Semiautomatic Benchmarking of Feature Vectors for Multimedia Retrieval

Tobias Schreck*
Technische Universität
Darmstadt, Germany

Jörn Schneidewind†
University of Konstanz,
Germany

Daniel Keim‡
University of Konstanz,
Germany

Matthew Ward§
Worcester Polytechnic
Institute, USA

Andrada Tatu¶
University of Konstanz,
Germany

ABSTRACT

Modern Digital Library applications store and process massive amounts of information. Usually, this data is not limited to raw textual or numeric data – typical applications also deal with multimedia data such as images, audio, video, or 3D geometric models. For providing effective retrieval functionality, appropriate meta data descriptors that allow calculation of similarity scores between data instances are requires. *Feature vectors* are a generic way for describing multimedia data by vectors formed from numerically captured object features. They are used in similarity search, but also, can be used for clustering and wider multimedia analysis applications.

Extracting effective feature vectors for a given data type is a challenging task. Determining good feature vector extractors usually involves experimentation and application of supervised information. However, such experimentation usually is expensive, and supervised information often is data dependent. We address the feature selection problem by a novel approach based on analysis of certain feature space images. We develop two image-based analysis techniques for the automatic discrimination power analysis of feature spaces. We evaluate the techniques on a comprehensive feature selection benchmark, demonstrating the effectiveness of our analysis and its potential toward automatically addressing the feature selection problem.

Keywords

Visual Analytics, Feature Vectors, Automatic Feature Selection, Self-Organizing Maps.

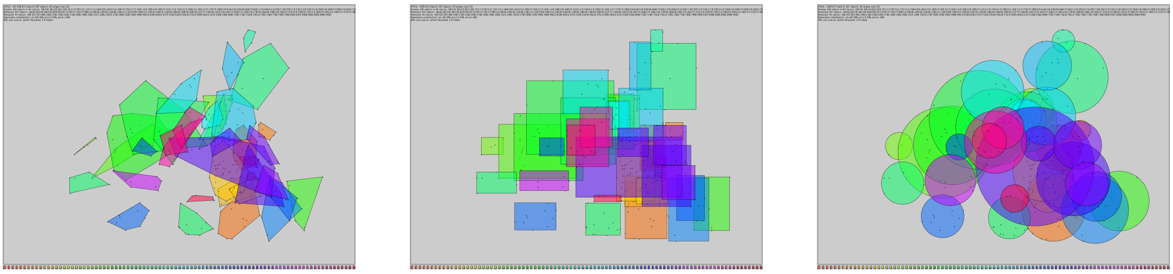
1. INTRODUCTION

Modern Digital Library and multimedia analysis applications store and process massive amounts of non-standard data. This data, not limited to raw textual or numeric records, may include complex data types from the field of multimedia (e.g., images, audio, video, geometric objects), or time related data streams (e.g., financial pricing streams, network monitoring streams). Methods for analysis and retrieval in such complex data typically rely on the *feature vector* (FV) paradigm [5], which describes the instances of any complex data type by vectors of characteristic numeric properties (features) extracted from the instances, allowing the calculation of *distances* between FV representations of the data objects [8]. The *similarity* between two data objects is then associated with the distance between their respective FV representations.

FVs are required by many important automatic data analysis algorithms like clustering, similarity search, or classification. We can informally define the *effectiveness* (or quality) of a FV extractor as the degree of resemblance between distances in FV space, and similarity relationships in object space. Extracting effective FVs for a given data type, i.e., features that describe relevant properties of the object instances and allow their meaningful discrimination, however, is a challenging task. It usually requires a lot of experimentation and supervised information, e.g., a human expert, or labeled training data for benchmarking and optimization of candidate FVs. However, in many data analysis scenarios, the data is neither fully labeled, nor has the analyst a–priori knowledge how to classify the data.

We propose a novel approach to measure the quality of a given FV space. We rely on the image-based analysis of certain views on the components of compressed versions of the candidate FV spaces. The key assumption underlying our analysis is that the degree of *heterogeneity* of features in a candidate FV space is an indicator for the discrimination power (effectiveness) in that FV space. Based on this hypothesis, we develop two image analysis functions allowing visual or automatic benchmarking of candidate FV spaces. The analysis aims at identifying the most effective FV space

*tschreck@gris.informatik.tu-darmstadt.de
†schneide@inf.uni-konstanz.de
‡keim@inf.uni-konstanz.de
§matt@wpi.edu
¶tatu@inf.uni-konstanz.de



(a) Convex hulls (b) Minimum boxes (c) Minimum discs

Figure 1: Supervised visual benchmarking of a given FV space using simple hulls formed around PCA-projected point clouds. Depending on the fidelity of the projection, important inter-class discrimination characteristics may be visually analyzed.

from a set of candidate FV spaces for a given data set. A key property of our analysis is that by relying on the Self-Organizing Map algorithm for clustering (cf. Section 3), it operates in a largely *unsupervised* way. Specifically, it does *not* require supervised training data.

2. BACKGROUND

In this Section, we review the feature vector approach for data analysis applications.

2.1 Feature Vector Approach

Similarity measures between complex data objects are usually implemented by two main approaches. The *transform* approach considers suitably defined costs of efficiently transforming one object into the other. E.g., the Edit or Levenshtein distance [5] is a distance measure for text based on insert, update, and delete operations. The second main approach for calculating object distances is the feature vector (FV) approach [5]. It extracts characteristic numeric values from the objects, forming vectors in high-dimensional FV space. E.g., text documents can be described by so-called $tf \times idf$ vectors based on term occurrence histograms [2]. Another example are 3D geometric models, which can be described by histograms of curvature, by volumetric properties, or by features derived from 2D projections, among others [3]. The similarity between objects is associated with the distance between their FV representations (e.g., using the Euclidean norm). FV-based applications rely on a representation of the input data in a discriminating FV space to produce meaningful results. These include content-based similarity search, where distances between a query object and candidate elements are used to produce answer lists. FV-based distances are also heavily used in Clustering and Classification applications [8, 5].

Unfortunately, for most data types to be supported, there is no absolute or optimal set of features known which should be used, but often, different features are equally promising candidates a-priori. Therefore, in practice a FV selection and optimization stage is engaged, which based on experimentation, identifies a set of features which work sufficiently well for a given application at hand. However, this stage is also usually very costly, as it relies heavily on the usage of supervised information, and on intensive experimentation and manual tuning of the feature vectors.

2.2 Measuring FV Space Quality

The FV selection problem is usually addressed by the benchmarking approach. A set of candidate feature vectors are calculated for a reference data set. Based on predefined classification information (a benchmark data set) or the judgment of a human expert, the quality of the candidate FV extractors is assessed by precision-recall statistics, or expert judgment regarding the degree of resemblance between distances in FV space and similarity relationships in object space. In a number of domains, reference benchmarks have been defined. E.g., in similarity search of 3D geometric models, the Princeton Shape Benchmark [15] consists of a database of 3D models with associated class labels. Using the benchmark, candidate 3D FV extractors can be compared in terms of precision-recall measurements observed when executing reference queries on the benchmark. While precision-recall statistics are the predominant way of performing feature vector benchmarking, also *visual benchmarking* is possible. Supervised visual benchmarking relies on classification information, using visual representations of the discrimination between object classes under different feature vector representations. E.g., in [14], a scheme relying on PCA-based projection of high-dimensional feature vector data to 2D display space was proposed. Together with simple shapes formed around the obtained point clouds, it was shown how inter-class object discrimination can be estimated from such plots, depending on the faithfulness of the projection. Figure 1 illustrates the idea.

Problematic is that the supervised approach is expensive, as it requires either a large labeled object collection, or a human expert to manually evaluate the quality of FV-based distances. Also, it is data-dependent: Whenever the underlying data changes, the benchmark needs to be updated to reflect the target data characteristics. Therefore, *unsupervised benchmarking* is very desirable, but to data a difficult problem. Certain statistical approaches were proposed for unsupervised FV space quality estimation [9, 1]. These works are of rather theoretical nature and to the best of our knowledge have not been practically leveraged yet. Therefore, approaches towards unsupervised benchmarking of competing feature vector spaces are desirable. In [12, 13], the distribution of distances between clusters found in FV space was used for FV quality estimation. In this work, we consider the distribution of individual components of cluster centers found in FV space.

3. FEATURE SPACE IMAGING

We next recall the Self-Organizing Map algorithm, a well-known data compression and projection algorithm. It is the basis for our FV space imaging technique developed in Section 4.

3.1 Self-Organizing Map Algorithm

The Self-Organizing Map (SOM) algorithm [10] is a combined vector quantization and projection algorithm well suited for data analysis and visualization purposes [16]. By means of a competitive learning algorithm, a network of reference (prototype) vectors is obtained from a set of input data vectors. The reference vectors represent clusters in the input data set and are localized on a low-dimensional (usually, 2D), regular grid. An important property of the algorithm is that the arrangement of prototype vectors on the grid approximately resembles the topology of data vectors in input space. The SOM is a compressed FV space representation obtained in an unsupervised way. Figure 2 illustrates two steps in the training of a SOM, during which data vectors are used to update the network of reference vectors..

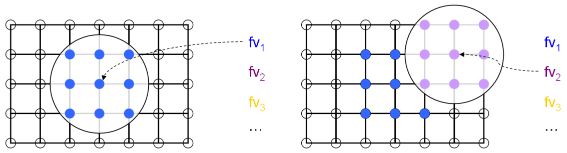


Figure 2: The Self-Organizing Map algorithm calculates a network of prototype vectors representing a set of input data vectors.

3.2 SOM Component Plane Images

Under the FV approach to similarity calculation, distances in object space are estimated by distances between FV space representations of the objects. E.g., the Euclidean distance, defined as $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ for two vectors $x, y \in \Re^n$ in n-dimensional vector space is widely used. It are ultimately the characteristics of the components (dimensions) in FV space which contribute to the calculated distances. To analyze the characteristics of the FV space components, we can visualize the individual dimensions by means of *Component Planes* (CPs) [16] obtained from the SOM representation. A CP visualizes the distribution of a given vector component over the calculated SOM. Recall that each SOM reference vector is located at a unique position on a regular grid. We can visualize the Component Plane image for component c by simply drawing a matrix of dimensionality corresponding to the SOM grid, color-coding each cell according to the normalized component value of the SOM reference vector at the respective SOM grid position. The values are normalized and color-coded such that the full component span $[c_{min}, c_{max}]$ is visualized.

Figure 3 illustrates three CPs from a FV space further discussed in Section 5. The images allow the efficient visual analysis of the distribution of component values. While the localization of component values on the SOM is not of primary concern here, their *overall distribution* is. As will be demonstrated, the *heterogeneity* of the component distribution may be used as an indicator for the discrimination power contained in a given FV space. This in turn is valuable for analyzing and evaluating a given FV space. Note that this analysis is unsupervised up to the setting of the SOM training parameters, for which in turn data-dependent heuristics and rules of thumb are known [11].

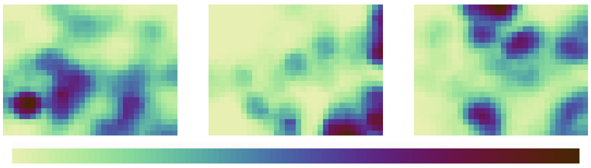


Figure 3: Three Component Plane (CP) images for a Self-Organizing Map of size 32×24 calculated from the VOX FV space (cf. Section 5). Applying $[min, max]$ normalization and applying the color scale shown below, each image visualizes the distribution of a given vector component on the SOM grid.

The characteristics of all components of a d-dimensional FV space may be visualized by laying out all d CP images obtained from the respective FV space's SOM in a matrix layout. This visualization (Component Plane Array, CPA), gives a compact image of the distribution of FV components. We can use the CPA (a) to visually assess overall component distribution characteristics, and (b) to identify the correlation structure of the respective FV space. Figure 4 shows the CPA of the CP images from the 343-dimensional VOX FV space (cf. Section 5).

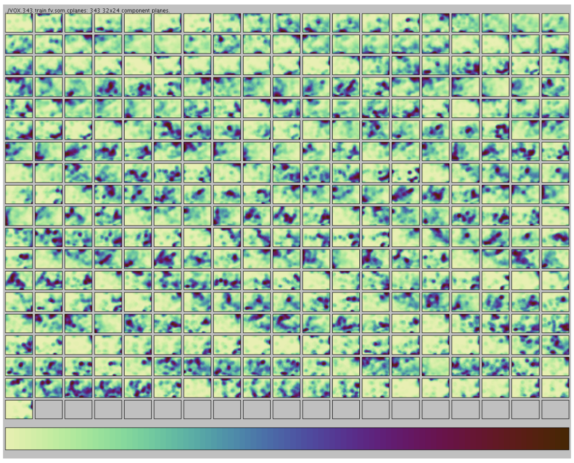


Figure 4: Component Plane Array (CPA) image of the 343-dimensional VOX FV space (cf. Section 5).

4. COMPONENT IMAGE ANALYSIS

In [13], it was proposed to use Component Plane Array images for the comparative visual analysis of discrimination power in different FV spaces. It was argued that the discrimination power contained in a given FV space can be estimated from the degree of heterogeneity of the individual FV space components in the respective SOM representation. The key hypothesis was that the more uniformly distributed the individual FV components are in the $[min, max]$ component intervals, the better the chances are that the given FV space meaningfully discriminates object clusters. In [13], ev-

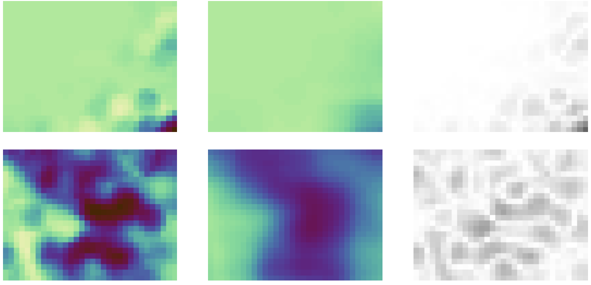


Figure 5: The *dtb* score is calculated over the difference image (right column) between an original Component Plane image (left column) and a blurred version of it (middle column). The top row shows a CP image of low heterogeneity, while the bottom row shows one containing more heterogeneity (the *dtb* scores amount to 17.84 and 81.14, respectively, in this example).

idence for this hypothesis was presented by visually relating the distribution of FV components with supervised ground-truth benchmark information. Here, we develop two image analysis functions capturing the described heterogeneity notion that was previously evaluated only informally by the user.

4.1 Function Based on Difference Image

The first function for measuring the degree of heterogeneity in a Component Plane image is based on the unsharp image filter, a standard digital image processing technique [7]. It measures the degree of CP image heterogeneity by the amount of image information lost when blurring the image. We implement the measure by considering a given Component Plane image as a gray-value image $CP(x, y)$ in the domain $[0, 1]$. We blur the image by moving an averaging kernel k over the image, replacing each gray value by the average over all pixels within the neighborhood k around that pixel. We then compare the original image with its blurred version $CP^k(x, y)$ by summing the absolute differences of the original and the blurred image pixels. Intuitively, in regions with low image heterogeneity, the values of the blurred pixels will be similar to the original values, yielding low differences. Conversely, in image regions with much heterogeneity, the blurring process will smooth out much of the image heterogeneity, resulting in higher differences.

We call this function *dtb* (difference to blurred) score, and parameterize it with the blurring kernel size k. It is defined as:

$$dtb(CP_i, k) = \sum_x \sum_y |CP_i(x, y) - CP_i^k(x, y)|, \qquad (1)$$

where $CP_i(x, y)$ is the gray value Component Plane image for FV component i, and $CP_i^k(x, y)$ is a blurred version obtained by applying the blurring kernel k on CP_i. Figure 5 illustrates the calculation of the *dtb* score for two CP images. The *dtb* score is easily extended to work on Component Plane Arrays of n CP images by averaging the *dtb* scores for all individual CPs:

$$dtb(CPA, k) = \frac{1}{n}\sum_{i=1}^{n} dtb(CP_i, k). \qquad (2)$$

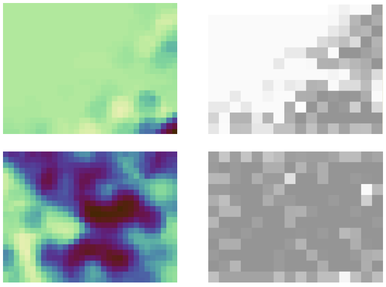


Figure 6: The Entropy score E measures Component Plane image heterogeneity by averaging the Entropy values calculated for all sub-images of a CP image. The top row shows a CP image of little heterogeneity, while the bottom row shows one containing more heterogeneity. The right column visualizes normalized entropy scores evaluated on 16×12 sub-images as a gray-value image. The E scores amount to 0.97 and 1.37, respectively, in this example.

4.2 Function Based on Image Entropy

Again we consider each Component Plane image CP as a gray value image in the domain $[0, 1]$. Since we are interested to assess the distribution of gray values H in the image, we are computing histograms over the gray levels. The histogram over gray values in a $2D$ image can be regarded as a $1D$ function $H(g)$ where the independent variable is the (appropriately quantized) gray value g, and the dependent variable is the number of pixels $H(g)$ with that gray value. Since all pixels in the image show a distinct gray value, the sum of the histogram bins must be equal to the number of image pixels $N = x * y = \sum_{g=G_{min}}^{G_{max}} H(g)$, and g corresponds to the index of quantized gray values, e.g., $G_{min} = G_0 = 0$ and $G_{max} = G_{255} = 255$ for a 8-bit quantization to 256 unique gray values. The histogram function is equal to the scaled probability distribution function $p(g)$ of gray levels in that image: $p(g) = \frac{1}{N}H(g)$ where $\sum_{g=G_{min}}^{G_{max}} p(g) = 1$. Based on the probability distribution we compute a measure for the information contained in the image. In general, any function $\sigma()$ can be used , but a common way of doing so is applying Shannon‘s Entropy E [6], which in theory is a measure for the number of bits required to efficiently encode an image [7]. If the probability of gray level g in a given image is represented as $p(g)$, the amount of information E contained is $E = -\sum_{g=G_{min}}^{G_{max}} p(g)\log_2(p(g))$. Maximum information content results if each gray level has the same probability (a uniform histogram corresponds to maximum information). Minimum Entropy results if the image contains only one single gray level.

Since the task is not only to analyze the whole image, but also analyze *local* patterns in the image, we use a regular grid gc of size $s = |gc|$ to partition the input image CP into s grid cells $gc_j(CP), j = 1, \ldots, s$, and then apply the method described above to compute the Entropy values for each grid cell as $E(gc_j(CP))$. We average over the local

Entropy scores to arrive at the global image Entropy score for a Component Plane image CP:

$$E(CP) = \frac{1}{s} \sum_{j=1}^{s} E(gc_j(CP)) \quad (3)$$

Figure 6 visualizes the Entropy-based analysis on two Component Plane images. To obtain the overall entropy score $E(CPA)$ for a Component Plane Array CPA, we finally average the Component Plane Entropy scores $E(CP_i)$, for all n Component Plane images CP_i contained in CPA:

$$E(CPA) = \frac{1}{n} \sum_{i=1}^{n} E(CP_i) \quad (4)$$

The higher the ranking score $E(CPA)$ of the Component Plane Array, the higher the heterogeneity we associate with the underlying FV space.

5. EVALUATION

Next we evaluate our analysis methods in terms of how good they resemble supervised analysis methods relying on human expert benchmarking. We base our evaluation on a FV vector benchmarking data set from the field of 3D similarity search, where the task is to define the most discriminating FVs for 3D geometric models, which in turn should allow the most effective similarity search using FV space distances. Equipped with a number of 3D FV spaces of significantly varying discrimination power, we generate Component Plane Array images, and compare their unsupervised image analysis scores with respective supervised benchmark scores.

5.1 Benchmark Dataset

The dataset used is the *train* partition of the *Princeton Shape Benchmark* (PSB-T) [15], popular for evaluating 3D similarity search algorithms. The PSB-T consists of 907 3D meshes modeling objects like animals, humans, vehicles, and so on. The models were manually grouped into 90 equivalence classes by shape similarity [15]. This constitutes the ground truth for evaluation of the retrieval precision of a given candidate FV space. Briefly, evaluation is done by using each object as a query against the benchmark. The list of answers obtained is evaluated by precision–recall statistics over the relevance of the answers [15, 2]. These statistics in turn are used to rank the effectiveness of the different FV extractors.

From a variety of FV extractors studied in previous 3D retrieval work [4, 3], we use a subset of 12 of the most robust methods to extract 3D FVs from the PSB-T benchmark. The individual methods consider geometric model properties such as curvature, volumetric- and image-based features and vary in dimensionality (tens to hundreds of dimensions). The individual FV spaces possess varying average discrimination power - some FV spaces work well for similarity searching, others perform poorer. Table 1 gives the used FV space names (FV name), along with respective FV dimensionalities (dim.) and R-precision (R-prec.) as the supervised discrimination precision score [4], relying on the PSB reference classification. Larger R-precision scores indicate better discrimination. Note that unlike in other data analysis domains (e.g., classifier analysis), in multimedia retrieval precision scores below 50% are not uncommon [15, 4], depending on the benchmark considered.

5.2 Analysis Score Calculation

For each of the 12 PSB-T FV spaces, we generated Component Plane Array images by first calculating Self-Organizing Maps for the FV spaces, using rectangular SOM grids of size 32 × 24. We iterated 150 times over all database elements during SOM calculation, stabilizing the SOM results. For each calculated SOM and vector component, we then generated a Component Plane image by scaling the respective component values linearly to the interval $[0, 1]$ and applying the color scale included in Figure 3. The actual Component Plane images were rendered as 320 × 240 checkboard-like raster images, where each component value was used to color-code the respective cell on the SOM grid.

We then apply our visual analysis functions introduced in Sections 4.1 and 4.2 on the generated images. We obtain an aggregate analysis score for each FV space by averaging the analysis values for each of the respective components. The *dtb* scores were calculated by applying Equation 2 from Section 4.1 using a rectangular kernel of 5 × 5 pixels for blurring. The *Entropy* scores were calculated by evaluating Equation 4 from Section 4.2 on the CPA images. 8 bit gray value quantization was used, and the sub-image grid *gc* for analyzing each Component Plane image was set to 16 × 12, yielding grid cell sizes of 20 × 20 pixels.

5.3 Results and Comparison

Table 1 lists the *dtb* and the E scores for each of the 12 FV space representations of the PSB-T benchmark. By their definition, increasing score values indicate increasing component heterogeneity. Comparing the scores with the R-precision values, we observe a high degree of resemblance of the R-precision scores by our analysis scores. This is an interesting result, as our analysis scores are based on purely unsupervised (i.e., automatically extracted information), while the R-precision scores rely on expert-generated supervised information (the PSB classification).

Table 1: FV spaces with supervised discrimination benchmark scores (R-precision) and unsupervised image-analysis scores.

FV name	dim.	R-prec.	dtb	E	comb.
DSR	472	42.61%	28.33	20.73	587.23
DBF	259	31.16%	27.15	21.46	582.30
VOX	343	31.13%	25.29	15.38	388.94
SIL	375	28.15%	31.94	21.30	680.26
CPX	169	27.08%	26.01	18.93	492.50
3DDFT	173	25.08%	20.41	18.31	373.76
GRAY	120	22.54%	28.66	19.41	556.22
RIN	155	22.52%	15.53	14.68	228.07
H3D	128	20.20%	25.07	18.19	456.06
SD2	130	18.36%	11.74	15.18	178.24
COR	30	15.75%	17.83	18.97	338.24
PMOM	52	14.82%	12.22	5.80	70.89

We take a closer look at the resemblance between the unsupervised and the supervised benchmark scores. Table 2

Table 2: Errors of the unsupervised ranking, measured against the supervised ranking.

FV name	R-prec.	dtb	E	comb.
DSR	1	+2	+2	+1
DBF	2	+2	-1	+1
VOX	3	+3	+6	+4
SIL	4	-3	-2	-3
CPX	5	0	+1	0
3DDFT	6	+2	+1	+2
GRAY	7	-5	-3	-3
RIN	8	+2	+3	+2
H3D	9	-2	-1	-3
SD2	10	+2	0	+1
COR	11	-2	-6	-2
PMOM	12	-1	0	0

presents the discrimination power ranks assigned to the individual FV spaces, for the R-precision evaluation, as well as the unsupervised CPA-based analysis. We use the R-precision ranking as the base line, and compare the deviation of the ranks assigned to the FV spaces by the image analysis functions. Again, the image-based analysis functions closely resemble the supervised ranking, deviating just one or two ranks positively or negatively from the supervised ranking, for most of the candidate FV spaces. Specifically, the best and the worst performing FV spaces, according to supervised benchmarking, are clearly identified by the automatic analysis. This avoids the risk of erroneously choosing one of the bad performing FV spaces when relying purely on the automatic discrimination power analysis for FV space selection.

While both analysis functions come close to the baseline supervised ranking, there are certain differences in the rankings. Considering the functions implement different heterogeneity definitions, a natural idea is to combine both scores into an *ensemble* score, unifying both "opinions" on FV space discrimination. Building ensembles by combining classifiers of different types is a well-known approach for improving classification accuracy. As both measures indicate increasing component heterogeneity by increasing scores, we are able to combine them simply by multiplication. The last columns in Tables 1 and 2 list the combined score results. The FV ranking based on the combined unsupervised score closely resembles the ranking based on the supervised benchmark, over- or undershooting only a few ranks for most of the FV spaces.

The correlation of the individual and the combined scores with the supervised rankings can be analytically compared by *Spearman's Rank Correlation Coefficient*, a normalized measure for the degree of correlation between sorted lists. According to this measure, *dtb* and *Entropy* achieve 74.8% and 64.3% rank correlation, respectively. The combined score improves over the individual scores, achieving a correlation of 79.8%. We also evaluated the correlation of the supervised and the unsupervised scores by means of regression analysis. Figure 7 gives the regression analysis of the R-precision and the *combined* scores using the logarithmic regression model. The correlation is confirmed at squared correlation coefficient $R^2 = 51\%$.

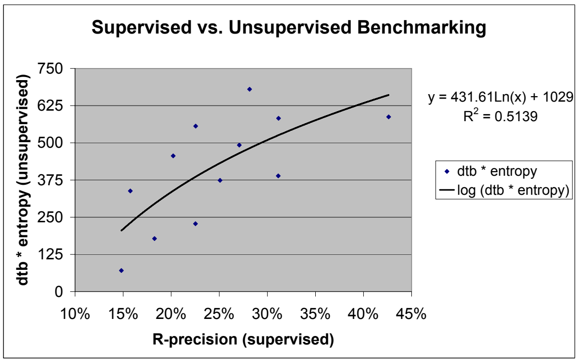


Figure 7: Regression analysis.

5.4 Discussion

Summarizing the experimental results, our image-based FV analysis closely resembles the supervised benchmarking of the PSB-T benchmark described in 12 candidate FV spaces. A combination of both individual analysis functions was found to yield the best correlation. The evaluation supports the idea that unsupervised FV space benchmarking is possible using image-based analysis of certain (SOM-)compressed FV space views. We state that we also performed extensive experiments on synthetically generated data sets that simulate FV spaces of varying discrimination power, validating our results. We propose to use the unsupervised estimator as a tool to complement or replace the supervised FV selection approach. An advantage of our method is that it is data-independent. Contrary to benchmark-based FV selection, which requires to define a new ground truth whenever the database content changes substantially, our method works automatically.

The presented discrimination power estimation is based on measuring heterogeneity among the *components* of cluster prototypes in the considered feature vector space. In [12, 13], we performed a similar estimation based on analysis of the heterogeneity of distances between the *cluster centers*. Noting that both estimators consider different characteristics of the same FV space representation, it seems promising to unify both discrimination power estimates into a *combined* component and distance-based estimator. Preliminary experiments indicate that the combination of both component-based and the distance based estimator further increase the estimation accuracy over the individual estimators. Future work will study how to best combine all three aspects into in a single estimation function, and quantify the additional improvements achievable.

6. CONCLUSIONS

FV space discrimination analysis is an important problem in many application domains relying on FV representations for similarity calculation. We introduced an approach for automatic, unsupervised FV space discrimination analysis based on analysis of certain component-based image representations of compressed FV spaces. The method allows unsupervised benchmarking of FV spaces. It is useful when there is no ground truth available on the data for which FVs need to be extracted. In case where supervised information is available, our approach is recommended as an additional unsupervised "opinion" on the discrimination power to expect in a given FV space. Experiments performed on a comprehensive data set showed that the FV ranking produced by the proposed method highly correlates with that of a corresponding supervised discrimination benchmark. An additional advantage of the method is that it has an intuitive visual representation (heterogeneity of the CPA images) that can be well understood and interpreted by the user.

We consider these results promising for future work. We like to refine the image-based analysis functions, and test them on additional benchmark datasets for other data types, e.g., benchmarks from image similarity search and classification. We here applied our analysis functions to the problem of automatically benchmarking a number of FV extractors of given vector dimensionality. The analysis is expected to be also applicable for the dimensionality selection problem, a task for which we like to test our method on.

7. ACKNOWLEDGMENTS

This work is based on a submission currently under review for International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG2008). This work was partially funded by the DELOS Network of Excellence on Digital Libraries (www.delos.info). We thank Dietmar Saupe and Dejan Vranic for providing the 3D FV extractors and for valuable discussion.

8. REFERENCES

[1] C. Aggarwal. On the effects of dimensionality reduction on high dimensional similarity search. In *Proc. ACM Symposium on Principles of Database Systems (PODS)*, 2001.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys (CSUR)*, 37:345–387, 2005.

[4] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic. An experimental effectiveness comparison of methods for 3D similarity search. *Int. Journal on Digital Libraries, Special Issue on Multimedia Contents and Management*, 6(1):39–54, 2006.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, New York, 2nd edition, 2001.

[6] M. Esteban and D. Morales. A summary of entropy statistics. *Kybernetika*, 31(4):337–346, 1995.

[7] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[8] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kauffman, 2nd edition, 2006.

[9] A. Hinneburg, C. Aggarwal, and D. Keim. What is the nearest neighbor in high dimensional spaces? In *Proc. Int. Conference on Very Large Data Bases (VLDB)*, pages 506–515, 2000.

[10] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

[11] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som_pak: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996.

[12] T. Schreck, D. Fellner, and D. Keim. Towards automatic feature vector optimization for multimedia applications. In *Proc. Annual ACM Symposium on Applied Computing, Multimedia and Visualization Track*, 2008. to appear.

[13] T. Schreck, D. Keim, and C. Panse. Visual feature space analysis for unsupervised effectiveness estimation and feature engineering. In *Proc. IEEE Int. Conference on Multimedia and Expo (ICME)*, 2006.

[14] T. Schreck and C. Panse. A new metaphor for projection-based visual analysis and data exploration. In *Proc. IS&T/SPIE Conference on Visualization and Data Analysis (VDA)*, 2007.

[15] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Proc. Int. Conference on Shape Modeling and Applications (SMI)*, 2004.

[16] J. Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.