Direct Sampling of Multiview Line Drawings for Document Retrieval

Cristopher Flagg
Cris@IR.CS.Georgetown.edu
Georgetown University

Ophir Frieder
Ophir@IR.CS.Georgetown.edu
Georgetown University

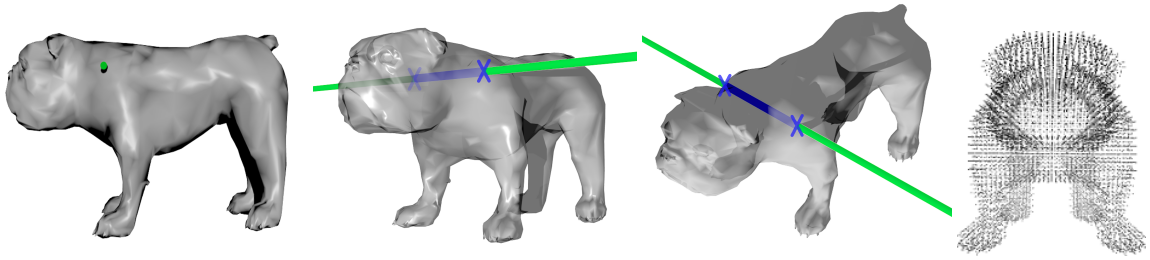


Figure 1: Sampling a point cloud from multiple images. Left: Source image with selected sampled point in green. Middle: Epipolar projections of the selected point onto orthogonal images. Right: Resulting sampled point cloud.

ABSTRACT

Engineering drawings, scientific data, and governmental document repositories rely on degraded two-dimensional images to represent physical three-dimensional objects. The collection of two-dimensional multiview images are generated from a set of known camera positions that are aimed directly at the target object. These images provide a convenient method for representing the original physical object but significantly degrades the interpretability of the object. The multiview images from the document repositories may be integrated to reconstruct an approximation of the original physical object as a point cloud. We show that retrieval methods for documents are improved by directly sampling point clouds from the multiview image set to reconstruct the original physical object. We compare the retrieval results from direct image retrieval, multiview convolutional neural networks (MVCNN), and point clouds reconstructed from sampled images. To evaluate these models, we trained them on line drawings generated from models in the ShapeNet Core data set. We show retrieval of the reconstructed object is more accurate than single image retrieval or the multiview image set retrieval.

DocEng '20, September 29-October 2, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-8000-3/20/09…$15.00
https://doi.org/10.1145/3395027.3419583

CCS CONCEPTS

• **Information systems** → **Retrieval effectiveness**; **Image search**; • **Computing methodologies** → *Point-based models*; *Mesh models*; Neural networks.

KEYWORDS

Document Repository, Document Retrieval, Point Cloud, Multiview Images

ACM Reference Format:
Cristopher Flagg and Ophir Frieder. 2020. Direct Sampling of Multiview Line Drawings for Document Retrieval. In *ACM Symposium on Document Engineering 2020 (DocEng '20), September 29-October 2, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3395027.3419583

1 INTRODUCTION

A typical representation for a model or a physical object in document-oriented repositories is a set of images of the object, taken from one or more distinct view points. This is referred to in the literature as a multiview representation of the object. Many databases used in industry today, such as national patent offices, describe physical models of objects (e.g. patents, industrial designs, and trademarks) as a multiview set of images. For a design patent, the United States Patent and Trademark Office states "[t]he drawings or photographs should contain a sufficient number of views to completely disclose the appearance of the claimed design, i.e., front, rear, right and left sides, top and bottom." [17]

Analysis of these images for the purpose of retrieval of similar objects poses a challenge to current systems. Multiview images fail to completely represent how the individual images interrelate to form the original object. We explore the background of multiview image collections and 3D model representations, ranging from

early systems that work directly with images to neural network models that seek to retrieve and classify point clouds based on their local and global structure. These methods focus on identifying important features from the image or model, identifying the correct classification for an unknown specimen, and retrieving similar images or models from a large document collection.

Figure 2 shows various representations of a 3D model. The original model is typically stored as a series of faces that create a mesh object (Figure 2a). This object is the ground truth for the physical object being considered. Many current techniques render this model in a series of views (Figure 2b) from a series of camera angles to create a smooth and well contoured image representation of the model. A collection of these images from multiple viewpoints is considered a set of multiview images. For most document repositories, the images provided are not rendered images, but line drawings in black and white (Figure 2c) that create an artist's rendition of the model. The contours of the model that are represented as gradations of shading in the rendered view are implied by single line contours in the line drawings. A silhouette (Figure 2d) of the model is defined as a binary image where the black portion of the image is considered 'inside' the model and the white portion is considered to fall 'outside' of the model and is considered background.

A point cloud (Figure 2e) that represents the original model is a collection of points in $\mathbb{R}^3$ which are distributed across the surface of the original 3D model. With the original mesh objects, the faces that define the outer object surface are neither uniform nor consistent. A naive method of generating a point cloud from a mesh model is to simply take the vertices of the model and use these as points in a point cloud. In the case of a cube, this would create only six points. No points would be located on the faces of the cube or along the cube's edges. Points located on the face and edge of the cube provide context to the overall shape of the object. To transform the ground truth models into point clouds, a uniform sampling of points are taken that lie on the surface of the model. This creates a shell of points lying on the faces of the model and not points internal to the model. Uniform sampling techniques are known and are one of the most common methods of transforming a 3D model from a mesh representation into a point cloud representation.

methods for creating a point cloud from existing mesh models are well known but cannot be applied when the original model is unavailable. Document repositories store only the multiview line drawing images that describe the shape of the model and not the original model itself. Steps must be taken to relate the multiview image collection so that the model may be reconstructed. We use the multiview line drawings of the original object to create a point cloud (2e) that closely approximates the surface of the original physical object. We show that this approximation of the original surface is similar to the original object and provides stronger retrieval than using the multiview line drawings provided within the repository.

We prove the following hypotheses:

- H1 - Directly sampling point clouds from multiview images creates reconstructed models that are sufficiently similar to the original model to perform retrieval.
- H2 - Document retrieval based on model reconstruction is more effective than retrieval using multiview images.

2 RELATED WORK

2.1 Patent Retrieval Systems

There exists some literature focused specifically on retrieval of documents from government databases, such as design patents and trademarks. The number of papers is limited and these approaches are applications of known methods to this particular domain.

An early attempt at a patent image retrieval framework [25] combines text and image feature extraction to retrieve patent images. The text data is used as a post retrieval concept filter to remove results that are in different technology domains.

While noting "design patent verification based on manual comparison is too labor-intensive, time-consuming and subjective," Zhu, et al., [28] uses Block-wise Dense SIFT (Block-DSIFT), Pyramid Histograms of Orientation Gradients (PHOG), and GIST as 2D image features. Features are extracted from the representative design images and clustered using K-Means and finally combined into an aggregate global feature for retrieval. They require all design images to be taken from a consistent view and does not address scaling or rotation invariant features.

Zeng and Yang [27] provide a synthesis design patent image retrieval method based on shape and color features. These moment invariant features are indexed for the query and related images are retrieved from the collection. The collection is a set of color images taken from a fixed view and does not address rotation or view invariant features when comparing design similarity.

Representing a physical object as a collection of images was an acceptable format when the total number of documents was relatively small and could be reviewed by manual inspection. As the number of documents increases, the ability to manually retrieve relevant documents becomes more difficult. Automated methods attempt to search these documents [1] [2] by isolating the representative images in the documents and then applying image retrieval techniques to create a set of searchable features for each document.

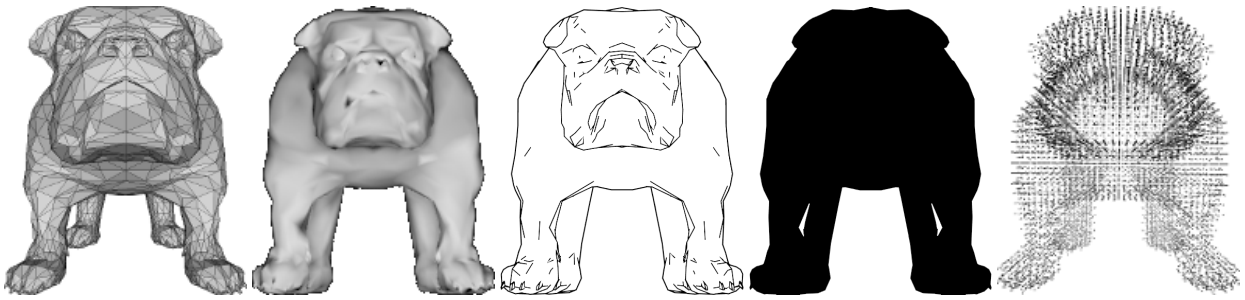
Lee, et al., [14] searches 3D trademarks comprising a collection of images. Discrete Fourier Transform is used to create the image feature for retrieval. The paper addresses rotation of the images by creating an additional set of rotated database images for each original image resulting in a much larger database.

Csurka [5] uses hand crafted features to identify the style of images from patents from the XRCE CLEF-IP 2011 [6] patent data set. The results are based on image style and do not attempt to classify the documents to which those images belong.

Song [22] uses ResNet-50 on a bespoke Chinese patent data set to classify the style of patent images. The work does not attempt any classification of the actual patents based on the images.

Flagg and Frieder [9] use features generated in an ad-hoc and offline manner and then combine them into features for model retrieval. This method demonstrated the strength of image reconstruction versus direct image based retrieval, but performed poorly compared to neural net methods applied to the same data set.

Yang [26] explores retrieval of technical diagram images and patent images and notes that "one of the main reasons and challenges [in diagram image retrieval] are lack of large representative benchmark data sets."



(a) Mesh Model (b) Rendered Image (c) Line Drawing (d) Silhouette (e) Point Cloud

Figure 2: Model representations.

2.2 Image Retrieval

Previous work focuses on either directly retrieving documents based on individual or multiview sets of images. The ImageNet Large Scale Visual Recognition Challenge 2012 provided the initial AlexNet [12] image recognition network and the ImageNet data set. Refinements to these networks have produced strong cross-domain image recognition and many publicly available implementations.

ResNet50 [10] is a 50 layer network that uses residual learning [10], which is closely related to the calculation of partial differential equations, to learn the difference of a feature from the input rather than computing the feature itself. This is achieved by using shortcuts between non-adjacent layers or blocks to compare the results of the calculated feature to the original data fed to the feature. This both decreases training time and addresses the vanishing gradients problem. Although Figure 3 represents the initial ResNet implementation, it illustrates both the repeated block format and the short cutting, referred to as identity mapping.

Many image analysis networks, such as ResNet [10], Inception [24], and VGG [21] are provided as pre-trained modules for use in neural network frameworks. These image modules may be integrated into other neural networks to perform image related tasks.

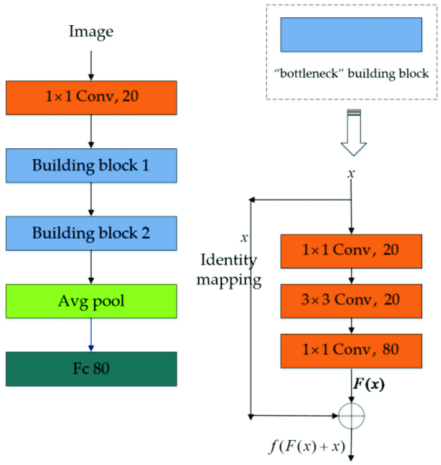


Figure 3: Example of ResNet architecture [7].

They typically include pre-trained weights, which allow the modules to be used without retraining.

For classification problems, the final layer is adjusted to match the number of classes in the data set. For retrieval problems, the network is trained on a classification problem. Once the network is trained, the next-to-last layer contains the feature vector that is appropriate to use for retrieval tasks.

2.3 Multiview CNN

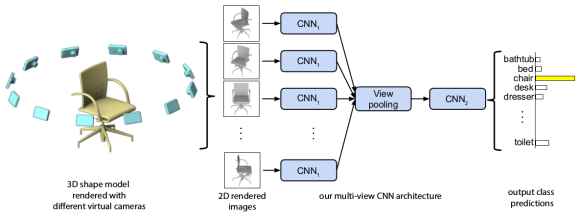


Figure 4: MVCNN architecture [23].

One of the initial and most direct methods of classification of 3D models was proposed by Su, et al., [23] whereby a Multi-View Convolution Neural Network (MVCNN) uses a multiview collection of images that represent the model. Identical cameras take pictures of the object from multiple non-orthogonal locations around the image (Figure 4). Each of these images is fed into a pre-trained VGG image recognition network. The feature vectors for each image are max-pooled and then used as input to a fully connected classification layer. This model attempts to directly classify the document using feature vector created from the multiview images.

2.4 Model Reconstruction from Images

Idesawa [11] uses the outer profiles of isometric drawing objects to extrude the parts before further refining the object. These views are then intersected to produce a final volume. This paper is focused primarily on generating polyhedra where the CAD drawings contain no additional surface detail. The points from the three-views are reconciled without the use of camera parameters or projections.

Laurentini [13] links silhouette-based images to the concept of a visual hull and defines visual hulls as the maximum space occupied by the object. The more silhouettes provided, the more refined the

visual hull becomes. This argued to be the closest approximation of the model available using intersections of volumes. The paper further defines active surfaces to extend the concept of the visual hull to non-convex surfaces.

Shum, et al., [20] looks to combine features from the image with extruded volumes estimated from the mutiview drawings provided. These methods use hidden lines and face reconstruction techniques for final construction of the models. Extruded volumes are intersected and are then combined with hidden line representations to form the final model.

Matusik, et al., [16] extends visual hulls to explore shading based on the contribution of multiple image views by matching pixels across an epipolar line. Computations are conducted in the image space and intersections are determined based on images taken from known locations in a specific order. The method creates an Image Based Visual Hull based on epipolar projections. The method attempts to reconstruct the entire hull by progressively scanning through all of the images and binning intersection points. This method reconstructs a mesh that represents the model shown in the images. While it shares some basis with the current method's epipolar focus and color sampling, it is computationally more complex and seeks to generate a model that is visually pleasing and not one that is focused on optimizing retrieval.

2.5 Point Clouds

In two of the foundational papers relating to point cloud manipulation, Qi et al., [18, 19] describe point clouds as unordered, locally related (the neighborhood of nearby points has meaning), and invariant under rigid uniform transformations. The model suggested learns to summarize point clouds as a sparse set of points and through repeated application of the algorithm allows an arbitrary number of points to be processed into a final feature vector.

The sampling layer uses iterative furthest point sampling, where an initial point is chosen at random. Then, iteratively, a new point is chosen that is furthest away from all points chosen thus far. This provides a uniform sampling of the point cloud.

In an improvement, pooling is used to deal with dense and sparse portions of the point cloud. Where the neighborhood of points is dense, layers are pooled and abstracted. Where layers are sparse, concatenation combines the points. Point alignment is factored into the model and based on a mini network of the local neighborhood and handled inline to the model.

Li, et al., [15] describes point clouds as irregular and unordered. This makes directly convolving kernels against features associated with the points difficult. To address these problems, a transformation is introduced to learn a mapping from the input points to a weighted set of input features associated with the points. This permutation transforms the points into a latent and potentially canonical order. The element-wise product and sum operations of the typical convolution operators are subsequently applied on the transformed features. This method generalizes typical CNNs to feature learning from point clouds.

3 SHAPENET CORE DATA SET

ShapeNet Core [4] is a publicly available, hence available for reproducibility, large-scale data set of 3D models collected from sources across the Internet. There are over 51,300 mesh models organized into 35,765 training models, 5,519 validation models, and 10,266 testing models. ShapeNet Core covers 55 common object categories (e.g., airplanes) and is commonly used to benchmark 3D tasks such as 3D model segmentation, 3D shape classification, 3D shape retrieval, and 3D model reconstruction from single images. The data set is available at https://shapenet.cs.stanford.edu/shrec16/.

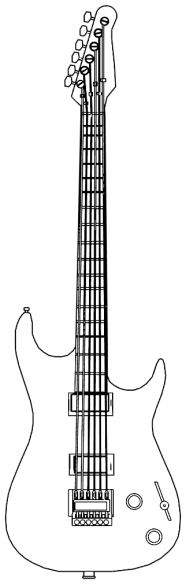


Figure 5: Line drawing of a guitar model.

The subject matter of design patents is diverse and broadly classified into 33 different areas. To simulate a design patent data set using the ShapeNet Core model collection, it is necessary to create a set of multiview images from the ShapeNet Core models. Rendered and line drawn images (Figure 5) are created for each model in the data set, as described in the Experimental Methods section. These multiview images are then used as the basis for image retrieval and model reconstruction. The classifications assigned to the original models are assigned to the generated multiview image collections. The original models are reserved as ground truth for comparison with models directly sampled from the multiview images.

4 EXPERIMENTAL METHODS

4.1 Line Drawings

Most reconstruction methods render mesh models under well defined lighting conditions. This shades the model contours as continual gradients and provides an image that is similar to a single color version of the original model. Documents pulled from document repositories are, however, rarely rendered in detail. Images are commonly hand drawn by skilled engineering illustrators. Given their wide use in both the engineering and patent setting, the models for this study were rendered as line drawings to best approximate the original repositories, as shown in Figure 6.

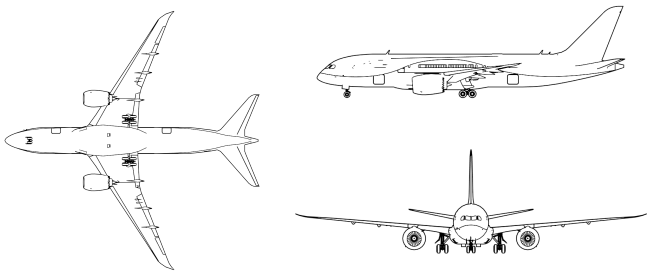


Figure 6: Individual three-view line drawings of an airplane.

To achieve a line drawn representation of the models that is similar to a technical or engineering line drawing, the models were rendered as solid white objects in Blender [3]. Black lines were added using Freestyle with visible edges and borders drawn in black. Additionally, edges with a crease angle above 2.44 radians were also added to the image. The results produce a close facsimile to actual line drawings of the objects as seen from a particular view. An example of a resulting line drawing is the guitar in Figure 5, which is generated as a front view of model 001014 from the training set. As a byproduct of the rendering process, the binary silhouettes are generated for each image and are defined as the non-transparent portions of the solid line drawn image. These images are used as the basis for both image retrieval and model reconstruction.

4.2 Image Retrieval

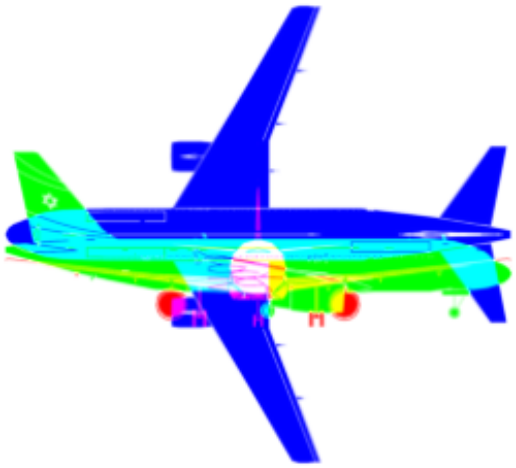


Figure 7: Combined three-view image. The front view is in the red image plane, the left view is in the green image plane, and the top view is in the blue image plane.

The previously described ResNet50 image model from the "torchvision" model library was used for image retrieval. ResNet50 takes a three plane RGB image while the images generated for use with the data set are single plane black and white line drawing images. The front, right, and top images are combined into a single RGB image as shown in Figure 7. The images were cropped and scaled to fit within a 244 square image, which is standard input for ResNet50. Additionally, each line image is arbitrarily rotated or flipped 25% of the time whenever a model is used in training or testing. This is recommended practice for image based recognition to reduce the orientation dependencies of the final classification. ResNet50 produces a 2048 feature vector as output. This is fed into a layer that reduces the output to the retrieval vector size of 256 and then produces a final output of 55, which is the ShapeNet Core class size. During training, the ResNet50 network was allowed to update its weights to better learn the specifics of the three-view images.

4.3 Multiview CNN

To combine more than three images as a single input, the previously described multiview CNN architecture was used. The front, back, left, right, bottom, and top images are provided as input to the MVCNN. Three of these views are shown in Figure 6. For consistency of comparison, the image analysis network used to generate feature vectors within the network was ResNet50. A feature vector is generated for each image and the resulting six vectors are max-pooled, so that the strongest element from each feature vector is represented in the combined feature vector. This combined feature vector is then fed into the standard set of fully connected layers described for the ResNet50 model. It is suggested in the paper and experimentally verified that a max-pool layer provides stronger retrieval results than an average-pool layer.

Since the pooling scheme originally described in the MVCNN paper [23] was improved upon in recent years, pooling is replaced in our implementation with an attention mechanism. Each of the six vector outputs $v_{1..6} \in \mathbb{R}^d$ from the six multiview images processed by ResNet50 are fed to a fully connected general attention layer $a = f(x \in \mathbb{R}^d)$. This layer learns a weighting for each feature $\sum_{n=1}^{6} (a \odot v)$ to allow vectors that contribute more strongly to the final answer to be weighted more highly. The learned weighting of the vectors allows more discriminative views to contribute more to the final image analysis results.

4.4 Model Reconstruction

4.4.1 Visual Hull. A visual hull is defined [13] where the visual hull $VH(S, R)$ of an object S relative to a viewpoint region R is a region of three dimensional space $\mathbb{R}^3$ such that, for each point $W \in \mathbb{R}^3$ that lies on the surface of $VH(S, R)$, if that point W is viewable within the viewpoint region R, then there is a point $X \in \mathbb{R}^2$ on the viewpoint region R where there is a vector starting at X and passing through W that contains at least a point of S. In the case of the current reconstruction method, multiple viewpoint regions R are each provided as uniform cameras positioned around the model. One common method of camera placement defines an orthogonal set of views where each camera is placed on the face of a cube surrounding the model. Another camera arrangement is in non-orthogonal placement which orients multiple cameras, each pointed at the centroid of the model, at a set elevation (e.g., 30 degrees about the x-y plane) and at equally spaced intervals around the z-axis. This is similar to the setup used in the MVCNN architecture. Additionally, planes may be defined as limits to the model when camera views do not fully encompass the model. In the non-orthogonal camera

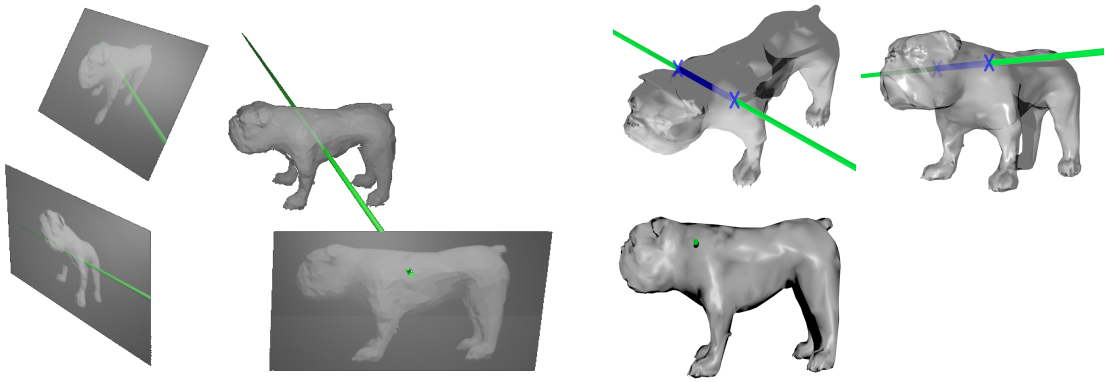


Figure 8: Epipolar projection of a selected point into non-orthogonal images. Left: Camera setup where images are taken from non-orthogonal views. The point chosen from the selected image is represented by a green dot and projected from this image into and through the visual hull. The cameras show the epipolar projection of the green line onto the camera images. Right: Epipolar intersections projected onto the camera images. The blue lines represent intersection of the visual hull within the projected images. The blue Xs represent the possible points located on the outside of the visual hull.

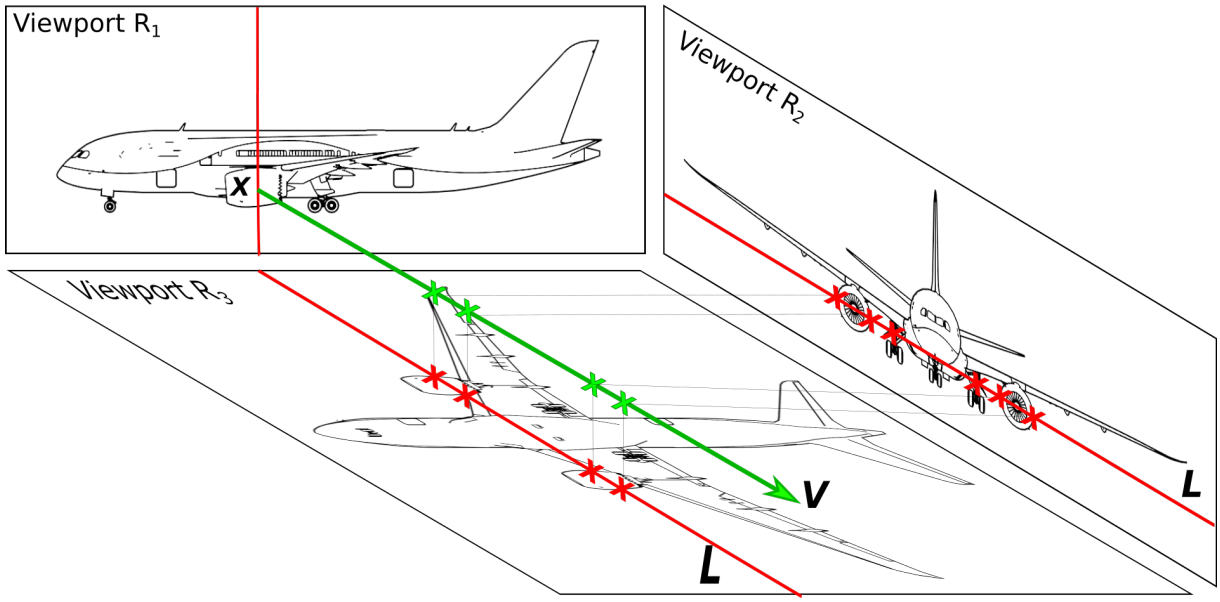


Figure 9: Projection of point X from the right side multiview image, viewpoint region R_S. The green line represents the vector projection L of X. The red lines on the front and bottom multiview images are the epipolar projections of L onto the image. The red Xs represent the intersections of the L with the image silhouette.

setup described, a floor plane may be required since no cameras are facing the bottom of the model. In all cases, each camera views a portion of the model. These cameras represent viewpoint regions R and allow a visual hull of the model to be reconstructed.

4.4.2 Epipolar Geometry. Given two viewpoint regions R_1 and R_2, with known focal points and camera characteristics, given a point $W \in \mathbb{R}^3$ that exists on the surface of $VH(S,R)$, a view $X \in \mathbb{R}^2$ of W in R_1 and a view $X' \in \mathbb{R}^2$ of W in R_2, it may be possible to triangulate the location of W. A vector V is projected from X orthogonal to the focal plane of R_1. The length of this vector is the distance from X with respect to the viewpoint region R_1 to point W. When the vector V is viewed from viewpoint region R_2, it appears as an epipolar line L in viewpoint region R_2, as shown in Figure 10. The location of the projection of W onto L in R_2 provides enough information to fix the length of V. If the point W is not viewable from R_2 then it is not possible to locate W from that view.

For document repositories and similar multiview image collections, the extrinsic camera parameters are known or may be inferred

and are consistent between cameras. Additionally, the spacing of the camera arrangement is known or may be inferred from the multiview images. Each image may be considered a viewpoint region R viewing model S. Any point within the image silhouette is considered within the projection of visual hull $VH(S, R)$ onto R and any point marked as background is not a projection of the visual hull $VH(S, R)$ onto R. We have additional grayscale value information in the form of the intensity of any of the points located within the projection of the visual hull onto R.

In the orthogonal view, two opposing viewpoint regions are unable to locate a point W in space since no epipolar line is projected from the source viewpoint region R_1 onto the other viewpoint region R_2. These corresponding views may not be used together to locate W. With the non-orthogonal camera setup this is not an issue and no viewpoint regions R are excluded from consideration.

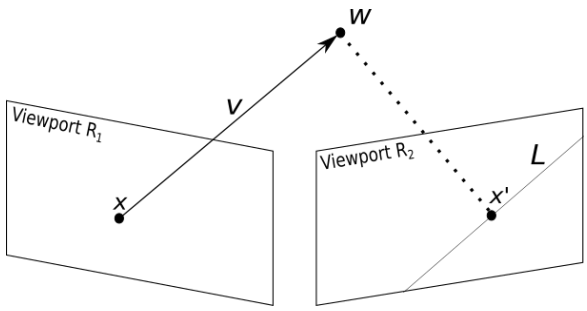


Figure 10: Projection of epipolar Line L into Viewpoint Region R_2 based on point X in Viewpoint Region R_1.

An example of these projections is shown in Figure 8 (non-orthogonal views). A view of the left side of the dog model S is used as the source viewpoint region R_s. A point X is chosen within the silhouette boundary of the model image. This point is projected as a green vector V from X towards the model S. The portion of the dog model S intersected by the green vector V is shown in blue. The two points at which the green vector V intersect the exterior surface of the model are shown as two blue Xs. Two other viewpoint regions R_1 and R_2 show views of model S. These views include the epipolar projection L of green vector V projected from point X. Due to the camera locations from which the projections of viewpoint regions R_1 and R_2 are taken, only one blue X is visible in each image. The projection of these two points along the epipolar projection L back into R^3 give a candidate sample point on the surface S.

An example of these projections which more closely relates to document repositories is shown in the orthogonal arrangement of the viewpoint regions R_1, R_2 and R_3 in Figure 9. These three views are taken from the line drawings provided in Figure 6. A point X is chosen within the silhouette boundary of the side view of the plane in viewpoint region R_1. This point is projected as a green vector V from X towards the space in which model S would reside. The portion of the plane model S that would be intersected by the green vector V is projected into two other orthogonal viewpoint regions R_2 (the front of the plane) and R_3 (the top of the plane). These views include the epipolar projection L (shown in red) of vector V (shown in green) projected from point X. The points at which the epipolar projection L intersects the image silhouettes are shown as red Xs on the epipolar lines. There are four points at which the epipolar projection in R_2 match the epipolar projection in R_3. These points are shown as green Xs on the vector V and are candidate sample points on the surface S of the 3D model.

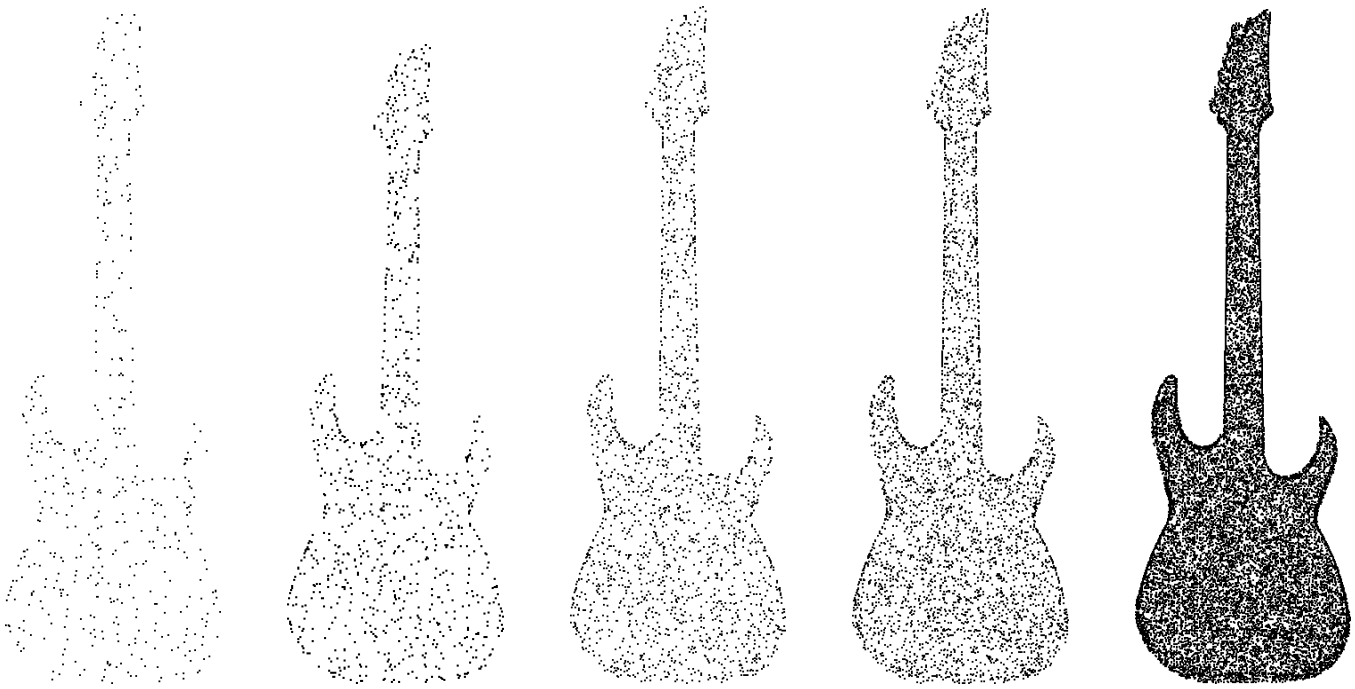
4.4.3 Sampling Methodology. Correspondence of points within multiple images may be matched and the resulting location fixed in space. Starting with multiview images where no original model exists, pairs of images may be used to identify possible points existing on the visual hull $VH(S, R)$. Each model S within the data set provides the ground truth for the reconstruction.

For two viewpoint regions R_1 and R_2 viewing model S, images from the viewpoint regions show the epipolar lines which have been projected onto both images. The intersection of the green vector projection L with the viewpoint regions R_1 and R_2 shows where the vector L intersects the visual hull $VH(S, R)$. In each region R_1 and R_2, the vector L intersects the silhouette of S projected into the viewpoint regions to create a line segment L_1s in viewpoint region R_1 and line segment L_2s in viewpoint regions R_2. The intersections L_1s and L_2s are possible points on the visual hull. Since we are reconstructing the point cloud, only points that are positioned on the outer shell of the visual hull $VH(S, R)$ are of interest.

When line segments L_1s and L_2s are projected back onto the green vector L they represent the possible projections of the original point X onto the visual hull. First, we are only interested in the points on L shared by both L_1s and L_2s. A point shared by both views would be a misprojection that matched the image silhouette in one viewpoint region but did not match the image silhouette in the other region. Additionally, only points that lie on the edge of the silhouettes represent points on the surface of visual hull $VH(S, R)$. Any points projected from L_1s and L_2s onto L that are on the edge of the silhouette are candidates for sampling.

To sample the points on the visual hull, first a candidate source point X is chosen at random from the interior of all of the image silhouettes from all of the viewpoint regions R. The viewpoint region from which X is sampled is considered R_s. From the other multiview images, two other images are chosen as viewpoint regions R_1 and R_2. In the case of orthogonally arranged cameras, neither R_1 nor R_2 may be chosen from the opposite viewpoint region (e.g. if the front image is chosen, neither R_1 nor R_2 may be the back view of the model). The previously described sampling methodology is used to identify the candidate silhouette edge points. If no points correspond to silhouette edges from R_1 or R_2, the point X chosen is ambiguous within viewpoint regions R_1 and R_2 and another X is sampled. If one or more points are candidates, then one is chosen at random. The point may be used with V and the viewpoint regions R_S camera parameters to determine the point X. Point X is added to the point cloud and the process is repeated until n distinct points are chosen, where n is less than or equal to the total sum of the pixels within the image silhouettes for all of the multiview images.

4.4.4 PointNet++ [19] and PointCNN [15]. PointCNN utilizes PointNet++ to identify local regions of a specific size. These regions are convolved using PointCNN to produce the aggregate point results. In our model, the PointCNN convolution is recursively applied to "project" information from neighborhoods into progressively larger contexts. Our model applies the following four point convolutions



(a) 512 points **(b) 1024 points** **(c) 2048 points** **(d) 4096 points** **(e) 20,000 points**

Figure 11: Point cloud model of a guitar sampled directly from multiview images using various point densities.

prior to final classification, where K refers to the kernel size and H refers to the number of hidden layers:

- Initial convolution over points to 38 features (k=5, H=32)
- Convolution from 48 to 384 features (K=12, H=64)
- Convolution from 384 to 1536 features (K=16, H=128)
- Convolution from 1536 to 3072 features (K=16, H=256)

With each convolution, the kernel generates features which are progressively larger to incorporate more of the local neighborhood of the point in the point cloud. The resulting feature vector is of length 3072 and has been progressively generated from a small initial set of features to a larger set of features. The specifics of the convolutional layer are detailed in the XConv implementation of PointCNN listed in the PyTorch Geometric [8] library.

In a similar structure to both the ResNet50 image classifier and MVCNN, this model takes the feature vector and sends it to a fully connected layer of the retrieval vector size of 256 and then produces a final output of 55, which is the ShapeNet Core class size.

4.4.5 Feature Vectors. A uniform classification portion of the network is constructed across all models to generate similar retrieval vectors. The final classification networks for each model consist of a layer of the retrieval vector size (in all cases a vector size 256). This layer is followed by a layer that reshapes the output of the retrieval vector later to the ShapeNet Core class size.

While classification over the 55 classes from ShapeNet Core is used for training of the ResNet50, MVCNN, and PointCNN networks, the final application of these networks is a retrieval problem. The networks are trained to match an image or model as closely as possible to the correct class. In doing so, later fully connected layers contain weights that help to discriminate between different classes based on their input. By using the activations of the next-to-last layer of the models as a retrieval vector, data about the models themselves is integrated with a class association. These vectors may be compared using a cosine distance to judge model similarity. No normalization of the final retrieval vectors is performed.

Because these networks are not trained in parallel to generate uniform retrieval vectors, it is not possible to compare a retrieval vector generated by ResNet50 with a retrieval vector generated by PointCNN and draw any meaningful conclusions. Since this paper compares the retrieval accuracy of multiview images versus reconstructed point clouds, it is not necessary to leverage a single classification network.

Initial experiments with ResNet50 shows that a retrieval vector of size 256 provides sufficient discriminatory power with minimal size vector. Larger vectors provide no increase in accuracy and smaller vectors provide degraded classification and retrieval results.

5 EVALUATION

All models are trained using the ShapeNet Core training and validation splits. Once trained, the testing split is used to generate retrieval vectors for each model. These retrieval vectors are compared pairwise using cosine similarity. The retrieval vector for each model is used to generate a list of sorted comparisons against the other testing models ranking similar models higher than dissimilar models. These similarity lists are evaluated using the ShapeNet evaluation tool to generate both micro and macro statistics about each model, as shown in Table 1.

	Micro					Macro				
	Precision	Recall	F1	mAP	NDCG	Precision	Recall	F1	mAP	NDCG
Original Models - PointCNN	*0.5084*	*0.8774*	*0.5819*	*0.7359*	*0.8162*	*0.1503*	*0.8515*	*0.2065*	*0.5359*	*0.5884*
Sampled Models- PointCNN	**0.4872**	**0.8355**	**0.5572**	**0.6772**	**0.7663**	**0.1432**	**0.7991**	**0.1961**	**0.4660**	**0.5270**
ResNet50 - Line	0.4672	0.7979	0.5338	0.6337	0.7276	0.1365	0.7463	0.1867	0.4193	0.4956
ResNet50 - Rendered	0.4575	0.7939	0.5256	0.6222	0.7197	0.1365	0.7577	0.1875	0.4323	0.5085
ResNet50 - Silhouette	0.4439	0.7681	0.5099	0.5792	0.6802	0.1324	0.7266	0.1815	0.3777	0.4513
MVCNN attention	0.3795	0.6509	0.4325	0.4559	0.5795	0.1109	0.6233	0.1524	0.2863	0.3776
MVCNN	0.2963	0.5342	0.3414	0.2803	0.4367	0.0918	0.5504	0.1284	0.1778	0.2741

Table 1: ShapeNet Core retrieval statistics. 'Original - PointCNN' provides an upper bounds for analysis since it is based on the original model and not image views rendered therefrom. Models sampled from images views provide superior retrieval across all metrics when compared with image based retrieval methods.

The retrieval results of the original point clouds provides an upper boundary on accuracy for the sampled point clouds. To evaluate the effectiveness of the proposed reconstruction method, we compare the original models to the generated models using well developed point cloud retrieval techniques. We extend the results of previous work by exploring document retrieval and model reconstruction using established techniques for image retrieval.

5.1 Comparison of Generated Models

For our experiments, we sampled all models in the ShapeNet Core data set using 2048 points. The models could be sampled at any point density, as shown in Figure 11. The resulting sampled point clouds are used as the inputs for the PointCNN network. Additionally, uniform sampling is used to create point clouds of the original models. A PointCNN model was trained using the point clouds sampled from the original models and a second PointCNN model is trained using the models sampled using our method.

An analysis of the results in Table 1 shows that the retrieval results of the original models (labeled as Original Models - PointCNN) when used with PointCNN and retrieval results of the point clouds sampled directly from the multiview images using our method (labeled Sampled Models - PointCNN) perform the best of the models considered. Since the same PointCNN model structure is the same for both sets of point clouds, the results confirm that the reconstructed models are sufficiently similar to the original point clouds for retrieval. This similarity in structure and retrieval accuracy confirms that, while multiview image reconstruction is not an exact reconstruction, the resulting structure is similar to the original.

5.2 Retrieval

The retrieval task is formulated as a 'Query by Example' problem. The original models and their corresponding classifications provide the ground truth for retrieval. While all retrieval is based on the original model, the goal is to simulate a collection of documents that contain a collection of multiview images.

In the case of image based retrieval, a collection of images is generated for each query model. Since this collection represents a single document, it may be classified under the ShapeNet Core classification scheme for the query model. A feature vector is generated from the image collection using one of the schemes described in the Experimental Methods section.

The ResNet50 and MVCNN models are trained using the line drawings generated from the ShapeNet Core training and validation splits. Once trained, the line drawings generated from the testing split are used to generate retrieval vectors for each model.

For ResNet50, separate models are also trained using silhouette images and rendered images from the ShapeNet Core data set. These results are also included in Table 1 and show that all three formats of multiview images provide similar retrieval results for both Micro and Macro NDCG rankings.

For MVCNN, both the original and attention based methods are evaluated using the generated line drawings.

For retrieval using the sampled point cloud, the query model is used to generate a multiview image collection, which is the basis from which to sample points (Figure 9) that could fall on the surface of the model. This sampled reconstruction may use at any number of points (see Figure 11). These points are fed into a version of PointNet++ to create a feature vector.

Once the feature vectors are created for any of the above methods, the retrieval vectors are compared pairwise using cosine similarity. For each query, a ranked list models from most similar to least similar is created. and evaluated using the ShapeNet Core evaluation tool to generate micro and macro statistics, as shown in Table 1.

The ShapeNet Core evaluation tool generates these statistics:

- **Micro and Macro Averaged**: Micro-averaged scores are averaged first within a category and then the categories are averaged to produce the final results. Micro averaged results give equal weight to classifications. Macro averaged scores give an unweighted average over the entire data set and give equal weight to the models.
- **Precision and Recall**: The evaluation tool uses only the first 1000 most similar records for these calcualtions.
- **F-score**: (Precision times recall) over (precision plus recall).
- **Mean Average Precision (mAP)**: Mean average precision, average of the precision@N for all results from 1 to N, where N is 1000.
- **Normalized Discounted Cumulative Gain (NDCG)**: The NDCG metric uses a graded relevance: 3 for perfect category and subcategory match, 2 for category and subcategory both being same as the category, 1 for correct category and a sibling subcategory, and 0 for no match. This is an attempt at capturing graded relevance between 3D models.

Point cloud based retrieval provides stronger retrieval results across all metrics. The retrieval results based on retrieval vectors from the sampled point clouds, while not as strong as retrieval based on point clouds from the original model, show improvement over other methods as well.

With respect to image based retrieval, rendered images of the models provide a similar result to both line drawings of the models and model silhouettes under both micro and macro NDCG. Surprisingly, the silhouette images include enough detail about the original object for a reasonable retrieval model. It is interesting to note that line drawings, which contain only binary data and simulated contours, perform as well as rendered images, which contain continual gradients to identify contour. This suggests retrieval for repositories based on line drawings are not significantly degraded from repositories that use rendered images.

Lastly, the combination of images proposed by MVCNN performs worse than the other image only methods. The original paper suggests a multiview set of twelve to eighty images of the object be used for classification and retrieval. The diminished results may be related to the limited number of views present in the orthogonal six-view images provided. The ResNet-50 models used for initial MVCNN classification may need additional retraining to improve performance. The addition of attention to the MVCNN network provided a boost to both micro and macro NDCG, but this method still did not perform as well as the other image based retrieval methods. Given the constraints of the document repository domain, direct retrieval based on three-view images of the model is more effective than either MVCNN or MVCNN with attention when applied to multiview images.

6 CONCLUSION

Retrieval of documents from large repositories is one of the canonical tasks in information retrieval. Documents based on multiview images present an additional difficulty. While direct image retrieval addresses the multimedia aspect of these documents, it does not provide the strongest retrieval context that may be used. When the original artifact described by the multiview document is reconstructed much greater retrieval accuracy is possible.

We described the direct sampling of point clouds from multiview images that represent a physical object. By reconstructing this object as a point cloud additional context may be gained and the retrieval and comparison of documents within the repository may be increased. The reconstructed model holds spatial information not present in the multiview images. When compared against direct image retrieval (ResNet50) and hybrid image retrieval (MVCNN), point cloud retrieval (PointCNN) provides improved retrieval accuracy.

REFERENCES

[1] Naeem Bhatti and Allan Hanbury. 2013. Image search in patents: a review. *International journal on document analysis and recognition* 16, 4 (2013), 309–329.

[2] Naeem Bhatti, Allan Hanbury, and Julian Stottinger. 2018. Contextual local primitives for binary patent image retrieval. *Multimedia Tools and Applications* 77, 7 (2018), 9111–9151.

[3] Blender Online Community. 2020. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam. http://www.blender.org

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

[5] Gabriela Csurka. 2017. Document image classification, with a specific view on applications of patent images. In *Current Challenges in Patent Information Retrieval*. Springer, 325–350.

[6] Gabriela Csurka, Jean-Michel Renders, and Guillaume Jacquet. 2011. XRCE's Participation at Patent Image Classification and Image-based Patent Retrieval Tasks of the Clef-IP 2011.. In *CLEF (Notebook Papers/Labs/Workshop)*, Vol. 2.

[7] Bei Fang, Ying Li, Haokui Zhang, and Jonathan Cheung-Wai Chan. 2018. Semi-supervised deep learning classification for hyperspectral image based on dual-strategy sample selection. *Remote Sensing* 10, 4 (2018), 574.

[8] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[9] Cristopher Flagg and Ophir Frieder. 2019. Searching Document Repositories using 3D Model Reconstruction. In *Proceedings of the ACM Symposium on Document Engineering 2019*. 1–10.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Masanori Idesawa. 1973. A system to generate a solid figure from three view. *Bulletin of JSME* 16, 92 (1973), 216–225.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[13] Aldo Laurentini. 1994. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence* 16, 2 (1994), 150–162.

[14] Chu-Hui Lee and Liang-Hsiu Lai. 2017. Retrieval of 3D Trademark Based on Discrete Fourier Transform. In *International Conference on Mobile and Wireless Technology*. Springer, 620–627.

[15] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*. 820–830.

[16] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. 2000. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 369–374.

[17] United States Patent and Trademark Office. 2019. Design Patent Application Guide. https://www.uspto.gov/patents-getting-started/patent-basics/types-patent-applications/design-patent-application-guide. February 3rd, 2019.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.

[20] Simon SP Shum, WS Lau, Matthew Ming-Fai Yuen, and Kai-Ming Yu. 2001. Solid reconstruction from orthographic views using 2-stage extrusion. *Computer-Aided Design* 33, 1 (2001), 91–102.

[21] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:cs.CV/1409.1556

[22] Gege Song, Xianglin Huang, Gang Cao, Wei Liu, Jianglong Zhang, and Lifang Yang. 2019. Enhanced deep feature representation for patent image classification. In *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, Vol. 11069. International Society for Optics and Photonics, 110690P.

[23] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*. 945–953. http://vis-www.cs.umass.edu/mvcnn/, code: https://github.com/jongchyisu/mvcnn_pytorch.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. arXiv:cs.CV/1512.00567

[25] Stefanos Vrochidis, Symeon Papadopoulos, Anastasia Moumtzidou, Panagiotis Sidiropoulos, Emanuelle Pianta, and Ioannis Kompatsiaris. 2010. Towards content-based patent image retrieval: A framework perspective. *World Patent Information* 32, 2 (2010), 94–106.

[26] Liping Yang, Ming Gong, and Vijayan K Asari. 2020. Diagram Image Retrieval and Analysis: Challenges and Opportunities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 180–181.

[27] Zhiyuan Zeng and Wenli Yang. 2012. Design Patent Image Retrieval Based on Shape and Color Features. *JSW* 7, 6 (2012), 1179–1186.

[28] Lei Zhu, Hai Jin, Ran Zheng, Qin Zhang, Xia Xie, and Mingrui Guo. 2011. Content-based design patent image retrieval using structured features and multiple feature fusion. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*. IEEE, 969–974.