Optica Applicata, Vol. XLV, No. 4, 2015
DOI: 10.5277/oa150411

Significance of features in object recognition using depth sensors

BOGDAN HARASYMOWICZ-BOGGIO*, ŁUKASZ CHECHLIŃSKI, BARBARA SIEMIĄTKOWSKA

Institute of Automatic Control and Robotics, Warsaw University of Technology,
ul. Św. Andrzeja Boboli 8, 02-525 Warsaw, Poland

*Corresponding author: mysticdrow@gmail.com

This article concerns a key topic in the field of visual object recognition – the use of features. Object recognition algorithms typically rely on a fixed vector of pre-selected features extracted from 2D or 3D scenes, which are then analyzed with various classification techniques. On the other hand, the activation of particular features in biological vision systems is hierarchical and data-driven. To achieve a deeper understanding of the subject, we have introduced several mathematical tools to estimate multiple RGB-D features' relevance for different object recognition tasks and conducted statistical experiments involving our database of high quality 3D point clouds. From the thorough analysis of the obtained results we draw conclusions that may be useful to design better, more adaptive object recognition algorithms.

Keywords: depth sensor, RGB-D features, 3D object recognition, Kinect.

1. Introduction

The entirety of operations performed by any real-life visual object recognition method can be regarded as the extreme reduction of a vast amount of input data, typically comprising up to several megabytes into a very limited number of values, which represent object positions or identities. The aim of such algorithms is, of course, to obtain accurate and useful information concerning the objects whose image is entangled in the analyzed data. We can differentiate roughly at least two steps in most recognition algorithms: feature extraction and object classification. This work focuses mostly on the former, analyzing various kinds of features, methods to measure their usefulness and the process of feature choice by a vision system designer.

Feature extraction usually consists of more than one stage. After low-level features are calculated over the input data, we face the difficult task of effective representation of regions (*i.e.*, point clouds, point clusters or image segments) which possibly contain known objects as vectors of limited, preferably low dimensionality. Such representation is necessary in order to classify objects with known methods. However, several

difficulties arise: an object view captured by a 2D or 3D camera, apart from often being very difficult to automatically extract from the environment, as stated before, comprises a vast amount of data, consisting of possibly thousands of points. For these points we may calculate many kinds of local or inter-point features regarded as simple feature vectors. Several methods can be used to simplify these clusters of features – the most common approach is to extract fixed-length vectors of features that, ideally, are affected primarily by the object's identity (or semantic class) and are invariant to the possible changes of irrelevant properties, such as the object's position on the scene. Multiple methods of this kind have been developed in 2D and 3D computer vision [1–7]. A more sophisticated, recent technique to simplify an image or point cloud region is to apply functions that describe its features without quantization [8–10].

An alternative, worth mentioning approach to object recognition (more commonly used in point cloud registration) is to use algorithms that try to geometrically match a relatively unsimplified object view to a model of a known object class [11–13] and measure the quality of matching in order to identify the object. However, this approach is significantly less popular than feature-based recognition, as registration is effective only for rigid, almost identical object instances and is computationally very demanding. Therefore, in this paper we focus on feature-based methods.

The common approach in computer recognition systems consists in extracting such features from the selected image or point cloud regions (segments) and reducing them to a single point in the chosen multi-dimensional feature space. However, it is intuitively understandable that different features can be better than others. In this paper we discuss and verify methods of estimating the usefulness of particular features and similarity metrics applied for tasks of 3D recognition of different object classes. Our focus lies on indoor objects common to the human environment, which could have a practical meaning in indoor robotics and intelligent building systems.

2. Features

For the presented research a set of easily interpretable local and global object features has been selected. Let us first describe the five chosen local features, which can be calculated at each point of the object surface. The first used feature is the angular surface inclination, which represents the magnitude of the angle between the surface normal vector and a global downwards-pointing vector (the gravitational acceleration which was measured using the built-in Kinect's accelerometer). This is described in the following equation:

$$\Theta(\mathbf{p}) = \operatorname{acos}(\hat{\mathbf{n}} \cdot \hat{\mathbf{g}}) \tag{1}$$

where $\hat{\mathbf{n}}$ is the surface normal vector at point $\mathbf{p}$ and $\hat{\mathbf{g}}$ is the normalized gravitational acceleration vector. This feature is invariant to translation of the object and to rotation around any vertical axis. Despite not being invariant in the general case (*i.e.*, for rota-

tions around non-vertical axes), this simple feature proves highly useful for recognition in indoor environments, as many objects have a well-defined base.

The next described geometric feature is surface convexity, which is a measure of outward surface curvature at a given point. If the surface is locally flat, the value of surface convexity is zero and if it is concave, the convexity is negative. In order to mathematically define this feature, let us consider the local coordinate system attached to $\mathbf{p}$ (in which $\mathbf{p} = [0, 0, 0]^T$ and where $\hat{\mathbf{n}} = [0, 0, 1]^T$ is the surface normal vector at $\mathbf{p}$. Using this coordinate system, we introduce the auxiliary quantity of surface convexity at $\mathbf{p}$ relative to point $\mathbf{p}_i$ defined with the following equation:

$$c(\mathbf{p}, \mathbf{p}_i) = (\widehat{\mathbf{e}_{xy}\mathbf{p}_i}) \cdot (\widehat{\mathbf{e}_{xy}\hat{\mathbf{n}}_i}) \cdot \operatorname{acos}\left(\frac{\mathbf{e}_z \hat{\mathbf{n}}_i}{\pi/2}\right) \tag{2}$$

where: $\mathbf{e}_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\mathbf{e}_z = [0 \ 0 \ 1]$ and ^ is the vector normalization operator and $\hat{\mathbf{n}}_i$ is the surface normal vector at point $\mathbf{p}_i$. Using this quantity, we can define surface convexity at $\mathbf{p}$ as:

$$c(\mathbf{p}) = \frac{1}{K} \sum_{\mathbf{p}_i \in N_p(r_1, r_2)} c(\mathbf{p}, \mathbf{p}_i) \tag{3}$$

where $N_p(r_1, r_2) = N_p(r_1)/N_p(r_2)$, $r_1 > r_2$, is the set-theoretic difference of two neighbourhoods of point $\mathbf{p}$ having radii of magnitude r_1 and r_2. The experimentally chosen values of these radii are correspondingly 10 and 8 mm; K is the cardinality of $N_p(r_1, r_2)$.

Surface convexity informs us only about the mean positive curvature of a surface nearby the given point. In order to capture more information about the shape of this curvature, we can use the concept of anisotropy of convexity, which measures the maximum spread of relative surface convexity in the proximity of the point of interest. We have defined this quantity with the following equation:

$$a(\mathbf{p}) = \max_{\mathbf{p}_i \in N_p(r_1, r_2)} [c(\mathbf{p}, \mathbf{p}_i)] - \min_{\mathbf{p}_i \in N_p(r_1, r_2)} [c(\mathbf{p}, \mathbf{p}_i)] - 1 \tag{4}$$

Therefore anisotropy is sensitive to the directionality of convexity – its value will be minimal both for a flat and a spherical surface, but higher for cylindrical, conical or angular surfaces. The information captured in the values of surface convexity and anisotropy is similar to the principal curvatures k_1, k_2. However, in our opinion, the interpretation of the proposed features is more intuitive. For convenience we have implemented modified versions of the previously described surface convexity and anisotropy features, which significantly enhance the differentiability for slightly

curved surfaces. In the following sections the terms surface convexity and anisotropy of convexity will refer to these practically modified versions of the original formulas. The transformed features present as follows:

$$c^{\chi}(\mathbf{p}) = \operatorname{sgn}[c(\mathbf{p})]\sqrt{c(\mathbf{p})} \tag{5}$$

$$a^{\chi}(\mathbf{p}) = 2\sqrt{\frac{a(\mathbf{p})+1}{2}} - 1 \tag{6}$$

Color features are widely used in 2D object recognition. In this work we decided to consider the color hue H and saturation S extracted from the HSV color representation, as these features are known to be less sensitive to changing lighting conditions than common RGB colors. As the used Kinect sensor has separate optical systems for color and depth images, color registration inaccuracy is quite common, which could affect the experimental results. However, we consider the H and S, features in the same manner as the geometric features.

In order to apply classic methods of comparison for any feature contained within different object clusters, we need to reduce the list of this feature's values given at each point to a limited, classifiable vector. To do this, we discretize the feature space and build histograms. Beside a simple 1D histogram for each of the 5 presented features, we decided also to consider histograms of pairs of these features, which can be regarded as 2D feature vectors calculated at each point. Accordingly, for these vectors we build 2D histograms. We do not consider higher dimensionality features, as our aim is to keep our experimental results easy to interpret. From all the 10 possible 2D feature combinations, we consider only 4, which we believe to be the most meaningful: inclination–convexity, inclination–anisotropy, convexity–anisotropy, and hue–saturation.

In addition to features obtained for each point of the scene, we decided to use shape descriptors calculated for whole object clusters (thus, global), based on Ref. [14]. These features are based on random sample vectors of points and are defined as follows: D2 is a histogram of distances between two random points belonging to the cluster of interest, D3 is a histogram of fields of triangles built on three random points and D4 is a histogram of volume of tetrahedrons built on four random cluster points. An important advantage of these descriptors is, besides capturing global shape properties, their tolerance to noise. Inaccurate, undulated surface measurement highly distorts such features as surface convexity or its anisotropy, but not the global shape descriptors.

3. Similarity and uniqueness

Since we have reduced the information given in each object view to a set of histogram features, we can explore several methods to measure similarity between these views.

For each histogram feature, we can define the similarity between histograms **x** and **y** corresponding to different object views using, among others, the following metrics:

– Manhattan similarity

$$S_{L^1}(\mathbf{x}, \mathbf{y}) = 1 - \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \right\|_1 \tag{7}$$

where $\|M\|_1$ means the L^1 norm of the vector M;

– Euclidean similarity:

$$S_{L^2}(\mathbf{x}, \mathbf{y}) = 1 - \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \right\|_2 \tag{8}$$

where $\|M\|_2$ means the L^2 norm of the vector M;

– Pearson's correlation:

$$S_P(\mathbf{x}, \mathbf{y}) = \frac{\operatorname{cov}(\mathbf{x}, \mathbf{y})}{\sigma_x \sigma_y} = \frac{E[(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_y)]}{\sigma_x \sigma_y} \tag{9}$$

– Bhattacharyya's similarity [15]:

$$S_B(\mathbf{x}, \mathbf{y}) = \sqrt{1 - \frac{1}{\sqrt{\mu_x \mu_y N^2}} \sum_I \sqrt{\mathbf{x}\mathbf{y}}} \tag{10}$$

We also introduce our own custom similarity metric, which is an attempt to correct the inconvenient properties of Pearson's correlation. We learned that some visible, but subtle patterns present in a histogram may carry important information which is ignored by Pearson's correlation. The proposed similarity is based on Pearson's correlation but additionally considers outliers in relative values of corresponding bins, so it is affected by details disregarded by Pearson correlation. We named this similarity detail susceptible correlation (DSC). To calculate the proposed correlation, we first perform a Gaussian blur and shift the input histograms, which improves robustness to noise. We have chosen to set the Gaussian blur value to $\sigma = 0.7$ in each dimension and the shift constant to $C_{\text{shift}} = 0.1$:

$$x_G = G_{\text{blur}}[\mathbf{x}] + C_{\text{shift}} \tag{11a}$$

$$y_G = G_{\text{blur}}[\mathbf{y}] + C_{\text{shift}} \tag{11b}$$

Then we calculate the relative altitude of bins:

$$\mathbf{z} = \frac{\min(x_G, y_G)}{\max(x_G, y_G)} \tag{12}$$

where min and max denote element-wise operations on corresponding vectors' values. Next we calculate the number of outliers and huge outliers:

$$k_{\text{outliers}} = N_T(\mathbf{z}, T_{\text{high}}) \tag{13}$$

$$k_{\text{huge}} = N_T(\mathbf{z}, T_{\text{low}}) \tag{14}$$

where the function $N_T(\mathbf{x}, T)$ returns the number of elements of $\mathbf{x}$ below threshold T. The threshold values are set to $T_{\text{high}} = 0.7$ and $T_{\text{low}} = 0.4$.

Then we define the Pearson-based factor of DSC as

$$k_{\text{Pearson}} = \begin{cases} 0 & \text{if} \quad S_P(\mathbf{x}, \mathbf{y}) < 0 \\ \sqrt{S_P(\mathbf{x}, \mathbf{y})} & \text{otherwise} \end{cases} \tag{15}$$

Finally we define DSC similarity as:

$$S_{\text{DSC}}(\mathbf{x}, \mathbf{y}) = k_{\text{Pearson}}(1 - k_{\text{outliers}})(1 - k_{\text{huge}})^2 \tag{16}$$

The similarity measures presented in this section do not show directly the usefulness of the analyzed histogram features to tell different objects classes and their instances apart. In order to measure these qualities, we have proposed several relative "uniqueness" magnitudes, comprising similarities of multiple object views. In order to calculate these quantities from a database of object views, let us organize the list of all the feature histograms as a 2D array $\mathbf{V}$, consisting of N class vectors $\mathbf{C}_i$, $i = 1, ..., N$, each having M_i views:

$$\mathbf{V} = [\mathbf{C}_i], \qquad \mathbf{C}_i = [v_{ij}] \tag{17}$$

We first define the mean class similarity matrix for a given measure of similarity S for each histogram feature f as

$$\mathbf{G} = [g_{ij}] = [\overline{S(v_{ip}, v_{jq})}], \qquad v_{ip} \in \mathbf{C}_i, \qquad v_{jq} \in \mathbf{C}_j, \qquad v_{ip} \neq v_{jq} \tag{18}$$

We then calculate the single-column class uniqueness matrix by applying a Mahalanobis -like distance function

$$\mathbf{U} = [u_i] = \left[\frac{g_{ii} - \overline{g_{ii}}}{\sigma_i}\right], \qquad i \neq j \tag{19}$$

where σ_i is the standard deviation of similarity calculated for all the available view pairs which belong to class i. This equation gives us a measure of the capacity of the analyzed feature to tell each object class apart from other classes. If this capacity is high, we expect the internal mean class similarity to be significantly higher than

the mean similarity to other classes. The choice to apply variance normalization is motivated by the fact that different features and similarity metrics may present different scales and internal class variances.

If the analyzed database contains many object classes, it would be also convenient to have a single value to measure each feature's overall usefulness (*e.g.*, for applications of limited computational resources). Instead of just taking the mean values of the uniqueness matrices, we introduce the class uniqueness score χ defined with the following equation:

$$\chi = \overline{2\,\mathrm{atan}\left(\frac{u_i}{2}\right)} \tag{20}$$

The purpose of using this function is to avoid overestimation of a feature's utility resulting from individual high uniqueness values. If we hypothesized a Gaussian distribution of similarity values, even large differences in high uniqueness are negligible, as high uniqueness indicates that all the views could be correctly classified using a given feature. The derivative of the averaged functions in the χ scores is 1 at $u_i = 0$ and remains approximately linear for small uniqueness values. Therefore, the proposed scores can be regarded as mean "useful" uniqueness.

Finally, we propose a tool for detecting correlations between similarities of features themselves, which would be helpful to avoid using redundant features (Table 1). Consider having two features with high uniqueness scores, but for which we suspect that both carry similar information. To check this, we can calculate Pearson's correlation

T a b l e 1. Class uniqueness scores χ overall uniqueness of class histograms for all features and similarity metrics combinations.

	Pearson	Bhattacharyya	DSC	L^1	L^2
I	2.18	1.96	1.99	2.01	1.95
C	2.37	2.02	2.22	2.09	2.02
A	2.09	1.74	1.81	1.77	1.76
H	1.33	1.12	1.12	1.15	1.15
S	1.37	1.30	1.23	1.29	1.20
I-C	2.13	1.97	2.24	2.00	1.89
I-A	2.10	1.97	2.20	2.00	1.85
C-A	2.26	2.01	2.33	2.00	1.96
H-S	1.33	1.25	1.40	1.27	1.21
D2	2.42	1.99	2.52	2.04	1.97
D3	2.35	1.77	2.38	1.83	1.86
D4	2.14	1.66	2.62	1.71	1.68

I – inclination, C – surface convexity, A – anisotropy of convexity, H – hue, S – saturation, I-C – inclination-convexity, I-A – inclination-anisotropy, C-A – convexity-anisotropy, H-S – hue-saturation.

of similarity values (*i.e.*, Pearson correlation, DSC, *etc.*) of all histogram pairs – which we called hypercorrelation. Let's denote the vector of similarities of all histogram pairs calculated for one feature as φ_1 and the corresponding vector calculated for another feature as φ_2. We define hypercorrelation as

$$\zeta = S_P(\varphi_1, \varphi_2) \tag{21}$$

We can obtain additional information by drawing a 2D plot of $\varphi_2(\varphi_1)$ points. On such plot we can observe some dependences between features which are difficult to measure, such as nonlinear functional dependences.

4. Experiments

In the carried experiment we captured 136 scenes containing 160 manually segmented object views. The objects are organized in 14 semantic classes. For the majority of the classes we included 4 different physical instances, each captured in several views. The scenes of the database have been taken using a Kinect sensor. The proposed database contains objects assigned to the following classes: apple, book, bottle, box, cornflakes, cup, deodorant, ironer, ketchup bottle, mouse, saucer, shoe, stamp, stapler. The sample objects are presented in Fig. 1.

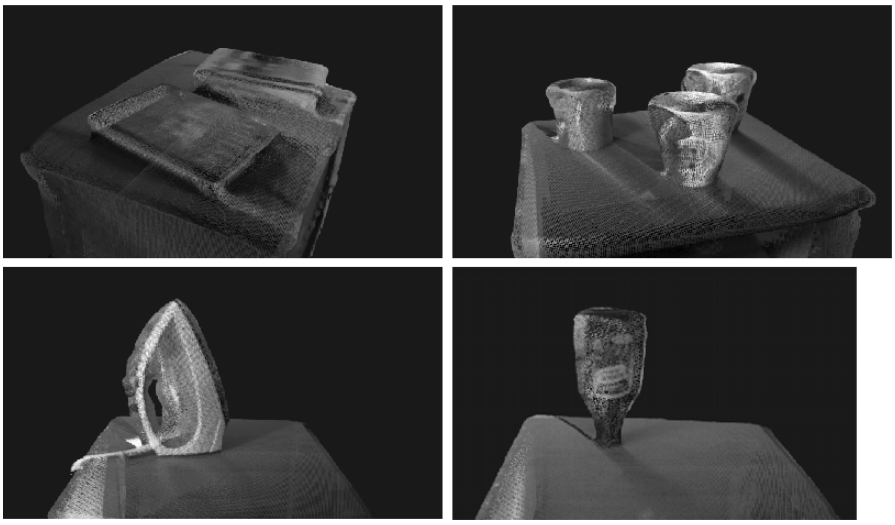

Fig. 1. Sample objects used in the experiments.

Using this database we have calculated the feature histograms: inclination, surface convexity, anisotropy of convexity, hue, saturation, inclination-convexity, inclination -anisotropy, convexity-anisotropy, hue-saturation, D2, D3, D4. To check which features and similarity measures can be useful in general, we calculated the mean instance self-similarity across the available scenes (a complete table of results is not presented

in this paper, but we draw conclusions using the full data). For some classes, such as ketchup bottle or ironer, the mean instance similarity shows the influence of object position (a ketchup bottle often stands upside down, an ironer has hot and cool bases). Shape distribution descriptors have the highest self-similarity values because they capture global shape properties and thus are more robust to cluttering on the object surface. Low self-similarity values for hue and saturation are the result of sensibility of these features to lighting variations and of the imperfect color registration of the Kinect sensor. Mean instance self-similarity for inclination is lower than for intrinsic features, as several objects included in the database have multiple bases.

DSC and Pearson's correlation achieved the best scores for different classes. However, for object recognition applications, obviously, it is not required to use the same similarity measure for all features. Note that Pearson's correlation achieves the best results for the first 5 features (all the simple 1D feature histograms), while DSC is the most suitable for the last 9 (2D histograms and shape distributions). Other similarity measures reached significantly lower results. We can see that the similarity metrics that perform best at particular features have also high corresponding instance self-similarities. Color-based features have significantly lower scores than the rest of the features, from which we can conclude that hue, saturation and their 2D combination are less reliable in 3D object recognition than geometric features. The rest of the features achieve relatively good scores between 2 and 2.7 (where D4 had the highest value). Having only overall scores we are not able to draw detailed conclusions.

Based on these results, in Table 1 we present the class uniqueness U defined in Eq. (19) calculated for each object class using the best similarity metrics. As motivated above, we chose Pearson's correlation for the first 5 features and DSC for the last 9.

The results corroborate the main thesis of this article – some features are better than others for particular objects, but there is no clear constant space of features that is best for every object class. Looking at the uniqueness of D4, which had the highest overall uniqueness score, we can see that even though for many object classes this feature performs unquestionably best, there are several classes where its uniqueness is very low. The reasons for this are case-specific: for example, slight curvatures of the almost flat saucers had a high impact on the volume enclosed by their surfaces, but little impact on the surfaces themselves, thus, D3 uniqueness was very high, but D4 uniqueness was almost insignificant. Both of these features, however, were ineffective for staplers, as the flat part of the used staplers was often imperfectly segmented from the flat table surface.

Despite a moderate overall score, surface convexity was the feature which most consistently detected class similarities for all semantic classes. None of the global geometric features (D2, D3, D4) was useful for measuring similarities of boxes, as the position of the cover strongly changed their global shape properties. The distribution of local convexity was also affected, but was stable enough to maintain significantly higher similarities between boxes than between boxes and other objects. As expected, inclination performed well for all rigid objects with a well-defined shape and base. Color features were mostly useful for detecting apples (of the same kind), ketchup

bottles (as they were all red) and cornflakes (as there was only one instance of this class used in the experiment). 2D histograms in some cases performed better than both of their 1D components, but more often inherited their disadvantages and had an increased sensibility to noise (as confirmed by Table 2). Therefore there is no clear advantage of using 2D over 1D histograms.

In the same manner we analyze the values of instance uniqueness. The values are omitted in this paper but it can be stated that they tend to be lower than for class uniqueness, which suggests that the differences between semantic classes are easier to detect than differences between instances of the same class. The shape distribution descriptors D2, D3, D4 performed well for instances, but were insufficient for differentiation in all cases. The features with high class uniqueness are not necessarily the same that have high instance uniqueness – these sets of features can be almost disjunctive. This is intuitively understandable, as some features may be irrelevant for a given class, but stable for a given instance (*e.g.*, color hue for cups and shoes or D4 for saucers).

Finally, we measured hypercorrelation ζ described in Eq. (21) and plotted the corresponding 2D scopes of similarity points with Person's correlation and DSC used as similarity metrics. The results are presented in Table 3. These scores for specific pairs of features are not significantly different for the considered similarity measures. The highest results are in line with our expectations: there is a strong dependence between 2D histograms and their 1D components and between 2D histograms which have a common feature.

More interesting conclusions can be drawn from hypercorrelation values between all D2, D3, D4 shape distribution features. D2 seems to be more independent from D3 and D4 than these last two features between themselves. Low hypercorrelation values can be found between local and global features, and also between color features and any other features. Two features (inclination-convexity and inclination-anisotropy) are strongly correlated and there seems to be no need to use both in an object recognition system. Comparing this finding with the uniqueness results shown before, we can conclude that it is advisable to use inclination-convexity over inclination-anisotropy, as it is less cluttered and gives consistently higher uniqueness.

5. Conclusions

In this paper we have considered several known similarity metrics to compare the feature histograms extracted from point clouds and also introduced our own metric detail susceptible correlation (DSC). DSC turned out to achieve the highest overall differentiation capacity (*i.e.*, uniqueness scores) both for semantic classes and instances. However, for specific features different metrics performed best. Furthermore, we proposed a method to detect correlations of the information carried by different features by measuring the correlation between values of feature histograms similarity (or correlation), which we called hypercorrelation. This quantity can be used to predict if the inclusion of specific features in an object recognition system will add new information (and possibly improve performance). As a final conclusion we would like to emphasize the po-

T a b l e 2. Class uniqueness *U* calculated applying the highest-score similarity metric for each feature.

	Apple	Book	Bottle	Box	Cornflakes	Cup	Deodorant	Ironer	Ketchup bottle	Mouse	Saucer	Shoe	Stamp	Stapler
I	7.52	0.54	11.35	–0.08	1.18	2.47	70.35	4.49	12.07	38.58	8.09	16.5	15.39	3.39
C	3.7	3.69	4.8	2.11	21.92	15.19	14.58	6.8	2.7	8.52	2.28	13.96	3.99	4.23
A	4.31	0.85	2.24	0.75	16.16	10.4	3.55	22.79	2.52	5.06	4.08	12.99	1.87	4
H	7.38	0.1	2.54	1.13	7.59	0.23	2.98	1.99	8.29	0	4.05	1.81	0.34	0.02
S	2.18	0.22	3.6	–0.15	5.62	0.67	1.34	9.31	10.58	0.12	1.86	2.79	2.33	0.27
I-C	6.11	0.41	10.44	0.38	2.48	3.55	32.94	3.83	8.31	27.93	6.03	25.13	7.15	5.12
I-A	5.8	0.45	8.34	0.32	2.22	4.04	12.93	3.64	9.91	12.79	6.51	24.37	5.74	5.48
C-A	6.12	1.52	3.32	2.13	35.41	12.26	9.61	8.22	2.18	10.96	4.14	17.89	3.33	3.54
H-S	4.94	0.19	2.85	0.14	6.53	0.76	2.74	4.81	14.2	0.1	2.75	2.16	0.84	0.22
D2	5.89	7.11	7.26	0.51	16.51	10.1	3.53	8	19.69	19.66	4.19	229.8	5.63	33.7
D3	4.29	23.7	1.78	0.11	47.7	33.18	93.4	29.2	20.9	49.55	182	101.7	3.49	0.05
D4	10.6	47.8	22.7	1.34	19.5	41.7	228	47.5	475	52.14	1.76	95.56	73.2	0.8

I – inclination, C – surface convexity, A – anisotropy of convexity, H – hue, S – saturation, I-C – inclination-convexity, I-A – inclination-anisotropy, C-A – convexity-anisotropy, H-S – hue-saturation.

T a b l e 3. Feature hypercorrelations. The presented values are calculated for Pearson's correlation above the diagonal and DSC below the diagonal.

	I	C	A	H	S	I-C	I-A	C-A	H-S	D2	D3	D4
I		0.53	0.32	0.24	0.1	0.92	0.91	0.6	0.32	0.01	0.35	0.4
C	0.48		0.31	0.28	0.12	0.66	0.57	0.83	0.32	0.05	0.42	0.38
A	0.36	0.5		0.33	0.16	0.35	0.59	0.72	0.31	0.37	0.4	0.21
H	0.21	0.24	0.3		0.26	0.26	0.31	0.38	0.77	0.22	0.32	0.14
S	0.24	0.21	0.2	0.31		0.13	0.16	0.16	0.64	0.16	0.11	0.05
I-C	0.85	0.6	0.38	0.21	0.21		0.92	0.7	0.35	0.03	0.36	0.4
I-A	0.83	0.56	0.61	0.28	0.23	0.9		0.76	0.38	0.17	0.45	0.42
C-A	0.54	0.82	0.79	0.33	0.23	0.67	0.74		0.4	0.21	0.5	0.38
H-S	0.32	0.29	0.31	0.76	0.63	0.31	0.37	0.38		0.21	0.29	0.19
D2	0.11	0.13	0.27	0.23	0.13	0.02	0.16	0.21	0.22		0.58	0.2
D3	0.21	0.36	0.37	0.27	0.15	0.25	0.36	0.41	0.24	0.54		0.7
D4	0.25	0.45	0.26	0.13	0.13	0.32	0.36	0.38	0.15	0.22	0.64	

I – inclination, C – surface convexity, A – anisotropy of convexity, H – hue, S – saturation, I-C – inclination-convexity, I-A – inclination-anisotropy, C-A – convexity-anisotropy, H-S – hue-saturation.

tential great benefits of developing more adaptive, nature-inspired computer vision methods, which, instead of relying on a fixed feature space and applying sophisticated classifiers, are able to choose (even autonomously) the most convenient features for a specific task.

Acknowledgements – This work was partially conducted within the project 2012/05/B/ST6/03094 financed by the Polish National Science Centre.

References

[1] Silberman N., Kohli P., Hoiem D., Fergus R., *Indoor segmentation and support inference from RGBD images*, [In] *ECCV*, 2012.

[2] Lavoué G., *Bag of words and local spectral descriptor for 3D partial shape retrieval*, [In] *Eurographics*, editor, [In] *Workshop on 3D Object Retrieval (3DOR)*, Eurographics, 2011.

[3] Sivic J., Zisserman A., *Efficient visual search of videos cast as text retrieval*, IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(4), 2009, pp. 591–606.

[4] Harasymowicz-Boggio B., Siemiatkowska B., *Object classification with metric and semantic inference*, [In] *2013 European Conference on Mobile Robots (ECMR)*, 2013, pp. 186–191.

[5] Dalal N., Triggs B., *Histograms of oriented gradients for human detection*, [In] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, Vol. 1, 2005, pp. 886–893.

[6] Hetzel G., Leibe B., Levi P., Schiele B., *3D object recognition from range images using local feature histograms*, [In] *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*, Vol. 2, 2001, pp. II-394–II-399.

[7] Swain M.J., Ballard D.H., *Color indexing*, International Journal of Computer Vision **7**(1), 1991, pp. 11–32.

[8] Liefeng Bo, Xiaofeng Ren, Fox D., *Kernel descriptors for visual recognition*, [In] *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 2010, pp. 244–252.

[9] Liefeng Bo, Xiaofeng Ren, Fox D., *Depth kernel descriptors for object recognition*, [In] *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 821–826.

[10] Xiaofeng Ren, Liefeng Bo, Fox D., *RGB-(D) scene labeling: features and algorithms*, [In] *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2759–2766.

[11] Buch A.G., Kraft D., Kamarainen J.-K., Petersen H.G., Kruger N., *Pose estimation using local structure-specific shape and appearance context*, [In] *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2080–2087.

[12] Rusu R.B., Blodow N., Beetz M., *Fast point feature histograms (FPFH) for 3D registration*, [In] *IEEE International Conference on Robotics and Automation, 2009, ICRA '09*, 2009, pp. 3212–3217.

[13] Schnabel R., Wahl R., Klein R., *Efficient RANSAC for point-cloud shape detection*, Computer Graphics Forum **26**(2), 2007, pp. 214–226.

[14] Osada R., Funkhouser T., Chazelle B., Dobkin D., *Matching 3D models with shape distributions*, [In] *SMI 2001 International Conference on Shape Modeling and Applications*, 2001, pp. 154–166.

[15] Bhattacharyya A., *On a measure of divergence between two statistical populations defined by their probability distributions*, Bulletin of the Calcutta Mathematical Society **35**(1), 1943, pp. 99–109.

Received April 22, 2015
in revised form June 28, 2015