Head-Pose Tracking with a Time-of-Flight Camera

Simon Meers and Koren Ward
University of Wollongong, Australia
meers@uow.edu.au, koren@uow.edu.au

Abstract

Intelligent interfaces that make use of the user's head pose or facial features in order to interpret the user's identity or point of attention, are finding increasing application in numerous fields. Although various techniques exist to passively track the user's gaze or head pose using monocular or stereo cameras, these systems generally cannot perceive in detail the characteristic three-dimensional (3D) profile of the user's head or face. Time-of-flight cameras, such as the Swiss Ranger SR-3000, are a recent innovation capable of providing three-dimensional image data from a single sensor. The advent of such sensors opens up new possibilities in the fields of head-pose tracking and face recognition. In this paper we propose a novel 3D time-of-flight camera system for robust and accurate head-pose tracking in real-time. Our system requires no manual calibration and is tolerant of varying lighting conditions, occlusion and facial expressions. The system can also be used to generate accurate 3D "face prints" suitable for use in face recognition with minimal data and search times. Preliminary experimental results are provided which demonstrate the potential of the system.

1 Introduction

1.1 Background

Intelligent user interfaces that involve tracking the user's head position or eye gaze direction are finding increasing applications such as view control in 3D simulation programs, video conferencing, interactive whiteboards, support systems for the disabled and haptic user interfaces for the blind. Generally such systems can be put into two categories. Firstly, those that involve the user having to wear a special helmet fitted with position estimation hardware or special spectacles incorporating exposed IR reflectors or LEDs that can be tracked by external sensors, *e.g.* [Hong and Park, 2005; NaturalPoint Inc., 2006; Meers *et al.*, 2006]. Secondly, those involving the use of monocular or stereo cameras to passively track the user's gaze or head pose, *e.g.* [Birchfield, 1998; Toyama, 1998; Heinzmann and Zelinsky, 1998].

Although wearing special helmets or IR spectacles can provide an inexpensive and accurate means of estimating of the user's head pose, such systems are often inconvenient or uncomfortable for the user to wear. Alternatively, various techniques have been developed to passively track the user's gaze or head pose using monocular or stereo cameras. Although monocular cameras have demonstrated considerable success at head pose tracking, (*e.g.* faceAPI [Seeing Machines, 2008a]), achieving high accuracy is lacking. More accurate results have been achieved with stereo disparity cameras. For example, FaceLab [Seeing Machines, 2008b], can track a face within 1° accuracy. However, this accuracy can be lost to some extent when the tracked facial features are partially obscured due to wider head movements. Furthermore, disparity cameras cannot determine the 3D profile of the whole face (due to the presence of featureless surfaces), which can facilitate tracking the user's head pose and determining the user's identity.

To improve on existing head pose tracking systems we have been experimenting with time-of-flight cameras, such as the Swiss Ranger SR-3000 [MESA Imaging, 2006b] which is capable of producing three-dimensional image data from a single sensor in real time. Our method involves detecting the user's nose tip in 3D space and then comparing spherical intersections with an internal 3D model of the user's face in order to determine the user's head pose direction. Our system has demonstrated itself to be robust and requires no manual calibration. It is also tolerant of variations in lighting conditions, occlusion and facial expressions. Our system has also demonstrated itself to be capable of generating accurate 3D "face prints" as a by-product which are suitable for use in face recognition. In the following sections,

we provide a brief description of the SwissRanger SR-3000 time-of-flight camera used in our system, followed by comprehensive descriptions of our nose tip detection method, our spherical intersection head pose tracking technique, and how we propose to adapt the system for face recognition. We conclude with a brief summary of our system and our preliminary experimental results which demonstrate the potential of the system.

1.2 The SwissRanger

In 2006, Swiss company MESA Imaging announced the release of the SR-3000 "SwissRanger" time-of-flight camera [MESA Imaging, 2006b]. The camera is surrounded by infrared LEDs which illuminate the scene, and allows the depth of each pixel to be measured based on the time of arrival of the frequency modulated infrared light in real-time. Thus for each frame it is able to provide a depth map in addition to a standard greyscale amplitude image. The amplitude image is based solely on reflected infrared light, and therefore is not affected by external lighting conditions.

Figure 1: SwissRanger SR-3000

(a) (b)

Figure 2: *(a)* Sample SR-3000 amplitude image, and *(b)* corresponding depth map

Despite the technological breakthrough that the SwissRanger has provided, it has a number of limitations. The sensor is QCIF (176x144 pixels), so the resolution of the data is low. The sensor also has a limited 'non-ambiguity range' before the signals get out of phase. At the standard 20MHz frequency, this range is 7.5 metres. However, given the comparatively short-range nature of our application, this limitation does not pose a problem for our system. The main limitation that has caused us some concern is *noise* associated with rapid movement. The SR-3000 sensor is controlled as a so-called 1-tap sensor. This means that in order to obtain distance information, four consecutive exposures have to be performed. Fast moving targets in the scene may therefore cause errors in the distance calculations; see [MESA Imaging, 2006a]. Whilst the accuracy of a depth map of a relatively stationary target is quite impressive (see Figure 3), the depth map of a target in rapid motion is almost unusable by itself (see Figure 4). We hope to overcome this problem to a considerable extent by using a combination of median filtering, time fusion and by combining the intensity image data with the depth map.

Figure 3: Point cloud for a stationary subject

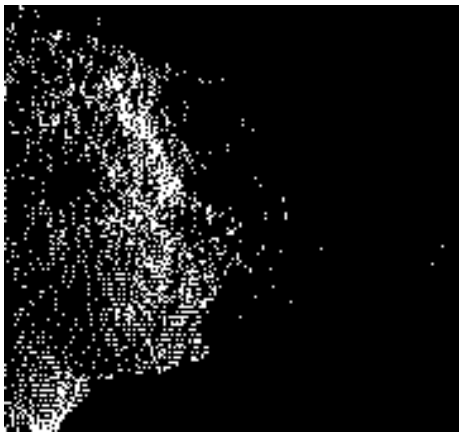

Figure 4: Point cloud for a subject in motion

2 Overview

Our system is required to be able to track an arbitrary human face. The diverse range of faces, hairstyles

and accessories makes this task considerably difficult. A feature-based approach would require the availability of at least three identifiable features in order to obtain an unambiguous three-dimensional orientation. For example, locating the eyes and nose would be sufficient to calculate the orientation. However the user's eyes may not always be visible to the camera, especially if the user is wearing sunglasses or spectacles. The ears could be used, however they may not be visible if the user has long hair. Both the eyes and ears could also be partially occluded from the camera when the user is looking away from the camera. The edges of the mouth could be used, though they can change with the facial expression, and could be obscured by facial hair. Likewise the chin or jaw-line (which are very easily detected in a depth map) cannot be guaranteed to be available on all users. Furthermore, a user may have a combination of feature-obscuring characteristics (*e.g.* long hair, facial hair, glasses, etc).

Consequently, the most universally available and identifiable feature on the human face is the nose, for a number of reasons. Firstly, it is rarely obscured. Secondly, if it is occluded the user can be assumed to be facing away from the camera. Thirdly, it is advantageously positioned near the centre of the face on an approximate axis of symmetry. Others researchers also consider the nose to be important in facial tracking, *e.g.* [Gorodnichy, 2002; Yin and Basu, 2001], and have devised systems to reliably detect the nose in both amplitude and depth images [Gorodnichy, 2002; Haker *et al.*, 2007]. The 'tip' of the nose (furthest protrusion from face) is considered to be the most important point in our system.

Although the nose can be considered to be the best facial feature to track due to its availability and universality, clearly more than this single feature is needed to obtain the orientation of the face in three-dimensional space. One approach would be to use an algorithm such as Iterative Closest Point (ICP) [Zhang, 1994] to match the facial model obtained in the previous frame with the current frame. This method may work but is expensive to do in real time. It may also fail if the head moves too quickly or if some frames are noisy and the initial fit is a considerable distance from optimal.

Alternatively, we could formulate an adaptive feature selection algorithm which automatically detects identifiable features within the depth map or amplitude image (or ideally a combination of the two) in order to detect additional features which can be used to perform matching. Here, a redundant set of features could theoretically provide the ability to match the orientation between two models with high accuracy. In practise however, the low resolution of the SwissRanger camera combined with the noisy nature of the depth map have caused this approach to prove unsuccessful. The features obtained in such an adaptive feature selection algorithm would also need to be coerced to conform to the target spacial configuration. To overcome these difficulties, we have developed a novel approach that simplifies the feature selection process whilst simultaneously removing the need for spatial coercion.

The premise is relatively simple: we only have one feature so far (ie the nose tip) and we need more, preferably of a known spatial configuration. Therefore, we intersect our model with a sphere of radius r centred on our nose feature. The resulting intersection profile contains all points on the model which are r units away from the central feature. Because of the spherical nature of the intersection, the resulting intersection profile is completely unaffected by the orientation of the model, and thus ideal for our purpose. It could simply be analysed for symmetry, if we could assume that the face is sufficiently symmetrical and that the central feature lies on the axis of symmetry, and an approximate head-pose could be calculated based on symmetry alone. However, given that many human noses are far from symmetrical, and up to 50% of the face may be occluded due to rotation, this approach will not always succeed. But if we save the model from the first frame, we can use spherical intersections to match it against subsequent frames and thus obtain the relative positional and rotational transformation. Multiple spherical intersections can be performed to increase the accuracy of the system.

Subsequently, the orientation matching problem is now reduced to that of aligning paths on spherical surfaces. This can be performed using ICP or a similar alignment optimisation algorithm. A good initial fit can be performed regardless of the orientation by simply optimising the alignment of the latitudinal extrema of the profiles (*i.e.* the top-most and bottom-most points). These should always be present, because at least 50% of the face is visible. If not, their absence can be easily detected by the fact that they lie on the end-points of the path, thus indicating that the latitudinal traversal was cut short. The latitudinal extrema are reliable features of the intersection profile due to the 'roll' rotational limits of the human head.

After having matched a subsequent 3D model to the position and orientation of the previous one, additional data becomes available. If the orientation of the face has changed, some regions that were previously occluded (*e.g.* the far-side of the nose) may now be visible. Thus, merging the new 3D data into the existing 3D model improves the accuracy of subsequent tracking. Consequently, with every new matched frame, more data is added to the target 3D model making it into a continuously evolving mesh.

To prevent the data comprising the 3D model from becoming too large we propose using polygonal simpli-

fication algorithms to adaptively reduce the complexity of the model. If a sufficient number of frames indicate that some regions of the target model are inaccurate, points can be adaptively moulded to match the majority of the data, thus filtering out noise, occlusion, expression changes, etc. In fact, regions of the model can be identified as being rigid (reliably robust) or noisy / fluid (such as hair, regions subject to facial expression variation, etc) and appropriately labeled. Thus, matching can be performed more accurately by appropriately weighting the robust areas of the 3D model. We expect to have this innovation incorporated into the system in time for our next publication.

This approach to head pose tracking clearly depends heavily on the accuracy of the estimation of the initial central feature point (*i.e.* the nose tip). If this is offset, the entire intersection profile changes. Fortunately, the spherical intersection profiles themselves can be used to improve the initial central point position. By using a hill climbing approach in three-dimensional space, the central point can be adjusted slightly in each direction to check for a more accurate profile match. This will converge upon the best approximation of the target centre point provided the initial estimate is relatively close to the optimal position.

Furthermore, the system can be used to differentiate or identify users. Each evolving mesh can be stored in a database, and a new model can be generated for a face which does not sufficiently match any existing models. Due to the simplified nature of spherical intersection profile comparisons, a database of faces can be searched with considerable efficiency. Spherical intersection profiles for each model could also be cached for fast searching.

3 Preprocessing

Several steps are used to prepare the acquired SwissRanger data for processing.

3.1 Median Filtering

As discussed in Section 1.2, the SwissRanger depth map is subject to considerable noise, particularly if the subject is in motion. Median filtering is applied to reduce the effects of noise in the depth map. The amount of noise in the depth map is also measured to identify frames which are likely to produce inaccurate results due to excessive noise.

3.2 Distance and amplitude thresholds

Some cross-correlation with the amplitude image can help eliminate erroneous data in the depth map. For example, pixels in the depth map which correspond to zero values in the amplitude image (most frequently found around the edges of objects) are likely to have been affected by object motion, and can be filtered out. Minimum and maximum distance thresholds can also be applied to the depth map to eliminate objects in the foreground or background.

3.3 Region of Interest

Whilst the user's torso can be of some value to a head-pose tracking application, it is generally desirable to perform calculations based solely upon the facial region of the images. Identifying the facial region robustly is non-trivial. Our initial prototype performed accurate jaw-line detection based on the greatest discontinuity in each column of the depth map in order to separate the head from the body. This approach sometimes failed due to extreme head rotation, excessive noise, presence of facial hair, occlusion, etc. Consequently, we opted for a more robust approach using a simple bounding-box. Given that a depth map is available, it is straightforward to determine the approximate distance of the user from the camera. This is achieved by sampling the first n non-empty rows in the depth map (the top of the user's head) and then calculating the average depth. By taking the approximate distance of the user's head, and anthropometric statistics [Poston, 2000], we determine the maximum number of rows the head is likely to occupy within the images. We then calculate the centroid of the depth map pixels within these rows and centre a region of interest of the appropriate dimensions on this point (see Figure 5). This method has proved 100% reliable in all sample sequences recorded to date.

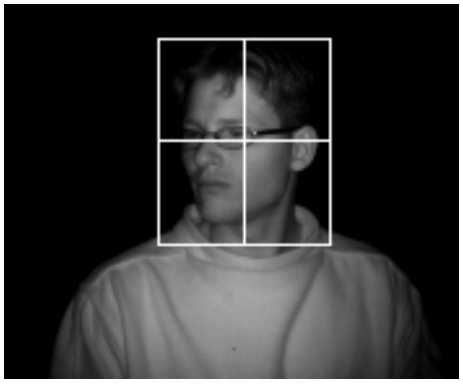

Figure 5: Example facial region-of-interest

4 Nose Tracking

Once the acquired data has been preprocessed, the central feature must be found in order to localise the models in preparation for orientation calculations. The rationale for choosing the nose tip as the central feature was discussed in Section 2. As Gorodnichy points out in [2002], the nose tip can be robustly detected in an amplitude image by assuming it is surrounded by a spherical Lambertian surface of constant albedo. However, by

using a SwissRanger sensor, we have an added advantage. Since the amplitude image is illuminated solely by the integrated infrared light source, we don't need to calculate complex reflectance maps to handle differing angles of illumination. We can also use additional data from the depth map such as proximity to camera and curvature (see Figure 6) to improve the search and assist with confirming the location of the nose tip. Figure 7 shows a typical frame with nose localisation data overlaid. Our preliminary results have shown that this approach is fast and robust enough to locate the nose within typical frame sequences with sufficient accuracy for our application.

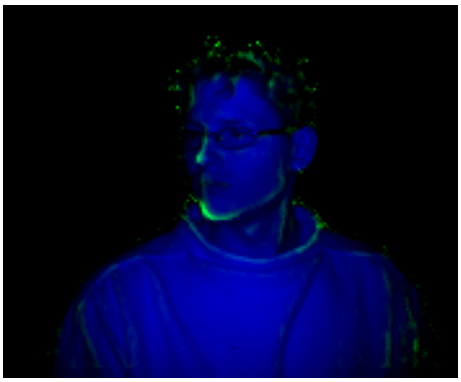

Figure 6: Amplitude image with curvature data overlaid. Greener pixels indicate higher curvature (calculated from depth map).

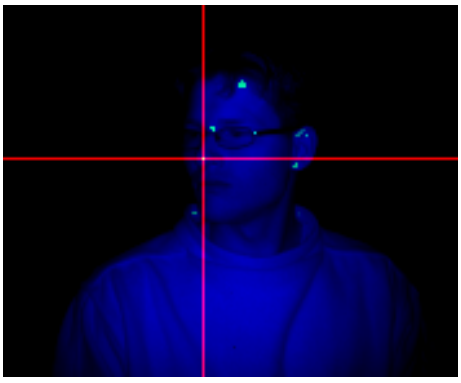

Figure 7: Sample frame with nose-tracking data displayed. Green pixels are candidate nose pixels; red cross indicates primary candidate.

5 Finding Orientation

Once the central feature (nose tip) has been located, the dimensionality of the problem has been reduced considerably by removing the translational component of the transformation. Now a single rotation about a specific axis in three-dimensional space (passing through our central feature point) will be sufficient to match the orientation of the current models to the saved target. Small errors in the initial location of the nose point can be iteratively improved using three-dimensional hill-climbing to optimise the matching of the spherical intersection profiles, as discussed in Section 2.

5.1 Spherical Intersection Algorithm

The intersection of a three-dimensional mesh with an arbitrary sphere might sound like an expensive operation to perform repeatedly, however our algorithm achieves this very efficiently (see Algorithm 1). In essence, it traverses the depth map from the centre point in the direction of the facial centroid until it finds a pair of projected pixels spanning the sphere boundary. It then adds the interpolated intersection point to a vector (SIP) and continues traversing the depth map along the intersection boundary until the end-points are found or the loop is closed. The average execution time of the algorithm on the sample sequence shown in the figures and Video 1 [Meers, 2008] on a dual-core 1.8GHz processor was $140\mu s$. The interpolation allows the profiles to be calculated with sub-pixel accuracy. Super-sampling of four-pixel groups was used to reduce the influence of noise.

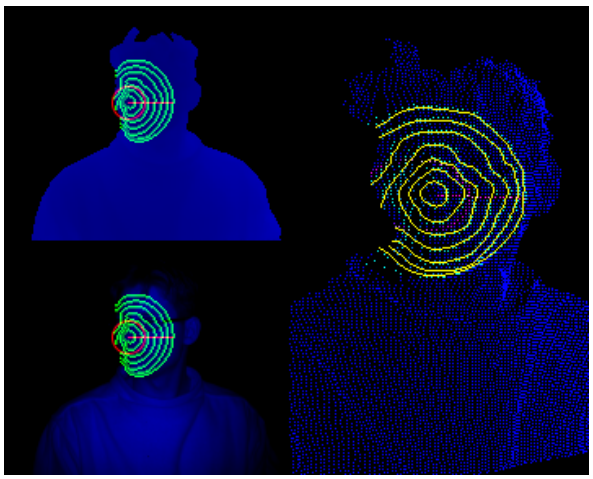

Figure 8: Example spherical intersection profiles overlaid on depth (top-left), amplitude (bottom-left) and 3D point cloud (right).

5.2 Profile Matching Algorithm

The profile matching algorithm is not yet fully implemented. The gaze calculation in Video 1 [Meers, 2008] is a crude approximation based on a least-squares fit of a line through the central feature and the point cloud formed by taking the midpoint of the latitudinal extrema

Algorithm 1 Find intersection profile of projected depth map with sphere of radius *radius* centred on pixel projected from $depthMap[noseRow][noseCol]$

```
r ⇐ noseRow, c ⇐ noseCol
if noseCol > faceCentroidCol then
  direction ⇐ LEFT
else
  direction ⇐ RIGHT
end if
found ⇐ false {No intersection found yet}
centre ⇐ projectInto3d(depthMap, r, c)
innerPoint ⇐ centrePoint
while r and c are within region of interest, and
distance(inner, centre) < radius do
  (r, c) ⇐ translate(r, c, direction)
  outerPoint ⇐ projectInto3d(depthMap, r, c)
  if distance(outer, centre) > radius then
    found ⇐ true
  end if
end while
if Not found then
  return No intersection
end if
for startDirection = UP to DOWN do
  direction ⇐ startDirection
  while Loop not closed do
    (r2, r2) ⇐ translate(inner.r, inner.c, direction)
    inner2 ⇐ projectInto3d(depthMap, r2, c2)
    (r2, c2) ⇐ translate(outer.r, outer.c, direction)
    outer2 ⇐ projectInto3d(depthMap, r2, c2)
    if inner2 or outer2 are invalid then
      break
    else if distance(inner2, centre) > radius then
      outer ⇐ inner2
    else if distance(outer2, centre) < radius then
      inner ⇐ outer2
    else
      inner ⇐ inner2
      outer ⇐ outer2
    end if
    id ⇐ distance(inner, centre)
    t ⇐ (radius − id)/((distance(outer, centre) − id)
    if startDirection = UP then
      Append (inner + (outer − inner) × t) to SIP
    else
      Prepend (inner + (outer − inner) × t) to SIP
    end if
    Update direction
  end while
end for
return SIP
```

of each intersection profile. Yet it can be seen that even this provides a relatively accurate gaze estimate.

In the final system, it is envisaged that the latitudinal extrema of each profile will be used only to provide an initial alignment for each profile pair, after which a new algorithm will measure the fit and optimise it in a manner similar to ICP [Zhang, 1994]. The fit metric provided by this algorithm will be used in the hill-climbing-based optimisation of the central point (nose tip).

It is advantageous that each profile pair should lead us to the same three-dimensional axis and magnitude of rotation required to align the model. Thus the resultant collection of rotational axes provides us with a redundant approximation of the rotation required to align the entire model. This can be analysed to remove outliers, etc, and then averaged to produce the best approximation of the overall transformation.

6 Building a Mesh

Once the current frame has been aligned with the target 3D model, any additional information can be used to evolve the 3D model. For example, regions which were originally occluded in the target model may now be visible, and can be used to extend the model. Thus a near-360° model of the user's head can be obtained over time. Adjacent polygons with similar normals can be combined to simplify the mesh. Areas with higher curvature can be subdivided to provide greater detail. For each point in the model, we maintain a running average of the closest points in the frames. This can be used to push or pull points which stray from the average and allow the mesh to 'evolve' and become more accurate with each additional frame. The contribution of a given frame to the running averages is weighted by the quality of that frame, which is simply a measure of depth map noise combined with the mean standard deviation of the intersection profile rotational transforms. Furthermore, a running standard deviation can be maintained for each point to allow the detection of noisy regions of the model, such as hair, glasses which might reflect the infrared light, facial regions subject to expression variation, etc. These measures of rigidity can then be used to weight regions of the intersection profiles to make the algorithm tolerant of fluid regions, and able to focus on the rigid regions. Non-rigid regions on the extremities of the model can be ignored completely. For example, the neck will show up as a non-rigid region due to the fact that its relation to the face changes as the user's head pose varies.

7 Face Recognition

The facial model is maintained within the system, and dynamically evolves as described in Section 6. When a

different user is presented to the system, it will be found that the intersection profiles for this user differ substantially from the current model, even in stable motionless frames. Therefore a new 3D model will subsequently be created and evolved. Each 3D model is saved in a database, together with a cached set of spherical intersection profiles. Thus, the system could potentially be used to store hundreds of thousands of users, and automatically recognise returning users. The system could also optimise the search by initially comparing only a subset of intersection profiles for each user, and only perform further comparisons if the initial metric indicates that this model is a potential match.

8 Conclusions

This paper describes a novel 3D time-of-flight camera system and method for robust and accurate head-pose tracking and identification of a user or subject. Although additional work is needed before our system is ready to apply to head pose tracking or facial recognition applications, our preliminary experiments have shown that our system can enable 3D time-of-flight cameras to locate the tip of the nose, construct a three-dimensional model of the face and obtain spherical intersections of the face in real time. Our experiments also show that the resulting spherical intersections and 3D model are suitable for accurately tracking the head pose of a user or subject and capable of performing facial recognition.

References

[Birchfield, 1998] S. Birchfield. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 232–237, 1998.

[Gorodnichy, 2002] D.O. Gorodnichy. On importance of nose for face tracking. In *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG 2002)*, pages 20–21, 2002.

[Haker *et al.*, 2007] M. Haker, M. Bohme, T. Martinetz, and E. Barth. Geometric invariants for facial feature tracking with 3d tof cameras. *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*, 1:1–4, July 2007.

[Heinzmann and Zelinsky, 1998] J. Heinzmann and A. Zelinsky. 3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm. *Proc. of the Int. Conf. on Automatic Face and Gesture Recognition*, pages 142–147, 1998.

[Hong and Park, 2005] S.K. Hong and C.G. Park. A 3D Motion and Structure Estimation Algorithm for Optical Head Tracker System. *American Institute of Aeronautics and Astronautics: Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[Meers *et al.*, 2006] S. Meers, K. Ward, and I. Piper. Simple, robust and accurate head-pose tracking with a single camera. In *The Thirteenth Annual Conference on Mechatronics and Machine Vision in Practice*, December 2006.

[Meers, 2008] S. Meers. Demonstration video, 2008. `http://acra08.simonmeers.com`.

[MESA Imaging, 2006a] MESA Imaging. SwissRanger SR-3000 Manual, Version 1.02, 2006.

[MESA Imaging, 2006b] MESA Imaging. SwissRanger SR3000 - miniature 3D time-of-flight range camera, 2006. `http://www.mesa-imaging.ch/prodview3k.php`.

[NaturalPoint Inc., 2006] NaturalPoint Inc. TrackIR, 2006. `http://www.naturalpoint.com/trackir`.

[Poston, 2000] Alan Poston. Department of defense human factors engineering technical advisory group (dod hfe tag), 2000. Human Engineering Design Data Digest – `http://hfetag.dtic.mil/hfs_docs.html`, Accessed June 24, 2008.

[Seeing Machines, 2008a] Seeing Machines. faceAPI, 2008. `http://www.seeingmachines.com/faceAPI.html`.

[Seeing Machines, 2008b] Seeing Machines. FaceLAB: A face and eye tracking system, 2008. `http://www.seeingmachines.com`.

[Toyama, 1998] K. Toyama. Look, ma-no hands! hands-free cursor control with real-time 3d face tracking. *Proc. Workshop on Perceptual User Interfaces (PUI98)*, 1998.

[Yin and Basu, 2001] L. Yin and A. Basu. Nose shape estimation and tracking for model-based coding. *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, 3, 2001.

[Zhang, 1994] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.