Salient Views and View-dependent Dictionaries for Object Recognition

Yi-Chen Chen

Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

Vishal M. Patel

Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

Rama Chellappa

Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

P. Jonathon Phillips

National Institute of Standards and Technology, Gaithersburg, MD 20899

Abstract

A sparse representation-based approach is proposed to determine the salient views of 3D objects. The salient views are categorized into two groups. The first are boundary representative views that have several visible sides and object surfaces that may be attractive to humans. The second are side representative views that best represent views from sides of an approximating convex shape. The side representative views are class-specific and possess the most representative power compared to other within-class views. Using the

Email addresses: cheny08@umiacs.umd.edu (Yi-Chen Chen), pvishalm@umiacs.umd.edu (Vishal M. Patel), rama@umiacs.umd.edu (Rama Chellappa), jonathon@nist.gov (P. Jonathon Phillips)

concept of characteristic view class, we first present a sparse representation-based approach for estimating the boundary representative views. With the estimated boundaries, we determine the side representative views based on a minimum reconstruction error criterion. Furthermore, to evaluate our method, we introduce the notion of view-dependent dictionaries built from salient views for applications in 3D object recognition and retrieval. The proposed view-dependent dictionaries encode information on geometry across views and representation of the object. Through a series of experiments on four publicly available 3D object datasets, we demonstrate the effectiveness of our approach compared to two existing state-of-the-art algorithms and one baseline method.

Keywords: Salient view, characteristic view class, view geometry, sparse representation, view-dependent dictionaries, object recognition.

1. Introduction

The concept of characteristic views was first proposed in [4], [12] for object recognition, which was defined as two views belonging to the same characteristic view class that are topologically equivalent, and can be related by a 3D transformation. The transformation consists of geometric rotation, translation and perspective projection [26]. [26] proposes a framework to partition the viewing space and to find the set of characteristic views for planar-faced solid objects. This work was later extended in [7], which essentially computes the characteristic views of objects with curved-surface.

There are a number of approaches for describing what is contained in a view [23], [3]. For view-based representations, human perceivers are influenced by factors such as familiarity with the object being viewed, the similarity of a given view to known views of visually-similar objects and the pose of the object [23]. Three-quarter views with all visible front, top and side, are often used as candidate views[1]. As noted in [20], three-quarter views are essentially views that most humans prefer when looking at an object. These views are also known as *canonical views* [3].

In [27], saliency refers to the novel information in an image relative to an existing representation. For an eigenspace representation, the saliency is computed as the residual error that is the amount of energy not captured by the basis set. A greedy algorithm was proposed for subset selection where the saliency of every ensemble view is first computed and then the view with the highest saliency is added to the subset. The subset is then modified using the eigenspace representation updating algorithm [5], [13] so that the task of salient view selection can be realized in a dynamic environment.

In recent years, the theory of sparse representations has emerged as a powerful tool for efficient processing of data. Motivated by its success in many computer vision and image processing applications [28], [11], we propose a sparse representation-based approach for selecting the salient views of an object [3], [20]. Given an object that is not necessarily convex, we assume it can be approximated by a simple convex shape with multiple number of sides. A side view class is defined as the set of all views of the corresponding

[1] In the viewing space there are in fact infinite number of viewpoints. Candidate views are views seen from a (possibly large but) finite subset of viewpoints [20]. Given a view descriptor, the objective in [20] is to find the maximum of this descriptor among the candidate views.

side of the shape, while a boundary view class refers to views where two or more sides can be seen simultaneously.

The motivation of our work originates from the concept of characteristic view class for convex planner-faced solid objects [26]. According to [26], for any given convex planner-faced solid object, planes obtained by expanding the object's faces partition the viewing space. We propose that through these viewing partitions, it is easier to define and find salient views of the object. An object in general cannot be both convex and planner-faced. For a convex but not a planer object, partitions of the viewing space are no longer "planner-faces"; for a general non-convex object, it is even much more involved to partition the viewing space. Our salient views are built upon the convexity assumption of a given object - even if the object is itself not convex, we assume it can be approximated by a convex shape. In this way, the viewing space partitions of the convex shape are based not only on the object's physical shape, but also the texture information. For example, partitions of a sphere, cylinder or lamp, can be roughly determined by the text, color or other texture contents shown on the surface. As what actually partition the viewing space may not necessarily be "planner-faces", we use "sides" to refer to those that partition the viewing space, and "boundaries" as the boundaries among different partitions.

Fig. 1 illustrates distinct regions of side view classes and boundary view classes given an approximate convex shape for an object. The shape consists of four sides, which give four side view classes and four boundary view classes under orthographic projection. These eight classes are exactly the eight characteristic view classes of the approximate convex shape. Using the

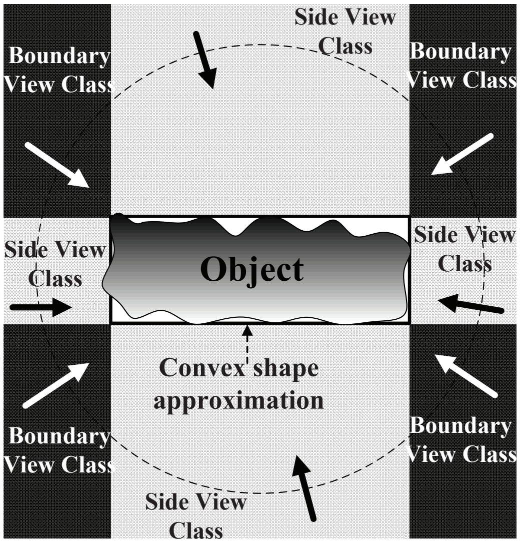


Figure 1: Convex shape approximation and the associated regions of side view classes and boundary view classes.

object's approximate convex shape and its sides, we categorize salient views into two categories: *boundary representative views* (BRVs) which have more visible sides and object surfaces, and therefore are more attractive from a human perception point of view; and *side representative views* (SRVs) which best describe the underlying side view classes. In Fig. 1, BRVs and SRVs are views seen from directions marked with solid white and solid black arrows, respectively. Fig. 2 shows the block diagram of the proposed two-stage approach for finding the salient views. Views are extracted from a video sequence, cropped and properly resized. In the first stage, the boundary scores are computed using a sparsity-based spread metric to estimate the BRVs and determine the side view classes. In the second stage, for each side, a set of SRVs that best represent a corresponding side are chosen by minimizing a representation error. In other words, our salient views consist of BRVs and

SRVs, where a BRV is found by maximizing the boundary score while a SRV is found by minimizing the sparse-to-full reconstruction error.

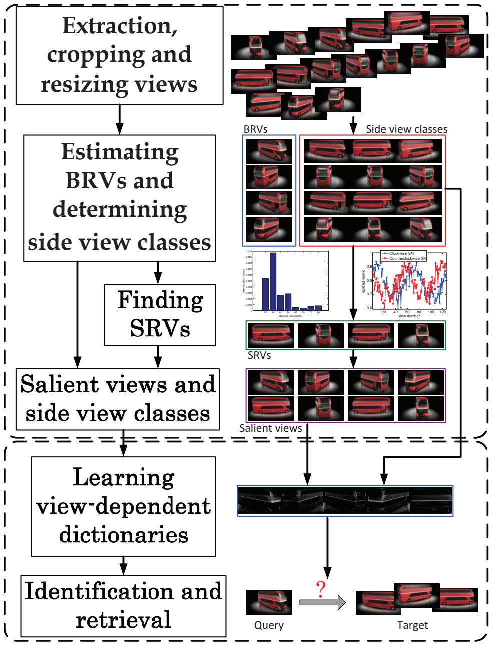


Figure 2: Block diagram of the proposed salient view selection approach.

Important applications of salient views include object recognition and retrieval. In these applications, objects are retrieved or classified from different perspectives. To show the effectiveness of our method, we introduce the notion of view-dependent dictionaries that are built using salient views and side view classes. These view-dependent dictionaries can then be used for 3D object recognition and retrieval applications.

Key contributions of our work are:

1. We propose a sparse representation-based approach for extracting the salient views of an object.

2. Our method is based on characteristic views. It selects representative views of visible sides and object surfaces.

3. We extend this work by introducing view-dependent dictionaries which are based on the salient views for object recognition and retrieval.

4. The view-dependent dictionaries are related by the geometry across views, and they represent the object in an informative way.

5. We demonstrate the effectiveness of our approach on four publicly available 3D object datasets.

Items **1** and **2** above summarize the preliminary version of this work that appeared in [8], while items **3**, **4** and **5** are extensions to [8].

1.1. Organization of the Paper

This paper is organized as follows. In section 2, we present our method for estimating the BRVs of an object. In Section 3, we describe our approach for determining the SRVs. In section 4, we detail the view-dependent dictionary learning method using salient views, and its application to object recognition and retrieval. Experimental results and discussions on recognition and retrieval using view-dependent dictionaries, as well as visual hull based view synthesis using the proposed salient views, are presented in section 5. Section 6 concludes the paper with a brief summary and future work.

1.2. Summary of acronyms and notations

We present in Tables 1 and 2 a summary of acronyms and notations used in this paper.

Table 1: Summary of acronyms.

Acronym	Original meaning	Where defined
BRV	Boundary Representative View	Section 2
SRV	Side Representative View	Section 2
VDD	View-dependent Dictionary	Section 4
VDDR	View-dependent Dictionary based Recognition/retrieval	Section 4.3
SVSR	Salient View selection based on Sparse Representation	Section 4.3
SS	Subset Selection [27]	Section 5.2
VS	Video Summarization [22]	Section 5.2
LOO	Leave-one-out	Section 5.2.1

2. Estimating boundary representative views

It has been shown from [26] that for a convex planar-faced solid object, planes obtained by expanding the object's faces partition the viewing space. These planes are used to partition the viewing space into regions called characteristic view domains. Whenever two views belong to the same characteristic view class, every viewable point in one view is also viewable in the other view, and vice versa. Using this idea on the assumed approximate convex shape of a given object, we use a metric called boundary score to select BRVs.

In this work, we consider only the 3D views of an object with respect to the Y axis rotation under the orthographic projection. To represent the saliency of a candidate view relative to an existing group, the proposed boundary score is given in a form of $1 - \text{SCI}$, where SCI stands for Sparsity Concentration Index [29], [19]. It is a measure of sparsity of the coefficient

Table 2: Summary of notations.

Variable	Definition	Where defined
$\mathbf{z}_j$	the view at position j in its column vectorized form	Section 2
$\mathbf{W}$	a sliding window with γ consecutive $\mathbf{z}'_j s$	Section 2
$\mathbf{C}_1$	the matrix that contains views (each in a column-vectorized form) in the first side view class as its columns	Section 3
$\mathbf{y}_1$	the column-vectorized form of $\mathbf{C}_1$	Section 3
$\mathbf{y}_s$	the column-vectorized form of a set of selected views from the first side view class	Section 3
$\mathbf{R}$	an object-dependent basis set of $\mathbf{y}_1$	Section 3
$\mathbf{R}_s$	a subset of $\mathbf{R}$ corresponding to $\mathbf{y}_s$	Section 3
$\boldsymbol{\sigma}$	a column of eigenvalues of $\mathbf{y}_1$ as entries	Section 3
$\{\mathbf{a}_l^i\}_{l=1}^{n_i}$	salient views (in column-vectorized form) of the ith object	Section 4.1
$\mathbf{A}_i$	the matrix whose columns are $\mathbf{a}_l^i s$	Section 4.1
$\mathbf{B}_i$	salient view VDD of the ith object	Section 4.1
$\mathbf{C}_{i,j}$	the matrix that contains views (each in a column-vectorized form) in the jth side view class of the ith object as its columns	Section 4.2
$\mathbf{D}_{i,j}$	the VDD of the jth side view class of the ith object	Section 4.2
$\mathbf{D}_i$	concatenation of side view class VDDs of the ith object	Section 4.2
$\mathbf{h}$	a query view (in column-vectorized form)	Section 4.2
$\mathbf{E}_i$	a general term for $\mathbf{B}_i$ or $\mathbf{D}_i$	Section 4.3
G	number of retrieved images for the query view $\mathbf{h}$	Section 4.3

representation of a vector in some basis. Low values of SCI indicate that the given view is fairly informative relative to the existing group. Therefore, when applied to the view selection, a high boundary score is a strong indication that the candidate view belongs to a characteristic view class that is different from the existing one. It can be shown that the boundary score falls in the range of $[0, 1]$.

In practice, as no knowledge on side view classes is given initially, we compute the boundary score of a candidate view relative to a set of views contained in a sliding window (i.e. a set of views with consecutive view indices) on the path of rotation (with respect to the Y axis). The BRVs are the views with the maximum boundary scores. In particular, given a view at position j, we use $\mathbf{z}_j$ to denote the view in its column-vectorized form. The boundary score of $\mathbf{z}_j$ is computed by

$$1 - \mathrm{SCI} = \frac{\gamma}{\gamma - 1}\left(1 - \max_{i \in \{1,2,\ldots,\gamma\}} \frac{\|\delta_i(\tilde{\mathbf{x}})\|_1}{\|\tilde{\mathbf{x}}\|_1}\right), \tag{1}$$

where $\tilde{\mathbf{x}}$ is the representation of the view $\mathbf{z}_{(j+\alpha)}$ under the sliding window $\mathbf{W}$ given by

$$\mathbf{W} \triangleq \left(\mathbf{z}_{(j-\beta-\gamma+1)}\ \mathbf{z}_{(j-\beta-\gamma+2)}\ \cdots\ \mathbf{z}_{(j-\beta)}\right). \tag{2}$$

We have

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ s.t. } \ \mathbf{z}_{(j+\alpha)} = \mathbf{W}\mathbf{x}. \tag{3}$$

In (1), $\delta_i(\tilde{\mathbf{x}})$ is a masked version of $\tilde{\mathbf{x}}$ such that its only nonzero entry is the one that corresponds to the i-th column of $\mathbf{W}$. We compute the spread metric of the view ahead of $\mathbf{z}_j$ by α units of indices, with respect to the set formed from the $(\beta+\gamma-1)$-th view up to the β-th view behind $\mathbf{z}_j$[2]. That is, this set is formed according to a β-index logged window with size γ.

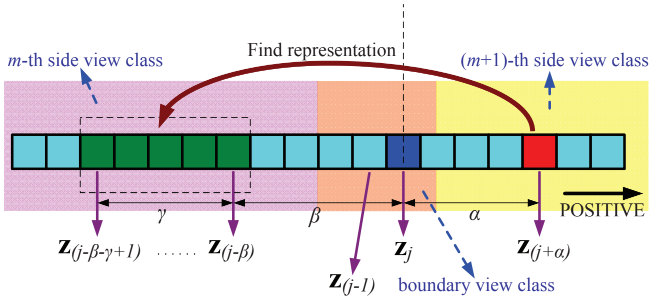


Figure 3: An illustration of finding the boundary score.

Fig. 3 illustrates the proposed method to compute the boundary score. Consider two side view classes: m-th side view class (in color purple) and $(m+1)$-th side view class (in color yellow), and one boundary view class in between. Since in the beginning no information on side view classes is provided, the choice of basis is unknown and we use a sliding window $\mathbf{W}$ (consisting of views in color green) with a predetermined size γ to find the boundary score at $\mathbf{z}_j$ by (1).

[2] A limitation of the proposed method is that α and β cannot be too small or too large. As we assume the training sequences with known BRVs or ground-truth visual-hull representations are not available, we experimentally set α and β such that α corresponds to 48° and β corresponds to 24°. If such training sequences can be acquired, we can first solve $\hat{\alpha},\hat{\beta} = \operatorname*{argmin}_{\alpha,\beta} \|\mathbf{x}\|_1$ $s.t.$ $\mathbf{z}_{(j^*+\alpha)} = \mathbf{W}^*\mathbf{x}$, where $\mathbf{W}^* = (\mathbf{z}_{(j^*-\beta-\gamma+1)}\ \mathbf{z}_{(j^*-\beta-\gamma+2)} \cdots \mathbf{z}_{(j^*-\beta)})$. Then we take the average over $(\hat{\alpha},\hat{\beta})$'s computed from all given BRVs and sequences as the experimental setting for test sequences in the same category.

2.1. Geometric interpretation of boundary representative views

There is a close relation between the proposed BRVs and a minimal set of representative views that can be used to successfully synthesize all other views of an object via visual hull based methods [14], [30]. Without explicitly rendering from the 3D model to reconstruct the synthesized view, these methods build correspondences between a given view and each synthesized view by performing pixel-to-pixel mapping based on spatial correspondences according to the geometry. All pixels that can be seen from synthesized view are mapped from the corresponding pixels of the given view.

Using the concept of visual hull, the minimal set of representative views[3] is selected such that for every view $\mathbf{z}_j$ of the given object, the intersection between the cone formed by projecting the silhouette image into the 3D space through the camera center of the view $\mathbf{z}_j$, and the shape of the object, is contained in the intersection between the visual hull formed by the corresponding cones[4] projected from camera centers of these representative views, and the shape of the object. A BRV is a view with as many as visible "sides" of the convex shape approximation on the object, and hence is a view whose characteristic view domain covers as many as viewpoints as possible. Therefore, the visual hull formed by cones of all viewpoints in the characteristic

[3] Given 2D images, we can estimate BRVs using the proposed algorithm. To find the minimal set of representative views, however, we need 3D geometric information of the object (including camera intrinsic and extrinsic parameters). The scenario where we find BRVs is based on a sequence of 2D images without knowing 3D parameters. Hence, in this work, we did not find the minimal set of representative views to replace BRVs.

[4] The visual hull is the intersection of these cones.

view domain of the BRV contains the intersection between the object shape and the corresponding cones of views that are as many as possible. In other words, BRVs are candidates of representative views, where the representative views are views that can be used to synthesize full views of the object. The relation of BRVs with geometry is further elaborated via the visual hull based view synthesis experiments presented in section 5.3.

3. Side representative view selection

Representative views can either be interpreted as a sparse representation (i.e., coefficients) under some basis, or can be used as sparse observation where sparse coefficients under some basis can be found. In this section, with representative views regarded as sparse observations, we propose a procedure for finding an object-dependent basis set. We assume that camera parameters are not known.

We assume that distinct side view classes are independent of each other. Without loss of generality, we consider the first side view class, denoted by $\mathbf{C}_1 = [\mathbf{z}_1\ \mathbf{z}_2\ ...\ \mathbf{z}_{k_1}]$. Its singular value decomposition (SVD) is $[\mathbf{z}_1\ \mathbf{z}_2\ ...\ \mathbf{z}_{k_1}] = \mathbf{V}\Sigma\mathbf{U}^T$, where $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_L]$ is an L-by-L matrix (L is total number of pixels of an image); $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ ...\ \mathbf{u}_{k_1}]$ is a k_1-by-k_1 matrix; and $\sigma_1, ...\sigma_{k_1}$ are the eigenvalues (i.e., first k_1 diagonal entries of Σ). Let $\mathbf{y}_1$ be the column-vectorized form of $\mathbf{C}_1$. It can be shown that

$$\mathbf{y}_1 = \mathbf{R}\boldsymbol{\sigma}. \tag{4}$$

where the ith column of $\mathbf{R}$ is the column-vectorized form of matrix $\mathbf{v}_i\mathbf{u}_i^T$. In (4), matrix $\mathbf{R}$ is an object-dependent basis set, and $\boldsymbol{\sigma} = [\sigma_1, ...\sigma_{k_1}]^T$ contains eigenvalues as coefficients.

Our objective is to select l_1 out of k_1 views as representative views that best represent the side view class, where $l_1 < k_1$. There are $\binom{k_1}{l_1}$ possible ways to select them. Let

$$\mathbf{R} = \begin{pmatrix} — & \mathbf{Q}_1 & — \\ — & \vdots & — \\ — & \mathbf{Q}_{k_1} & — \end{pmatrix}, \tag{5}$$

where each $\mathbf{Q}_j$ ($j \in \{1, ..., k_1\}$) is a L-by-k_1 matrix corresponding to $\mathbf{z}_j$. Consider one way in which the selected views $\mathbf{z}_{s_1}, ..., \mathbf{z}_{s_{l_1}}$ form a column vector $\mathbf{y}_s$, and the corresponding matrices $\mathbf{Q}_{s_1}, ..., \mathbf{Q}_{s_{l_1}}$ form a $\mathbf{R}_s$ such that

$$\mathbf{y}_s = \begin{pmatrix} \mathbf{z}_{s_1} \\ \vdots \\ \mathbf{z}_{s_{l_1}} \end{pmatrix}, \qquad \mathbf{R}_s = \begin{pmatrix} \mathbf{Q}_{s_1} \\ \vdots \\ \mathbf{Q}_{s_{l_1}} \end{pmatrix}. \tag{6}$$

We solve the following equation using the ℓ_1 norm:

$$\hat{\mathbf{x}}_s = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y}_s = \mathbf{R}_s\mathbf{x}. \tag{7}$$

Since $l_1 < k_1$, fewer constraints are involved in solving (7) than in (4), and we would expect $\hat{\mathbf{x}}(\mathbf{y}_s)$ to be sparser than $\mathbf{w}$. Among all possible $\binom{k_1}{l_1}$ ways, the one which gives the least sparse-to-full reconstruction residual is chosen. In other words, we seek

$$\hat{\mathbf{y}}_s = \arg\min_{s} \|\mathbf{y}_1 - \mathbf{R}\hat{\mathbf{x}}_s\|_2. \tag{8}$$

The corresponding best reconstruction is closest to $\mathbf{y}_1$, and can be thought of as the one directly reconstructed using sparse observations from these l_1 representative views. It has sparse representation $\hat{\mathbf{x}}(\hat{\mathbf{y}}_s)$ under the basis $\mathbf{R}$ defined in (4).

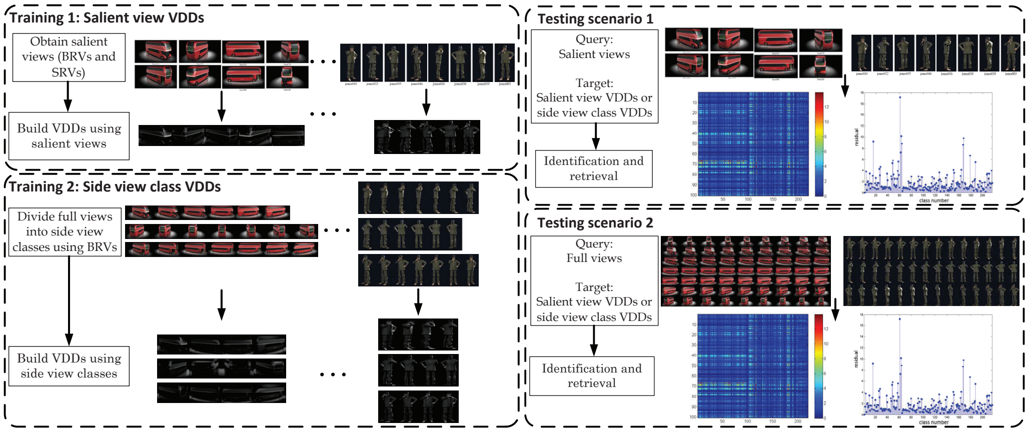
There is a close relation between the proposed salient views and the characteristic views proposed in [4], [12], [26]. For a convex shape, the salient view of a characteristic view class is an important view of that class, in the sense that: When the characteristic view class is a boundary view class, the corresponding salient view is a BRV, found by maximizing the boundary score given by (1); When the characteristic view class is a side view class, the corresponding salient view is a SRV, found by minimizing the reconstruction error given by (8).

4. View-dependent Dictionaries

Important applications of salient views include object recognition and retrieval where one wants to recognize or retrieve images having the same object while taken from different perspectives [17], [9], [10]. In this section, we introduce the notion of view-dependent dictionaries for this application.

With the convexity of an object (or the convexity by approximation if the object is not convex), the corresponding characteristic view classes can be categorized into boundary view classes and side view classes. As shown in sections 2.1 and 3, BRVs are candidates of views that can be used for synthesizing the original full views of the object via visual hull based methods, while each SRV is selected to minimize spare-to-full reconstruction error within the corresponding side view class. Hence, BRVs and SRVs are designed to reflect the view geometry and represent the object in an informative way. As these views are more representative compared to other views, the dictionaries learned from them, called view-dependent dictionaries (VDDs), encode information on geometry across views and representation of the object. The VDDs are designed to represent a 3D object based on views taken from the object's full geometric perspectives, and meanwhile, remove the 3D redundancy. Therefore, they are useful in object recognition and retrieval applications.

The VDDs can be built either directly from salient views (i.e., BRVs and SRVs), or using views belonging to side view classes. Here, we refer to dictionaries built from salient views and side view classes by "salient view VDD" and "side view class VDD", respectively[5].


Training 1: Salient view VDDs
Obtain salient views (BRVs and SRVs)
Build VDDs using salient views
Training 2: Side view class VDDs
Divide full views into side view classes using BRVs
Build VDDs using side view classes
Testing scenario 1
Query: Salient views
Target: Salient view VDDs or side view class VDDs
Identification and retrieval
Testing scenario 2
Query: Full views
Target: Salient view VDDs or side view class VDDs
Identification and retrieval
residual
class number

(a) (b)

Figure 4: Illustration of different training and testing scenarios for recognition/retrieval. (a) Two training scenarios: salient view VDDs and side view class VDDs. (b) Two testing scenarios: salient views (as query) vs. VDDs (as target), and full views (as query) vs. VDDs (as target).

[5] Dictionaries of salient views and side view classes are denoted by $\mathbf{B}_i$ (in section 4.1) and $\mathbf{D}_{i,j}$ (in section 4.2), respectively.

4.1. Salient view VDDs

Let the salient views of the ith object be denoted by $\{\mathbf{a}_l^i\}_{l=1}^{n_i}$. Then, the following optimization problem can be solved to obtain the corresponding salient view VDD denoted by $\mathbf{B}_i$:

$$(\mathbf{B}_i, \mathbf{\Lambda}_i) = \arg \min_{\mathbf{B},\mathbf{\Lambda}} \|\mathbf{A}_i - \mathbf{B}\mathbf{\Lambda}\|_F^2, \text{ s.t. } \|\boldsymbol{\lambda}_l\|_0 \leq T_0,$$
$$\forall l \in \{1, ..., n_i\}, \forall i \in \{1, ..., P\}, \tag{9}$$

where P is the total number of objects (i.e. classes) in the target gallery, $\boldsymbol{\lambda}_l$ represents the l^{th} column of $\mathbf{\Lambda}$, $\mathbf{A}_i$ is the matrix whose columns are $\mathbf{a}_l^i s$ and T_0 is the sparsity parameter. Here, $\|\mathbf{A}\|_F$ is the Frobenius norm of matrix $\mathbf{A}$ defined by $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{ij}^2}$, and the norm $\|\boldsymbol{\lambda}\|_0$ counts the number of non-zero elements in $\boldsymbol{\lambda}$.

Various methods can be used to solve the above optimization algorithm. In this paper, we use the K-SVD algorithm for learning the view-dependent dictionaries due to its simplicity and fast convergence [2]. The K-SVD algorithm alternates between sparse-coding and dictionary update steps. In the sparse-coding step, $\mathbf{B}$ is fixed and the representation vectors $\boldsymbol{\lambda}_{i,l}$s are found for each example $\mathbf{a}_l^i$ by solving the following sparse coding problem

$$\min_{\boldsymbol{\lambda}_{i,l}} \|\mathbf{a}_l^i - \mathbf{B}\boldsymbol{\lambda}_{i,l}\|_2^2 \text{ such that } \|\boldsymbol{\lambda}_{i,l}\|_0 \leq T_0,$$
$$\forall l \in \{1, .., n_i\}, \forall i \in \{1, .., P\}. \tag{10}$$

As solving (10) is NP-hard, approximate solutions are usually sought [6], [25]. Greedy pursuit algorithms such as matching pursuit and orthogonal matching pursuit [18] are often used to find the approximate solutions to the above sparse coding problem [24]. In the dictionary update step, the dictionary is updated atom-by-atom in an efficient way. The K-SVD algorithm has been observed to converge in a few iterations [2].

4.2. Side view class VDDs

Let $\mathbf{C}_{i,j}$ be the matrix that contains views (each in a column-vectorized form) in the jth SVC of the ith object as its columns. Using the K-SVD algorithm, we learn a sub-dictionary $\mathbf{D}_{i,j}$ that best represents $\mathbf{C}_{i,j}$ by solving the same optimization problem as in (10). We then concatenate $\mathbf{D}_{i,j}$s to form a side view class VDD, $\mathbf{D}_i$. In other words, $\mathbf{D}_i = [\mathbf{D}_{i,1}\ \mathbf{D}_{i,2}\ ...\ \mathbf{D}_{i,m_i}]$.

4.3. View-based Object Recognition and Retrieval

In this section, we show how the VDDs can be used for view-based object recognition and retrieval.

4.3.1. Recognition

Given a query $\mathbf{h}$, in a particular view, we project it onto the span of the atoms in each view-dependent dictionary. Let $\mathbf{E}_i$ be the ith target class's view-dependent dictionary (either $\mathbf{B}_i$ or $\mathbf{D}_i$, depending on applications). The approximation and residual vectors can then be calculated as

$$\mathbf{h}^i = \mathbf{E}_i \mathbf{E}_i^{\dagger} \mathbf{h}, \tag{11}$$

and

$$\mathbf{r}^i(\mathbf{h}) = \mathbf{h} - \mathbf{h}^i = (\mathbf{I} - \mathbf{E}_i(\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T)\mathbf{h}, \tag{12}$$

respectively, where $\mathbf{E}_i^{\dagger} \triangleq (\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T$ is the pseudoinverse of $\mathbf{E}_i$, and $\mathbf{I}$ is the identity matrix. As $\mathbf{E}_i$ leads to the best representation for the ith target object, it is assumed that $\|\mathbf{r}^i(\mathbf{h})\|_2$ will be small if $\mathbf{h}$ belongs to the ith class and larger for the other classes. Therefore, if

$$i^* = \arg\min_{1 \leq i \leq P} \|\mathbf{r}^i(\mathbf{h})\|_2, \tag{13}$$

then $\mathbf{h}$ is identified as belonging to the i^*th class in the target gallery as the corresponding view-dependent dictionary gives the minimum reconstruction error.

4.3.2. Retrieval

For image retrieval, we search for the relevance of $\mathbf{h}$ among views belonging to the i^*th target class by a G-nearest-neighbor criterion, where G is the number of retrieved images for $\mathbf{h}$. The resulting View-Dependent Dictionary-based Recognition/retrieval algorithm is denoted as VDDR.

Fig. 4 (a) illustrates two training scenarios for building salient view VDDs and side view class VDDs, respectively. In the testing phase, the query views can either be salient views or full views. These two testing scenarios are illustrated by Fig. 4 (b). We refer to our Salient View selection based on Sparse Representation approach as SVSR. Algorithm 1 summarizes the overall procedure of the proposed SVSR with VDDR for object recognition and retrieval using salient views and view-dependent dictionaries.

5. Experimental Results

In this section, we demonstrate the performance of our method in finding salient views as well as object recognition and retrieval on 3D video sequences. All the 3D video sequences used in our experiments are sequences of still

Algorithm 1: The proposed SVSR with VDDR.

```
Input: Full 3D views of the target gallery, and query views.
Algorithm:
1. Use (1) to compute the boundary score. Choose views with the highest boundary scores as BRVs.
2. Use BRVs to divide the full views into side view classes. Use (8) to find SRVs.
Training:
3. Collect salient views and side view classes. Use (9) to learn $\mathbf{B}_i$ and $\mathbf{D}_i$.
4. Repeat 1, 2 and 3 for all objects in the target gallery.
Testing:
5. Recognition and retrieval - For each query view, determine the closest target class by (13). The relevances are found by the nearest neighbor criterion.
Output:
1. Salient views, side view classes and view-dependent dictionaries of the target gallery.
2. The closest target class and the relevance to each the query views.
```

images taken at regular intervals of $0^\circ \sim 360^\circ$ and $0^\circ \sim 180^\circ$ (with respect to the Y axis) for objects and faces, respectively.

5.1. Salient Views

We selected three available sequences of 3D videos for our experiments on salient view selection: the BUS sequence[6], the HEAD sequence[7] and the JONES sequence[8]. A given video is converted into a set of images, each of which is one view of the object at some particular rotation angle with respect to the Y axis, ranging from 0° to 360°. Images are cropped and resized in the preprocessing stage. Fig. 5 shows these sequences of images. There are 126 views (2.85° increment per view), 32 views ($\sim 11.25^\circ$ increment per view), and 51 views ($\sim 7.05^\circ$ increment per view) for BUS, HEAD and JONES sequences, respectively. In these figures, the sequence of images going from the left to the right in each row, and then from the top row to the bottom row, corresponds to the (camera) clockwise direction. We calculate the spread metrics with $W_{\beta,\gamma}$ sliding in both clockwise (positive) direction, and counterclockwise (negative) direction.

By assuming that the approximate convex shape has four perceptible sides for the object in each of these sequences, we pick four peaks from spread metric scores. In addition, we use the fact that any two peaks shall be separated by a certain gap, otherwise peaks may be located within the same boundary view class (the gap is 22.5° for the BUS sequence, and 30° for HEAD and JONES sequences). Fig. 6, 7 and 8 show the results. It is

[6] http://vimeo.com/3066167

[7] http://vimeo.com/15198240

[8] http://www.youtube.com/watch?v=vq1UeTW6uKE

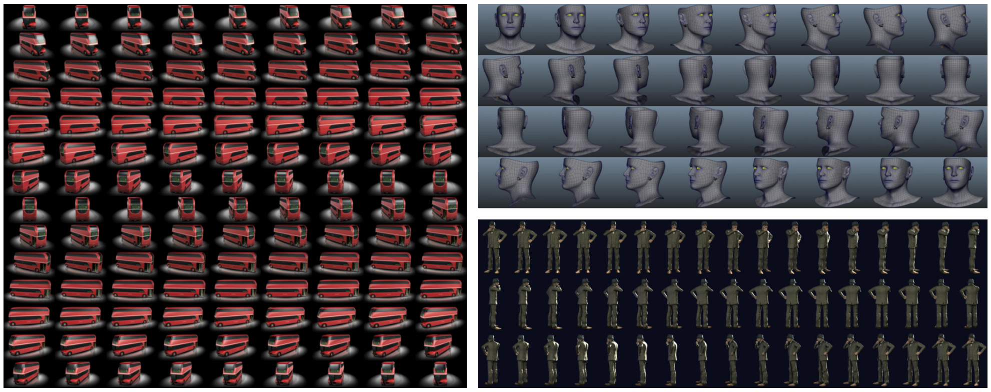
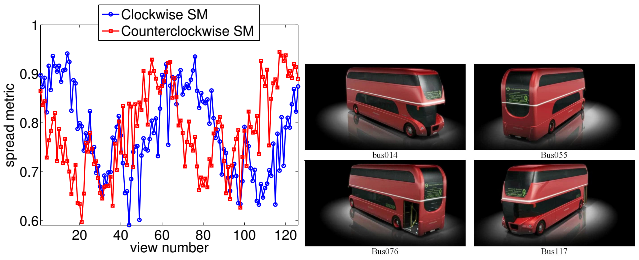


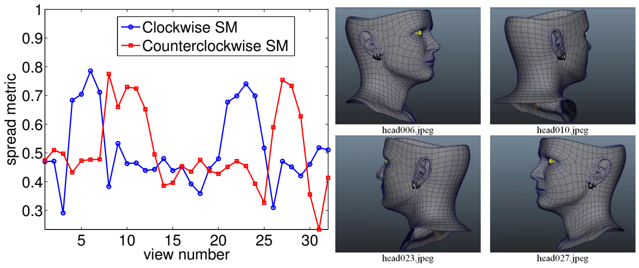
Figure 5: Sequences of 3D views. Left: the BUS sequence (126 views); right top: the HEAD sequence (32 views); right bottom: the JONES sequence (51 views).



(a) (b)

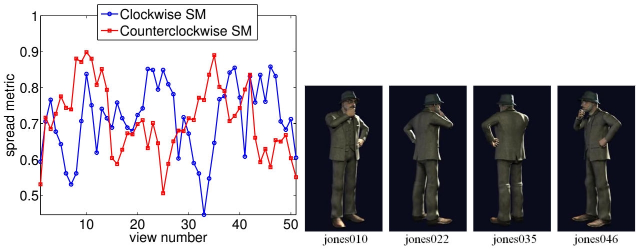
Figure 6: Finding BRVs for the BUS sequence: (a) Clockwise spread metric and counterclockwise spread metric. (b) Estimated BRVs.

expected that these BRVs are those with more sides and visible surfaces as suggested in [3], [20], and hence human perceivers are more sensitive to them.



(a) (b)

Figure 7: Finding BRVs for the HEAD sequence: (a) Clockwise spread metric and counterclockwise spread metric. (b) Estimated BRVs.



(a) (b)

Figure 8: Finding BRVs for the JONES sequence: (a) Clockwise spread metric and counterclockwise spread metric. (b) Estimated BRVs.

Fig. 9 shows the four side view classes which are separated using the estimated BRVs. Taking into account the overall computational load, we evenly down-sample views in each class, such that each class has no greater than nine views. In Fig. 9, we use green lines to mark distinct side view classes. It can be seen that for most cases, views belonging to the same side view class come with more similar poses than those of views that are from distinct side view classes. Fig. 10 shows the resulting SRVs. For each side view class, we pick only one view with the minimum sparse-to-full reconstruction error (i.e., $l_1 = 1$). The results of the BUS sequence are shown in Fig. 10(a), where views with numbers 034, 070, 096 and 126 are obtained with the minimum residuals calculated by (8) and are representatives of side view classes shown in the first row up to the fourth row at the left top of Fig. 9, respectively. Similarly, for the HEAD sequence, views in Fig. 10(b) with numbers 009, 014, 027 and 031 are obtained as SRVs of the left bottom 4 rows in Fig. 9, whereas views in Fig. 10(c) with numbers 016, 030, 039 and 003 are SRVs of those 4 rows of side view classes shown at the right of Fig. 9, for the JONES sequence.

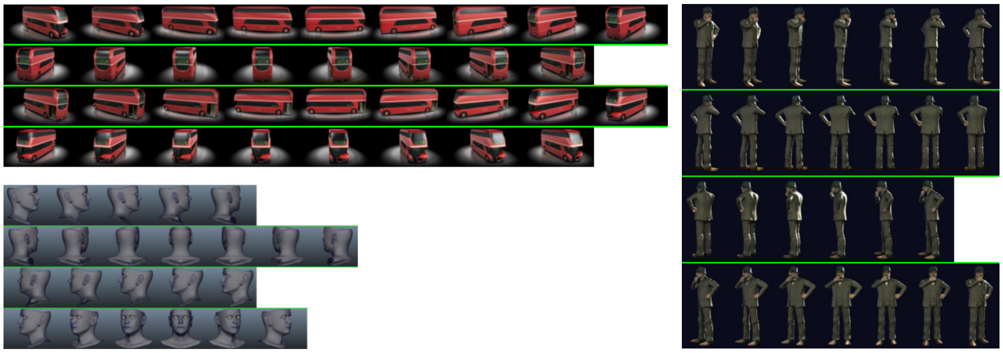


Figure 9: Estimated 4 side view classes with down-sampled views. Left top: the BUS sequence; left bottom: the HEAD sequence; right: the JONES sequence.

To link the proposed salient view selection with sparse representation, we use (7) to formulate the problem as $\mathbf{y}_s = \mathbf{R}_s\mathbf{x}$, where $\mathbf{y}_s$ is a sparse view in contrast with the full view $\mathbf{y}_1$. In other words, given a sparse view (i.e., an incomplete observation of the full view), our goal is to acquire the original full view. Equations (4)-(8) illustrate the proposed sparse representation-based approach. The principal component analysis (PCA), on the other hand, is indeed different from our approach. This can be explained by the fact that the PCA does not impose a sparsity constraint through ℓ_1-minimization, and it does not necessarily give the minimum reconstruction error of the full view, required by both (7) and (8). In addition, eigen-images[9] obtained by PCA in general are not *real* views, which does not constitute a sparse view scenario for view selection considered in our work. Tables 3, 4 and 5 compare the ℓ_2-norm reconstruction errors of the proposed SRVs and eigen-images obtained by PCA. As shown, compared to SRVs, the eigen-images do not give better reconstruction of full views. In addition, these images shown in Fig. 11 are not real views.

Intuitively, one would expect a SRV to be the side view that capture the most energy compared to other within-class views, and thus have minimum sparse-to-full reconstruction residuals according to (7) and (8). It is not hard to see this phenomenon by comparing representative views in Fig. 10 with their associated classes in Fig. 9. For all these sequences, the SRVs are generally pretty close to side views: frontal view, left-side view, right-side view and back view. Finally, the salient views are selected from both BRVs and SRVs.

5.2. Object Recognition and Retrieval using View-dependent Dictionaries

In this section, we demonstrate the performance of our method in object recognition and retrieval on four datasets: Humster3D videos, Princeton 3D models [21], Vetter's 3DFS database [1] and Human ID database [16]. For

[9] Here we use eigen-images to refer to images of eigenvectors.

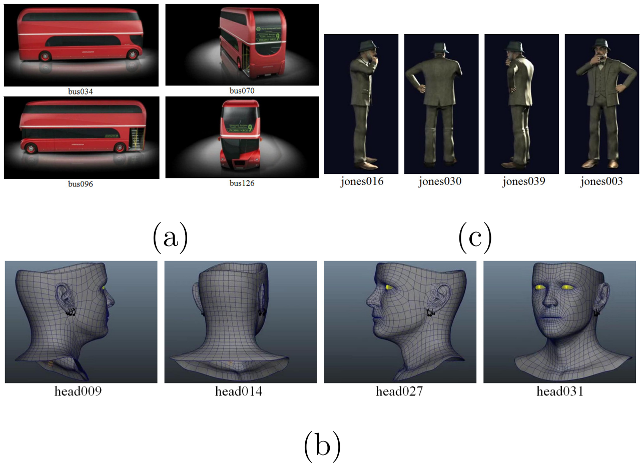


Figure 10: SRVs of (a) the BUS sequence (b) the HEAD sequence (c) the JONES sequence.

The BUS sequence	1st side view class	2nd side view class	3rd side view class	4th side view class
SRV reconstruction error	0.0008	0.0004	0.0001	0
PCA reconstruction error	12.8804	11.6513	13.7120	14.5873

Table 3: Reconstruction errors using SRVs and eigen-images by PCA of the BUS sequence (view size: 32 × 60 pixels).

The HEAD sequence	1st side view class	2nd side view class	3rd side view class	4th side view class
SRV reconstruction error	0.0001	0.0003	0.0001	0.0001
PCA reconstruction error	3.0541	7.7308	2.6140	6.6715

Table 4: Reconstruction errors using SRVs and eigen-images by PCA of the HEAD sequence (view size: 40 × 50 pixels).

The JONES sequence	1st side view class	2nd side view class	3rd side view class	4th side view class
SRV reconstruction error	0.0002	0.0001	0.0005	0.0002
PCA reconstruction error	5.2547	3.8546	6.4123	5.6018

Table 5: Reconstruction errors using SRVs and eigen-images by PCA of the JONES sequence (view size: 60 × 32 pixels).

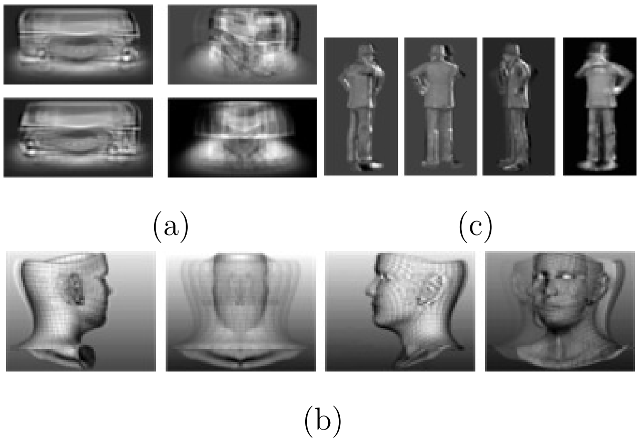


Figure 11: Eigen-images by PCA from side view classes of (a) the BUS sequence (b) the HEAD sequence (c) the JONES sequence. These images are not real views.

each view of the four datasets, we took its grayscale image as the input feature.

We compare the proposed SVSR with two other state-of-the-art approaches proposed in [27] and [22], and one baseline approach. In [27], Winkeler *et al.* proposed a greedy algorithm for subset selection. In their work, saliency was defined as the amount of energy not captured by the basis set for an eigenspace representation. The saliency of every ensemble view is computed and the one with the highest saliency is added to the subset. In [22], Shroff *et al.* proposed a video summarization algorithm to select exemplar frames. Their algorithm optimizes a linear combination of *diversity* and *square error*, where *diversity* represents the scatter of exemplars to their mean, while the *square error* represents the summation of all class scatters.

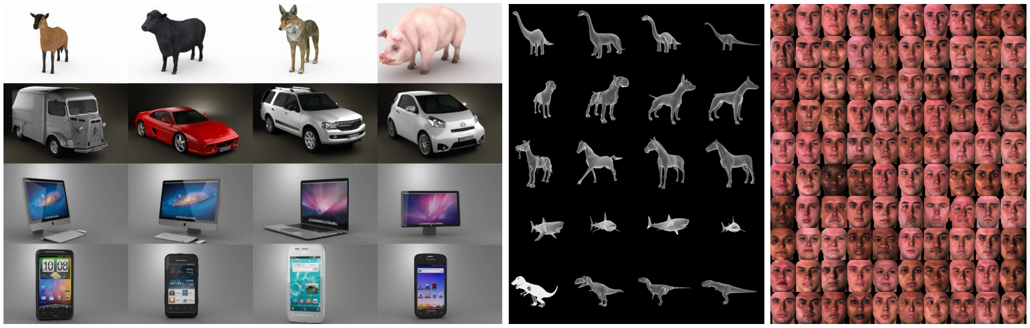
We refer to the methods in [27] and [22] by SS (for Subset Selection) and VS (for Video Summarization), respectively. For fair comparisons, the SS, VS and SVSR methods are all followed by the VDDR algorithm for building view-dependent dictionaries. On the other hand, the baseline method is the one that randomly selects salient views, followed by a nearest neighbor (NN) classifier without using dictionaries. This baseline-NN is provided in contrast with the SS-VDDR, VS-VDDR, and SVSR-VDDR methods. For each model, we selected 8 salient views using SVSR (4 BRVs and 4 SRVs, i.e. $n_i = 8$ and $m_i = 4$), VS, SS and baseline algorithms. Unless otherwise stated, the number of dictionary atoms is set equal to 8 for the salient view VDDs, and 20 for the side view class VDDs. Moreover, as the baseline-NN selects salient views randomly, we reported its average performance over 20 trials.

We evaluate the methods in terms of identification and retrieval performances. Given a certain number of retrieved images, the average retrieval performance [15], [10] of a class is defined as the average number of relevant retrieved images over all query images of that particular class. The overall average retrieval performance is the mean of average retrieval performance over all classes.

5.2.1. Humster3D videos

Humster3D videos[10] contain a wide range of videos of 3D models including vehicles (1068), furniture (375), electronics (104), animals & plants (30) and life & leisure (28). We selected a subset containing 16 videos in the following 4 categories for our experiments: animals (4), vehicles (4), LCDs (4) and i-phones (4). Each video contains 100 views and each view was resized to 24 × 42 pixels. Fig. 12(a) shows example images from these 16 videos, and

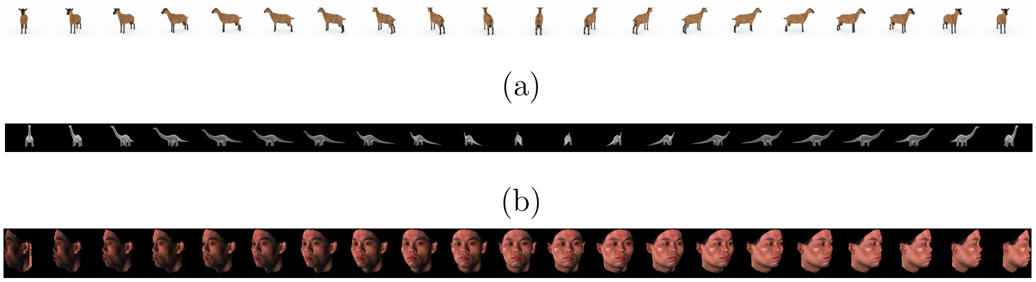
[10] http://humster3d.com/



(a) (b)

(c)

Figure 12: Example images from 3D datasets. (a) Humster3D videos. First row: animals; second row: vehicles; third row: LCDs; fourth row: i-phones (b) Princeton 3D models. First row: apatosauruses; second row: dogs; third row: horses; fourth row: sharks; fifth row: trexes (c) Vetter's 3DFS database (100 subjects).



(a)

(b)

(c)

Figure 13: Example of down-sampled views from 3D datasets. (a) Humster3D videos. (b) Princeton 3D models. (c) Vetter's 3DFS database.

Fig. 13(a) shows a series of down-sampled views of the first (top-left) video shown in Fig. 12(a).

Table 6 shows the rank-1 recognition rates under various combinations of query views and view-dependent dictionaries. The query views can be either full views or salient views, and the target gallery is given by either the salient view VDDs or the side view class VDDs. "Full views vs. salient view VDDs" means that full views are query and salient view VDDs are target. In addition, we further conducted leave-one-out (LOO) tests, where the view-dependent dictionary associated with the true class of the query object is excluded from the target gallery. As salient view VDDs are built using few salient views, tests with and without LOO were both conducted. On the other hand, since side view class VDDs are built using almost all full views (with BRVs excluded), only LOO tests were conducted.

As shown in Table 6, SVSR-VDDR obtained the highest rank-1 recognition rate. It also obtained the highest category (among animals, vehicles, LCDs and i-phones) recognition rates for most tests. Compared to SS-VDDR and VS-VDDR, the baseline-NN was able to obtain better performances because the between-category distances possessed in the target gallery are large enough. The overall average retrieval rates among 16 classes (eight target views retrieved for each query image) of baseline-NN, SS-VDDR, VS-VDDR and SVSR-VDDR are 6.29, 5.61, 6.75 and 7.04, respectively. The SVSR-VDDR obtained the best the overall average retrieval performance.

5.2.2. Princeton 3D models

Princeton 3D models (version 1) [21] contain a database of 1814 3D polygonal models collected from the internet. We selected a subset containing 20

Experiments \ Algorithms	baseline-NN	SS-VDDR	VS-VDDR	SVSR-VDDR
Full views vs. salient view VDDs	99.84	96.90	**100**	**100**
Full views vs. salient view VDDs (LOO)	**93.25**	75.60	88.75	90.35
Full views vs. side view class VDDs (LOO)	93.96	89.80	90.85	**92.50**
Salient views vs. salient view VDDs (LOO)	**92.03**	76.56	89.06	91.41
Salient views vs. side view class VDDs (LOO)	92.11	85.94	92.19	**93.75**
Average	94.24	84.96	92.17	**93.60**

Table 6: Rank-1 recognition rates (%) on the Humster3D videos.

models across the following 5 animal categories for experiments: apatosaurus, dog, horse, shark, and trex. We extracted 90 views from each model and each view was resized to 30 × 30 pixels. Fig. 12(b) shows example images from these 20 models, and Fig. 13(b) shows a series of down-sampled views of the first (top-left) model shown in Fig. 12(b).

Table 7 shows rank-1 recognition rates. While the proposed SVSR-VDDR obtained the second highest recognition rate, it ranks the highest in a majority (3 out of 5) of category recognition tests. Moreover, the baseline-NN obtained the lowest average recognition rate. This can be explained by the fact that the between-category distances among the target classes are no longer large. In fact, compared to the Humster3D videos, more class outliers may exist in the target gallery from this dataset. The overall average retrieval rates among 20 classes (eight target views retrieved for each query image) of baseline-NN, SS-VDDR, VS-VDDR and SVSR-VDDR are 5.48, 4.81, 6.36 and 6.18, respectively.

5.2.3. Vetter's 3D face database

Vetter's 3D face database [1] contains 100 face models. We extracted 60 views (rotated from $0^\circ \sim 180^\circ$ with respect to the Y axis) from each model and resized each view to 30 × 30 pixels. Fig. 12(c) shows example images from all 100 models, and Fig. 13(c) shows a series of down-sampled views of the first (top-left) model shown in Fig. 12(c).

Table 8 shows face recognition rates. The proposed SVSR-VDDR obtained the highest recognition rate. The proposed SVSR-VDDR obtained the best average retrieval performance. In particular, the overall average retrieval rates among 100 classes (eight target views retrieved for each query

Experiments \ Algorithms	baseline-NN	SS-VDDR	VS-VDDR	SVSR-VDDR
Full views vs. salient view VDDs	85.60	77.93	89.91	**91.42**
Full views vs. salient view VDDs (LOO)	57.50	54.86	63.11	**63.99**
Full views vs. side view class VDDs (LOO)	61.64	76.30	76.71	**77.41**
Salient views vs. salient view VDDs (LOO)	64	**75**	73.75	68.13
Salient views vs. side view class VDDs (LOO)	70.56	**84.38**	82.50	82.50
Average	67.86	73.69	**77.20**	76.70

Table 7: Rank-1 recognition rates (%) on the Princeton 3D models.

image) of baseline-NN, SS-VDDR, VS-VDDR and SVSR-VDDR are 2.52, 2.65, 2.71, 2.75, respectively.

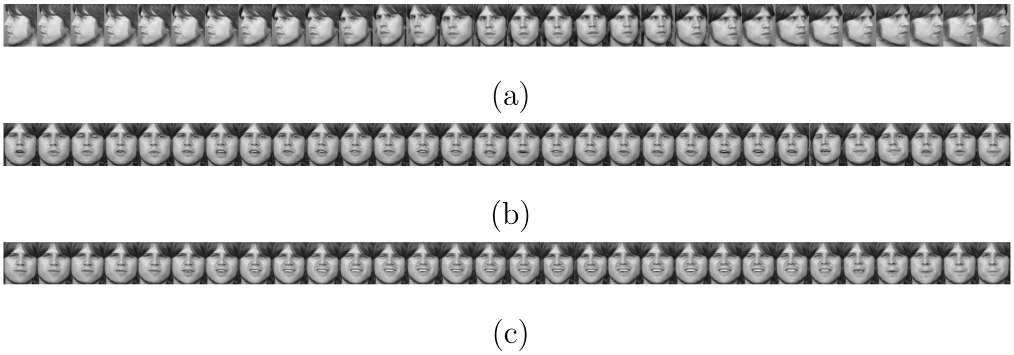
Experiments \ Algorithms	baseline-NN	SS-VDDR	VS-VDDR	SVSR-VDDR
Full views vs. salient view VDDs	31.47	33.16	33.87	**34.41**

Table 8: Rank-1 recognition rates (%) on Vetter's 3D face database.

5.2.4. Human ID database

In this subsection, we demonstrate the effectiveness of the proposed SVSR-VDDR on video-based face recognition for real people. The Human ID database [16] contains videos of human faces and people, which is useful for testing algorithms for face and person recognition. Complete data sets in this database are available for 284 subjects. In our experiment, videos of a subset of 60 out of 284 subjects were chosen. For each of these selected subjects, there are videos of moving facial mug shots, facial speech and dynamic facial expressions. Fig. 14 (a), (b) and (c) shows 30 cropped face images of one subject from its moving facial mug shots, facial speech and dynamic facial expression videos, respectively. Similar to the Vetter's database used for our experiments in section 5.2.3, the facial mug shot video contains poses from the left side pose to the right side pose

The face region of each frame extracted from the selected videos was properly cropped and resized to 30 × 24 pixels as a view. We used salient views from the moving facial mug shot videos to construct salient view VDDs, and evaluated these dictionaries using query full views from the same subject's



(a)

(b)

(c)

Figure 14: Example cropped face images from videos in the Human ID database. Types of videos include: (a) moving facial mug shot (b) facial speech and (c) dynamic facial expression.

moving facial mug shot videos, facial speech videos, and dynamic facial expression videos. Table 9 shows rank-1 face recognition rates among 60 classes. As shown, the proposed SVSR-VDDR obtained the highest (average) recognition rates. Comparing different video types, we observe that faces of subjects in the speech and expression videos appear in a single frontal pose, which can be accounted for by the view-dependent dictionaries as the moving facial mug shot videos also contain frontal face images. However, low recognition rates were obtained on these videos. This can be explained by the fact that these videos contain facial variations that are novel to the original facial mug shot videos, and hence are more challenging for recognition.

5.2.5. Discussion

Among all compared methods, we observed VS-VDDR and the proposed SVSR-VDDR obtained close performances. This can be explained by the fact that both VS and SVSR aim to find object representative views. The slight difference is that the VS minimizes the cost as a linear combination

Experiments \ Algorithms	baseline-NN	SS-VDDR	VS-VDDR	SVSR-VDDR
Full views of moving facial mug shot videos vs. salient view VDDs	67.63	68.91	77.87	**82.45**
Full views of facial speech videos vs. salient view VDDs	20.52	15.00	43.33	**53.33**
Full views of dynamic facial expression videos vs. salient view VDDs	28.71	20.00	**45.00**	**45.00**
Average	38.95	34.64	55.40	**60.26**

Table 9: Rank-1 recognition rates (%) on the Human ID database.

of *diversity* and *square error*, while the proposed SVSR finds representative views that either contain more sides (BRVs) or minimize the reconstruction error (SRVs). The SS, on the other hand, defines the saliency as the information relative to a representation. It turns out that the SS finds discriminative views. While discriminative views are not necessarily representative, the SS is not optimal for object recognition and retrieval applications. Furthermore, the baseline-NN works well only when the within-class variation is small and between-class distances are large enough in the target gallery. It is sensitive to a few outliers in the target gallery that either increases the within-class scatter and/or decreases the between-class distances.

5.3. Visual Hull-based View Synthesis

As shown in section 2.1, BRVs are candidates of representative views that can be used to successfully synthesize full views of the object. In this section, we experimentally evaluate the performance of visual hull based view synthesis for all compared algorithms. Our experiments were conducted on Vetter's 3DFS database [1].

Given a view, we use image-based visual hull [14], [30] to build correspondences between the view and each synthesized view. All pixels that can be seen from synthesized view are mapped from the corresponding pixels of the given view. When the objective view is located between two given views, it can be reconstructed using the two synthesized views from the given views, according to the relative perspectives between itself and the two given views. The number of pixel columns from either of the synthesized views is determined by the ratio of the perspective between the objective view and one synthesized view, to the perspective between the objective view and the other synthesized view. To illustrate this idea further, let $\mathbf{z}_{\theta_d}$ be the objective view, and two given views be denoted by $\mathbf{z}_{\theta_1}$ and $\mathbf{z}_{\theta_2}$, where $\theta_1 \leq \theta_d \leq \theta_2$ are the view perspectives with respect to the Y axis. Let $\hat{\mathbf{z}}_{\theta_1}$ and $\hat{\mathbf{z}}_{\theta_2}$ denote the two synthesized views from $\mathbf{z}_{\theta_1}$ and $\mathbf{z}_{\theta_2}$, respectively. Let C be the number of columns of $\mathbf{z}_{\theta_d}$ in its 2D matrix form. Then, at θ_d, the reconstructed view, $\tilde{\mathbf{z}}_{\theta_d}$ is synthesized in such a way that its right $\lfloor C\frac{\theta_d-\theta_1}{\theta_2-\theta_1}\rfloor$ columns are mapped from the same right $\lfloor C\frac{\theta_d-\theta_1}{\theta_2-\theta_1}\rfloor$ columns of $\tilde{\mathbf{t}}_{\theta_2}$, while its left $\lceil C\frac{\theta_2-\theta_d}{\theta_2-\theta_1}\rceil$ columns are mapped from the same left $\lceil C\frac{\theta_2-\theta_d}{\theta_2-\theta_1}\rceil$ columns of $\tilde{\mathbf{t}}_{\theta_1}$. On the other hand, if the objective view is not located between two given views, then all columns of its reconstructed view are directly mapped from the synthesized view of the closet given view.

Fig. 15 illustrates an example of the reconstructed view at 0° using two synthesized views from two given views at $-45°$ and 30°, denoted by "SV1" and “SV2”, respectively. The number of columns from the left of the reconstructed view contributed using the same columns of the synthesized view at $-45°$, and the number of columns from the right of the reconstructed view contributed using the same columns of the synthesized view at 30°, have a ratio of 30 to 45, which are perspectives between the objective view to the given views at 30° and $-45°$, respectively. The reconstructed view at 0° is observed to have a shorter distance to its ground-truth than the other two synthesized views, either of which is synthesized using only one given view.

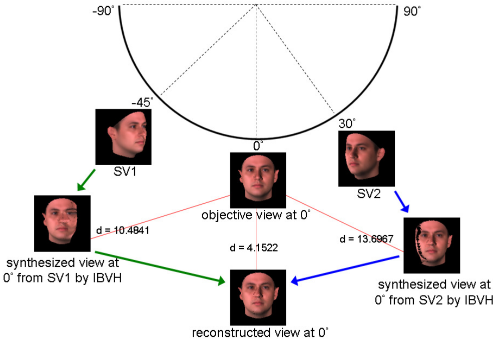


Figure 15: Visual hull based reconstruction. The reconstructed view at 0° using two synthesized views from two views at $-45°$ and 30°, has a shorter distance to its ground-truth than the other two synthesized views, either of which is synthesized using only one given view.

Table 10 shows average reconstruction errors using two given views produced from different algorithms on the Vetter’s 3D face database. Each view is resized to 112×95 pixels. For our SVSR method, two BRVs with the highest boundary scores computed using (1) are selected as given views. The reconstruction error is computed using the ℓ_2-norm distance between the desired view and the reconstructed view in the normalized grayscale. The “baseline1” refers to the scenario that two given views are randomly selected, while “baseline2” refers to the scenario that two given views are fixed at $-45°$ and $45°$.Fig. 16 shows the average reconstruction errors versus subject indices $(1 \sim 100)$ and perspectives $(-90° \sim 90°)$, respectively. As shown in Table 10 and Fig. 16, the proposed SVSR obtained the lowest average reconstruction errors among other compared methods.

Experiments \ Algorithms	baseline1	baseline2	SS	VS	SVSR
Average reconstruction errors	20.4584	18.8793	19.0119	19.0483	**18.6331**

Table 10: Average reconstruction errors on Vetter’s 3D face database. The reconstruction error is computed as the ℓ_2-norm distance between the reconstructed view and its ground-truth view in the normalized grayscale.

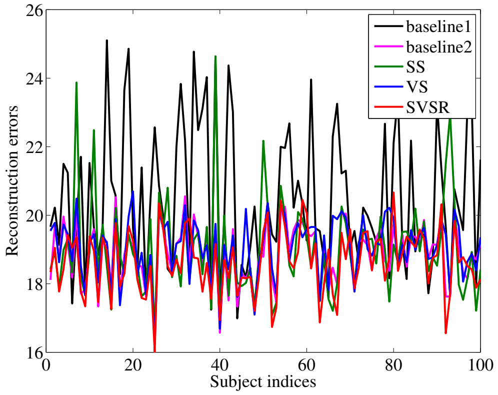


Figure 16: Average reconstruction errors versus subject indices on Vetter’s 3D face database.

6. Conclusion

We presented a two-stage approach based on sparse representation to find the salient views of an object. The first stage computes the spread metric and boundary scores to estimate boundary representative views. Using these estimated representative views, full views are roughly partitioned into different side view classes. In the second stage, side representative views are determined that have minimum class sparse-to-full reconstruction residuals. We constructed view-dependent dictionaries using the salient views and side view classes for applications in 3D object recognition and retrieval. We related the view-dependent dictionaries with the geometry across views. These dictionaries can represent the object in an informative way. Through a series of experiments on four publicly available 3D datasets, we demonstrated the effectiveness of our approach compared to the two existing state-of-the-art algorithms and one baseline method.

We are currently extending our work to view selection among the full 3D views taken at all perspectives (rotations with respect to all three axes) in various distances. Another important research direction is to extract features that are both class representative and class discriminative for 3D view selection and object recognition. We will also evaluate the robustness of our approach to noise and occlusions.

Acknowledgment

The work of YCC was partially supported by a Co-operative Agreement from the National Institute of Standards and Technology (NIST) under the Grant 70NANB11H023. The work of VMP and RC was partially supported by an ONR grant N00014-12-1-0124. The identification of any commercial product or trade name does not imply endorsement or recommendation by NIST.

[1] 3dfs-100 3 dimensional face space library. *University of Freiburg, Germany*, 3rd version, 2002.

[2] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, 2006.

[3] Volker Blanz, Michael J. Tarr, and Heinrich H. Bülthoff. What object attributes determine canonical views? *Perception*, 28:575–599, 1999.

[4] I. Chakravarty and Herbert Freeman. Characteristic views as a basis for three-dimensional object recognition. *Proceedings of SPIE*, 336:37–45, 1982.

[5] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, 1997.

[6] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.*, 20(1):33–61, 1998.

[7] Shuang Chen and Herbert Freeman. Characteristic-view modeling of curved-surface solids. *International Journal of Pattern Recognition and Artificial Intelligence - IJPRAI*, 10:537–560, 1996.

[8] Yi-Chen Chen, Vishal M. Patel, Rama Chellappa, and P. Jonathon Phillips. Salient view selection based on sparse representation. *IEEE International Conference on Image Processing (ICIP)*, October 2012.

[9] Yi-Chen Chen, Vishal M. Patel, P. Jonathon Phillips, and Rama Chellappa. Dictionary-based face recognition from video. *European Conference on Computer Vision*, October 2012.

[10] Yi-Chen Chen, C. S. Sastry, V. M. Patel, P. J. Phillips, and Rama Chellappa. In-plane rotation and scale invariant clustering using dictionaries. *IEEE Trans. on Image Processing*, 22(6):2166–2180, June 2013.

[11] M. Elad, M.A.T. Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, June 2010.

[12] Herbert Freeman and I. Chakravarty. The use of characteristic views in the recognition of three dimensional objects. *Pattern Recognition in Practice*, 1980.

[13] B. S. Manjunath, S Chandrasekaran, and Y. F. Wang. An eigenspace update algorithm for image analysis. *Proceedings of International Symposium on Computer Vision - ISCV*, pages 551–556, 1995.

[14] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. *SIGGRAPH*, pages 369–374, July 2000.

[15] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[16] A. J. O'Toole, J. Harms, S. L. Snow, D. R. Hurst, M. R. Pappas, J. H. Ayyad, and H. Abdi. A video database of moving faces and people. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):812–816, May 2005.

[17] Vishal M. Patel, Tao Wu, Soma Biswas, P. Jonathon Philips, and Rama Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–965, June 2012.

[18] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *1993 Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[19] J. K. Pillai, V. M. Patel, Rama Chellappa, and N. K. Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1877–1893, September 2011.

[20] Oleg Polonsky, Giuseppe Patané, Biasotti Silvia, Craig Gotsman, and Michela Spagnuolo. What's in an image? towards the computation of the ”Best” view of an object. *The Visual Computer*, 21(8-10):840–847, 2005.

[21] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The Princeton Shape Benchmark. *Shape Modeling International, Genova, Italy*, June 2004.

[22] N. Shroff, P. Turaga, and R. Chellappa. Video pr*é*cis: Highlighting diverse aspects of videos. *IEEE Transactions on Multimedia*, 12(8):853–868, December 2010.

[23] Michael J. Tarr and David J. Kriegman. What defines a view? *Vision Research*, 41:1981–2004, 2001.

[24] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. on Information Theory*, 50(10):2231–2242, October 2004.

[25] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.

[26] R. Wang and Herbert Freeman. Object recognition based on characteristic view classes. *Proceedings of IEEE International Conference on Pattern Recognition*, pages 8–12, 1990.

[27] Jay Winkeler, B. S. Manjunath, and S. Chandrasekaran. Subset selection for active object recognition. *IEEE CVPR*, 2:511–516, 1999.

[28] J. Wright, Yi Ma, J. Mairal, G. Sapiro, T.S. Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, June 2010.

[29] John Wright, Allen Y. Yang, Arvinda Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:210–227, 2009.

[30] Zhanfeng Yue and Rama Chellappa. Synthesis of silouettes and visual hull reconstruction for articulated humans. *IEEE Trans. on Multimedia*, 10(8):1565–1577, December 2008.