described is not for a general purpose depth camera. Whilst this method cannot replace commodity depth sensors for general use, our hope is that it will enable 3D face and hand sensing and interactive systems in novel contexts.

References

AHMED, A. H., AND FARAG, A. A. 2007. Shape from shading under various imaging conditions. In *Proc. CVPR*, IEEE, 1–8.

AMIT, Y., AND GEMAN, D. 1997. Shape quantization and recognition with randomized trees. *Neural Computation 9*, 7.

BARRON, J. T., AND MALIK, J. 2013. Shape, illumination, and reflectance from shading. Tech. Rep. UCB/EECS-2013-117, EECS, UC Berkeley, May.

BATLLE, J., MOUADDIB, E., AND SALVI, J. 1998. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern Recognition 31*, 7, 963–982.

BEN-ARIE, J., AND NANDY, D. 1998. A neural network approach for reconstructing surface shape from shading. In *In Proc. ICIP 98.*, vol. 2, IEEE, 972–976.

BESL, P. J. 1988. Active, optical range imaging sensors. *Machine vision and applications 1*, 2, 127–152.

BLAIS, F. 2004. Review of 20 years of range sensor development. *Journal of Electronic Imaging 13*, 1.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. *Proc. ACM SIGGRAPH*.

BREIMAN, L. 2001. Random forests. *Machine Learning 45*, 1.

BROWN, M. Z., BURSCHKA, D., AND HAGER, G. D. 2003. Advances in computational stereo. *PAMI 25*, 8, 993–1008.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI 24*, 5.

CRIMINISI, A., AND SHOTTON, J. 2013. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer.

FREDEMBACH, C., AND SUSSTRUNK, S. 2008. Colouring the near-infrared. In *Color and Imaging Conference*, vol. 2008, Society for Imaging Science and Technology, 176–182.

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG) 30*, 6, 129.

GIRSHICK, R., SHOTTON, J., KOHLI, P., CRIMINISI, A., AND FITZGIBBON, A. 2011. Efficient regression of general-activity human poses from depth images. In *Proc. ICCV*.

GUAN, P., WEISS, A., BALAN, A., AND BLACK, M. 2009. Estimating human shape and pose from a single image. In *Proc. ICCV*.

GURBUZ, S. 2009. Application of inverse square law for 3d sensing. In *SPIE Optical Engineering+ Applications*, International Society for Optics and Photonics, 744706–744706.

HERNÁNDEZ, C., VOGIATZIS, G., AND CIPOLLA, R. 2008. Multiview photometric stereo. *IEEE Trans. PAMI 30*, 3, 548–554.

HERTZMANN, A., AND SEITZ, S. 2005. Example-based photometric stereo: Shape reconstruction with general, varying BRDFs. *PAMI 27*, 8.

HOIEM, D., EFROS, A., AND HEBERT, M. 2005. Automatic photo pop-up. In *Proc. ACM SIGGRAPH*.

HORN, B. K. 1975. Obtaining shape from shading information. *The psychology of computer vision*, 115–155.

IDESES, I., YAROSLAVSKY, L., AND FISHBAIN, B. 2007. Real-time 2D to 3D video conversion. *J. of Real-Time Image Processing 2*, 3–9.

JIANG, T., LIU, B., LU, Y., AND EVANS, D. 2003. A neural network approach to shape from shading. *International journal of computer mathematics 80*, 4, 433–439.

KARSCH, K., LIU, C., AND KANG, S. 2012. Depth extraction from video using non-parametric sampling. In *Proc. ECCV*.

KESKIN, C., KIRAÇ, F., KARA, Y., AND AKARUN, L. 2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proc. ECCV*.

KHAN, N., TRAN, L., AND TAPPEN, M. 2009. Training many-parameter shape-from-shading models using a surface database. In *Proc. ICCV Workshop*.

KIM, D., HILLIGES, O., IZADI, S., BUTLER, A. D., CHEN, J., OIKONOMIDIS, I., AND OLIVIER, P. 2012. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM, 167–176.

KRISHNAN, D., AND FERGUS, R. 2009. Dark flash photography. In *ACM Transactions on Graphics, SIGGRAPH 2009 Conference Proceedings*, vol. 28.

LANMAN, D., AND TAUBIN, G. 2009. Build your own 3D scanner: 3D photography for beginners. In *ACM SIGGRAPH 2009 Courses*, ACM, 8.

LIAO, M., WANG, L., YANG, R., AND GONG, M. 2007. Light fall-off stereo. In *Proc. CVPR*.

LIU, C. P., CHENG, B. H., CHEN, P. L., AND JENG, T. R. 2011. Study of three-dimensional sensing by using inverse square law. *Magnetics, IEEE Transactions on 47*, 3, 687–690.

MARSCHNER, S. R., WESTIN, S. H., LAFORTUNE, E. P., TORRANCE, K. E., AND GREENBERG, D. P. 1999. Image-based BRDF measurement including human skin. In *Rendering Techniques 99*. Springer, 131–144.

MULLIGAN, J., AND BROLLY, X. 2004. Surface determination by photometric ranging. In *Proc. CVPR Workshop*.

NEWCOMBE, R. A., IZADI, S., ET AL. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, IEEE, 127–136.

PRADOS, E., AND FAUGERAS, O. 2005. Shape from shading: a well-posed problem? In *Proc. CVPR*, vol. 2.

REMONDINO, F., AND STOPPA, D. 2013. *ToF range-imaging cameras*. Springer.

ROTHER, C., KIEFEL, M., ZHANG, L., SCHÖLKOPF, B., AND GEHLER, P. V. 2011. Recovering intrinsic images with a global sparsity prior on reflectance. In *Proc. NIPS*.

SAXENA, A., SUN, M., AND NG, A. 2009. Make3D: Learning 3D scene structure from a single still image. *PAMI 31*, 5, 824–840.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IJCV*.

SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2006. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. ECCV*.

SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. 2011. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*.

SIMPSON, C. R., KOHL, M., ESSENPREIS, M., AND COPE, M. 1998. Near-infrared optical properties of ex vivo human skin and subcutaneous tissues measured using the monte carlo inversion technique. *Physics in Medicine and Biology 43*, 2465–2478.

SMITH, W. A., AND HANCOCK, E. R. 2008. Facial shape-from-shading and recognition using principal geodesic analysis and robust statistics. *International Journal of Computer Vision 76*, 1, 71–91.

TUNWATTANAPONG, B., FYFFE, G., GRAHAM, P., BUSCH, J., YU, X., GHOSH, A., AND DEBEVEC, P. 2013. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on Graphics (TOG) 32*, 4, 109.

VINEET, V., ROTHER, C., AND TORR, P. 2013. Higher order priors for joint intrinsic image, objects, and attributes estimation. In *Proc. NIPS*, 557–565.

VISENTINI-SCARZANELLA, M., STOYANOV, D., AND YANG, G.-Z. 2012. Metric depth recovery from monocular images using shape-from-shading and specularities. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, IEEE, 25–28.

VOGEL, O., BREUSS, M., LEICHTWEIS, T., AND WEICKERT, J. 2009. Fast shape from shading for Phong-type surfaces. In *International Conf. Scale Space and Variational Methods*.

WANG, X., AND YANG, R. 2010. Learning 3D shape from a single facial image via non-linear manifold embedding and alignment. In *Proc. CVPR*.

WEI, G.-Q., AND HIRZINGER, G. 1996. Learning shape from shading by a multilayer network. *IEEE Transactions on Neural Networks 7*, 4, 985–995.

ZHANG, Z., TSA, P.-S., CRYER, J. E., AND SHAH, M. 1999. Shape from shading: A survey. *PAMI 21*, 8, 690–706.

ZHANG, Z. 2000. A flexible new technique for camera calibration. *IEEE Trans. PAMI 22*, 11, 1330–1334.

ZHANG, S. 2010. Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. *Optics and lasers in engineering 48*, 2, 149–158.