MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition

Rogerio Feris, Matthew Turk, Ramesh Raskar, Kar-Han Tan, Gosuke Ohashi

TR2004-099 June 2004

Abstract

We present a novel method for automatic fingerspelling recognition which is able to discriminate complex hand configurations with high amounts of finger occlusions. Such a scenario, while common in most fingerspelling alphabets, presents a challenge for vision methods due to the low intensity variation along important shape edges in the hand image. Out approach is based on a simple and cheap modification of the capture setup: a multi-flash camea is used with flashes strategically positioned to cast shadows along depth discontinuities in the scene, allowing efficient and accurate hand shape extraction. We then use a shift and scale invariant shape descriptor for fingerspelling recognition, demonstrating great improvement over methods that rely on features acquired by traditional edge detection and segmentation algorithms.

RTV4HCI 2004

Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition

Rogerio Feris[1] Matthew Turk[1] Ramesh Raskar[2] Karhan Tan[3] Gosuke Ohashi[4]

[1]University of California, Santa Barbara
{rferis,mturk}@cs.ucsb.edu

[2]Mitsubishi Electric Research Labs
raskar@merl.com

[3]University of Illinois Urbana-Champaign
tankh@vision.ai.uiuc.edu

[4]Shizuoka University
tegooha@ipc.shizuoka.ac.jp

Abstract

We present a novel method for automatic fingerspelling recognition which is able to discriminate complex hand configurations with high amounts of finger occlusions. Such a scenario, while common in most fingerspelling alphabets, presents a challenge for vision methods due to the low intensity variation along important shape edges in the hand image. Our approach is based on a simple and cheap modification of the capture setup: a multi-flash camera is used with flashes strategically positioned to cast shadows along depth discontinuities in the scene, allowing efficient and accurate hand shape extraction. We then use a shift and scale invariant shape descriptor for fingerspelling recognition, demonstrating great improvement over methods that rely on features acquired by traditional edge detection and segmentation algorithms.

1. Introduction

Sign language is the primary communication mode used by most deaf people. It consists of two major components: 1) word level sign vocabulary, where gestures are used to communicate the most common words and 2) fingerspelling, where the fingers on a single hand are used to spell out more obscure words and proper nouns, letter by letter. Facial expressions can also be employed to distinguish statements, questions and directives.

Over the past decade, great effort has been made to develop systems capable of translating sign language into speech or text, aiming to facilitate the interaction between deaf and hearing people. Extensive research has been done in both word level and fingerspelling components.

Previous approaches to word level sign recognition rely heavily on statistical models such as Hidden Markov Models (HMMs) [17, 18, 4]. Excellent recognition rates were obtained for small word lexicons, but scalability is still an issue for glove-free sign recognition. For fingerspelling recognition, most successful approaches are based on instrumented gloves, which provide information about finger positions. Lamar and Bhuiyant [8] achieved letter recognition rates ranging from 70% to 93%, using colored gloves and neural networks. More recently, Rebollar et al. [14] used a more sophisticated glove to classify 21 out of 26 letters with 100% accuracy. The worst case, letter 'U', achieved 78% accuracy.

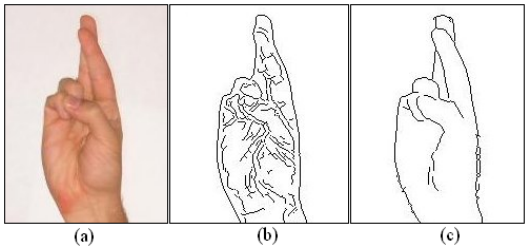

Figure 1: *(a) Letter 'R' in ASL alphabet. (b) Canny edges. Note that important internal edges are missing, while edges due to wrinkles and nails confound scene structure. (c) Depth edges obtained with our multi-flash technique.*

In general, non-intrusive vision-based methods, while useful for recognizing a small subset of convenient hand configurations [7, 1], are limited to discriminate configurations with high amounts of finger occlusions - a common scenario in most fingerspelling alphabets. In such cases, traditional edge detectors or segmentation algorithms fail to detect important internal edges along the hand shape (due to the low intensity variation in skin-color), while keeping edges due to nails and wrinkles, which may confound scene structure and the recognition process (see Figure 1b). Also, some signs might look very similar to each other, with small differences on finger positions, thus posing a problem for appearance-based approaches [7].

We address this problem by using a technique we have recently proposed for conveying shape in non-photorealistic rendering [13]. Our approach is based on a simple and cheap modification of the capture setup: a multi-flash camera is used with flashes strategically positioned to cast shadows along depth discontinuities in the scene, allowing efficient and accurate hand shape extraction, as shown in Figure 1c. Our method was also extended to handle dynamic scenes, being suitable for real-time processing.

We show that depth discontinuities (aka depth edges)

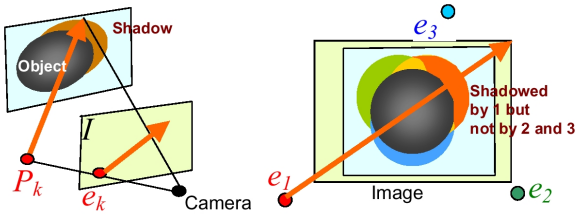

Figure 2: *Imaging geometry. Shadows of the gray object are created along the epipolar ray. We ensure that depth edges of all orientations create shadow in at least one image while the same shadowed points are lit in some other image.*

may be used as a signature to reliably discriminate among complex hand configurations in the ASL alphabet, which would not be possible with current glove-free vision methods. For classification, we have used a shape descriptor similar in spirit to shape context matching [2], which is invariant with respect to image translation and scaling.

The remaining of this paper is organized as follows: in Section 2 we describe our multi-flash technique for extraction of depth edges. Section 3 covers our shape descriptor and classification method. We report our experimental results in Section 4 and discuss issues and related work of our tecnique in Section 5. Section 6 describes our implementation details. Finally, conclusions and future work are addressed in Section 7.

2. Multi-flash Imaging

The technique for detecting shape features in images was recently described in [13], for non-photorealistic rendering. For completeness we review the basic idea here.

The method is motivated by the observation that when a flashbulb (*close* to the camera) illuminates a scene during image capture, thin slivers of cast shadow are created at depth discontinuities. Moreover, the position of the shadows is determined by the relative position of the camera and the flashbulb: when the flashbulb is on the right, the shadows are create on the left, and so on. Thus, if we can shoot a sequence of images in which different light sources illuminate the subject from various positions, we can use the shadows in each image to assemble a depth edge map using the shadow images.

2.1 Imaging Geometry

In order to capture the intuitive notion of how the position of the cast shadows are dependent on the relative position of the camera and light source, we examine the imaging geometry, illustrated in Figure 2. Adopting a pinhole camera model, the projection of the point light source at P_k is at pixel e_k on the imaging sensor. We call this *image* of the light source the *light epipole*. The images of (the infinite set

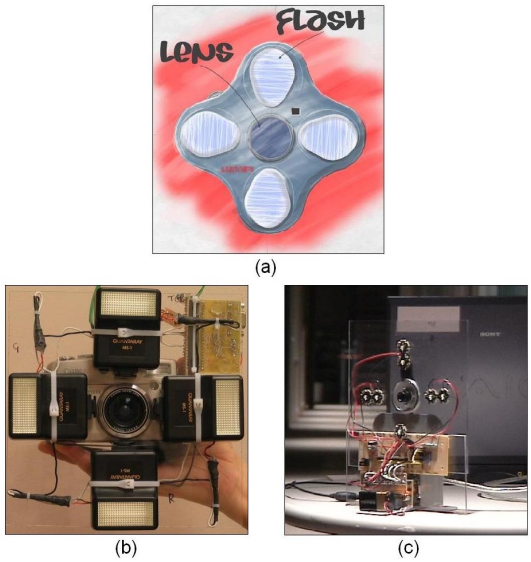

Figure 3: *(a) Our prototype to capture depth discontinuities. (b) Setup for static scenes. (c) Setup for dynamic scenes.*

of) light rays originating at P_k are in turn called the *epipolar rays*, originating at e_k. We use the terms depth discontinuities and depth edges interchangeably here.

There are two simple observations that can be made about cast shadows:

- A shadow of a depth edge pixel is constrained to lie along the epipolar ray passing through that pixel.
- When a shadow is induced at a depth discontinuity, the shadow and the light epipole will be at opposite sides of the depth edge.

These two observations suggest that if we can detect shadow regions in an image, then depth edges can be localized by traversing the epipolar rays starting at the light epipole and identifying the points in the image where the shadows are first encountered.

2.2 Removing and Detecting Shadows

Our approach for reliably removing and detecting shadows in the images is to position lights so that every point in the scene that is shadowed in some image is also captured without being shadowed in at least one other image. This can be achieved by placing lights strategically so that for every light, there is another on the opposite side of the camera to ensure that all depth edges are illuminated from two sides. Also, by placing the lights close to the camera, we minimize changes across images due to effects other than shadows.

To detect shadows in each image, we first compute a *shadow-free image*, which can be approximated with the MAX composite image, which is an image assembled by choosing at each pixel the maximum intensity value among

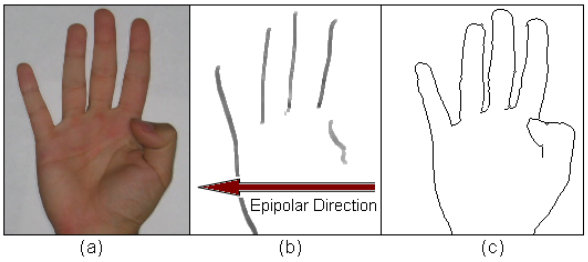

Figure 4: *Detecting depth edges. (a) Hand image. (b) Ratio image (right flash). (c) Detected edges.*

the image set. The shadow-free image is then compared with the individual shadowed images. In particular, for each shadowed image, we compute the *ratio image* by performing a pixel-wise division of the intensity of the shadowed image by the intensity of the MAX image. The ratio image is close to 1 at pixels that are not shadowed, and close to 0 at pixels that are shadowed. This serves to accentuate the shadows and remove intensity transitions due to surface material changes.

2.3 Algorithm

Codifying the ideas discussed we arrive at the following algorithm. Note that the shadowed images I_k in the algorithm below have ambient component I_0 removed, where I_0 is an image taken with only ambient light and none of the n light sources on.

Given n light sources positioned at $P_1, P_2...P_n$,

- Capture ambient image I_0
- Capture n pictures $I_{k,0}$, $k = 1..n$ with a light source at P_k
- Compute $I_k = I_{k,0} - I_0$
- For all pixels x, I_{max}(x) = $\max_k(I_k(x))$, $k = 1..n$
- For each image k,
  - ▷ Create a ratio image, R_k, where $R_k(x) = I_k(x)/I_{max}(x)$
- For each image R_k
  - ▷ Traverse each epipolar ray from epipole e_k
  - ▷ Find pixels y with step edges with negative transition
  - ▷ Mark the pixel y as a depth edge

We propose using the following configuration of light sources: four flashes at left, right, top and bottom positions (Figure 3). This setup makes the epipolar ray traversal efficient. For the left-right pair, the ray traversal is along horizontal scan lines and for the top-bottom pair, the traversal is along vertical direction. Figure 4 illustrates depth edge detection.

We have also extended our method to dynamic scenes. As in the static case, we bypass the hard problem of finding the rich per-pixel motion representation and focus directly on finding the discontinuities i.e., depth edges in motion. We refer to [13] for a description of the algorithm. The setup is similar to the static case with flashes around the camera, but triggered in a rapid cyclic sequence, one flash per frame (see Figure 3c).

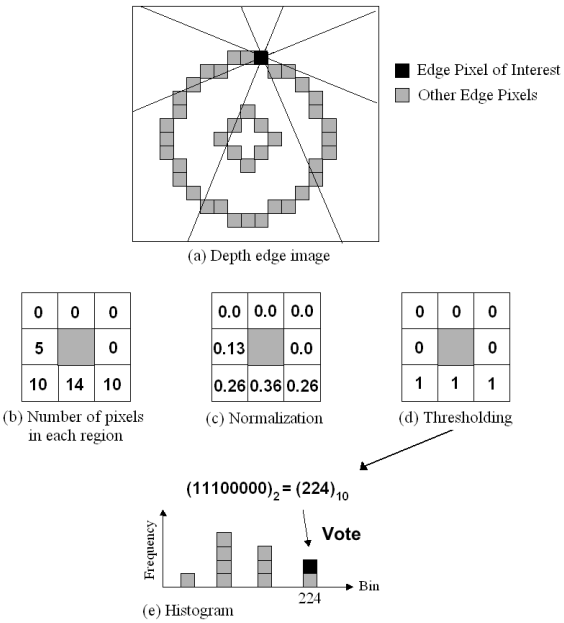

Figure 5: *Shape descriptor used for classification.*

3. Shape Descriptor and Classification

In this section, we present a shape descriptor for depth edges which is invariant with respect to image translation and scale. Our approach is simple and yet very effective. It has been recently evaluated on a large dataset for content-based image retrieval [11].

The basic idea is illustrated on Figure 5. For each edge pixel of interest, we first analyze its context by counting the number of other edge pixels in eight neighboring regions, as shown in Figure 5(a). This gives us a vector of eight elements $C_i, 1 \leq i \leq 8$ (Figure 5b). We then normalize each element for scale invariance (Figure 5c) by denoting $S_i = C_i/C$, where $C = \sum_i^8 C_i$. Finally, thresholding is applied (Figure 5d), so that each element encodes the information of either high or low density of edge pixels along a specific direction of the pixel of interest. The threshold value 0.15 is obtained empirically.

Inspired by the concept of Local Binary Patterns [12] in the field of texture analysis, the values ”0”s and ”1”s are arranged counter-clockwise from a reference region (in our example, the bottom-right region) to express an 8-bit binary number. The correspondent decimal number $d, 0 \leq d \leq 255$ is used to vote for the respective bin in the histogram

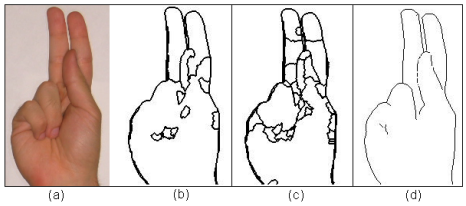

Figure 6: *(a) Letter 'K' of ASL alphabet. (b),(c) Mean Shift segmentation algorithm with different parameter settings. (d) Output of our method.*

shown in Figure 5e. A 256-dimensional feature vector is then obtained by applying the above mentioned process to all edge pixels in the depth edge image.

Since the descriptor is based on the relative position of edge pixels, it is clear that it is invariant with respect to image translation. Scale invariance is obtained in the normalization step. The descriptor can also be made rotation invariant [11]. However, this may not be appropriated for some fingerspelling alphabets (e.g., Japanese Sign Language), which might have letters that are rotated versions of the others.

We have used a nearest-neighbor technique for classification. Initially, supervised learning is carried out by acquiring a set of images for each letter in the fingerspelling alphabet. Depth edges are then extracted and the shape descriptor technique is applied, so that a training database comprised of labeled 256-dimensional feature vectors is formed. Given a test image, features are extracted and the class of the best match training sample according to Euclidean distance is reported.

4. Experiments

We compared the hand contours obtained using our technique with the output of a traditional Canny edge detector [3] and a state-of-the-art Mean Shift segmentation algorithm [5]. We refer to Figure 1 for a comparison of our method with Canny edges. Changing parameter settings in the Canny algorithm could reduce the amount of clutter, but important edges along the hand shape would still not be detected. Figure 6 shows a comparison with Mean Shift algorithm. Clearly, due to the low intensity skin-color variation in the inner hand region, the segmentation method is not able to detect important boundaries along depth discontinuities. Our method accurately locates depth edges and also offers the advantage that no parameter settings are required.

We realized that depth edges are good features to discriminate among signs of fingerspelling alphabets. Even when the signs look very similar (e.g., letters 'E','S' and 'O' in ASL alphabet), the depth edge signature is quite discriminative (see Figure 7). This poses an advantage over vision methods that rely on appearance or edge-based representations. Note that our method does not detect edges in finger boundaries with no depth discontinuity. It turns out that this is helpful to provide more unique signatures for each letter.

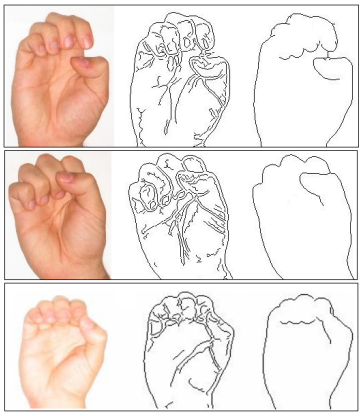

Figure 7: *From left to right: input image, Canny edges and depth edges. Note that our method misses finger boundaries due to the absence of depth discontinuities. This turns out to be helpful to provide unique signatures for each letter.*

In order to quantitatively evaluate the advantages of using depth edges as features for fingerspelling recognition, we considered an experiment with the complete ASL alphabet, except letters 'J' and 'Z', which require motion analysis to be discriminated. We collected a small set of 72 images using our multi-flash camera (three images per letter, taken at different times, with resolution 640x480). The images showed variations in scale, translation and slight variations in rotation. The background was plain, with no clutter, since our main objective is to show the importance of obtaining clean edges in the interior of the hand. It is worth mentioning that textured but flat/smooth backgrounds would not affect our method, but would make an edge detection approach (used for comparison) much more difficult.

For each image, features were extracted as described in Sections 2 and 3. For sake of comparison, we also considered shape descriptors based on Canny edges. Recognition rate was obtained using a leave-one-out scheme in the collected dataset. Our approach achieved 96% of correct matches, compared with 88% when using Canny edges.

Rebollar [14] mentioned in his work that letters 'R', 'U' and 'V' represented the worst cases, as their class distributions overlap significantly. Figure 8 shows these letters and their corresponding depth edge signatures. Note that they are easily discriminated with our technique. In the experiment described above, the method based on Canny edges fails to discriminate them.

Figure 9 shows a difficult case for traditional methods, where our method also fails to discriminate between letters 'G' and 'H'. In this particular case, we could make use of

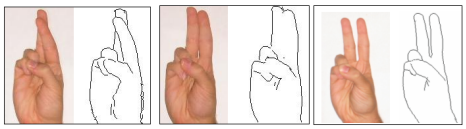

Figure 8: *Letters 'R', 'U' and 'V', the worst cases reported in [14]. Note that the use of a depth edge signature can easily discriminate them.*

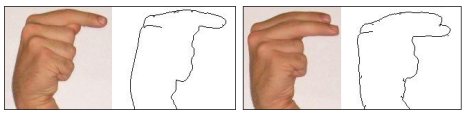

Figure 9: *A difficult case for traditional algorithms (letters 'G' and 'H'), where our method may also fail.*

additional information, such as the intensity variation that happens between the index and the middle finger in letter 'H' and not 'G'.

All the images in our experiment were collected from the same person. We plan to build a more complete database with different signers. We believe that our method will better scale in this case, due to the fact that texture edges (e.g., wrinkles, freckles, veins) vary from person to person and are eliminated in our approach. Also, shape context descriptors [2] have proven useful for handling hand shape variation from different people. For cluttered scenes, our method would also offer the advantage of eliminating all texture edges, thus considerably reducing clutter (see Figure 10)

For segmented hand images with resolution 96x180, the computational time required to detect depth edges is 4ms on a Pentium IV 3GHz. The shape descriptor computation requires on average 16ms. Thus, our method is suitable for real-time processing. For improving hand segmentation, depth edges could be computed in the entire image. In this case, the processing time for 640x480 images is 77ms.

We intend to adapt our method for continuous sign recognition in video. Demonstration of detection of depth edges in motion are showed in our previous work [13]. We are currently exploiting a frequency division multiplexing scheme, where flashes with different colors (wavelength) are triggered simultaneously. We hope this will allow for efficient on-line tracking of depth edges in sign language analysis.

5. Discussion

In this section, we discuss issues related to our method and propose ways to overcome failure situations. Then we follow with a brief discussion on related work.

There is a tradeoff in choosing the baseline distance between camera and light sources. A larger baseline is better to cast a wider detectable shadow in the internal edges of the

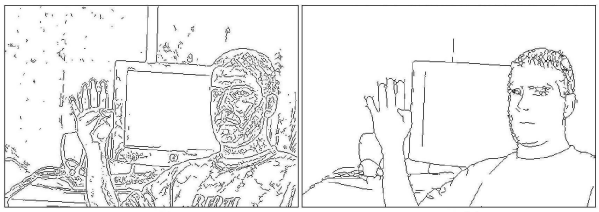

Figure 10: *(a) Canny edges (b) Depth edges. Note that our method considerably reduces the amount of clutter, while keeping important detail in the hand shape.*

hand, but a smaller baseline is needed to avoid separation of shadow from the fingers (shadow detachment) when the background is far away. The width of the abutting shadow in the image is $d = fB\,(z_2 - z_1)/(z_1 z_2)$, where f is the focal length, B is baseline in mm, and z_1, z_2 are depths, in mm, to the shadowing and shadowed edge. Shadow detachment occurs when the width, T, of the object is smaller than $(z_2 - z_1)\,B/z_2$. Fortunately, with rapid miniaturization and sophistication of digital cameras, we can choose a small baseline while increasing the pixel resolution (proportional to f), so that the product *fB* remains constant.

What if there is no cast shadows due to lack of background? In these cases only the outermost depth edge, the edge shared by foreground and distant background, is missed in our method. This could be detected with a foreground-background estimation technique. The ratio of I_0/I_{max} (image acquired with no flash over max composite of flash images), is near 1 in background and close to zero in interior of the foreground.

Another solution for both problems cited above is to consider a larger baseline and explore it to detect only internal edges in the hand, while using traditional methods (such as skin-color segmentation or background subtraction) to obtain the external hand silhouette.

We noticed that depth edges might appear or disappear with small changes in viewpoint (rotations in depth). This was in fact explored in the graphics community with the concept of *suggestive contours* [6]. We believe this may be a valuable cue for hand pose estimation [1].

A common thread in recent research on pose estimation involves using a 3D model to create a large set of exemplars undergoing variation in pose, as training data [16, 1]. Pose estimation is formulated as an image retrieval problem in this dataset. We could use a similar approach to handle out-of-plane hand rotations. In this case, a 3D hand model would be used to store a large set of depth edge signatures of hand configurations under different views.

5.1. Related Work

Shadows have already been exploited for gesture recognition and interactive applications. Segen and Kumar [15] de-

scribes a system which uses shadow information to track the user's hand in 3D. They demonstrated applications in object manipulation and computer games. Leibe et al. [9] presented the concept of a *perceptive workbench*, where shadows are exploited to estimate 3D hand position and pointing direction. Their method used infrared lighting and was demonstrated in augmented reality gaming and terrain navigation applications.

These approaches consider light sources far away from the camera center of projection and casted shadows are separated from the objects. In contrast, our approach consider light sources with small baseline distance from the camera, allowing them to be built in a self-contained device, no larger than existing digital cameras.

We have not seen any previous technique that is able to precisely acquire depth discontinuities in complex hand configurations. In fact, stereo methods for 3D reconstruction would fail in such scenarios, due to the textureless skin-color regions as well as low intensity variation along occluding edges.

Many exemplar-based [1] and model-based [10] approaches rely on edge features for hand analysis. We believe that the use of depth edges would lead to significant improvements in these methods. Word level sign language recognition could also benefit from our technique, due to the high amounts of occlusions involved. Flashes in our setup could be replaced by infrared lighting for user interactive applications.

6. Implementation

Our basic prototype for static scenes (Figure 3b) makes use of a 4 MegaPixel Canon Powershot G3 digital camera. The four booster (slaved Quantarray MS-1) 4ms duration flashes are triggered by optically coupled LEDs turned on sequentially by a PIC microcontroller, which in turn is interrupted by the hot-shoe of the camera. For dynamic scenes, our video camera (Figure 3c) is a PointGrey DragonFly camera at 1024x768 pixel resolution, 15 fps which drives the attached 5W LumiLeds LED flashes in sequence.

7. Conclusions

We have introduced the use of depth edges as features for reliable, vision-based fingerspelling recognition. We basically bypass dense 3D scene reconstruction and exploit only depth discontinuities, which is a valuable information to recognize hand postures with high amounts of finger occlusions, without making use of instrumented gloves.

Our method is simple, efficient and requires no parameter settings. We demonstrated preliminary but very promissing experimental results, showing that the use of depth edges outperforms traditional Canny edges even considering simple scenarios with uncluttered background. In more complex scenarios, our technique significantly reduces clutter by eliminating texture edges and keeping only contours due to depth discontinuities.

We are currently evaluating our method in a large database with different signers. We also plan to address the problem of continuous signing in dynamic scenes.

Acknowledgements This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48

References

[1] V. Athitsos and Stan Sclaroff. Estimating 3D hand pose from a cluttered image. In *International Conference on Computer Vision and Pattern Recognition*, Madison, USA, 2003.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[4] Y. Chen, W. Gao, G. Fang, C. Yang, and Z. Wang. CSLDS: Chinese sign language dialog system. In *International Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, 2003.

[5] C. Christoudias, B. Georgescu, and Peter Meer. Synergism in low level vision. In *International Conference on Pattern Recognition*, Quebec City, Canada, 2002.

[6] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3):848–855, 2003.

[7] M. Kolsch and M. Turk. Robust hand detection. In *International Conference on Automatic Face and Gesture Recognition (to appear)*, Seoul, Korea, 2004.

[8] M. Lamar and M. Bhuiyant. Hand alphabet recognition using morphological PCA and neural networks. In *International Joint Conference on Neural Networks*, pages 2839–2844, Washington, USA, 1999.

[9] B. Leibe, T. Starner, W. Ribarsky, Z. Wartell, D. Krum, J. Weeks, B. Singletary, and L. Hodges. The perceptive workbench: Toward spontaneous and

natural interaction in semi-immersive virtual environments. *IEEE Computer Graphics and Applications*, 20(6):54–65, 2000.

[10] S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *International Conference on Computer Vision and Pattern Recognition*, Madison, USA, 2003.

[11] G. Ohashi and Y. Shimodaira. Edge-based feature extraction method and its application to image retrieval. In *7th World Multi-conference on Systemics, Cybernetics and Informatics*, Florida, USA, 2003.

[12] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[13] R. Raskar, K. Tan, R. Feris, J. Yu, and M. Turk. A non-photorealistic camera: Depth edge detection and stylized rendering with multi-flash imaging. In *SIGGRAPH 2004 (to appear)*.

[14] J. Rebollar, R. Lindeman, and N. Kyriakopoulos. A multi-class pattern recognition system for practical fingerspelling translation. In *International Conference on Multimodal Interfaces*, Pittsburgh, USA, 2002.

[15] J. Segen and S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In *International Conference on Computer Vision and Pattern Recognition*, pages 479–485, Fort Collins, USA, 1999.

[16] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *International Conference on Computer Vision*, Nice, France, 2003.

[17] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk- and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[18] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *International Conference on Computer Vision*, pages 363–369, Mumbai, India, 1998.