An automated calibration method for non-see-through head mounted displays

Stuart J. Gilson[1], **Andrew W. Fitzgibbon**[2] **and Andrew Glennerster**[3]
[1]Department of Physiology, Anatomy and Genetics, University of Oxford, Oxford, UK
[2]Microsoft Research Ltd., Cambridge, UK
[3]School of Psychology and Clinical Language Sciences, University of Reading, Reading, UK

Abstract—*Accurate calibration of a head mounted display (HMD) is essential both for research on the visual system and for realistic interaction with virtual objects. Yet, existing calibration methods are time consuming and depend on human judgements, making them error prone. The methods are also limited to optical see-through HMDs. Building on our existing HMD calibration method [1], we show here how it is possible to calibrate a non-see-through HMD. A camera is placed inside an HMD displaying an image of a regular grid, which is captured by the camera. The HMD is then removed and the camera, which remains fixed in position, is used to capture images of a tracked calibration object in various positions. The locations of image features on the calibration object are then re-expressed in relation to the HMD grid. This allows established camera calibration techniques to be used to recover estimates of the display's intrinsic parameters (width, height, focal length) and extrinsic parameters (optic centre and orientation of the principal ray). We calibrated a HMD in this manner in both see-through and in non-see-through modes and report the magnitude of the errors between real image features and reprojected features. Our calibration method produces low reprojection errors and involves no error-prone human measurements.*

Keywords: head mounted display; non-see-through; calibration; photogrammetry

1. Introduction

A head mounted display (HMD) can be modelled in a similar way to a conventional camera. Like a camera, it has both *intrinsic* parameters – focal length, aspect ratio, centre pixel (also called a frustum) – and *extrinsic* parameters – position of the optic centre and orientation of the principal ray. All 3D rendering systems include the notion of a software camera (or frustum) at their core, and this frustum must be configured with the same parameters as the physical display in order to render the scene correctly.

The issue, then, is to find the intrinsic and extrinsic parameters for each eye's display in a HMD. HMD manufacturer specifications tend to be inadequate for this task, so the only other solution is to attempt to measure these display properties. Unlike calibrating a monitor display, it is usually difficult to get sufficient physical access to a HMD display in order to make accurate measurements. Instead, we describe here a method based on photogrammetry (camera calibration) techniques.

HMD displays fall into two categories: see-through and non-see-through. Of the see-through variety, there are two sub-categories: optical-see-through and video-see-through. Video-see-through displays are very popular in augmented reality applications, where a video camera mounted within the HMD sends digitized images of the real world to the graphics computer, which can then overlay computer graphics onto the images before sending them to the HMD to be displayed to the observer. Such displays are generally straight-forward to calibrate [2], [3], since the issue of calibrating a conventional camera is well understood [4]. However, the optic centre of the camera is not at the observer's eye, and the resulting calibrated display will differ from that which the observer would see if they removed the HMD. For some applications, this discrepancy is acceptable (e.g. navigation, gaming, architectural walk-throughs), while for other applications involving interaction with real and virtual objects the offset between hand and eye may be important.

Optical-see-through displays generally use a half-silvered mirror placed in front of the observer's eyes, with a display device (cathode-ray tube or liquid crystal) mounted on the HMD. The half-silvered mirror permits rays of light from the real world to reach the observer, while also reflecting images from the display device. The observer sees a composite of the two sources, but with several limitations. Notably, the computer graphics (CG) image is effectively blended with the real world image and, as such, can never completely obliterate the real world. Hence, making virtual objects occlude real ones is impossible. Also, dark details in the CG image will become washed out by bright areas of the real world. Of more relevance here, is that there is no digital record of the real world and, so, existing camera calibration methods cannot be used. Yet, without an accurate calibration, virtual objects will register poorly with real objects, making optical-see-through a poor choice for augmented reality.

Non-see-through HMDs usually place the display device directly in front of the observer's eye, and are thus optically much simpler than either of the other two types of HMD.

This does not make them any easier to calibrate, though. While there is no real world visible to the observer, and so registering virtual objects with real world ones is not an issue, a correct calibration is still important. Failure to calibrate correctly can lead to observers misinterpreting the virtual world (for example, they often underestimate distances to objects which can be a symptom of inadequate calibration [5]). Inadequate calibration can also lead to users experiencing premature fatigue and possible onset of nausea.

Thus, there is a demand for calibrating both optical-see-through and non-see-through HMDs. To our knowledge, there are no quantitative methods for calibrating a non-see-through HMD [5]. Optical see-through HMDs have received more attention in calibration research, probably because of the ability to see the real world and CG image simultaneously. The most extensively covered calibration methods, SPAAM (single point active alignment method), uses a human observer to calibrate the display by wearing the HMD and positioning their head in order to align HMD image points with the known locations of real world objects [6]. When this alignment is achieved, the HMD position and pose is recorded from the tracker, and the procedure is repeated with more image/world coordinate pairs until sufficient data has been gathered to estimate the frustum parameters. This is a time-consuming process, requiring a skilled observer to make numerous, potentially erroneous judgements. Also, there can be high variability in the results due to the difficulty in performing such an alignment task with a free-moving head, and the problem of registering virtual and real coordinates at different distances (and, hence, with different accommodative demands).

Owen *et al.*[7] used a calibration method similar to ours. They placed a camera inside the HMD and recorded images of a CG calibration grid, which they use to map between camera and HMD image coordinates. They then synthesized calibration data for a subsequent conventional camera calibration procedure. They recommend that any HMD users perform additional fine-tuning of the calibration using a SPAAM-like procedure.

Our method has several distinct advantages over that of Owen *et al*'s. Firstly, we used tracked real world objects for our calibration data (instead of synthesized data) which had the advantage that all subsequent parameters generated by the calibration method were in the coordinate frame of the tracker, avoiding any need for explicit physical measurements. Using tracked real world objects also meant we could compute their 2D projections on the HMD image and compare those re-projections with their actual projections recorded by the camera. This gave a useful, quantitative measure of calibration accuracy. Additionally, our method allows us to include non-linear optical distortions (such as radial and tangential distortions) in our solution, although we do not include such distortions here (but see [8]). Finally, we have found no need to perform the manual "phase II" step

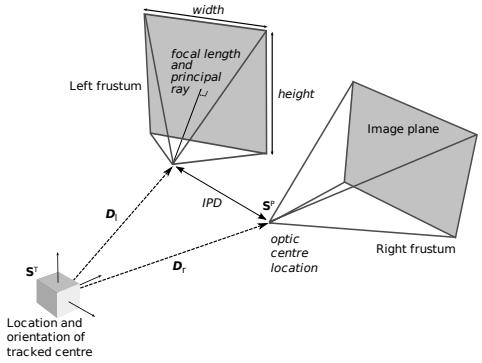

Fig. 1: Calibration aims to find values for the display frustums, and also their position and orientation with respect to the HMD's tracked centre.

described in Owen's paper – our calibration worked well for all tested users and we thus avoid this human-centered and error-prone aspect.

2. Methods

Our aim was to find estimates of the intrinsic and extrinsic matrices which define the HMD display (shown pictorially in figure 1). The intrinsic matrix comprises the focal length (f, in both horizontal and vertical directions, thus denoting aspect ratio), centre pixel location (c) and the near- and far-clipping planes (which are application specific and not covered here):

$$\mathbf{P} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & \text{near} & \text{far} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The extrinsic matrix describes the principal ray of the camera as a rotation and translation:

$$\mathbf{S}^P = \begin{bmatrix} \mathbf{R} & \mathbf{T}^{'} \\ 0 & 1 \end{bmatrix} \tag{2}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{T}$ is a 1×3 translation matrix.

We calibrated an nVision datavisor 80 HMD, which has an optical see-through mode, but can be made completely non-see-through by attaching felt 'socks' over the HMD optics. A camera (AVT Pike, 1388×1038 pixels resolution, ≈65° FOV) was placed on a tripod and positioned beneath the

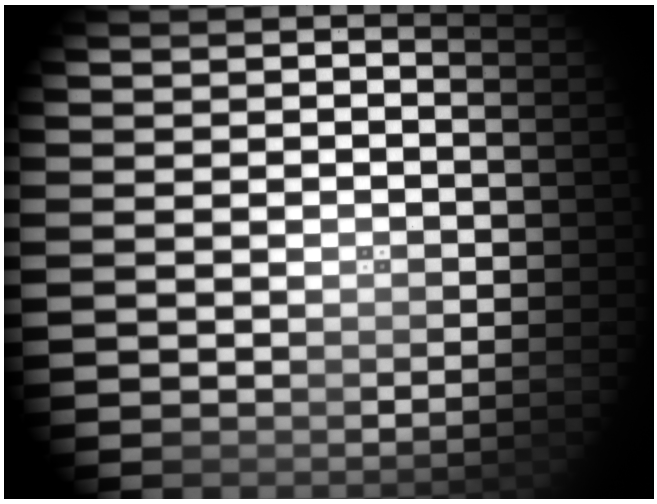

Fig. 2: Chequerboard pattern displayed in HMD and captured by the camera. Note spots indicating the logical centre of HMD display.

Fig. 3: Image of Tsai grid captured by the camera.

independently supported HMD in such a way that it could capture as much of the HMD image as possible. The location and pose of the HMD, $\mathbf{S}^T$, was recorded using a 6 degree-of-freedom real time optical tracking system (Vicon Motion Systems MX3). The HMD displayed a simple chequerboard pattern (41×41 squares) and an image of this was captured using the camera (figure 2). The vertices of the chequerboard in this image were extracted using image processing. Using a salient feature in the middle of the HMD image, we were able to relate the known vertices of the HMD chequerboard to corresponding vertices in the camera image.

This allowed us to generate a mapping between camera image and HMD coordinates. If the HMD vertices are denoted by:

$$\mathbf{g}^{HMD} = \{(x_i^{HMD}, y_i^{HMD}) \,|i = 1...441\} \tag{3}$$

where x_i^{HMD} and y_i^{HMD} are HMD coordinates, then, for each vertex i there exists:

$$\mathbf{g}^{CAM} = \{(x_i^{CAM}, y_i^{CAM}) \,|i = 1...441\} \tag{4}$$

where x_i^{CAM} and y_i^{CAM} are the coordinates of the corresponding vertex in camera coordinates. Now any camera coordinate can be converted to a HMD coordinate using interpolation. If $\mathbf{x}_j^{CAM}$ denotes a point in the jth camera image, then we found the smallest triangle of the chequerboard that encompassed it, whose vertices we denote $\mathbf{g}_i$, $\mathbf{g}_h$ and $\mathbf{g}_v$. We then used linear interpolation to re-express the coordinate in HMD coordinates using the basis vectors $(\mathbf{g}_h^{CAM} - \mathbf{g}_i^{CAM})$ and $(\mathbf{g}_v^{CAM} - \mathbf{g}_i^{CAM})$ and their equivalents $(\mathbf{g}_h^{HMD} - \mathbf{g}_i^{HMD})$ and $(\mathbf{g}_v^{HMD} - \mathbf{g}_i^{HMD})$ respectively. Expressed in terms of these basis vectors, $\mathbf{x}_j^{CAM}$ and $\mathbf{x}_j^{HMD}$ are equivalent points [1].

We removed the HMD from the camera and – crucially – ensured that the camera did not move. We then captured 10 images of a Tsai-grid [9] (with 25mm spacing, see figure 3), with the grid moved and re-oriented between images. The Tsai grid had reflective markers attached which allowed it to be tracked by the tracking system. For each image, the location and pose of the Tsai grid was recorded by the tracking system. The corners of the Tsai grid were extracted from these images, to give a set of 2D projections in camera coordinates, $\mathbf{x}_j^{CAM}$.

Our method must also know the 3D location of the Tsai-grid vertices. However, the tracking system only reported the position of the geometric centre of the reflective markers attached to the Tsai-grid. We thus synthesized a set of 3D coordinates with an appropriate spacing and position relative to the reflective markers, and then transformed this plane by the rotation and translation reported by the tracker. Thus, for each image we had a set of world coordinates $\mathbf{X}_j$ (x, y and z triples) and a corresponding set of projections, $\mathbf{x}_j^{CAM}$.

Before these coordinates could form the input to the camera calibration routine, all 2D image locations were transformed into HMD coordinates using the basis vectors described above. This was a critical step, since without this, the subsequent photogrammetry would produce an intrinsic model of the camera, not the HMD.

We estimated initial values for the intrinsic and extrinsic matrices by finding a single homography that mapped each $\mathbf{x}_j^{HMD}$ onto the corresponding $\mathbf{X}_j$ for each image. The resulting estimates for focal length, aspect ratio, centre pixel, optic centre location and principal ray direction were then used as a starting point for a simplex minimization [10]. The cost function was the reprojection error – that is, the root-mean-square (RMS) difference between the original Tsai-grid projections $\mathbf{x}^{HMD}$ and the new projections computed by:

$$(x_j, y_j, z_j, w_j) = \mathbf{P}\mathbf{S}^P[\mathbf{X}_j 1]' \quad (5)$$

The extrinsic matrix $\mathbf{S}^P$ defines the location and pose of the camera. We can use this information to find the HMD display optic centre and principal ray by finding the difference between this matrix and the matrix representing the HMD location and pose recorded from the tracker:

$$\mathbf{D} = \mathbf{S}^P \times \text{inv}\left(\mathbf{S}^T\right) \quad (6)$$

This simplicity arises from $\mathbf{S}^P$ and $\mathbf{S}^T$ being in the same coordinate frame. We now have a projection matrix (of the HMD display) and a modelling matrix which can be used directly in a 3D programming language such as OpenGL by post-multiplying it with the modelling matrix from the tracker:

$\mathbf{D}$ can be decomposed into its rotational and translational components. The component of translation along the interocular axis can be explicitly manipulated to suit different inter-pupillary distances for different users.

```
// Switch to projection matrix
glMatrixMode(GL_PROJECTION);

// Load intrinsic matrix, P
glLoadMatrix(P);

// Switch to modelling matrix
glMatrixMode(GL_MODELVIEW);

// Load tracker_to_HMD transform
glLoadMatrix(D);

// Incorporate tracker transform
glMultMatrix(S_T);
```

This procedure was repeated for both eyes' displays in the binocular HMD. Each display was calibrated independently – we have found no need to perform an explicit stereo calibration.

To test the generalization of any given calibration solution, we collected four data sets for each HMD display. Between each data set, both the HMD and camera were moved to a new location within the tracked volume. Thus, the position of the camera relative to the HMD also changed, within the limits imposed by the constraint that the camera image should capture as much of the HMD display as possible. Calibration accuracy was quantified as the RMS error measured for all features across all images in the data set. If the computed calibration was too specific to the training images, then it would produce reprojections with high RMS errors in the test images. Conversely, a good generalized calibration would perform comparably well across all the test images.

It may seem counter-intuitive to use reprojections as an error measure here, since a non-see-through HMD has no real-world image in which to make such reprojections. The important point, of course, is that the camera did not move between capturing the HMD chequerboard and the Tsai-grid images and, thus, each camera pixel corresponded to the same ray irrespective of whether the HMD was present or not. Because we have used a HMD with both see-through and non-see-through modes, we collected see-through data in order to provide a direct comparison of both calibration methods.

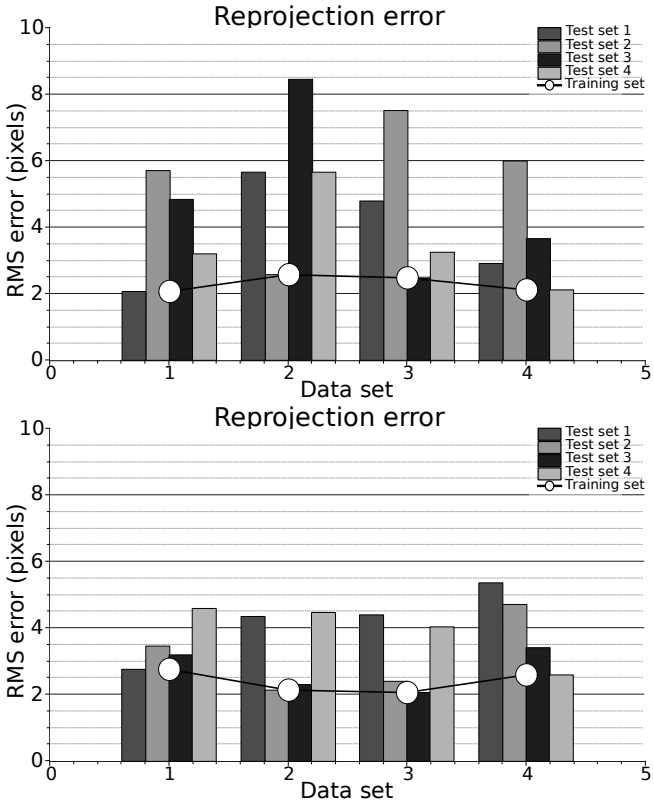

Fig. 4: Reprojection errors for non-see-through calibration (left and right displays). Grey bars show errors for each calibration when tested on the other three ('test') data sets. The black line shows the native error for a calibration tested on it's own training images.

3. Results

We first present calibration data for the non-see-through mode. Figure 4 shows reprojection errors for four 'test' data sets for each display, i.e. when the calibration is tested using images other than those used for generating the calibration. The mean error per feature averaged across all data sets is 3.9 pixels for the left display and 5.9 pixels for the right display. We also show the minimized reprojection error for each training data set, i.e. that obtained during each calibration (2.4 pixels for left display and 2.6 pixels for right display).

Figure 5 shows the analogous results obtained for the HMD calibrated in see-through mode. Mean RMS pixel error test images for the left and right displays were 5.1 and 5.7 respectively, while errors for training images were 2.3 pixels

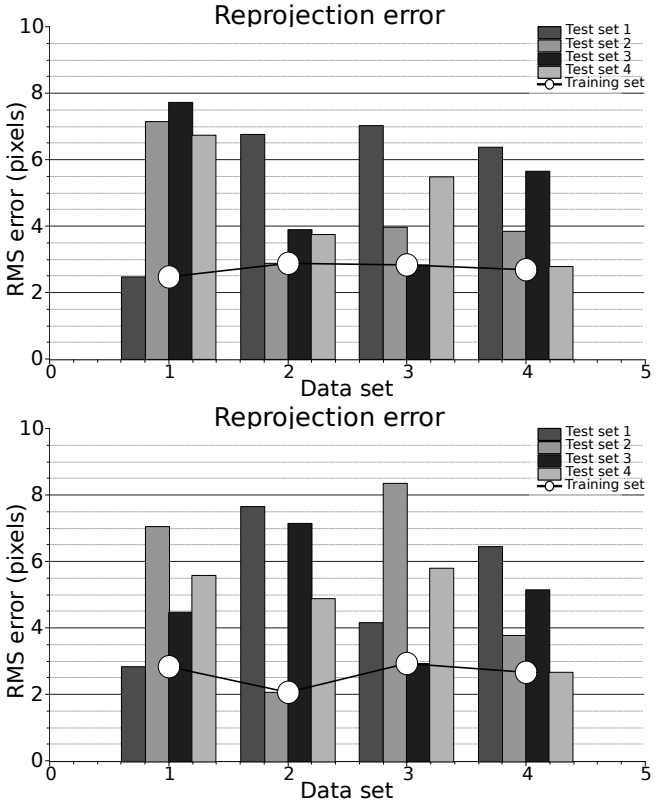

Fig. 5: Reprojection errors for see-through calibration (left and right displays).

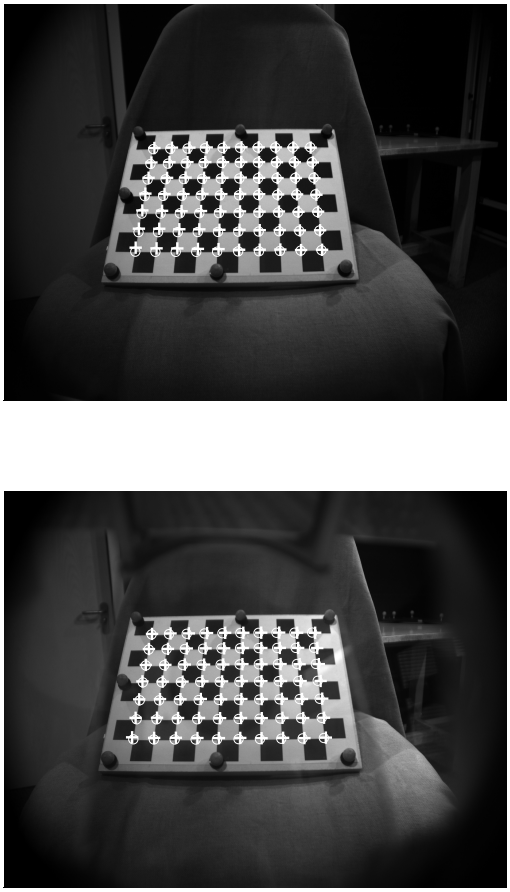

Fig. 6: Reprojections (white crosses) and original image features (white circles) for non-see-through (top) and see-through (bottom) calibration modes. The RMS errors in these two images are 4.2 pixels (calibration from data set 2, tested on an image from data set 1, figure 4) and 5.2 pixels (calibration from data set 2, tested on an image from data set 1, figure 5) respectively.

and 2.7 respectively[1].

For the left display, using the see-through method, we captured a further four data sets without moving either the HMD or the camera between sets (data not shown). We obtained a mean RMS error of 3.0±0.1 pixels for the four test data sets, and 2.2±0.1 pixels for the four training sets. These data allow us to partition the errors in figure 5, at least approximately, into: a baseline measure of the extent of error (approximately 2 pixels); an estimate of the additional error that is introduced by testing with a different set of images than those used in the calibration, when other factors remain constant (approximately 1 pixel); and the additional error introduced by moving the HMD and camera between collection of data sets (approximately a further 1-2 pixels, to bring the total RMS error to values reported in figure 4).

Figure 6 shows the Tsai-grid vertices re-projected back into the same images as figure 3 for both non-see-through and see-through calibration modes.

[1] In [1], there was a mistake in the reported magnitude of RMS errors. Figure 5 in the present work shows similar data but with the correct magnitude.

4. Conclusion

We have shown here a calibration method for non-see-through HMDs, which is based on our existing method for optical see-through HMDs [1]. Unlike existing calibration methods [6], [7], ours yields a quantitative measure of calibration accuracy. Our method produced mean reprojection errors in the order of 5 pixels, a similar magnitude to that found for see-through mode.

Acknowledgement

This work was funded by the Wellcome Trust.

References

[1] S. J. Gilson, A. W. Fitzgibbon, and A. Glennerster, "Spatial calibration of an optical see-through head-mounted display," *Journal of Neuroscience Methods*, vol. 173, no. 1, pp. 140 – 146, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6T04-4SM1T95-1/2/fd29f968171f9401edbd5c0cb1457230

[2] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers, "Calibration requirements and procedures for a monitor-based augmented reality system," *IEEE Trans. Vis. Comput. Gr.*, vol. 1, pp. 255–273, 1995.

[3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Comput. Graph.*, vol. 21, pp. 34–47, 2001.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, UK, 2001.

[5] S. A. Kuhl, W. B. Thompson, and S. H. Creem-Regehr, "HMD calibration and its effects on distance judgments," in *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization*. New York, NY, USA: ACM, 2008, pp. 15–22.

[6] M. Tuceryan, Y. Genc, and N. Navab, "Single point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality," *Presence-Teleop. Virt.*, vol. 11, pp. 259–276, 2002.

[7] C. B. Owen, J. Zhou, A. Tang, and F. Xiao, "Display-relative calibration for optical see-through head-mounted displays," in *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2004.

[8] S. J. Gilson, A. W. Fitzgibbon, and A. Glennerster, "Extensions to a fully automatic head mounted display calibration method," *Journal of Vision*, 2009.

[9] R. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision." *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 364–374, 1986.

[10] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, pp. 112–147, 1998.