# Multi-touchless: Real-Time Fingertip Detection and Tracking Using Geodesic Maxima

Philip Krejov and Richard Bowden
Center for Vision, Speech and Signal Processing
University of Surrey, Guildford, UK
Email: {p.krejov, r.bowden}@surrey.ac.uk

***Abstract*—Since the advent of multitouch screens users have been able to interact using fingertip gestures in a two dimensional plane. With the development of depth cameras, such as the Kinect, attempts have been made to reproduce the detection of gestures for three dimensional interaction. Many of these use contour analysis to find the fingertips, however the success of such approaches is limited due to sensor noise and rapid movements. This paper discusses an approach to identify fingertips during rapid movement at varying depths allowing multitouch without contact with a screen. To achieve this, we use a weighted graph that is built using the depth information of the hand to determine the geodesic maxima of the surface. Fingertips are then selected from these maxima using a simplified model of the hand and correspondence found over successive frames. Our experiments show real-time performance for multiple users providing tracking at 30fps for up to 4 hands and we compare our results with state-of-the-art methods, providing accuracy an order of magnitude better than existing approaches.**

## I. Introduction

Multitouch technology is now commonplace across modern devices but limited to the two-dimensional plane of the display. The detection and tracking of fingertips in three dimensions, opens up multitouch to new applications. Tracking the fingertips in three space independently of display plane has applications in areas of computing ranging from medical analysis in sterile environments, to home entertainment and gaming. Detecting and tracking the fingertips in real-time is difficult due to the variability in environment, the fast motion of the hands and the high degree of freedom of finger movement. We propose a robust real-time approach that uses geodesic maxima instead of visual appearance which is both efficient to compute and robust to both pose and environment.

With the arrival of time-of-flight (TOF) and structured light cameras, depth information has become available, and has brought about pioneering methods of capturing user interaction. It also facilitates simplified techniques to background subtraction and clustering; processes pertaining to robust tracking. Using the depth information, we use a graph-based approach to find the geodesic extrema of the hand showing that these extrema correspond to the tips of extended fingers. We also demonstrate a method for eliminating points incorrectly identified as fingertips.

## II. Related Work

There is a large body of work that focuses on the task of analysing hands for use in computer interaction. This includes systems developed to track hands or fingertip locations as hand tracking is often the first stage in finding the fingertips. Using a region-growing technique Chen et al [4] locates and tracks the center of the hand. Nanda and Fujimura [11] uses Potential Fields to track a model contour over depth sequences that can be applied to hands. Van Den Bergh and Van Gool[22] track hands using depth segmentation and a broad RGB color model to segment the hand.

Model-based tracking is a popular approach capable of inferring the location of the fingertips [12], and can account for large scale occlusion [7]. These techniques can achieve 15 frames per second, for a single hand but this is still too slow for smooth interaction. Our approach is capable of four hands at 30 frames per second.

Many researchers have chosen to optimise model fitting with the assistance of devices that are attached to a user. Motion capture is a common example of this, and works using markers placed at key locations across the hand, sometimes including the wrist. Aristidou and Lasenby [2] use markers to reduce the complexity of mapping a model to the full structure of the hand. A colored glove is employed by [24] to simplify pose inference from a colour image to a dataset of "tiny" images. Matching is performed using nearest neighbor, providing an approximate pose of the hand, which is then improved using blending.

One prerequisite of seamless interaction is the use of gloveless detection, which relies on the structure of the hand, often constraining the palm to have to face the camera. Many systems examine contour curvature to determine the most probable location of the finger tips. Lee and Hollerer [10] improve on Argyros and Lourakis's [1] method of finger detection by fitting an ellipse to candidate points, and selecting the tip as the furthest point of the ellipse's principle axis. Pan et al [14] uses curvature analysis to initialise a multi-cue hand tracker which, once initialised, tracks using 100 KLT features. A scanning window can also find candidate fingertips, Oka et al uses a template of a rectangle with a semi-circular tip [13]. Oka et al demonstrates a method for tracking assignment, using the hand's orientation to reduce the permutations of matching points from one frame to another. The space between fingers also aids in detection when using a convex hull around the hand [5]. Other methods use detection based on finger edges [19] and the narrow nature of fingers [8][20][6].

Using depth information obtained from a structured light

camera Raheja et al first removes the background. Following this he uses a large circular filter to remove the palm and obtain the fingers' masks. Each finger's tip is located by finding the closest point to the camera under each finger mask [16]. Hackenberg et al [6] uses the understanding that the fingers comprise of tips and pipes, to find fingers. Keskin et al [9] uses features first applied to body pose estimation, to train a random forest using a labeled model. This random forest then classifies the regions of a captured hand on a per pixal basis. Pose is then estimated using a support vector machine. For simple single finger interaction Takahashi et al [21] takes the closest region to the camera.

In terms of real-world interaction, these methods heavily constrain the hand to facing the camera. This is due to the requirement for clear separation between fingers and limits gestures to two-dimensions coplanar with the image plane. By using depth we can relax this constraint, allowing fingertip detection for complex out-of-plane hand poses using the geodesic distance along the surface of the hand.

The use of geodesic distance was applied to the task of locating the body skeleton by Plagemann et al [15] using a time of flight camera. Later work by Schwarz et al [18] and Baak et al [3] built on this work proposing optimisation and recovery schemes. We employ the same idea of using geodesic extrema but in the context of finger detection and use Dijkstra's algorithm to efficiently identify these candidate regions.

### A. Contribution

In this paper we propose a system for the detection and tracking of multiple hands and fingertips acquired using a Kinect that operates at 30fps. Running on a multi-core desktop PC it is capable of processing four hands simultaneously while maintaining this performance. The first contribution incorporates a graph based approach, to accurately retrieve candidate fingertips. Our implementation of Dijkstra's uses less depth heuristics than those previously applied in body pose estimation [3] reducing the need to compute a graph union while maintaining robustness. Secondly, we demonstrate an approach to filtering wrist points wrongly identified in the process of selecting fingertips. Finally we use euclidean real world coordinates to eliminate retrieved points on the fist and closed fingers and provide smoother trajectories than filtering in the image domain.

The remainder of this paper is organized as follows: Sect. III, defines the methods used in the detection and tracking framework. Sect. IV details our experimental findings and parametric validation. Sect. V discusses potential future work and concludes. The resulting approach is both faster and more accurate by an order of magnitude than competing approaches.

## III. Method Description

Depth images are captured using a Microsoft Kinect through OpenCV, which provides calibrated depth information. For the purpose of formulation we define each pixel in this depth image as a point tuple, containing the pixel and corresponding real world coordinates, defined as p = $(x^c, y^c, x^w, y^w, z^w)$. $w(p) = (x^w, y^w, z^w)$, to access individual world coordinates, and use $c(p) = (x^c, y^c)$ for image coordinates. In both cases we use subscript $w_i(p)$, $i \in \{x, y, z\}$ as short hand for a specific coordinate.

### A. Method Overview

We begin by capturing the depth image and calibrated point cloud, derived from the intrinsic properties of the kinect. We locate and separate hand blobs with respect to the image domain. Processing each hand in parallel, we build a weighted graph from the real world point information for the surface of each hand. Our method uses an efficient Dijkstra's shortest path algorithm to traverse the graph in order to find N candidate fingertips. These candidates are then filtered based on their location relative to the center of the hand and the wrist. Points that remain are then used in a Kalman-smoothed model of the fingertip locations. The correspondence between the model's prediction and detected points is found using the least sum of squared differences. Detected points are added to the model as a new finger if no corresponding point is found in the model. Points in the model that do not have an assignment either receive a blind update, or are removed.

### B. Hand and Forearm Segmentation

For each colour frame provided by the Kinect, the user's face is located using a standard Viola Jones detector [23]. The face detection provides a point location for each frame, with multiple face detections removed using non-maximal suppression. Using a point cloud derived from the calibrated depth information (distance from the Kinect), the bounding box of the closest face $f$ is found and is considered the working user. The depth is smoothed over a small temporal window of 20ms, and defines the back plane for segmentation, which allows the removal of the body and background from the depth image. The remaining depth space is considered the working area for interaction. Any points within this space are clustered into connected blobs for further analysis. Each point in a blob forms $p = (x^c, y^c, x^w, y^w, z^w)$, hence a hand can be described as the following set;

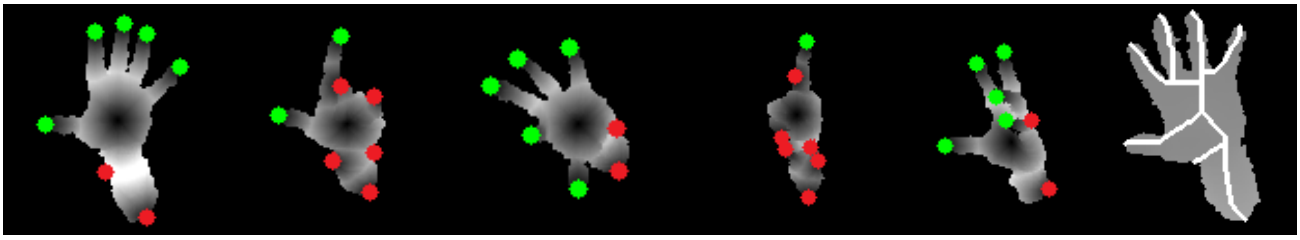
$$\mathbf{P} = \{p_i\}_{i=1}^{|\mathbf{P}|} \tag{1}$$

Blobs that are smaller than potential hand shapes are ignored using $w_z(f)$ to account for the users distance from the Kinect. The remaining blobs are considered candidates for the user's arms. Points in $\mathbf{P}$ are then divided as belonging to either a hand subset or wrist subset defined as $\mathbf{P}^H \subset \mathbf{P}, \mathbf{P}^W \subset \mathbf{P}$ and $\mathbf{P} = \mathbf{P}^H \cup \mathbf{P}^W$. The classification of these points is performed using a depth cut-off $W_{depth}$ positioned at one quarter of the arms total depth which is calculated using the following:

$$W_{depth} = Z_{max} - \frac{(Z_{max} - Z_{min})}{4} \tag{2}$$

where,

$$Z_{max} = \max_{p \in \mathbf{P}}(w_z(p)) \tag{3}$$



(a) Open Palm (b) Pinching (c) Open Pinching (d) Pointing (e) Self Occlusion (f) Extrema Paths

Fig. 1. Multiple hand shapes and their first seven extremities: (a) Open hand with two extrema belonging to the wrist. (b) and (c) Are examples of pinching. A single pointing finger where only one other extrema approaches the tip (d). (e) Demonstration of a hand with self occlusion. (f) Shows the paths for each extrema

and,

$$Z_{min} = \min_{p \in \mathbf{P}}(w_z(p)) \tag{4}$$

This heuristic was found experimentally while being constrained anthropometrically to lie between the forearm and wrist. This also ensures that $|\mathbf{P}^W| \neq 0$ . Points in $\mathbf{P}$ further from the camera than $W_{depth}$ form the forearm set defined as;

$$\mathbf{P}^W = \{p \mid p \in \mathbf{P}, w_z(p) >= W_{depth}\} \tag{5}$$

Points closer than $W_{depth}$ belong to the hand,

$$\mathbf{P}^H = \{p \mid p \in \mathbf{P}, w_z(p) < W_{depth}\} \tag{6}$$

*1) Wrist Centroid:* We model the wrist using the set of points $\mathbf{P}$ to form an elipse detailed in section III-D that is centered around the wrist centroid. This wrist Centroid $p^{\widehat{W}}$ is found as the average;

$$p^{\widehat{W}} = \left( \frac{1}{|\mathbf{P}^W|} \sum_{p \in \mathbf{P}^W} c_x(p), \frac{1}{|\mathbf{P}^W|} \sum_{p \in \mathbf{P}^W} c_y(p) \right) \tag{7}$$

*2) Hand Centre Localisation:* Finding the centre of the hand is necessary as it is used later as the seed location for Dijkstra's algorithm. Simply using the centroid of points $\frac{\sum \mathbf{P}^H}{|\mathbf{P}^H|}$ would result in the seed point shifting when the hand is opened and closed. We use the chamfer distance of the set of points $\mathbf{P}^H$from the external boundary to find a stable centre for the hand[13]. Using this, a hands center $p^{\widehat{H}}$ is found to be the point in $\mathbf{P}^H$ with the greatest distance in the chamfer image.

### C. Candidate Fingertip Detection

The fingertips can be considered as extremities of the hand. This means that by mapping the surface of the hand and searching for five extremities, the fingertips can be found, excluding those that are closed. In practice however, the wrist also forms additional extremities that are of similar geodesic distance. For this reason we greedily compute the first seven extremities, hence each fingertip is accounted for, including two false-positives. In the case where fingers are closed, we do not consider the tip to be of interest, as the finger contributes to the formation of the fist. This coincides with the understanding that it is no longer an extremity, and is no longer found using Dijkstra's approach. The extremity normally associated with a folded finger now forms an additional false-positive to be filtered at a later stage.

In order to find the candidate fingertips we first build a weighted undirected graph of the hand, based on the points of $\mathbf{P}^H$ discovered in section III-B. Each point present in the hand represents a vertex in the graph. These vertices are connected with neighbouring vertices in an 8-neighbourhood fashion, with the edge cost being derived from the euclidean distance between their world coordinates.

Using the hand's centre $p^{\widehat{H}}$ as the seed point, we compute the first geodesic extremity of the hand graph. The remaining extrema are then found iteratively using Dijkstra's with a non-initialised distance map [3] reducing the computational intensity. The seven extremities found for various hand shapes are shown in figure 1. We define these first seven extremities as $\mathbf{E} = \{e^1, ..., e^7\}, e^i \in \mathbf{P}^H$. Each extrema is associated with its shortest path, shown in figure 1f . Formally we define this path for the $i^{th}$ extrema ($e^i$) as an ordered set of vertices $\mathbf{V}^i = \{v_1, ..., v_{|\mathbf{V}^i|}\}$ where, $v_1 = e^i$ and $v_{|\mathbf{V}^i|} = p^{\widehat{H}}$.

### D. Non-Fingertip Rejection

Once the set $\mathbf{E}$ of fingertip candidates has been found the next stage is to filter these points to a subset of valid fingertips. This is performed using a combination of a penalty metric derived from the path taken during Dijkstra's and a candidate's position relative to the centre of the hand. The aforementioned penalty criterion, is used specifically to remove falsely identified tips that reside around the wrist.

Finding the penalty for each candidate requires the covariance of the hand and arm points, after it is translated to the wrist location. Using the covariance to map the wrist in this manner, takes in to consideration the variability of hand shapes. The covariance $cov(\mathbf{P})$ is found using,

$$cov(\mathbf{P}) = \frac{1}{|\mathbf{P}|} \sum_{p \in \mathbf{P}} (c(p) - c(p^{\widehat{H}}))(c(p) - c(p^{\widehat{H}}))^T \tag{8}$$

This covariance is then translated to form a masking ellipse centred around the wrist. Pixels that have a Mahalanobis distance within three standard deviations of the wrist are marked as 1, while pixels outside of this boundary are marked

(a) Open Palm (b) Closed Fist (c) Two Extended Fingers

Fig. 2. Covariance of several hand shapes once it is translated to the wrist. (a) Shows a long principle axis for a full hand and arm. A circular boundary can be seen for a closed fist in (b). (c) demonstrates rotation in the wrist.

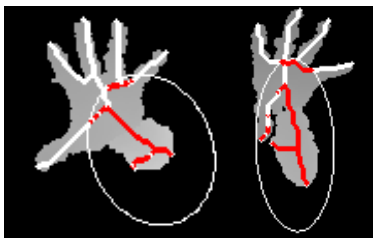
as 0, which is shown in Figure 2. The following equation details the formulation of this mask,

$$\mathbf{M}(p) = \begin{cases} 1 & \text{if } (c(p) - p^{\widehat{W}})cov(\mathbf{P})^{-1}(c(p) - p^{\widehat{W}}) < 9, \\ 0 & Otherwise. \end{cases} \tag{9}$$

Using both the path from the extrema to the hand's centre, and the elliptical mask, the penalty $S(\mathbf{V})$ can be found. The paths score is incremented for each vertex with increasing depth through the mask ellipse, and is then normalised using the complete path's length. It is important to mention that when moving along this path, a vertex is only included in the penalty if the current vertex's $v_i$ depth is less then the next $v_{i+1}$ ie, $w_z(v_i) < w_z(v_{i+1})$. This heuristic is consistent with the understanding that the wrist has a greater depth than the centre of the hand, hence we only consider vertices that traverse with increasing depth. This is found for the path associated with each candidate in $\mathbf{E}$ with the following;

$$S(\mathbf{V}) = \frac{1}{|\mathbf{V}|}\sum_{i=1}^{|\mathbf{V}|}\mathbf{M}(v_i)\mathbb{I}[w_z(v_i) < w_z(v_{i+1})] \tag{10}$$

Figure 3 shows the vertices in red that contribute to an increased score.



(a) (b)

Fig. 3. The complete shortest path for each extrema. The points along each path that contribute to an increase in score are coloured in red. (a) shows that vertices outside of the ellipse do not contribute to the penalty. The path of the thumb in (b) demonstrates the need to only penalise vertices that traverse with increasing depth.

The candidates are then filtered using Equation 11 down to a subset of candidates $\mathbf{E}' \in \mathbf{E}$, that exclude wrist extrema, where the value of $\theta$ is determined using parameter tuning detailed in section IV.

$$\mathbf{E}' = \{e' : e' \in \mathbf{E}, S(\mathbf{V}) < \theta\} \tag{11}$$

The remaining candidates are reduced using the euclidean distance of the fingertip, to the centre of the hand $\mathbf{E}'' \in \mathbf{E}'$;

$$\mathbf{E}'' = \{e'' : e'' \in \mathbf{E}', d(w(e'), w(p^{\widehat{H}})) > \beta\} \tag{12}$$

where distance $d()$ is the L2 Norm and $\beta$ is the cut-off radius. This in practice forms a sphere around the user's hand. When a fingertip is detected outside of this sphere, the tip is considered a true-positive. This hard cut-off was chosen so as to give a clear condition of when a fingertip is detected, to improve user interaction. The value of $\beta$ was chosen using parametric tuning across multiple users and multiple hand shapes, which is detailed in Section IV.

### *E. Finger Assignment and Tracking*

A Kalman filter is used to track and assign the fingertips between frames. This model is updated using the point correspondence that minimises the change between consecutive frames. We found the need to check all possible permutations when matching points, as our tracking is performed in three dimensions. When assigning five or less points, searching all permutations ($O(n!)$) requires less operations than using the Hungarian algorithm($O(n^3)$). As the hand is limited to five fingers it consists of only 120 permutations in the worst case.

The world position of each detected fingertip $w(e'')$ is used to update a bank of three dimensional Kalman filters that persist from the previous frame forming $\mathbf{K}_{t-1}$. To update this model, points from $\mathbf{E}''$ are paired with the predictions of $\mathbf{K}_{t-1}$ by selecting the assignment between points that have the lowest cost. This requires an indexing set to map from one set of points to the other. This map is constructed using the permutations of the smaller of these two sets ${}^{\mathbb{N}^0}\mathcal{P}_{|A|}$ where, $|\mathbf{A}| = min(|\mathbf{E}''|, |\mathbf{F}|)$ and, $\mathbb{N}^0$ is non-negative natural numbers. This forms a matrix where each row represents one of the possible permutations. Iterating through each permutation we build the assignment set $\mathbf{A} = (a_1, ..., a_{|\mathbf{A}|}) \in {}^{\mathbb{N}^0}\mathcal{P}_{|A|}$ where $a$ is the index used to associate a fingertip to a Kalman filter. The final correspondence between fingertips is the permutation $\mathbf{A}^{BEST}$ which has the lowest sum of squared differences from $\mathbf{E}''$ to the predictions of $\mathbf{K}_{t-1}$, as shown here;

$$\mathbf{A}^{BEST} = \begin{cases} \underset{\mathbf{A}}{\text{argmin}} \sum_{i=1}^{|\mathbf{A}|}(w(e_i'') - k_{ai})^2 & \text{if } |\mathbf{E}''| \leq |\mathbf{K}_{t-1}| \\ \underset{\mathbf{A}}{\text{argmin}} \sum_{i=1}^{|\mathbf{A}|}(k_i - w(e_{ai}''))^2 & \text{if } |\mathbf{E}''| > |\mathbf{K}_{t-1}| \end{cases} \tag{13}$$

Any Points in $\mathbf{E}''$ that are not matched initialise a new point $k$ that is introduced into the model. This is to account for the presence of new fingers. For predictions derived from $\mathbf{K}_{t-1}$ that were not matched, their Kalman filter is updated using the previous predicted position. This blind update is performed on the condition that the prediction's confidence

(a) (b) (c) (d) (e) (f)

Fig. 4. Multiple fingertips and the centre of the hand tracked over ten frames. The length of each line demonstrates the distance covered in the last ten frames.

does not diminish considerably, at which point it is removed from the model.

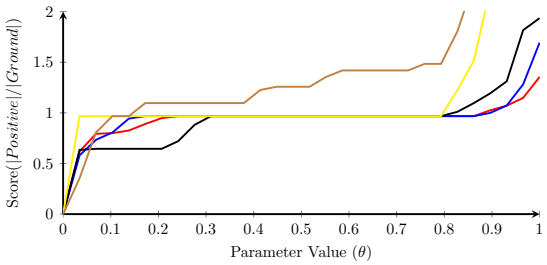
The Kalman smoothed model can then be used to output each of the fingertips as a three dimensional coordinate in millimeters. These coordinates can also be projected back to the image domain for the purpose of displaying the points as shown in Figure 4. It is important to track the points in three dimensions as the addition of depth allows for sub-pixel accuracy. While it is possible to track the points in the image domain, the resulting fingertips are prone to jitter due to the lower resolution.

## IV. Experiments

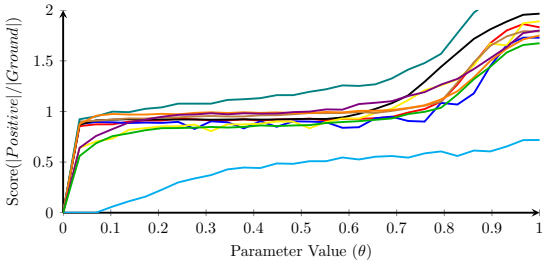
We test and demonstrate the proposed method using a Microsoft Kinect. Written in C++, the testing is performed on a standard desktop PC with a 3.4 GHz i7 CPU. In order to evaluate performance it was necessary to capture our own dataset, as current hand datasets are more related to gesture acquisition. While it would have been possible to label an existing dataset manually, for example Rens [17] gesture set, they do not contain information regarding transitions between hand shapes. For our dataset we captured ten sequences of point cloud and RGB data. The data is captured using five seated users performing multiple actions with blind capturing, with hands up to a meter from the Kinect. Each user was asked to perform their first sequence with a constrained rate of movement, to asses the generalisation across multiple users. They were then requested to perform actions at a faster rate with gestures that were less restrained. This was to replicate more realistic movement that is suitable for interaction, which included transitions between gestures and natural resting poses. Another set of sequences consisting of just gestures was captured for parametric tuning. For the purpose of testing, all sequences were semi-autonomously labelled using course finger detection that was then manually corrected. This data is available at http://personal.ee.surrey.ac.uk/Personal/P.Krejov/.

### A. Wrist Exclusion

To asses the effect of the parameters $\theta$ and $\beta$ performance was evaluated over a small set of sequences before evaluation in section IV-C on a larger unseen testing set. For wrist exclusion, the parameter $\theta$ can have a possible range between 0 and 1. As the value of $\theta$ increases, filtering of wrist points




(a) scores for $\theta$ for each hand shape




(b) scores for $\theta$ for each user sequince

Fig. 5. The results show parameter tuning of the variable $\theta$. Varying hand shapes are plotted in (a). The plot (b) is results across various users, where the lower line is for sequence $5a$ where the hand is lost during tracking.

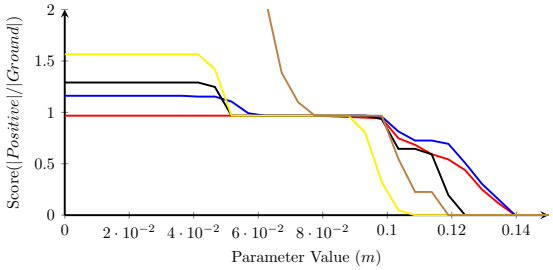
relaxes, and more false-positives are detected. The ratio of fingers detected over the number of ground labeled fingers was recorded. Figure 5 shows the ratio of detections over ground track points vs $\theta$ for various hand shapes and users. It can be seen that there is a broad plateau where the value of $\theta$ does not affect performance, showing that $\theta$ generalises well across both hand shapes and users. We selected a value of 0.37 from within this region for use in our experiments.

### B. Fist Exclusion

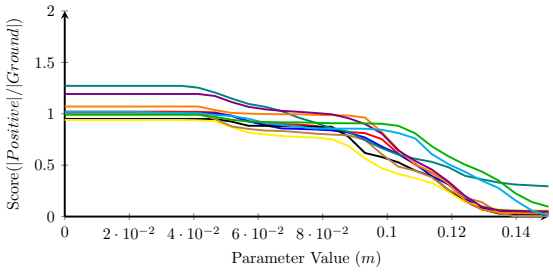
Using the same principle for fist circumference, $\beta$ was tested across multiple hand shapes and users. Figure 6 shows a wide plateau over which parameter selection does not affect performance. We selected a value of 7mm for our experiments.

### C. Evaluation

There are two main considerations when assessing the performance of the approach: the number of fingers detected with relation to the correct amount, and the precision of




(a) Open Pinching




(b) Self Occlusion

Fig. 6. The results recorded while parameter tuning the variable $\beta$. For varying hand shapes (a). The plot (b) is the results across various users, and show stability for a range of values.

the estimated position. Combining these metrics to quantify the overall performance of the system is avoided, as an appropriate weighting depends on the application. For this reason we represent the performance independently for both. Correct fingertips are identified as being within (1cm) of the ground truth fingertip. The results of this can be seen for both sequences for each user in Table I.

| User | Error(cm) | TP(%) | TP(%) Div |
|---|---|---|---|
| 1 | 2.20 | 75.62 | 20.64 |
| 2 | 1.17 | 83.15 | 16.01 |
| 3 | 1.60 | 88.37 | 19.79 |
| 4 | 2.65 | 77.45 | 21.49 |
| 5 | 3.05 | 69.51 | 24.25 |
| 6 | 4.68 | 75.63 | 22.59 |
| 7 | 3.57 | 70.74 | 23.65 |
| 8 | 0.83 | 87.62 | 18.92 |
| 9 | 3.34 | 74.40 | 17.54 |
| 10 | 2.26 | 79.78 | 21.02 |
| Average | 2.48 | 78.23 | 19.11 |

TABLE I
THE ERRORS FOR EACH SEQUENCE IN CM ERROR AND THE PERCENTAGE OF CORRECTLY IDENTIFIED FINGER TIPS.

We found that there is varying difficulty across each of the sequences, with the most difficult being user 5. In this particular sequence the user used very rapid movements for interaction. This motion caused extensive blurring in the footage, and the structure of the hand was completely lost for many frames.

The average error across all sequences for each hand is 2.48cm. However, this does not provide a fair representation of the performance as this includes outliers that offset the hand error. A better understanding is taken from the histogram of all errors per finger, which is shown in Figure 7, with 80 percent of the errors within 5.7mm. It is also found that

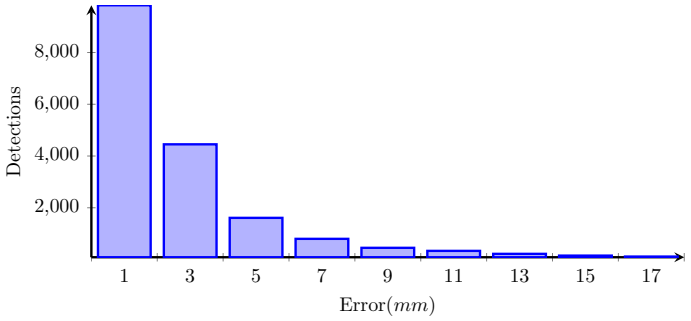



Fig. 7. A histogram of the detections across all user sequences, showing the majority of detections are within 5mm of there ground truth.

73% of the finger tips are within the 4mm boundary. A comparison with existing work can be attempted, as the results of Oikonomidis are given in terms of millimeters. It must be noted that their approach is optimised for pose estimation, not finger tip localisation. For a temporal sequence their total error is 5-7.5mm for varying depths, while for single frame estimation 74% of their results are within 40mm. Our detection improves on this, with 80% of detections being within 5.7mm or 5.1mm for instantaneous detection. This demonstrates that our approach does not rely heavily on temporal smoothing with accuracy that is an order of magnitude greater for static estimation. While their approach is applied to more complex hand poses, they achieve real-time performance through the use of GPU optimisation. Our approach concentrates on finger tip detection but allows 30fps for multiple hands with out gpu optimisation.

## V. Conclusions and Future Work

We proposed a novel graph based method for fingertip detection and tracking for use in computer interaction. The data captured using a Kinect provides the data for precise localisation of the possible fingertips through the use of Dijkstra's algorithm. With the use of robust filtering, candidates are tracked using a Kalman filter to provide accurate fingertip localisation. The method is able to process multiple hands at frame-rates in excess of 30 fps, providing multiple users the ability to interact using a Multitouchless interface. For future work we look to improve this system using a skeletal tracker to improve the robustness of the hand and wrist localisation. In addition we aim to improve detection and filtering using machine learning techniques.

## VI. Acknowledgments

This work was funded by the EPSRC project Making Sense EP/H023135/1

## REFERENCES


[1] Antonis A. Argyros and Manolis I. A. Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *Proceedings of the 2006 international conference on Computer Vision in Human-Computer Interaction*, ECCV'06, pages 40–51, Berlin, Heidelberg, 2006. Springer-Verlag.

[2] A. Aristidou and J. Lasenby. Motion capture with constrained inverse kinematics for real-time hand tracking. In *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*, pages 1 –5, march 2010.

[3] Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE 13th International Conference on Computer Vision, to appear*, November 2011.

[4] Chia-Ping Chen, Yu-Ting Chen, Ping-Han Lee, Yu-Pao Tsai, and Shawmin Lei. Real-time hand tracking on depth images. In *Visual Communications and Image Processing (VCIP), 2011 IEEE*, pages 1 –4, nov. 2011.

[5] V. Frati and D. Prattichizzo. Using kinect for hand tracking and rendering in wearable haptics. In *World Haptics Conference (WHC), 2011 IEEE*, pages 317 –321, june 2011.

[6] G. Hackenberg, R. McCall, and W. Broll. Lightweight palm and finger tracking for real-time 3d gesture control. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 19 –26, march 2011.

[7] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1475 –1482, 29 2009-oct. 2 2009.

[8] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 441–450, New York, NY, USA, 2011. ACM.

[9] C. Keskin, F. Kirac, Y.E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1228 –1234, nov. 2011.

[10] Taehee Lee and T. Hollerer. Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 83 –90, oct. 2007.

[11] H. Nanda and K. Fujimura. Visual tracking using depth data. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, page 37, june 2004.

[12] Nikolaos Oikonomidis, Iason. Kyriazis and Argyros Antonis. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11. BMVA Press, 2011.

[13] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *Computer Graphics and Applications, IEEE*, 22(6):64 – 71, nov/dec 2002.

[14] Zhigeng Pan, Yang Li, Mingmin Zhang, Chao Sun, Kangde Guo, Xing Tang, and S.Z. Zhou. A real-time multi-cue hand tracking algorithm based on computer vision. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 219 –222, march 2010.

[15] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108 –3113, may 2010.

[16] J.L. Raheja, A. Chaudhary, and K. Singal. Tracking of fingertips and centers of palm using kinect. In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*, pages 248 –252, sept. 2011.

[17] Zhou Ren, Jingjing Meng, and Junsong Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1 –5, dec. 2011.

[18] L.A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab. Estimating human 3d pose from time-of-flight images based on geodesic distances and optical flow. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 700 –706, march 2011.

[19] Gihan Shin and Junchul Chun. Vision-based multimodal human computer interface based on parallel tracking of eye and hand motion. In *Convergence Information Technology, 2007. International Conference on*, pages 2443 –2448, nov. 2007.

[20] G. Simion, V. Gui, and M. Otesteanu. Finger detection based on hand contour and colour information. In *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, pages 97 –100, may 2011.

[21] Masaki Takahashi, Mahito Fujii, Masahide Naemura, and Shin'ichi Satoh. Human gesture recognition using 3.5-dimensional trajectory features for hands-free user interface. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, ARTEMIS '10, pages 3–8, New York, NY, USA, 2010. ACM.

[22] M. Van den Bergh and L. Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66 –72, jan. 2011.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001.

[24] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 63:1–63:8, New York, NY, USA, 2009. ACM.