Single Spin Image-ICP matching for Efficient 3D Object Recognition

Arvid Halma
arvid.halma@tno.nl*

Frank ter Haar
frank.terhaar@tno.nl†

Ernst Bovenkamp
ernst.bovenkamp@tno.nl*

Pieter Eendebak
pieter.eendebak@tno.nl*

Adam van Eekeren
adam.vaneekeren@tno.nl†

* TNO Science and Industry, Delft, The Netherlands
† TNO Defence, Security and Safety, The Hague, The Netherlands

ABSTRACT

A robust and efficient method is presented for recognizing objects in unstructured 3D point clouds acquired from photos. The method first finds the locations of target objects using *single* spin image matching and then retrieves the orientation and quality of the match using the iterative closest point (ICP) algorithm. In contrast to classic use of spin images as object descriptors, no vertex surface normals are needed, but a global orientation of the scene is used. This assumption allows for an efficient and robust way to detect objects in unstructured point data. In our experiments we show that our spin matching approach is capable of detecting cars in a 3D reconstruction from photos. Moreover, the application of the ICP algorithm afterwards allows us (1) to fit a query model in the scene to retrieve the car's orientation and (2) to distinguish between cars with a similar shape and a different shape using the residual error of the fit. This allows us to locate and recognize different types of cars.

Categories and Subject Descriptors

I.2.10 [**Computing Methodologies**]: Vision and Scene Understanding—*3D/stereo scene analysis*; I.4.8 [**Computing Methodologies**]: Image Processing and Computer Vision—*Scene Analysis*

Keywords

3D Object recognition, Spin image, Iterative Closest Point

1. INTRODUCTION

Today, a range of techniques exist to create a 3D model from an environment. Laser scanning and visual reconstruction are examples that result in unstructured point data. Various methods for the recognition of objects in an acquired scene are proposed. In this context spin images proved to be a powerful description to detect target objects in cluttered and occluded scenes [4]. Spin images are viewpoint independent shape descriptors that are created by mapping 3D points into a 2D histogram.

One disadvantage of the classical approach is the number of spin images that need to be matched, resulting in high computational costs. Another disadvantage is the need for surface normals at the observed points. Usually these surface normals are not available from the aforementioned acquisition methods. One way to overcome this is by estimating surface normals and then creating spin images only for object locations that are selected by some cheaper shape similarity filtering step [1]. However, estimating normals is far from trivial and other shape similarity measures often cannot handle partially observed models properly. Instead of adding different measures for filtering, we make the assumption that the global "up" direction of the scene and target objects are known. This prior knowledge can then be used to efficiently scan the scene for target objects.

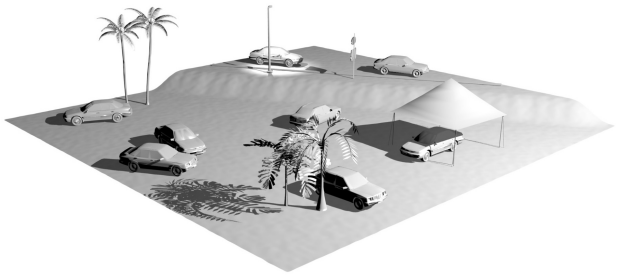

Figure 1: Scene with eight cars.

We follow [3] by first using spin images to detect the locations of target objects and then to use the iterated closest point (ICP) algorithm to retrieve the orientation. These methods complement each other, because spin images can be used to efficiently find objects in a larger scene independent of the object orientation, whereas the ICP algorithm is able to reconstruct the orientation of two equally sized objects. In addition this combination allows the parameters for spin image creation to be safely optimized for robustness as the ICP algorithm is not only used to orient the model, but is also used to verify the quality of the match; if the ICP match error between fitted model and scene object exceeds a predefined threshold, the match is discarded.

3DOR'10, October 25, 2010, Firenze, Italy.

This is an advantage over using spin images for object detection alone. In choosing parameters for spin images, there is always a trade-off between discriminative power and robustness. Smaller bin sizes result in descriptions which are better suited to match the finer object details, whereas larger bin sizes are less sensitive to noise and thus result in more robust detections.

The way spin images are used here is different from the way they were introduced [4]. Usually multiple spin images are used to represent the known model. The method proposed here uses only a *single* spin image that covers the known model with the origin located on the top-most point of the model and with a normal in the global direction (Figure 3). As mentioned before, no normal vectors per point are considered. The "support angle" to determine whether a point contributes to the spin image is therefore also discarded. For this special case a more efficient implementation of spin image creation is given.

Although we assume a known global direction, matching objects does not reduce to a 2D problem by projecting the points onto a single plane. Target objects can reside on different heights and can be occluded when seen from an arbitrary angle.

To demonstrate our approach we try to detect cars of different type in both an artificial and a real world scene. Figure 1 shows the artificial scene with two kinds of cars and other objects such as trees. Figure 8 shows the real world scene. For these scenes the concept of a known global "up" direction clearly makes sense: for all cars it holds that the roof of cars is likely to be above its wheels. Note, however, that as a consequence, cars that appear up side down in the scene will not be detected.

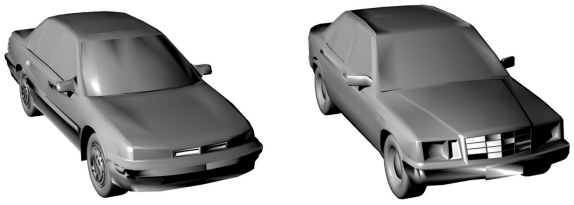

Figure 2: Two car types found in the scene. Respectively: car model 1 and 2.

2. APPROACH

Using a single spin image direction reduces the number of possible object poses significantly as only rotations around the z-axis are considered. Thus the matching problem is reduced from 6 dimensions (x, y, z, r_x, r_y, r_z) to 4 dimensions (x, y, z, r_x, r_y, r_z). It is a realistic assumption that a global direction is available when the observations were made under known conditions. However, when this global direction is not known, one can also attempt to find the ground-plane and then to retrieve the global direction from this plane.

In the rest of this section we assume the known global direction to be the vector $(0, 0, 1)^T$, i.e. the z or "up" direction, without loss of generality. When a different normal vector is preferred, both the model and scene point data need to be rotated [1]

Before scanning the scene P_{scene} for possible occurrences of a model P_{model} a spin image of P_{model} is to be created. The spin image is positioned at the point in P_{model} with the largest z-value. The size of the spin image $(\alpha_{max}, \beta_{min})$ is chosen such that the entire model is covered. The size of each bin ρ of the spin image resembles the resolution of the spin image. An example of the spin image for car model 1 is given in Figure 3.

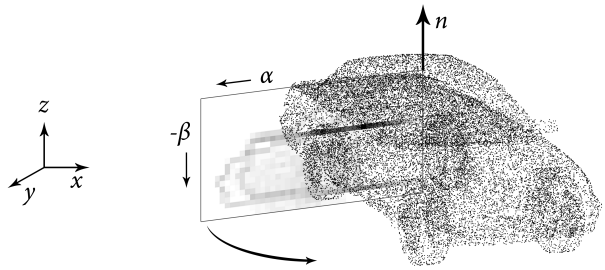

Figure 3: Single spin image for a car.

Given the input scene P_{scene} and the known model P_{model}, which are both unstructured point clouds, all locations T and orientations R of P_{model} in P_{scene} are now to be found. Algorithm 1 shows the steps which are taken to retrieve T and R.

In step 1 of Algorithm 1 all points $\vec{p}$ in the observed scene P_{scene} are taken as origin/pivot for the creation of a spin image. If the scene point cloud is much denser than the model point cloud one could use a uniform sampling of the points as origins. An efficient implementation for creating a single spin image at a given origin $\vec{o}$ reduces to Algorithm 2 when the global direction $(0, 0, 1)^T$ is assumed. Each spin image from the scene is compared with the spin image from the model. Two spin images A and B, each having $k = nm$ bins, are compared by determining the linear correlation coefficient [3]:

$$corr(A, B) = \frac{k \sum a_i b_i - \sum a_i \sum b_i}{\sqrt{(k(\sum a_i^2 - (\sum a_i)^2))\,(k(\sum b_i^2 - (\sum b_i)^2))}}$$

When a spin image created at a point $\vec{o}$ in the scene has a linear correlation coefficient greater than a certain threshold $corr_{thresh}$, that point is added to the set of detections D.

In general there will be multiple neighboring points for a single object that match the model well enough. In the experiments described in the next section, matching points are found to be centered at the roof of a car. In step 2 of Algorithm 1 detections that can be captured in the range of a single spin image are considered a cluster LD. Since the number of target objects in the scene is not known in advance, not all clustering algorithms are suitable. In our approach, clustering is based on the Delaunay tessellation created from the scene points, i.e. a mesh which connects all scene points. In this mesh all edges longer than α_{max}

[1]When the global direction is represented by some vector $\vec{m}$ instead of $\vec{n} = (0, 0, 1)^T$, a point $\vec{p}$ should be rotated using the quaternion product: $q\vec{p}q^*$, with $q = [(\vec{m} \times \vec{n})/\|(\vec{m} \times \vec{n})\|, \arccos(\vec{m} \cdot \vec{n})]$

Algorithm 1 Outline

1. Find set of detection locations, D, using spin images
$I_{model} \leftarrow spin(P_{model}, \alpha_{max}, \beta_{min}, \rho, top(P_{model}))$
$D = \emptyset$
for all $\vec{p} \in P_{scene}$ **do**
  $I_{scene} \leftarrow spin(P_{scene}, \alpha_{max}, \beta_{min}, \rho, \vec{p})$
  $corr \leftarrow correlation(I_{model}, I_{scene})$
  if $corr > corr_{thresh}$ **then**
    $D \leftarrow D \cup \{\vec{p}\}$
  end if
end for

2. Add cluster label to detections: $\vec{p} \rightarrow [\vec{p}, l]$
$LD = DelaunayCluster(D)$

3. Determine centroid for each cluster, C
$C = \emptyset$
for all $l \in labels(LD)$ **do**
  $D_l \leftarrow \{\vec{p} \mid [\vec{p}, k] \in LD \wedge l = k\}$
  $C_l \leftarrow 1/\|D_l\| \sum_{\vec{p} \in D_l} \vec{p}$
end for

4. Verify quality, find final locations T and orientations R
$Objects \leftarrow \emptyset$
for all $\vec{c} \in C$ **do**
  $P_{cutout} \leftarrow \{\vec{p} \in P_{scene} \mid \|\vec{p}_{x,y} - \vec{c}_{x,y}\| < \alpha_{max} \wedge \beta_{min} < \vec{c}_z - \vec{p}_z < 0\}$
  $(T, R, \epsilon) \leftarrow icp(P_{model}, P_{cutout})$
  if $\epsilon < \epsilon_{thresh}$ **then**
    $Objects \leftarrow Objects \cup \{(\vec{T}, \vec{R}, \epsilon)\}$
  end if
end for
return $Objects$

Algorithm 2 $I = spin(Pointset, \alpha_{max}, \beta_{min}, \rho, \vec{o})$

Require: spin normal in z-direction, i.e. $(0, 0, 1)^T$
$m \leftarrow \lceil \alpha_{max}/\rho \rceil$
$n \leftarrow \lceil \beta_{max}/\rho \rceil$
$I \leftarrow \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{m \times n}$
for all $(x, y, z) \in Pointset$ **do**
  $\alpha \leftarrow \sqrt{(\vec{o}_x - x)^2 + (\vec{o}_y - y)^2}$
  $\beta \leftarrow z - \vec{o}_z$
  if $\alpha < \alpha_{max}$ **and** $\beta_{min} < \beta < 0$ **then**
    $i \leftarrow \lfloor \alpha/\rho \rfloor$
    $j \leftarrow \lfloor \beta/\rho \rfloor$
    $I_{i,j} \leftarrow I_{i,j} + 1$
  end if
end for
return I

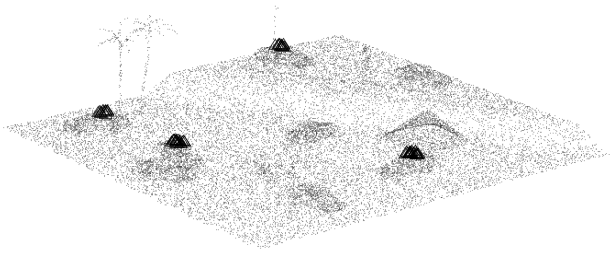

Figure 4: Triangles denote locations with well matching spin images.

are removed which results in a set of isolated connected components. Every point in such a connected component is assigned the same cluster label l.

In step 3 the center of gravity C_l is determined for each cluster l which is the mean location of the points in cluster l. The center is used as the unique detection location per object. Notice that for this procedure to work, at least three adjacent detections per object need to be found.

In step 4 points P_{cutout} in the neighborhood of a detection are selected from the scene. These points are then aligned using the ICP algorithm. The alignment procedure results in a final estimation of location T, orientation R and a fitting error. If the error is smaller than a certain threshold, the location and orientation are added to the final results. In more detail, after the detection of potential cars the orientation of the car remains unknown, because spin-images are rotation invariant. To this aid we employ the Iterative Closest Point (ICP) algorithm to establish the correct correspondence between the object model P_{model} and a part of the scanned scene P_{cutout}. Since the ICP algorithm requires an initial alignment, we initialize the ICP algorithm with twelve different orientations of the car model. The ICP algorithm then optimizes each of the initial orientation and position of the car model in the scene, and we keep the ICP refinement with the lowest Root Mean Square (RMS) distance. The twelve orientations that we use are different rotations around the up-vector in steps of 30 degrees. The ICP algorithm that we apply optimizes the RMS distance between closest point pairs of the model's vertices to the scanned scene.

$$d_{rms}(P_{model}, P_{cutout}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} e_{min}(p_i, P_{cutout})^2}$$

where e_{min} is the Euclidean distance between a point p of the P_{model} to its closest point in the P_{cutout}. For time-efficiency we use a Kd-tree of P_{cutout} for fast closest point pair selection and only fifteen iterations for the ICP algorithm. In the ICP algorithm we select eighty percent of the best matching point pairs are used to estimation the best location T and orientation R. These are common variants of the ICP algorithm as described in [5]. The assumption is that a model fits only well to scanned objects with the same 3D shape properties. When the residual ICP distance between the model and the scene is small enough, then we have an actual segmentation of the 3D shape.

3. EXPERIMENTS

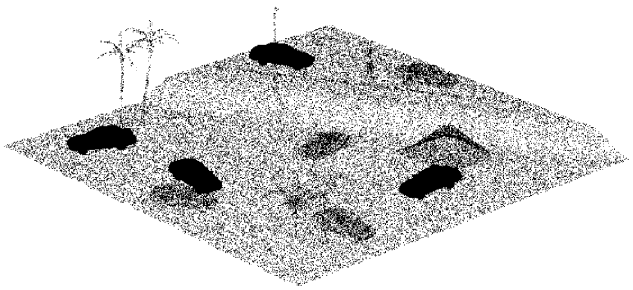

Figure 5: Models fitted in the scene.

In this section the method proposed above is applied to the detection of cars in different scenes. The robustness of the method is demonstrated, first by making an artificial scene more realistic and secondly, by applying it to real world data.

3.1 Artificial Dataset

The scene depicted in Figure 1 was created with a 3D modeling software program. The scene contains four cars of model 1 and four cars of model 2. All meshes in the entire scene were then uniformly sampled to create a point cloud containing 250,000 points. Each car in the scene contained roughly 6,000 points. In the same way a point cloud for car model 1 as shown in Figure 2 was created, which contains 25,000 points.

A spin image of the model was created with a resolution ρ of 0.1 m. The radius of the spin image was set such that the 4.2 m car easily fits in the image: $\alpha_{max} = 2.5$ m. The height of the spin image was taken to be smaller than the height of the car model, in order to discard the ground in the matching process: $\beta_{min} = -1.0$ m. The spin image was positioned at the point with the largest z-value, i.e. the rooftop. Figure 3 shows the spin image for this car model.

In the matching process, each point in the scene acts as a sampling point at which a scene spin image is made. All sampling points with a spin image correlation coefficient greater than $\epsilon_{thresh} = 0.6$ were considered in the clustering step. The geometric meaning of the threshold parameter is not a very intuitive and was therefore chosen somewhat arbitrarily. This fixed value was successfully used in all subsequent experiments from which we conclude that the method is not very sensitive to the choice of this parameter.

All four cars of model 1 were successfully detected. Figure 4 shows the detected cars in black. With coarser settings, a spin image bin size of $\rho = 0.2$ m, sufficient details are lost to consider the two car models equivalent and detect all eight cars in the scene. In other words, this allows for generalized searches for unknown models.

3.2 Data from a Single Viewpoint

As a bridge to apply the method to a real world dataset, we anticipate that a point cloud may be acquired from a single viewing direction. In particular, we assume the scene to be observed from above, resulting in a sparse or absent point sampling from the sides of objects in the scene. The representation of the model to be fitted can use this prior knowledge by preventing matching error for parts of the model that can not be seen from that direction. Figure 6 shows a point cloud of the car model sampled at faces that can be seen from above.

Figure 6: Car model sampled as if it was scanned from above.

The consequence of this step is that cars located below other objects will be lost. In particular, this holds for the rightmost car under the tent in Figure 1.

3.3 Data with Gaussian Noise

In addition to using projected data, the robustness of single spin matching in noisy data is tested. Different levels of normal distributed noise are added to the artificial dataset containing 7 cars (the partly occluded car is not counted). For each noise level it is analyzed how many object detections are found. The results are depicted in Figure 7.

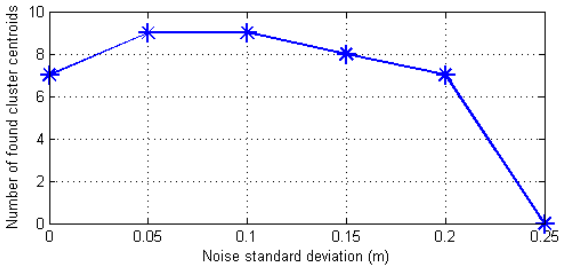

Figure 7: Influence of noise on Spin Matching.

It can be seen that the breakdown point lies at a noise level of approximately $\sigma = 25$ cm. For this amount of noise no cars are found anymore. Furthermore it is observed that for low noise levels false detections, where no car is present, are found. Note that all parameters of our method (including the threshold) are not changed for this experiment.

The subsequent ICP model fitting step is presented thoroughly when our method is applied on the real world dataset in the next section.

3.4 Real World Dataset

To obtain a real-life dataset visual reconstruction was performed using Bundler [6] and Patch-based Multi-view Stereo Software (PMVS2) [2] on several locations. In this paper we present results from the balcony of the TNO building. The balcony has a height of roughly 20 meters, and a width of nearly 80 meters. The reconstruction yielded a point density of roughly 1000 points/m^2 (Figure 8).

For detecting and fitting cars in this dataset, the same car model as for the artificial scene is used. The scene contains 13 cars. With the same parameters as before, 18 locations are marked by the spin image matcher, of which 12

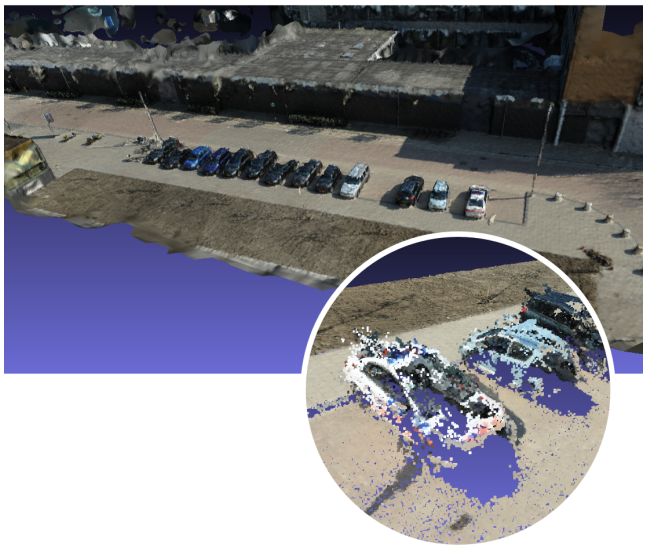

Figure 8: Visual reconstruction for the Stieltjesweg. From the right viewpoint the first impression of the reconstruction quality is good. The inset shows, however, that the cars are noisy and incomplete.

Figure 9: Two matching cars found at Stieltjesweg. The purple and brown car models are fitted in the point cloud.

are indeed good matches. At these 18 locations, the ICP algorithm tries to fit the car model. Based on the residual RMS distance we determine whether or not a car similar to our car model is present at the potential locations. Furthermore, it is important to investigate the effect of the number of model samples (n) used in the ICP algorithm.

0	$n = 100$	$n = 500$	$n = 1000$	$n = 2000$	init
1	172	197	193	199	wrong
2	106	114	124	121	good
3	173	175	179	180	wrong
4	150	158	170	162	good
5	140	161	164	159	good
6	112	114	123	117	good
7	135	158	159	159	good
8	128	116	122	123	good
9	142	158	158	160	wrong
10	84	94	100	106	good
11	135	139	150	143	wrong
12	119	145	147	159	good
13	103	98	108	105	good
14	83	88	87	91	good
15	183	189	192	176	good
16	148	167	167	170	wrong
17	131	145	159	148	good
18	152	185	187	188	wrong

Table 1: The residual RMS distance (mm) is determined for each model fit in the scene. The rows represent the fitted cars from left to right. On the potential cars locations 2, 6, 8, 10, 13, and 14 the model fits well. The columns show the results for a different amount of samples. The last column shows whether or not a good initialization was available.

In the following experiment we vary the number of random samples n selected from the top part of the car model in order to determine a good trade-off between a fast alignment and a robust alignment. Four sets of samples were used; 100, 500, 1000, 2000 random sets of samples, also shown in Figure 12. For each of these smaller subsets we use the ICP algorithm to acquire the optimal alignment on a spin location. Table 1 shows for each of the spin locations how well the ICP algorithm was able to fit the model in the scene. These results correspond to the fitted cars shown in Figure 10 from left to right. These results show that locations 2, 6, 8, 10, 13, and 14 are locations where the model fits very well and the residual RMS distance is low. An important observation is that for this set of six detections it does not matter if 500, 1000, or 2000 samples were used, the residual RMS distances are in these cases distinctive enough from the other 12 detections. The results for 100 samples appear to be less discriminative. Nevertheless, this means that the segmentation of this specific type of car can be done very efficiently with the use of only 500 surface samples and a RMS-threshold of 115 mm. The accepted car segmentations are shown in Figure 11.

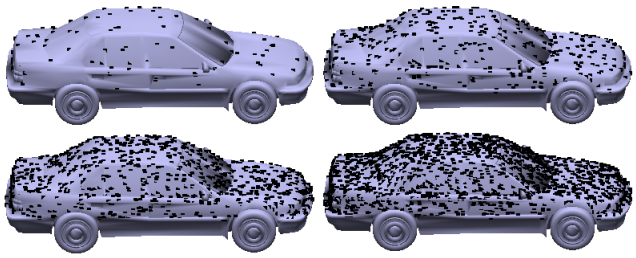

Figure 10: The selected sets of samples used in the ICP algorithm to detect the orientation of the cars.

Figure 11: The fitting results (n=500) on the selected spin locations. Several detections have wrong locations and orientations. Note that for the visualization the original 3D model is used.

Figure 12: The selected fitting results (n=500) based on the RMS threshold.

4. CONCLUSION AND FUTURE WORK

The proposed method successfully exploits the prior knowledge, the global scene orientation, to efficiently recognize objects in a scene. Whether the assumption of having a known global direction and therefore finding objects with 5 instead of 6 degrees of freedom is realistic, depends on the application domain. As demonstrated, an outdoor scene is an example for which this can be the case.

The proposed method is less computationally expensive than classical spin image matching. Firstly, it only considers objects rotated over a single axis instead of arbitrary rotations, and is therefore a problem with less degrees of freedom. As a consequence an optimized algorithm for spin image creation in this specialized case was given. In addition, it does not require expensive normal estimation from a point cloud to generate a spin image. Secondly, for each possible object location a single spin image match is performed, as opposed to comparing sets of spin images. It therefore is an attractive alternative for detecting objects in larger scenes.

The robustness of detecting objects was tested on an artificial scene containing Gaussian noise and in a real-world scene. The experiments showed that under controlled distortions of an artificial scene, the proposed method performed very well. With unadapted parameters, it was also able to retrieve equivalent cars in a sparser real-world dataset. As far as we could verify, this is the first time general object recognition and model fitting is presented in a 3D reconstruction from photos.

Depending on the viewpoint and occlusions in the scene cars may not be completely covered in the 3D reconstruction, in our future work we will tackle these challenges with the use of partial matching.

5. REFERENCES

[1] A. Patterson IV, P. Mordohai, and K. Daniilidis. Object detection from large-scale 3d datasets using bottom-up and top-down descriptors. In *European Converence on Computer Vision*. European Converence on Computer Vision 2009, October 2008.

[2] Yasutaka Furukawa and Jean Ponce. PMVS - Patch based Multi-View Stereo software. `http://grail.cs.washington.edu/software/pmvs/`.

[3] Andrew E. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 1997.

[4] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, may 1999.

[5] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *3DIM*, pages 145–152, 2001.

[6] Noah Snavely. Bundler - Structure from Motion software. `http://phototour.cs.washington.edu/bundler/`.