Sliced Curvature Scale Space for Representing and Recognizing 3D objects

Billy Okal
Social Robotics Laboratory
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 74, D-79110 Freiburg
Email: okal@informatik.uni-freiburg.de

Andreas Nüchter
Robotics and Telematics
University of Würzburg
Am Hubland, D-97074 Würzburg
Email: andreas@nuechti.de

***Abstract*—Perception plays a key role in the development of intelligent autonomous systems. In particular object recognition and registration tasks are crucial to any intelligent autonomous system such as autonomous cars or personal robots. The representation of 3D object sensor measurements largely affects the choice of higher level processing possible on the sensor data. We explore the use of scale space theory via the curvature scale space and extend it to represent 3D objects in our new SCSS (Sliced Curvature Scale Space) framework. We further develop techniques of further processing the SCSS representation including feature extraction and dimensionality reduction for use in learning frameworks. We perform an array of experiments to validate the effectiveness of our method and demonstrate recognition performance using support vector machines. The results indicate that our new representation retains the nice qualities of the original curvature scale space method while being robust and compact for 3D object representation and recognition.**

I. Introduction

In the development of intelligent autonomous systems, interaction ability among such systems and the users-humans is a key indicator of success. These interactions are largely dependent on the ability to perceive and interpret the environment these systems operate in. The key challenges in achieving the aforementioned result include development of appropriate sensors to acquire environmental information, design of robust representation schemes for the collected data and finally developing means of decoding such data into meaningful cues for interaction. While there has been significant advances n sensor mechatronics allowing acquisition of environmental data at very high rates, representation and interpretation are still not adequate as evidenced by the ongoing research in these areas. This is even more pronounced in the 3D case, meaning the impact of recent success like the development of the Kinect devices are held back by lack of appropriate representation and interpretation of the depth data for use in intelligent robotics tasks.

Scale space theory provides a promising framework for representing signals in arbitrary dimensions at multiple scales. A nice overview of scale space theory is given in [1], [2]. The multi-scale representation is motivated by the fact that when no prior information about the scale at which sensor measurements were taken, then it is only sensible that the measurement signals be represented at all possible scales. This representation is achieved by smoothing a signal with a kernel of varying width and has connections the mammalian receptive field operation as shown in neurophysiological studies by [3], [4]. The curvature scale space (CSS) by [5] has been successfuly applied in 2D cases with demonstrated success and provides a suitable basis for expansion to 3D. We are interested in leveraging this basis for developing a new representation for 3D objects.

II. Related Work

Given the interesting nature of this task, there have been a number of attempts in developing a representation for 3D object sensor data, each making a fair share of trade-offs in terms of expressiveness, computational needs, robustness to viewpoint, scale, affine transformations, occlusions etc. Early approaches used CAD based representation which utilizes combinations of standard geometric primitives to model objects as explained in [6]. Other approaches have used range images [7] especially in robot navigation, polygonal meshes as in computer graphics applications. Point clouds are commonly used in robotics domain as explained in [8], [9]. Most the approaches described so far are typically not expressive enough to easily draw high level cues on and require further refinements such as filtering and additional feature extraction. Chua and Jarvis have also proposed point signatures [10] for representing 3D objects and these encode the structural neighbourhood of a point although the choice of this neighbourhood is not trivial. A related approach is the use of shape signatures [11] by Hamza and Krim which exploits the geodesic shape of an object in a probability distribution and uses information theoretic measures to compute object similarities. One of the drawbacks of their method is that the measure they propose is not symmetric and it not always defined hence requiring additional checks in practical use. The use of wavelets [12] and radial basis functions is also common in signal processing related applications, however these scheme usually have complicated procedures for computing object properties that make them less practical. Spin images [13] by Andrew Johnson and Martial Herbert are another representation with properties including invariance to pose and simplicity of computation. Other common representation schemes are reviewed in [6] survey. We are mainly interested in the curvature scale space representation and extending it to 3D domain.

Object recognition of the other hand has enjoyed a wealth of attempts in the past and most approaches are dependent on the underlying representational power. The main categories of methods for object recognition include appearance based methods, contour methods, model-based methods, methods involving exhaustive search and correlation filter methods. In appearance based methods, the object is usually represented as a point in some high dimensional space and statistical techniques applied to compute distances between such points to get object similarities. These methods are reviewed in [6]. Contour based methods aim to exploit the geometry of object like boundaries and this generally makes them view dependent unless special care is taken. Mokhatarian and Murase developed a complete isolated object recognition system based on silhouettes in [14] that also employs the curvature scale space representation. The same representation was also used by image retrieval in [15]. Extending their approach to more realistic multiple object scenarios is not trivial. Stiene et al. have also used an extension of CSS called Eigen-CSS for object recognition in range scans [16]. They use support vector machines to learn to classifly objects. Eigen-CSS [17] improves upon the original CSS by making it robust to rotations. Correlation filter methods utilize the correlation between certain properties of objects to evaluate similarities and are common in 2D domain where image correlations are directly used. Exhaustive search methods like RANSAC (Random Sample Consensus) are common in registration related applications where point correspondences are directly evaluated. Most of these approaches can be aided by leaning techniques when sufficient features are available. We adopt the learning approach in evaluating our representation's recognition power.

The rest of the paper is organized as follows; in Section III we delve into the details of our representation including how to compute it and extraction of features. In Section IV we show some of the results of using the SCSS representation and finally we conclude and mention further work in Section V.

III. Our Approach

In this section, we briefly review the concept of curvature scale space and extend it to develop our SCSS representation. A more comprehensive theory of CSS is available in [18].

A. *Curvature Scale Space (CSS)*

The curvature scale space representation which is also included in the MPEG-7 format [5] is developed by repeatedly convolving a signal with a Gaussian kernel. Given a 2D curve parametrized as $\Gamma = (x(s), y(s))$, the curvature $\kappa(s, \sigma)$ at level σ is computed using Equation 1. The process is known as *evolution*.

$$\kappa(s,\sigma) = \frac{X_s(s,\sigma)Y_{ss}(s,\sigma) - X_{ss}(s,\sigma)Y_s(s,\sigma)}{(X_s(s,\sigma)^2 + Y_s(s,\sigma)^2)^{3/2}} \quad (1)$$

where

$$X_s(s,\sigma) = x(s) \otimes g_s(s,\sigma) \quad (2)$$
$$Y_s(s,\sigma) = y(s) \otimes g_s(s,\sigma) \quad (3)$$
$$X_{ss}(s,\sigma) = x(s) \otimes g_{ss}(s,\sigma) \quad (4)$$
$$Y_{ss}(s,\sigma) = y(s) \otimes g_{ss}(s,\sigma) \quad (5)$$

and $g_s(s,\sigma), g_{ss}(s,\sigma)$ are the first and second partial derivatives of the Gaussian kernel respectively. The CSS image is then made by plotting the location of zero-crossings of the curvature signal $\kappa(s, \sigma)$ for all levels. Such an image is shown in Figure 2.

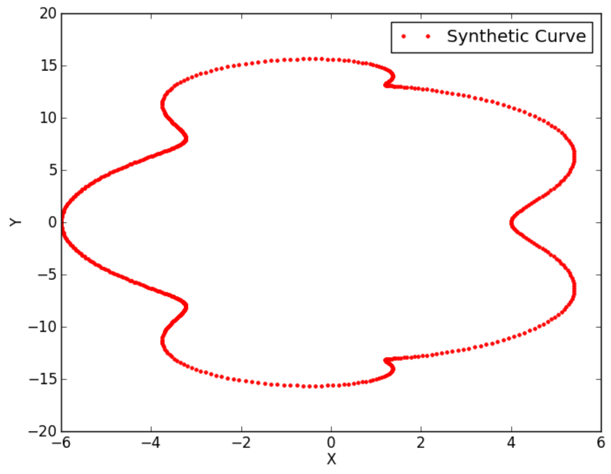

Fig. 1. Simple synthetic curve given by $x(t) = 5\cos(t) - \cos(6t), y(t) = 15\sin(t) - \sin(6t)$

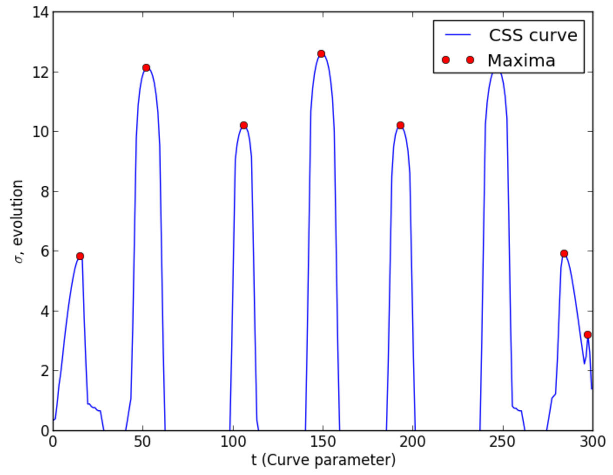

Fig. 2. CSS example for a simple synthetic curve shown in Figure 1

The CSS representation works well for 2D cases with properties like robustness to scale, various transformations, partial occlusion etc. We now extend this framework to work in 3D with point clouds and meshes.

B. *Sliced Curvature Scale Space (SCSS)*

The CSS representation described above works well for 2D cases but extending it to 3D is not trivial. One possible approach is to use range images which encode some information about the depth dimension, but this does not represent a 3D system. We therefore adopt a slicing mechanism whereby we view a 3D object as a set of infinitely close thin plate slices packed together. We can then apply the ordinary CSS technique for each of the slices and devise a way of combining the results into a coherent feature which we call the SCSS (Sliced

Curvature Scale Space). It is important to note that the resulting feature is independent of the slicing direction making the representation invariant to viewpoints. Alternative approach to slicing is to directly model the object surface but this generally requires data of high resolution and surface closedness both is which are rare in real world robotics applications. The overall procedure of computing the SCSS is described in the following sections.

Given a 3D object represented as a mesh or point cloud, we take a fixed number of slices along an arbitrary direction and extract the boundary curve of these slices. The choice of slicing resolution is not trivial and has to be experimentally checked to ensure fine features are not missed with the discretization. Sometimes the points clouds and meshes of object are not empty on the inside and hence it is necessary to add additional measure to extract only the boundary contour of the slices. Furthermore, depending on the sampling density of the points clouds or meshes used, additional re-sampling of the extracted boundary curves may be required. We call these boundary contour curves *slice signals*. Re-sampling the slice signals can be done using many approaches. We explored the use of Kochanek-Bartels splines [19] and re-sampling via Fourier transformation in the frequency domain by zero-padding. The Kochanek-Bartels are cubic Hermit splines with additional control of the tangent based on continuity, bias and tension parameters. One can also 'follow' the curve and add new samples by checking the distance between existing samples. The resulting set of slice signals are then used to compute the SCSS feature by repeatedly packing CSS features along the chosen slicing direction. The curvature in the SCSS representation is then dependent on the slice value and is given by Equation 6.

$$\kappa(s,\bar{c},\sigma) = \frac{X_s(s,\bar{c},\sigma)Y_{ss}(s,\bar{c},\sigma) - X_{ss}(s,\bar{c},\sigma)Y_s(s,\bar{c},\sigma)}{(X_s(s,\bar{c},\sigma)^2 + Y_s(s,\bar{c},\sigma)^2)^{3/2}} \tag{6}$$

where $\bar{c}$ is the slice value and the remaining terms are defined analogous to Equation 1. The resulting SCSS representation is a matrix whose row are the CSS images for the slice signals at given slice values.

Some of the key properties of the original CSS representation include the fact that closed object remain closed under the evolution process. Furthermore, noise in the signal may create small contour changes but the maxima derived remain the same and local information about the object is preserved. The representation is also compact at only a few integers are needed to represent an object. We posit that these feature directly carry over to the SCSS representation with minimal performance reduction. The compactness property is evident in the fact that we still need only a few integers representing the maxima in the SCSS matrix to encode an object. This is number is definitely higher than in the CSS case, but we develop techniques to reduce the number in the next sections using dimensionality reduction. Also in the limiting case as the number of slices increases to infinity, the SCSS representation captures all the local information on a 3D object surface.

C. Feature Extraction

In order to use the SCSS representation developed in the preceding sections in object recognition and registration set-ups, the amount of information used to encode a single object need to be reduced for efficient processing and also to eliminate potential noise. It is important to remove only unwanted elements and keep all informative pieces and is possible the structural relation between them. We explore standard dimensionality reduction techniques of principal component analysis and non-negative matrix factorization to achieve this. We also investigate the extension of Eigen-CSS to 3D.

1) EigenSCSS: The original CSS representation in 2D domain has one problem that if the contour curve is rotated by some angle, the resulting CSS image representation is shifted horizontally by a proportional amount which makes object comparison non-trivial as the starting point has to be first located. The Eigen-CSS method corrects this by computing marginal sums of the CSS image to get a row sum $\mathbf{r}$ and column sum $\mathbf{c}$. The row sum is further made rotation invariant by phase correlation using Equation 7. The column sum is already rotation invariant. Together they are then used in object comparison.

$$\tilde{\mathbf{r}} = |\mathcal{F}^{-1}(\ |\mathcal{F}(\mathbf{r})|\ )| \tag{7}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier Transform and its inverse respectively. It is straightforward to extend this operation to the SCSS representation to rid of the rotation issue. The resulting row and column are then combined as a feature vector $\mathbf{x} = [\tilde{\mathbf{r}}\mathbf{c}]^T$ and $\mathbf{x}$ projected onto its eigenspace by a procedure summarized in [16]

2) Dimensionality Reduction: To further process the data in the SCSS representation we treat the SCSS matrix as image and each pixel as feature and then utilize principal component analysis to select a smaller subset of features that maximally explain the object in terms of variance. The choice of the number of principal components to retain is arrived at experimentally by plotting the cumulative eigenvalues. In particular, we use the non-linear version Kernel-PCA detailed in [20] in order not to impose linearity constraints.

Because the SCSS representation does not contain any negative values and the features sought are local in nature, it is suits matrix factorization methods like NMF (Nonnegative Matrix Factorization) for use in dimensionality reduction. The NMF method which is detailed in [21] seeks to decompose the original matrix $\mathbf{X}$ of features into smaller matrices $\mathbf{W}$ and $\mathbf{H}$ so that the original vectors $\mathbf{x}$ can be represented as linear combinations of columns of $\mathbf{W}$ weighted by components of $\mathbf{H}$. In fact is has been shown by [22], [23] that NMF can produce a parts based representation of the data hence giving interpretable models. Given a matrix $\mathbf{A}$ containing feature vectors, NMF wishes to approximate this matrix by other low rank matrices $\mathbf{W}$ and $\mathbf{H}$ by solving the following non-linear optimization problem given in Equation 8.

$$\min ||\mathbf{A}_{m\times n} - \mathbf{W}_{m\times k}\mathbf{H}_{k\times n}||_F^2, \quad \text{s.t.} \quad \mathbf{W} \geq 0, \mathbf{H} \geq 0 \tag{8}$$

where k is the required rank. There is no global solution for the minimization problem since it is only convex in either $\mathbf{W}$ or $\mathbf{H}$ but not both. In practice multiple starting positions are tried out while checking the error difference of found local minima using metrics such as Frobenius norm described in the next section. The resulting matrices $\mathbf{W}$ and $\mathbf{H}$ are usually sparse meaning they can be stored efficiently. There are number of algorithms for solving the optimization problem available in standard literature accompanied by efficient implementations in most common languages.

D. Classification of Objects

In order to recognize object using the features that we have developed in the preceding sections, we employ a discriminative learning approach to test the discriminative power of the features. We use support vector machines (SVM) detailed in [24] for classification of 3D objects in a multi-class classification setup. Generally, given a dataset of observations $\mathbf{x}$ and labels $\mathbf{y}$, SVMs aim to find a model given by Equation 9.

$$y(\mathbf{x}) = \omega^T \phi(\mathbf{x}) + b \tag{9}$$

where ω is the set of parameters of the model, b is the bias and $\phi(\mathbf{x})$ is feature space transformation using various kernels. The most common kernels include radial basis functions (RBF), linear and polynomial kernels. These common kernels are shown in Equations 10, 11 and 12 for linear, polynomial and RBF kernels.

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}' \tag{10}$$

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^d \tag{11}$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2) \tag{12}$$

In practice, the performance of a set of kernels is compared to decide of which one to use. Other parameters of the SVM are also tuned during experimentation.

IV. Results

We demonstrate performance in object similarities by showing similarities across a number of common objects. We use the Frobenius norm as a metric to compute the similarities between objects. Concretely, given a $m \times n$ sized matrix $\mathbf{A}$, the Frobenius norm is defined as the square root of the sum of its absolute elements.

$$\mathbf{A}||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{Tr}(\mathbf{A}\mathbf{A}^H)} \tag{13}$$

where Tr is the trace and $\mathbf{A}^H$ is the conjugate transpose. In order to compare two matrices $\mathbf{A}$ and $\mathbf{B}$ for similarity, we assume that the matrices have similar sizes and that their columns or equivalently rows correspond i.e. columns/rows one matrix can be represented as a permutation of columns/rows of the other matrix. Hence computing $\varpi = ||\mathbf{A} - \mathbf{B}||_F$ gives a measure of how similar they are. If ϖ is close to 0, then the matrices are very similar and vice versa. Figure 3 shows SCSS representation of a cup both viewed as an image and in 3D, the extracted maxima and the correspondence of the handle feature indicated with the red box.

The object similarities are shown in Table I with the common objects; coffee cup, banana, beer bottle, cereal box and a football. It is worth nothing that the football is unusually similar to a banana. This is because the banana without the end features is just a football that is scaled and stretched along one axis. These similarities are even better illustrated in Figure 4.

	Cup	Banana	Cereal Box	Beer Bottle	Football
Cup	0	20027.477	19744.158	19768.158	19632.135
Banana	20027.477	0	14065.791	13633.769	**9771.715**
Cereal Box	19744.012	14065.791	0	16423.921	16040.761
Beer Bottle	19768.158	13633.769	16423.921	0	14622.983
Football	19632.135	**9771.715**	16040.761	14622.983	0

TABLE I. Similarities between object based on SCSS images

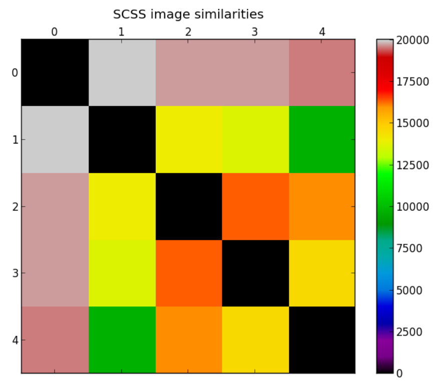

Fig. 4. Illustration of similarities between five objects (0=cup, 1-banana, 2=cereal box, 3=beer bottle, 4=football)

We also performed experiments to check the repeatability of the SCSS representation when the slices are taken from varying viewpoints of an object. In particular, we use the Stanford bunny model and show two SCSS images generated along different axes in Figure 5 and Figure 6. It is easy to notice that the SCSS images are very close even though the bunny has a very irregular surface, generating very different contours for the different viewpoints. This irregular surface gives the different impresion between the two images. We also computed the numerical similarity between the two images using the Frobenius norm discussed above and found 3470.917 which was much less than similarity values with the other five objects and shown in Table II with the key value in bold. The numerical comparison results serve to reinforce our claim about the repeatability.

We also perform further experiments in learning for object recognition. Since we have five object classes, we employ a one-vs-the-rest strategy for each classification experiment and repeat this for every dimensionality reduction method. The results are shown in Table III and Figure 7. In each of the cases the SVM was run with parameters $C = 1.2$, $\gamma = 0.7$ and degree of the polynomial equal to 3. The 'original' data in the table and plot correspond to dimensionality reduction by rescaling of the original SCSS images. The SVM implementation was based on the standard libsvm [25] library via the scikit-learn [26] python framework with tolerance of 0.001. The associated dimensionality reduction procedures were also implemented in scikit-learn. From the results, one can notice that there is a slight improvement in recognition when the features are first transformed using Kernel PCA. The difference

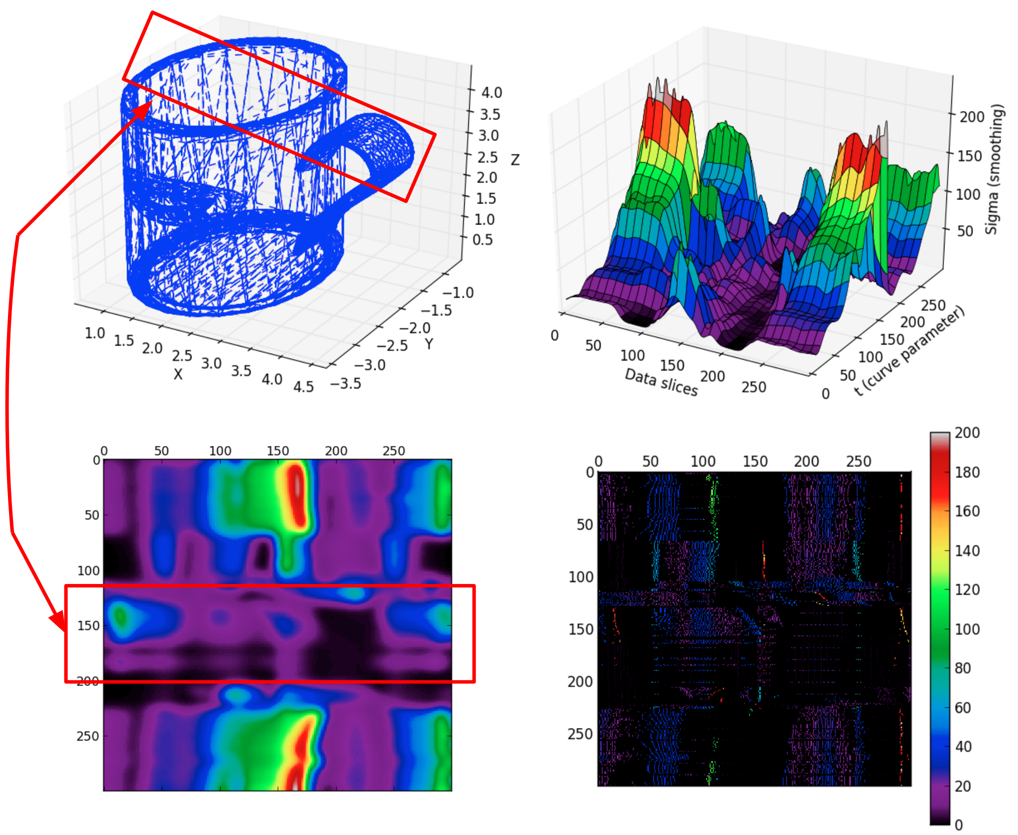

Fig. 3. Demonstration of maxima feature extraction and their correspondence to original object's attributes (TOP LEFT: Original cup 3D points, TOP RIGHT: The SCSS representation in 3D, BOTTOM LEFT: SCSS Image and BOTTOM RIGHT: Extracted maxima features)

	Bunny side	Cup	Banana	Football	Cereal box	Beer Bottle
Bunny front	**3470.917**	18414.378	10470.791	12464.294	13478.597	15797.838

TABLE II. SCSS SIMILARITY ON THE SAME OBJECT (BUNNY) FROM DIFFERENT VIEWPOINTS

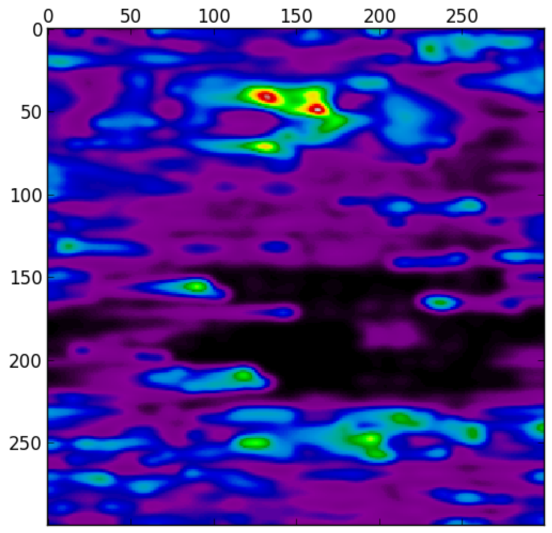

Fig. 5. Bunny SCSS from front viewpoint, see Table II for numerical comparison wth other objects

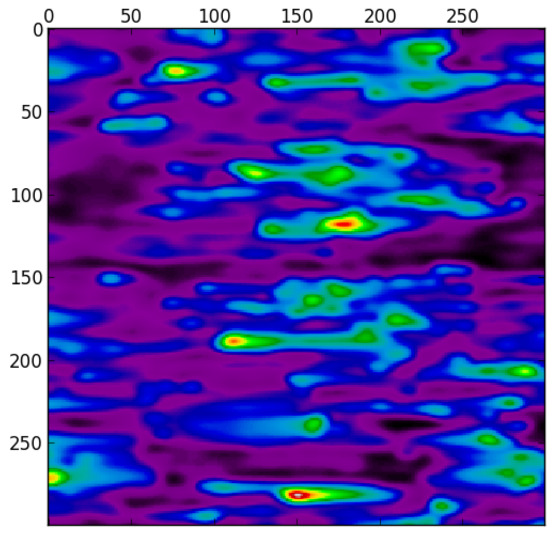

Fig. 6. Bunny SCSS from side viewpoint, see Table II for numerical comparison with other objects

Kernel Type/Dim method	Polynomial	RBF kernel	Linear
Original data	0.92	0.92	0.88
Kernel PCA	0.96	0.96	0.95
NMF	0.32	0.36	0.52

TABLE III. CLASSIFICATION RATES USING DIFFERENT SVM CONFIGURATIONS AND DIMENSIONALITY REDUCTION METHODS

is small because the features processed are localized on the object and hence simple by averaging neighbors is sufficient. We attribute the poor performance of NMF dimensionality reduction to insufficiency of training data given the strictness in the optimization of NMF and the fact that NMF does not provide a unique factorization. We however believe that with more data the procedure may perform even better that Kernel PCA given its parts based representation.

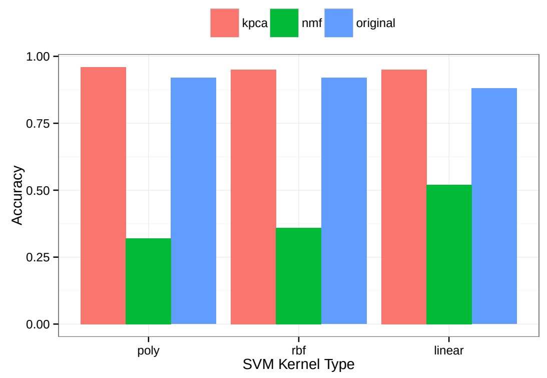

Fig. 7. Recognition rates with various SVM kernel types and dimensionality reduction methods

V. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed and developed the Sliced Curvature Scale Space (SCSS) as a means for representing 3D objects using the sensor information acquired. The SCSS representation is anchored on the already successful CSS representation in 2D cases and directly 'carries over' the key properties making it a robust means of representing 3D objects. As a drawback, the SCSS representation contains large amounts of data that it not efficient for processing and we have therefore proposed a number of standard and non-standard ways of reducing the amount of data per object to be stored when using SCSS representation. We have demonstrated this using principal component analysis, non-negative matrix factorization and simple rescaling of SCSS images and found favourable results. We have further demonstrated that the features extracted from our representation scheme are indeed discriminative by way of classifying various objects using the well known support vector machines framework. The experiments with classification also serve to reinforce our belief that the features are indeed localized on the object surfaces which allows simple dimensionality reduction schemes to be effective.

In the future, we believe that using a paralleled version of our method for computing the SCSS representation will provide better speeds since all the key parts can be paralleled. We also believe that having better trained learning models using much more training data will improve the recognition rates. We also wish for better theoretical work on connecting SCSS with other scale space approaches that could provide insights into new ways of generating SCSS and learning metrics for directly comparing SCSS elements in higher dimensional spaces.

ACKNOWLEDGMENT

The authors would like to thank Herbert Jaeger and Girum Demisse for informative comments on this work and discussions on dimensionality reduction.

REFERENCES

[1] T. Lindeberg, "Scale space theory," *Encyclopedia of Mathematics*, 1997.

[2] ——, "Scale-space for discrete signals," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 3, pp. 234–254, 1990.

[3] G. DeAngelis, I. Ohzawa, and R. Freeman, "Receptive-field dynamics in the central visual pathways," *Trends in neurosciences*, vol. 18, no. 10, pp. 451–458, 1995.

[4] R. Young, "The gaussian derivative model for spatial vision: Retinal mechanisms," *Spatial vision*, vol. 2, no. 4, pp. 273–293, 1987.

[5] F. Mokhtarian and M. Bober, *Curvature scale space representation: theory, applications, and MPEG-7 standardization*. Springer Publishing Company, Incorporated, 2011.

[6] R. Campbell and P. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.

[7] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252–1262, 2007.

[8] R. Rusu, Z. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[9] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[10] C. Chua and R. Jarvis, "Point signatures: A new representation for 3d object recognition," *International Journal of Computer Vision*, vol. 25, no. 1, pp. 63–85, 1997.

[11] A. Hamza and H. Krim, "Geodesic object representation and recognition," in *Discrete Geometry for Computer Imagery*. Springer, 2003, pp. 378–387.

[12] L. Pastor, A. Rodriguez, J. Espadero, and L. Rincon, "3d wavelet-based multiresolution object representation," *Pattern Recognition*, vol. 34, pp. 2497–2513, 2001.

[13] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433 – 449, May 1999.

[14] F. Mokhatarian and H. Murase, "Silhouette-based object recognition through curvature scale space," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, May 1993, pp. 269 –274.

[15] S. Abbasi, F. Mokhtarian, and J. Kittler, "Curvature scale space image in shape similarity retrieval," *Multimedia systems*, vol. 7, no. 6, pp. 467–476, 1999.

[16] S. Stiene, K. Lingemann, A. Nuechter, and J. Hertzberg, "Contour-based object detection in range images," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*. IEEE, 2006, pp. 168–175.

[17] M. S. Drew, T. K. Lee, and A. Rova, "Shape retrieval with eigen-css search," *Image and Vision Computing*, vol. 27, no. 6, pp. 748–755, 2009.

[18] F. Mokhtarian and A. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 789–805, 1992.

[19] D. H. Kochanek and R. H. Bartels, "Interpolating splines with local tension, continuity, and bias control," in *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3. ACM, 1984, pp. 33–41.

[20] H. Hoffmann, “Kernel pca for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[21] D. Seung and L. Lee, “Algorithms for non-negative matrix factorization,” *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[22] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[23] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[24] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer Series in Statistics, 2001, vol. 1.

[25] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.