

A Geometry-Based Local Descriptor for Range Data

Fredrik Viksten
Image Coding Group
Dept. of Electrical Engineering
Linköping university, Sweden
viksten@isy.liu.se

Klas Nordberg
Computer Vision Laboratory
Dept. of Electrical Engineering
Linköping university, Sweden
klas@isy.liu.se

Abstract

We present a novel local descriptor for range data that can describe one or more planes or lines in a local region. It is possible to recover the geometry of the described local region and extract the size, position and orientation of each local plane or line-like structure from the descriptor. This gives the descriptor a property that other popular local descriptors for range data, such as spin images or point signatures, does not have. The estimation of the descriptor is dependant on estimation of surface normals but does not depend on the specific normal estimation method used. It is shown that is possible to extract how many planar surface regions the descriptor represents and that this could be used as a point-of-interest detector.

1. Introduction

Analysis of local image properties in 2D images have a long history, and a large variety of properties and estimation methods have been presented over the last two or three decades. Investigated properties include local orientation, phase, frequency and curvature [2, 7, 12, 8] etc. The estimation of such properties is related to detection of points-of-interest [13, 19, 21]. In the last decade, local descriptors [24, 17, 1, 20, 15] have seen much interest, mostly due to its success in recognition [17, 20] and similar areas such as pose estimation [15, 30, 31].

In the case of sparse 3D data, which describes points in a scene, e.g., produced by active range sensors or methods based on motion or base-line stereo, similar methods for describing local features have been developed. A very coarse division could be into model-free and model-based descriptors. The general approach for the model-free descriptors is based on detecting planar surfaces or other geometrical primitives to which a reference direction can be related, thus achieving rotation invariance. Once the reference direction has been established a descriptor, not bound by a mathe-

matical model is extracted [16, 3, 27]. When it comes to the model-based descriptors the previous work is a bit more diverse. Some are a bit more global [18] using bigger parts of the object and then relating them to each other on a symbolic level [5, 6]. This is closely related to the area of range data segmentation which saw much exploration in the 80s and 90s, see e.g., [32, 14] etc. Some related research is similar to the local image properties for 2D images in that they describe local properties such as curvature [29, 28].

In this paper we propose a representation / descriptor of the local region around each point in the data set. This descriptor can be analyzed in order to determine if the point is on a planar surface or close to or on an edge or a corner. Furthermore, in each of these cases the representation can be further analyzed to provide explicit descriptions of the parameters of the plane, or planes which meet in the edge or corner. We call this novel representation the *scene tensor* and it can be seen as an extension to the orientation tensor, [2, 11, 7]. The orientation tensor is a representation of local orientation, but it has its shortcomings since it only describes orientation of a single hyperplane. We will here describe a way of extending the orientation tensor so that it can contain more information. To be able to store more information than what is possible with the orientation tensor we wanted to

- Raise the order of the tensor above second order
- Use projective geometry.

Each of the steps have been investigated separately previously but the combination of them seems new. Results on the first point can be found in [26] and the use of projective geometry in tensors has been published in [25]. In this presentation we are mostly interested in the form of representation rather than the ways of estimating it. Even-though, ways of estimation are presented to show that the representation works as described. We have previously shown it to work in 2D [23, 31].

In the remaining part of this paper we will first try to motivate the development of the scene tensor. Following this

we will, in section 3, introduce some concepts from projective geometry, which is one of the building blocks in the construction of the representation. In section 4.1 and 4.2, we then show how points and planes can be represented in terms of symmetric matrices or tensors on projective spaces. These are then combined into the final representation in section 4.3 which also discusses overall structure and analysis of the representation. The final part of the theory, given in section 4.4, discusses how to analyse a specific tensor to find the geometric information it represents. Where as the presented representation can describe both plane and line-like structures this paper focuses on the plane-like structures.

2 Motivation

Recent developments in vision research, [9, 10], suggest that a future successful cognitive vision system will be dependent on effective information representations. Having many different representations for different descriptors could present a problem when designing a learning structure for a system where you want to be able to handle all those descriptors. As a result of this, we wanted to design a representation that can handle complex features, i.e. features that are combinations of other lower-level features. Having a representation that is the same for many combinations of local structures goes a way to achieve this. We might not have come all the way but nonetheless a new representation scheme has been developed.

Furthermore, a representation of this type could also play a role in managing the so-called bootstrapping problem where a robot with no knowledge of the outside world could use general descriptions of it to determine how it can interact with it to learn about various objects. If explicit 3D shape descriptors are available, the grasping and picking-up parts of this problem can be significantly reduced [4].

3 Points and hyperplanes

A point $\mathbf{x} \in \mathbb{R}^n$ is said to lie in a hyperplane described by the vector $\mathbf{l} = \|\mathbf{l}\|\hat{\mathbf{l}}$, $\mathbf{l} \in \mathbb{R}^n$, if

$$\mathbf{x}^T \hat{\mathbf{l}} = l_1 x_1 + \dots + l_n x_n = \|\mathbf{l}\| \quad (1)$$

with x_i and l_i being components of the vectors \mathbf{x} and $\hat{\mathbf{l}}$.

Given a vector representation of a point $\mathbf{x} \in \mathbb{R}^n$ we can extend this to projective space by the homogeneous representation

$$\mathbf{x}_H = \begin{pmatrix} x_0 \\ \mathbf{x} \end{pmatrix} \quad (2)$$

where the real number $x_0 \neq 0$. The choice of x_0 can be said to influence how large a change in the data that is needed for the vector \mathbf{x}_H to change in a significant way. We can

also extend the representation of the plane by \mathbf{l} into a dual homogeneous form by

$$\mathbf{l}^H = \begin{pmatrix} -\|\mathbf{l}\| \\ l_0 \hat{\mathbf{l}} \end{pmatrix} \quad (3)$$

where l_0 just as x_0 can not equal zero and $\mathbf{l}^H \in \mathbb{R}^{n+1}$.

If x_0 and l_0 were chosen to be equal to one, equation 1 would turn into

$$\mathbf{x}_H^T \mathbf{l}^H = -\|\mathbf{l}\| + \mathbf{x}^T \hat{\mathbf{l}} = 0 \quad (4)$$

If however x_0 and l_0 were chosen to be some other non-zero real number we need to define the scalar product of the space by

$$\mathbf{G} = \begin{pmatrix} l_0 & \mathbf{0}^T \\ \mathbf{0} & x_0 \mathbf{I} \end{pmatrix} \quad (5)$$

so that equation 4 would be expressed as

$$\mathbf{x}_H^T \mathbf{G} \mathbf{l}^H = \mathbf{x}_H^T \begin{pmatrix} l_0 & \mathbf{0}^T \\ \mathbf{0} & x_0 \mathbf{I} \end{pmatrix} \mathbf{l}^H = x_0 l_0 (\mathbf{x}^T \hat{\mathbf{l}} - \|\mathbf{l}\|). \quad (6)$$

The reason for calling the representation in equation 2 and equation 3 for ordinary and dual homogeneous form is that they actually lie in separate spaces. The only implication of this that we have to be wary of in this paper is that if we change the coordinate system of the space of \mathbf{x}_H we have to change the space of \mathbf{l}^H in the ‘‘opposite way’’. Let’s say that we transformed \mathbf{x}_H by the transformation \mathbf{H}_x , we would then need to transform \mathbf{l}^H by

$$\mathbf{H}_l = \mathbf{G}^{-1} \mathbf{H}_x^{-T} \mathbf{G} \quad (7)$$

where $^{-T}$ is the inverse and transpose. It should be noted that in a homogeneous format, ordinary or dual alike, multiplication with a non-zero scalar does not change the information that the vector carries. It is always a simple task to get the original \mathbf{x}_H or \mathbf{l}^H since the scalars x_0 and l_0 are assumed to be known and can be used for normalization.

4 Scene tensor in theory

We will in this paper use matrices to represent tensors. The tensor product will be marked as \otimes and can be thought of as an outer product between vectors representing tensors.

4.1 Second order contra-variant tensor

The second order contra-variant tensor, denoted \mathbf{S}_{20} for its two contra-variant indexes and zero co-variant ones, is defined as

$$\mathbf{S}_{20} = w \mathbf{x}_H \otimes \mathbf{x}_H = w \mathbf{x}_H \mathbf{x}_H^T \quad (8)$$

where $w \geq 0$ is some scalar used as a weight. The descriptor found in equation 8 only describes one point and in its null-space we find information on all hyperplanes passing through that point. To make the descriptor more interesting we need to integrate information in a local region \mathbf{R} around a point \mathbf{p} according to

$$\mathbf{S}_{20}^{\mathbf{R}}(\mathbf{p}) = \sum_{\mathbf{i} \in \mathbf{R}(\mathbf{p})} w(\mathbf{i}) \mathbf{x}_H(\mathbf{i}) \mathbf{x}_H^T(\mathbf{i}) \quad (9)$$

where w this time is a weighting function. This results in that the range of $\mathbf{S}_{20}^{\mathbf{R}}$ has a higher dimensionality than the range of \mathbf{S}_{20} . Looking at the result of equation 9

$$\mathbf{S}_{20}^{\mathbf{R}} = k \begin{pmatrix} x_0^2 & x_0 \mathbf{x}_c^T \\ x_0 \mathbf{x}_c & \mathbf{C} + \mathbf{x}_c \mathbf{x}_c^T \end{pmatrix} \quad (10)$$

where k is some constant, we can interpret its matrix form as being made up of weighted means, \mathbf{x}_c , of the points $\mathbf{x} \in \mathbf{R}$ and of the covariance \mathbf{C} for the same points. The null-space holds information on all planes going through the points in the tensor. The last operation must have decreased the dimensionality of the null-space since the range has increased. Therefore the set of possible planes passing through all points in $\mathbf{S}_{20}^{\mathbf{R}}$ must be smaller than for \mathbf{S}_{20} . In the following we will use the designation \mathbf{S}_{20} for $\mathbf{S}_{20}^{\mathbf{R}}$.

4.2 Second order co-variant tensor

The second order co-variant tensor is denoted \mathbf{S}_{02} and is related to the orientation tensor as it describes orientation. It is defined and calculated as

$$\mathbf{S}_{02} = w \mathbf{l}^H \otimes \mathbf{l}^H = w \mathbf{l}^H [\mathbf{l}^H]^T \quad (11)$$

with w being some weight. The difference to the orientation tensor is that \mathbf{l}^H holds information on position as well as orientation, whereas in the orientation tensor, only information on orientation is contained. This tensor has been described in [25]. Similarly to the \mathbf{S}_{20} tensor, \mathbf{S}_{02} holds information on hyperplanes. However \mathbf{S}_{02} does this by keeping the normal $\hat{\mathbf{l}}$ for the plane and the orthogonal distance $\|\mathbf{l}\|$ to the plane in its range. In the null-space of \mathbf{S}_{02} information on all points that lie within this plane can be found. With matrix notation \mathbf{S}_{02} is expressed as

$$\mathbf{S}_{02} = w \begin{pmatrix} \|\mathbf{l}\|^2 & -l_0 \|\mathbf{l}\| \hat{\mathbf{l}} \\ -l_0 \|\mathbf{l}\| \hat{\mathbf{l}} & l_0^2 \hat{\mathbf{l}} \hat{\mathbf{l}}^T \end{pmatrix} \quad (12)$$

Integration over a region is not so easily interpreted in form of weighted means and covariances as for the \mathbf{S}_{20} tensor. However if a number of hyperplanes are within a region that is integrated over we will end up with a \mathbf{S}_{02} tensor of the form

$$\mathbf{S}_{02} = w_1 \mathbf{l}_1^H \otimes \mathbf{l}_1^H + w_2 \mathbf{l}_2^H \otimes \mathbf{l}_2^H + \dots \quad (13)$$

where w_i are some weights. We see that the range of \mathbf{S}_{02} spans the space where the corresponding hyperplanes lie and that the null-space of \mathbf{S}_{02} is the set of all points \mathbf{x}_H which lie in all of the hyperplanes \mathbf{l}_i^H .

4.3 Fourth order tensor

If, at a point \mathbf{p} , we have an estimate of $\mathbf{S}_{20}(\mathbf{p})$ and one of $\mathbf{S}_{02}(\mathbf{p})$ we can combine these into a fourth order tensor by

$$\mathbf{S}_{22}(\mathbf{p}) = \mathbf{S}_{20}(\mathbf{p}) \otimes \mathbf{S}_{02}(\mathbf{p}) \quad (14)$$

Since \mathbf{S}_{20} holds information of where and \mathbf{S}_{02} of what, it should be clear that \mathbf{S}_{22} holds information on both the hyperplanes position and orientation. By reshaping the second order tensors to vectors we can again represent the fourth order tensor by a matrix according to

$$\mathbf{S}_{22}(\mathbf{p}) = \mathbf{S}_{20}(\mathbf{p}) \mathbf{S}_{02}^T(\mathbf{p}). \quad (15)$$

In section 2 we mentioned that the fourth order tensor should be capable of holding information on several features. To realize this property of \mathbf{S}_{22} we have to integrate over a local region \mathbf{R}

$$\mathbf{S}_{22}^{\mathbf{R}}(\mathbf{p}) = \sum_{\mathbf{i} \in \mathbf{R}(\mathbf{p})} w(\mathbf{i}) \mathbf{S}_{20}(\mathbf{i}) \mathbf{S}_{02}^T(\mathbf{i}) \quad (16)$$

which we assume to contain a number of different hyperplanes present. If there are k different hyperplanes within the region \mathbf{R} with corresponding \mathbf{S}_{20} and \mathbf{S}_{02} descriptors we would have

$$\mathbf{S}_{22}^{\mathbf{R}} = \sum_k w_k \mathbf{S}_{20,k} \otimes \mathbf{S}_{02,k} = \sum_k w_k \mathbf{S}_{22,k}, \quad (17)$$

i.e., a summation of different fourth order tensors each representing one hyperplane. In the following we will denote $\mathbf{S}_{22}^{\mathbf{R}}$ found in equation 16 and equation 17 by \mathbf{S}_{22} , i.e. we will assume that an integration in a local region has taken place.

4.4 Analysis of \mathbf{S}_{22}

One base property for the idea behind the fourth order tensor can be expressed in the following way. First, if we have a sum like

$$\mathbf{S}_{22} = \sum_i^n k_i \mathbf{S}_{20,i} \mathbf{S}_{02,i}^T = \sum_i^n k_i \mathbf{S}_{22,i} \quad (18)$$

for some k_i , where both sets $\{\mathbf{S}_{20,i}\}$ and $\{\mathbf{S}_{02,i}\}$ are linearly independent, the rank of \mathbf{S}_{22} should be n . If however

say $\mathbf{S}_{02,i}$ is linearly dependent for $k = 1$ and $k = 2$ the sum should be

$$\mathbf{S}_{22} = \sum_i^n k_i \mathbf{S}_{20,i} \mathbf{S}_{02,i}^T = (k_1 \mathbf{S}_{20,1} + k_2 \mathbf{S}_{20,2}) \mathbf{S}_{02,1}^T + \sum_{i>2}^n k_i \mathbf{S}_{20,1} \mathbf{S}_{02,1}^T \quad (19)$$

and the rank of \mathbf{S}_{22} should be $n - 1$.

If we have the case that both the sets $\{\mathbf{S}_{20,i}\}$ and $\{\mathbf{S}_{02,i}\}$ are orthogonal we could by using the singular value decomposition retrieve the second order tensors that make up \mathbf{S}_{22} . However, if they are not perpendicular a singular value decomposition will not suffice to give us the original second order tensors in the local neighborhood. Using the assumption that the co-variant tensor \mathbf{S}_{02} is of rank 1, methods for analysis have been developed, implemented and tested. There is one analysis method for when the rank of \mathbf{S}_{22} is 2 and one for when it is 3. The analysis methods tries to find weights with which it can recombine the singular values and singular vectors to form the original $\mathbf{S}_{22,i}$ in equation 18. When these weights are found it is possible to extract the second order tensors that made up the \mathbf{S}_{22} that was analysed. The analysis stage is mainly done to evaluate that the fourth order tensors hold information on what they should. For technical documentation of the analysis the reader is referred to [22].

5 Understanding the representation

If we look at the two second order tensors described in section 4.1 and section 4.2 we find that they will have different ranks depending on the underlying geometrical structure. Moreover the rank of \mathbf{S}_{22} is dependent on the number of geometrical entities described by \mathbf{S}_{20} and \mathbf{S}_{02} that has been added to it, see section 4.3. In this section there is a short discussion on the topic of rank for both the second order tensors and for the fourth order tensor since this will probably be of help in understanding the tensor representation.

5.1 Rank for \mathbf{S}_{20}

For the second order contra-variant tensor, \mathbf{S}_{20} , a few examples of rank are given in table 1. In the table above the single line we have equivalence between the interpretation and rank, i.e. this is how we have decided to interpret the rank. Below the single line there is only implication from left to right. This is due to that there can be many different shapes that give the same rank. The \mathbf{S}_{20} tensor can have a maximum of rank 4 in the 3D case, however when it has rank 4 it is no longer possible to use the information

Shape of local region	Rank of \mathbf{C}	Rank of \mathbf{S}_{20}
Point	0	1
Line	1	2
Plane	2	3
2 parallel lines	2	3
Cylinder (curved part)	3	4
2 non-parallel planes	3	4

Table 1. Examples of rank for \mathbf{S}_{20}

Shape of local region	Rank of \mathbf{D}	Rank of \mathbf{S}_{02}
Point	3	3
Line	2	2
Plane	1	1
2 parallel lines	2	3
Cylinder (curved part)	2	3
2 non-parallel planes (=line)	2	2

Table 2. Examples of rank for \mathbf{S}_{02}

since there are several shapes that could have resulted in rank 4. Basically the more directions in the local neighborhood there is information in, the higher rank the \mathbf{S}_{20} tensor gets. An example of what happens when you add two \mathbf{S}_{20} tensors is the following. Consider one tensor for a line in 3D space and then another tensor for a line going in another direction. The underlying information that has been added up in the respective tensors covariance part, equation 10, does in their combined local neighborhood exist in two directions. If these two tensors representing the lines were added the new tensor would end up with the same rank as for a full plane-like shape in the data.

5.2 Rank for \mathbf{S}_{02}

For the second order co-variant tensor a few examples of rank for some geometrical structures are given in table 2, where \mathbf{D} refers to $l_0^2 \hat{\mathbf{I}} \hat{\mathbf{I}}^T$ in equation 12. In this table equivalence and implication works the same way as for table 1. Two example cases for the \mathbf{S}_{02} tensor can be seen in figure 1. In the left of figure 1 we see an example for the case of a plane. We see that all normals $\hat{\mathbf{I}}$ are the same. Moreover the perpendicular distance $\|\mathbf{1}\|$ is the same for all positions on the plane. Therefore \mathbf{I}^H has no variation in any of its elements which gives that \mathbf{S}_{02} has a rank of 1 after it has been integrated over such a region.

In the right of figure 1 we have the case that the local neighborhood is a curved part of a cylinder. In this case the normals $\hat{\mathbf{I}}$ all vary within one plane in 3D space, i.e. in a 2 dimensional sub-space of \mathbb{R}^3 , so \mathbf{D} has a rank of 2. In this case however, $\|\mathbf{1}\|$ can vary freely so \mathbf{I}^H can vary in at least a 3 dimensional sub-space of \mathbb{R}^4 but there is still a vector \mathbf{v}

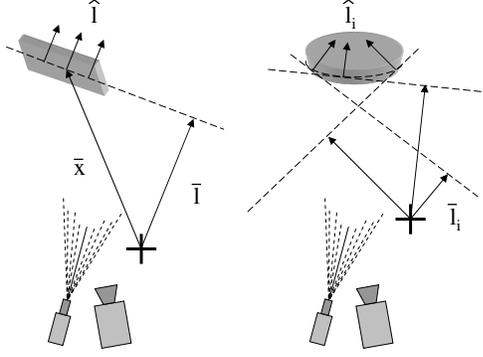


Figure 1. Examples for rank of \mathbf{S}_{02} .

which is perpendicular to $\hat{\mathbf{n}}$ for all planes on the surface of the cylinder. This gives that

$$\mathbf{S}_{02} \begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix} = \left(\sum_i \mathbf{1}_i^H \mathbf{1}_i^{H^T} \right) \begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix} = \mathbf{0}, \quad (20)$$

i.e. \mathbf{S}_{02} still has a null-space, so the rank of \mathbf{S}_{02} is a maximum of 3 in this case.

5.3 Rank for \mathbf{S}_{22}

A complication when it comes to the scene tensor in 3D as compared to in 2D is that both the line and plane case, of what $\mathbf{S}_{20,k}$ and $\mathbf{S}_{02,k}$ in equation 17 can represent, do exist. We do not know what happens to the rank of \mathbf{S}_{22} if \mathbf{S}_{22} is made up of a combination of \mathbf{S}_{22} rank 1 tensors for the line case and \mathbf{S}_{22} rank 1 tensors for the plane case. As mentioned in section 4.4 the analysis function implemented assumes that $\mathbf{S}_{02,k}$ is of rank 1, i.e. that it represents a plane. Due to this and that planes are the most common geometrical structure found in sparse 3D data we will only experiment with the plane case in this work. In this case the rank should equal the number of different planes that are present in the local neighborhood where \mathbf{S}_{22} was estimated.

5.4 Rank measure

If we have good rank measures for rank 2 and 3, we could use non maximum suppression to find peaks which would give us a crude point-of-interest detector. Even though some other kind of POI-detector would be preferable due to heavy computational load, the analysis mentioned in section 4.4 and detailed in [22] still needs to know the rank of the matrix to be analyzed. A measure of rank that is used in many of the steps to be presented was derived from the properties trace, determinant and norm for a 3×3 matrix. We do however apply this rank measurement to larger matrices also by, at each $\bar{\mathbf{p}}$, inserting the singular

values $\sigma_1 \geq \sigma_2 \geq \sigma_3$ from the SVD of the matrix to be analyzed into the following expressions:

$$c_1 = \frac{9d - 4qt + t^3}{3d - 3qt + t^3} \quad c_2 = \frac{-9d + qt}{3d - 3qt + t^3} \\ c_3 = \frac{3d}{3d - 3qt + t^3} \quad (21)$$

where

$$t = \sigma_1 + \sigma_2 + \sigma_3 \quad d = \sigma_1 \sigma_2 \sigma_3 \\ q = \sigma_1 \sigma_2 + \sigma_2 \sigma_3 + \sigma_3 \sigma_1 \quad (22)$$

By defining the measures c_k in this way, we get

- $c_1 = 1, c_2 = 0, c_3 = 0$ if $\sigma_2 = \sigma_3 = 0$.
- $c_1 = 0, c_2 = 1, c_3 = 0$ if $\sigma_1 = \sigma_2$ and $\sigma_3 = 0$.
- $c_1 = 0, c_2 = 0, c_3 = 1$ if $\sigma_1 = \sigma_2 = \sigma_3$.

with the additional property that $c_1 + c_2 + c_3 = 1$ which means that c_k can be used as a measure of confidence of rank k .

6 Estimation methods

Estimation of the scene tensor from range data is mainly about estimating \mathbf{S}_{02} . It could therefore be said that it basically is about estimation of normals for a local surface patch. Here we present one method that has been implemented and tested. The estimation of the representation of the local neighborhood as described in equation 10 is done by applying the following steps at each point $\bar{\mathbf{p}}$ in the range image:

1. Form the vector $\mathbf{x}_H = (x_0 \ x \ y \ z)^T$ from the range data, where x_0 is a suitably chosen constant.
2. Compute $\mathbf{S}_{20}(\bar{\mathbf{p}}) = \mathbf{x}_H \mathbf{x}_H^T$ in every point $\bar{\mathbf{p}}$.

We now have an estimate of \mathbf{S}_{20} at each point $\bar{\mathbf{p}}$ that is not integrated over a local region. To compute \mathbf{S}_{02} using polynomial expansion [7]:

1. Set the signal model to a linear one

$$\hat{g} = a_0 + a_1 x + a_2 y \quad (23)$$

and solve

$$\epsilon = |z - \hat{g}| \quad (24)$$

by methods detailed in [7] for every point $\bar{\mathbf{p}}$.

2. At all points $\bar{\mathbf{p}}$ form

$$\mathbf{l} = \frac{1}{\beta} \begin{pmatrix} a_2 \\ -a_1 \\ -\beta \end{pmatrix} \quad (25)$$

where β is the mean of the difference of neighboring x and y coordinates in $\mathbf{x}(\bar{\mathbf{p}})$. Or in other words, β is a measure of the dynamic range per pixel step in the x and y directions of the data. Use \mathbf{l} to in turn form

$$\mathbf{T} = \kappa \hat{\mathbf{l}} \hat{\mathbf{l}}^T \quad (26)$$

where $\hat{\mathbf{l}}$ is a normalized version of the vector \mathbf{l} and κ is a constant given by the norm of the \mathbf{l} vectors.

3. It then follows from equation 1 that

$$\mathbf{T} \mathbf{x} = \kappa \hat{\mathbf{l}} \hat{\mathbf{l}}^T \mathbf{x} = \kappa \hat{\mathbf{l}} \|\mathbf{l}\| = \kappa \mathbf{l} \quad (27)$$

which means that we can compute $\kappa \mathbf{l} = \mathbf{T} \mathbf{x}$

4. From this follows directly that we can compute $\kappa \|\mathbf{l}\|^2 = \mathbf{x}^T \mathbf{T} \mathbf{x}$.

5. We now have all the elements necessary for the matrix \mathbf{S}_{02} according to

$$\mathbf{S}_{02} = \kappa \begin{pmatrix} -\|\mathbf{l}\|^2 & -l_0 \mathbf{l}^T \\ -l_0 \mathbf{l} & l_0^2 \hat{\mathbf{l}} \hat{\mathbf{l}}^T \end{pmatrix} \quad (28)$$

where l_0 is a suitable chosen constant.

The estimation of the representation of the local neighborhood, as described in equation 16, is then computed as a variant of equation 16

$$\mathbf{S}_{22}(\mathbf{p}) = \sum_{\mathbf{i} \in \mathbf{R}(\mathbf{p})} \frac{w(\mathbf{i})}{\epsilon(\mathbf{i})} \mathbf{S}_{20}(\mathbf{i}) \mathbf{S}_{02}^T(\mathbf{i}) \quad (29)$$

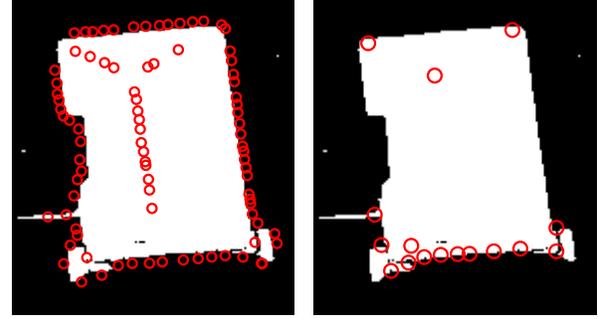
where $\epsilon(\bar{\mathbf{p}})$ is the residual when trying to fit a plane to the local region, see [7].

7 Experiments

Presenting results for such a high-dimensional descriptor as the scene tensor is a problem. This chapter will depend heavily on images to show the results.

7.1 Used range data

For the range data experiments two different inputs were used. The first one is data scanned with an industrial laser-based range sensor. The scan is not calibrated and we do not have ground truth. It is therefore not possible to give any numbers on the exactness of the tensor representation in respect to some parameters, e.g. in respect to extracted angles. It is however possible to use it to get a feel for how well the representation and estimation works by overlaying results in images. The second input used in this section was a synthetic version of the well-known ‘‘blox’’ image. The synthetic image was rendered with OpenGL and the depth buffer was extracted and used as range data. For the x and y values we generated data so that they were of the same dynamic range as the depth component.



(a) Rank 2 detected.

(b) Rank 3 detected.

Figure 2. Detected points of interest.

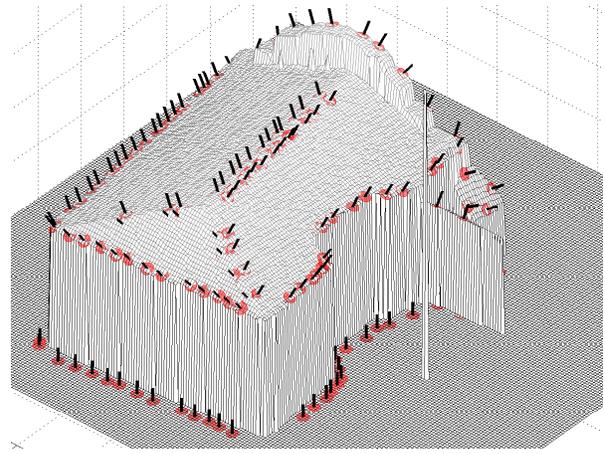


Figure 3. Information from rank 2 tensors.

7.2 Information extraction

We have selected the tensors that were detected by the POI-detection method in section 5.4 and then analysed them to find out what they represent. In figure 3 and figure 4 we have used the extracted information to overlay the range scan of the miniature house with vector graphics where a base (disc), and a line is drawn to show the planes, normals and positions that were extracted from each tensor. In figure 3 each tensor detected in figure 2(a) has resulted in two base/normal pairs and in figure 4 each tensor detected in figure 2(b) has resulted in three such base/normal pairs.

7.3 Detecting the wrong rank

An interesting question to answer is what would happen if we detected the wrong rank and used that wrong rank as input to the analysis function. An example of miss-detected rank can be seen in figure 5. In figure 5 a few points of interest from figure 2(a) have been analyzed with the rank 3

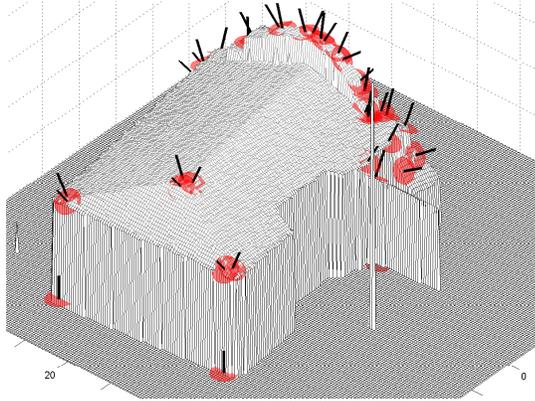


Figure 4. Information from rank 3 tensors.

analysis method and a few points from figure 2(b) have been analyzed with the method for rank 2 analysis. One such case would be if we detected a too low rank. It would mean analyzing a tensor that represents more local properties than we try to extract, e.g. trying to extract only 2 patches from a rank 3 tensor. Experiments on this has showed that the extracted information contains one plane that is almost correct. The “correct” plane usually has some small errors. The other extracted plane seems to be an average of the other two planes. The other case would be to try to extract more features from one tensor than it represents, e.g. extract 3 planes from a rank 2 tensor. In this case the correct information is extracted, sometimes with small errors, and one additional plane is also extracted. The additional plane is usually very close in position to one of the correct planes. The orientation for the additional plane does however seem to vary quite much.

7.4 Noise tests

To test how much the information extracted from tensors change due to noise we will use the depth buffer of the synthetic version of the “blox” image as input and add Gaussian noise to it. Gaussian noise was added to the z -component of the data and the signal to noise ratio for the data in dB is defined by the function

$$\text{SNR} = 10 \log \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2} \quad (30)$$

where σ is the standard deviation. The angle between ground truth normals and normals estimated from noisy data was calculated by an ordinary scalar product. The average of the errors for the normals for each noise level and for each estimation method is presented in table 3 for the rank 2 and rank 3 cases. The angles in the tables are in degrees.

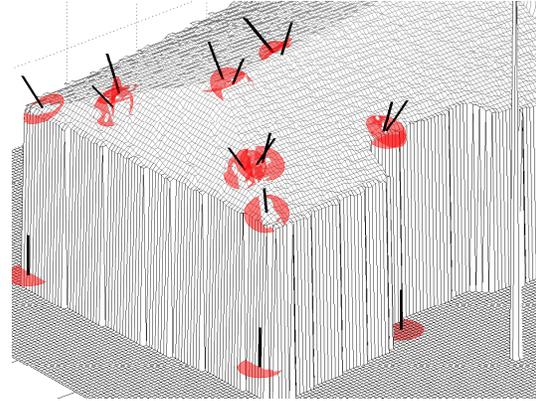


Figure 5. Using wrong rank.

SNR:	2	15	30	44	inf
Rank					
2	4.5	4.7	4.7	4.6	4.1
3	9.4	9.3	8.1	7.0	6.0

Table 3. Errors for noise test.

8 Conclusions and future work

We have shown that the representation works in 3D when used together with range data, and that the local geometrical properties are recoverable. There is however still some work to be done. To be able to use the descriptor in object recognition it needs to be invariant to rotation. This could probably be achieved in the same way as in [16, 3, 27]. A faster POI-detector should also be used. More research should be directed towards exploring the description of line-like structures as well as the mix between them.

References

- [1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded-up robust features. In *9th European Conference on Computer Vision*, Graz, Austria, 2006.
- [2] J. Bigün. *Local Symmetry Features in Image Processing*. PhD thesis, Linköping University, Sweden, 1988. Dissertation No 179, ISBN 91-7870-334-4.
- [3] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *Int. J. Comput. Vision*, 25(1):63–85, 1997.
- [4] D. Cliff, I. Harvey, and P. Husbands. *Explorations in Evolutionary Robotics, In Adaptive Behavior*, vol. 2, no. 1, 1993, pages 71–104.
- [5] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-d shape recovery using distributed aspect matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

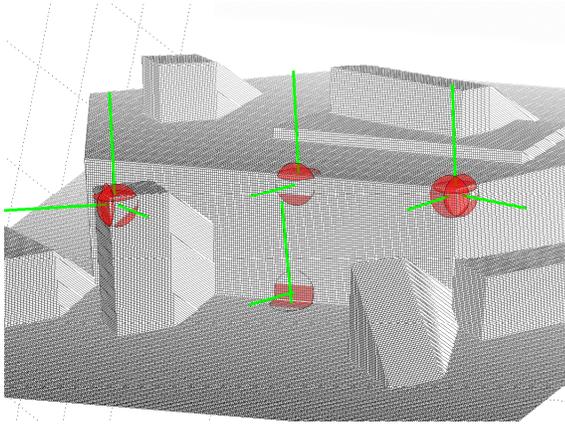


Figure 6. Select tensors from the noise test.

- [6] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1140–1157, 1989.
- [7] G. Farneback. *Polynomial Expansion for Orientation and Motion Estimation*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 2002. Dissertation No 790, ISBN 91-7373-475-6.
- [8] M. Felsberg. *Low-Level Image Processing with the Structure Multivector*. PhD thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002. TR no. 0203, available at <http://www.informatik.uni-kiel.de/reports/2002/0203.html>.
- [9] G. Granlund. Organization of architectures for cognitive vision systems. In *Proceedings of Workshop on Cognitive Vision*, Schloss Dagstuhl, Germany, October 2003.
- [10] G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.
- [11] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
- [12] R. M. Haralick and L. Watson. A facet model for image data. *Computer Graphics and Image Processing*, 15(2):113–129, February 1981.
- [13] C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, September 1988.
- [14] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. K. Bowyer, D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.
- [15] B. Johansson and A. Moe. Patch-duplets for object recognition and pose estimation. In *2nd Canadian Conference on Robot Vision*, 2005. Accepted.
- [16] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [18] G. Medioni and A. Francois. 3d structures for generic object recognition. pages Vol I: 30–37, 2000.
- [19] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [21] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.
- [22] K. Nordberg. A fourth order tensor for representation of orientation and position of oriented segments. Technical Report LiTH-ISY-R-2587, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, May 2004.
- [23] K. Nordberg and R. Söderberg. Detection and representation of complex local features. In *2005 International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, March 2005. IEEE.
- [24] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
- [25] S. M. Seitz and P. Anandan. Implicit representation and scene reconstruction from probability density functions. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 28–34, 1999.
- [26] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *Proceedings of the 10th International Conference on Pattern Recognition*, volume 1, pages 274–278, 1990.
- [27] Y. Sun, J. K. Paik, A. Koschan, D. L. Page, and M. A. Abidi. Point fingerprint: A new 3-d object representation scheme. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(4):712–717, 2003.
- [28] C.-K. Tang and G. Medioni. Curvature-augmented tensor voting for shape inference from noisy 3d data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):858–864, 2002.
- [29] G. Taubin. Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1115–1138, 1991.
- [30] F. Vikstén and A. Moe. Local single-patch features for pose estimation using the log-polar transform. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Estoril, Portugal, June 2005. IAPR. Accepted.
- [31] F. Vikstén, R. Söderberg, K. Nordberg, and C. Perwass. Increasing Pose Estimation Performance using Multi-cue Integration. In *International Conference on Robotics and Automation*, Orlando, Florida, USA, May 2006.
- [32] N. Yokoya and M. D. Levine. Range image segmentation based on differential geometry: A hybrid approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):643–649, 1989.