Müller, Wolfgang; Zech, Markus; Henrich, Andreas; Blank, Daniel

VisualFlamenco : Dependable, Interactive Image Browsing Based on Visual Properties

Date of secondary publication: 03.03.2025

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-1068751

Primary publication

Müller, Wolfgang; Zech, Markus; Henrich, Andreas; Blank, Daniel (2008): VisualFlamenco : Dependable, Interactive Image Browsing Based on Visual Properties, in: Proceedings: 2008 International Workshop on Content-Based Multimedia Indexing : CBMI 2008 ; London, United Kingdom, 18 - 20 June 2008, Piscataway, NJ: IEEE, pp. 568–575, doi: 10.1109/CBMI.2008.4564998.

VISUALFLAMENCO: DEPENDABLE, INTERACTIVE IMAGE BROWSING BASED ON VISUAL PROPERTIES

Wolfgang Müller, Markus Zech, Andreas Henrich, Daniel Blank

Media Informatics
University of Bamberg
Germany
wolfgang.mueller@uni-bamberg.de

ABSTRACT

Faceted search as a method for exploratory search is a search interface technique for Boolean retrieval that provides a good compromise between the wish to guide the user and the need to give freedom to the user. Both, guidance and freedom are needed in order to enable him (or her) to accomplish his search goal as quickly and efficiently as possible. The search goal can be both, to retrieve a target image (lookup search) or to get an overview of a collection as the basis for accomplishing complex search tasks (exploratory search).

VisualFlamenco is about bringing content-based image retrieval and faceted search together. While there is some preexisting work towards faceted search based on automatically extracted metadata, the base innovation of our work is to *visualize* the meaning of facets and let the user pick visualized facets directly.

The main point is this research is to give the user the right *expectations* about what the system can achieve. We see this as a step towards *dependable* multimedia search, search interfaces that do not promise more than they can achieve. A mismatch of expectations and possibilities has been denoted by some as one of the main problems in current research about multimedia search.

Within this paper we present a visual faceted search tool that seeks to provide dependable image search. We describe its base idea, the features used, as well as user experiments done using the system.

1. INTRODUCTION

Content-based Image Retrieval (CBIR) has been a topic of research for a long time. For an excellent overview of the area, see [1]. The motivation for CBIR is the fact that textual annotation of images for the goal of retrieval is costly and inherently incomplete. This reason becomes even stronger in the times of user-generated data, present in sites such as flickr.com, youtube.com and the like. Annotation (tags) are the current way out of the problem, but in our experience with flickr.com, gained with (presently unpublished) measurements on flickr.com crawls with 10'000 to 100'000 images, tags are rarely used to describe visual properties of images, and $\approx 30\%$ remain untagged.

Hierarchical Faceted Search (HFS [2]) is a way to express Boolean information retrieval queries in a manner useful for exploratory search (for a description of different types of search, see *e.g.* [3]). It seeks to combine the benefits of hierarchical search (hierarchies guide the search) and Boolean queries that essentially give the user the freedom to combine search terms. In HFS each object is considered to be a leaf in multiple hierarchies. For example, the first author of this article is a man. The gender of a person is a facet, a one-level hierarchy with two leaves. He lives in Bamberg, which is in Bavaria, which is situated in Germany, which in turn is situated in Europe (a four-level facet). He is a computer scientist, concerned with multimedia, as well as indexing (this makes him several leaves in the *scientific interest* facet).

A HFS search interface enables the user to select an inner node of a facet. It will then return as a result all leaves (documents) that are in the subtree of the selected node. When selecting inner nodes in multiple subtrees, the *intersection* of leave (document) sets is returned as a result.

Thus, a HFS system would both support users in finding female researchers of any subject from China (researcher, woman, Asia/China) or physicists from Europe (researcher/physicist, Europe). Which hierarchies are to be searched is at the discretion of the user.

HFS-like techniques have been used in image search. For example MetaXa, as described in [4], seeks to provide facets by extracting simple semantic labels (indoors/outdoors, sharp, etc.) from the image. The results are very convincing. However, we do feel that in the current setting, there is still *a need for visual features.* When looking for a good photo, visual properties matter, too.

As an example, just imagine that you are looking for images to illustrate a calendar. You will tend to choose images by the people that are on the images, sure, but you will also

This work is funded by the Deutsche Forschungsgemeinschaft DFG HE 2555/12-1.

try to choose images that have given visual properties, that will look good on a calendar. Being able to choose images taken outdoors surely helps finding your best image made on your favourite walk (for instance). But being able to tell the system that you are looking for greenish images with much structure in them will get you much farther. Being able to do so is the goal of VisualFlamenco.

This paper is organized as follows: In the next section we describe visual faceted search (VFS), in section 3 we will discuss the features used in our prototype. Sec. 4 will show an example query. Sec. 5 will describe our user experiments. Section 6 finally considers our thoughts about improving the efficiency of faceted search. Finally, section 7 summarizes our lessons learned and points towards future work.

2. VISUAL FACETED SEARCH

Within this section, we will describe visual faceted search, discussing the differences to other approaches.

Semantic learning and the expectation mismatch: Recognition of semantic categories using off-line learning techniques has been at the center of attention in recent years, especially as part of the TrecVid video retrieval benchmark (*http://trecvid.nist.gov*). While the overall current results are impressive, there is still a considerable mismatch between promise and reality. The user is led to think that the system visually recognizes semantic concepts. However, often precision and recall are below 50%, *i.e.* the system will miss out on the majority of matches, and the user does not have any possibilities to *help* the system.

Among others, Alan Hanjalic criticized this behavior at a panel at ACM MM 2007, asking for more *dependability*, *i.e.* giving promises to the user he can depend on.

VFS: Low expectations, good results: *The approach of visual faceted search (VFS) strongly contrasts what concept learning tries to achieve.* In VFS each image is described by a list of Boolean visual features. One possible feature is *e.g.* the *dominant color* of the whole image or of a particular region.

Using this feature, the user can choose to see only those images that contain given dominant color in a given region. Thus, by one click they can narrow down the set of search results considerably. In a few clicks, using few features, the user will have found his search target. This search often requires only a few steps although the non-semantic interface does not give any promises comparable to semantic learning-based systems.

The fact that features are presented *as is* to the user encourages the user to help the system bridge the semantic gap. The usefulness of this approach is best described when comparing it to *query by visual example (QbvE)*. In QbvE, the user is asked to provide a query image to a system that he perceives as a black box. The black box then provides a query result. The user can only guess what mechanism provides the results.

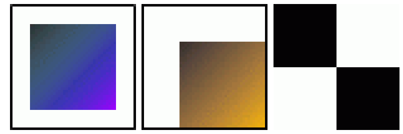

Fig. 1. Three example visual facets for VisualFlamenco. Left: Most pixels in the center are to be blue. Center: Orange should be the dominant color in the lower right of the image. Right: the structure of the image should be rather coarse.

This weakens his ability to provide useful feedback, severely limiting the usefulness of this method.

VFS: An *intuition* about visual features: In VFS, however, the user is shown a) which features are at his disposal, and b) by using the system click by click he can get a *feeling* which feature provides which type of result. So indeed, the user can *learn* how to use the system *better*.

Combining visual, EXIF and semantic features: In the previous lines we have made a case for our contribution, *i.e.* the use of visual features as facets in faceted search. However, as you may have noted, nothing precludes the use of several types of features in one system: Visual facets can be mixed with facets based on EXIF data (*e.g.* the creation date of each photo) and semantically learned features. The *combination* of all features probably will give the best results. VFS provides a *common framework* for use of all these features.

Difference to QBIC and query by sketch: It must be noted that there are several systems that enable the user to freely specify visual aspects of images. However, we feel that VFS provides two non-trivial, useful advantages over such systems.

Query by sketch systems enable the user to *draw* the query example and then perform a query by example on the drawing. In our view of the literature (as a starting point use for example [5, 1]), the main weakness of such systems is that the systems do not provide the user with a knowledge of how the drawing is used to find the query result, and thus which changes of the drawing will bring the user closer to the desired query result.

QBIC (query by image content) [6] is a system enabling users to specify the look of regions. In some respects this is close to VFS.

However, what is missing in both QBIC and query by sketch is the *guidance* for the user. Facets make it easy to decide for the user how to go about a search. Another important point is that the faceted *data organization* has many advantages for efficient search, thus carrying the promise for good performance while being interactive and intuitive.

While we have outlined the principal ideas and motivations in this section, we will go into more detail below, describing our VFS prototype in detail.

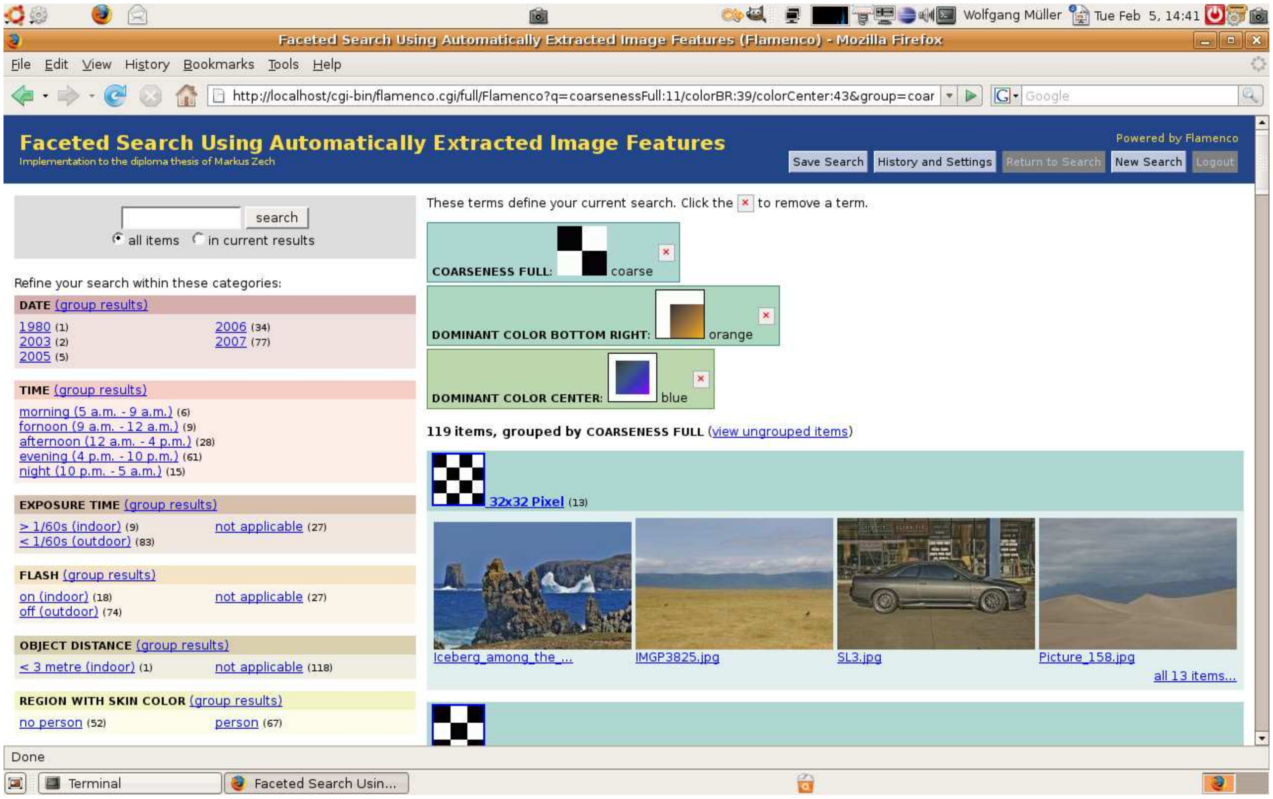

Fig. 2. Searching our collection with the facets given in Fig. 1.

Type	Facet description
EXIF (*i.e.* non-visual)	Date
EXIF	Time
EXIF	Exposure time
EXIF	Flash
EXIF	Object distance
EXIF	Camera model
full, region	Dominant color
full, region	Coarseness [8]
full, region	Contrast
full	Coherent color region
full	Region with skin color
full	Straight edges
full	Directionality horizontal
full	Directionality vertical

Table 1. Facets used in current prototype.

3. OUR VFS PROTOTYPE

Our prototype called VisualFlamenco [7] explores the combination of visual and semantic features. It is based on the original Flamenco prototype (*http://flamenco.berkeley.edu/*). With a small software change we were able to use it also for visual facets. See some facets in Fig. 1.

We wrote JAVA software (*http://java.sun.com/*) automatically extracting features from images, and inserting each image in the proper manner into VisualFlamenco. The software described here thus extracts high- and low-level features, as well as EXIF data from JPEG images and feeds these features to Flamenco as facet specifications. Flamenco then uses these specifications for creating a user experience.

For describing better the facets used in our prototype, let us turn our attention to Tab. 1. In the table, the facet type denotes features derived from EXIF data of the image as "EXIF", features that are obtained from one of 5 image regions (center, upper left, lower left, upper right, lower right) as "region", and features derived from the full image as "full".

We first describe the most important visual facets, continuing with a short description of the less important visual facets. The usefulness of these is derived from our user experiments.

Dominant color in region: a simple visual facet: The most simple to derive, and (as the experiments outlined below suggest) most simple to use visual facet is the dominant color in a region.

In order to extract this facet, for each image and each image region the most frequent color is extracted. For this purpose we obtain 36-dimensional color histograms for each re-

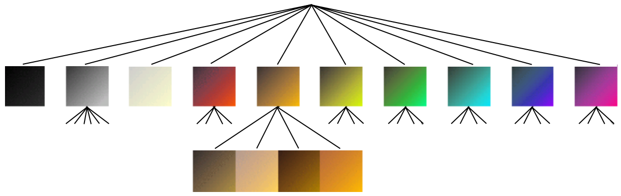

Fig. 3. Color feature hierarchy used in our prototype.

gion and pick the bin that corresponds to the highest number of pixels.

Each color bin is either an inner node or a leaf in a two-level color hierarchy (see Fig. 3). The user can choose to specify the most frequent color bin of an image region as a *hue* (coarse color specification). He then can refine this color specification as a *region* in HSV color space.

One important point for usability is the *visualization* of the color facet. Each dominant color facet corresponds to an *image region* but also to a *region in color space*. While it is straightforward to visualize the region, we found that we gain in usability if we *visualize the color region* by showing a *color gradient* in the facet visualization as depicted in Fig. 1 on the left and in the middle.

Dominant coarseness level: visualizing texture properties: Colleagues whom we describe the system tend to agree to the idea that *color* can be visualized such that a user has an idea how to use the system. However, often it is doubted that texture features can be used in a similar manner. We found the *coarseness* of an image a useful facet in this respect. The coarseness feature derived from [8] describes the approximate size of the *dominant structure* in the image. Coarse images, such as portraits usually have few dark/bright changes, fine-grained (mostly landscape) images contain structures such as forests seen from the distance (*i.e.* many small trees).

Similar to color facets described in the previous section, we first enable the user to distinguish fine-grained, medium and coarse images in the first level of the facet. He then can choose to specify more precisely the coarseness of each image region. A visualization example is given in Fig. 1 on the top right. Figure 2 shows a screenshot of the prototype.

A short description of the other facets:

EXIF facets: The following facets were extracted from EXIF data.

- **Date:** The date the photo was taken. The granularity of the facet is chosen to be by year, month, day. Note that this might be split into more facets, as it would make sense to be able to look for photos taken in the end of the year, or photos taken on a given weekday, or photos taken on Easter or other religious holidays that are not on the same day each year.
- **Exposure time:** The exposure time used to take the image. Long exposure indicates darkness. Many current cameras can change aperture only slightly. The "work" is done by changing the effective ISO number of the sensor and the exposure time.
- **Flash:** Did the flash fire when the photo was taken? The use of flash is well-correlated with the scene being an indoor-scene.
- **Object distance:** This reflects the camera-to-object distance. Above a certain distance, the probability is high that the scene is an outdoor scene [9].
- **Camera model:** While in many scenarios this facet is of little use, it does make sense in situations where users tend to do a certain type of photos only with a given camera.

Full image and image region visual facets: The following visual facets were extracted from the full image, *as well as* from the following regions: upper left quarter, lower left quarter, upper right quarter, lower right quarter, center quarter of the image.

- **Dominant color:** The dominant color as described above.
- **Dominant Coarseness:** The dominant coarseness as described above.
- **Contrast:** This facet enables the user to choose between images with many short-range changes of brightness, and images with softer changes of brightness.

Full image visual facets: The following visual facets were extracted from the full image.

- **Coherent color region:** Size and color of the largest coherent color region.
- **Region with skin color:** A given color range was identified as probable skin color. The user can choose images that are largely or less largely composed of skin color. This way, one can discern portrait images from other images. Of course, this feature is inherently ethnocentric. The "skin color" color range has to be picked in function of the majority of the people that are likely to appear on the photos. The feature can work well in regions with a clear majority skin color. In melting pots this is less likely to be a useful feature. A possible extension of such a feature would be to be able to choose what type of skin one is looking for.
- **Straight edges:** This feature seeks to characterize images by the number and length of straight edges in the image. It is quite costly to obtain, all in all difficult to use, but it does help the user in finding images with many buildings on them.

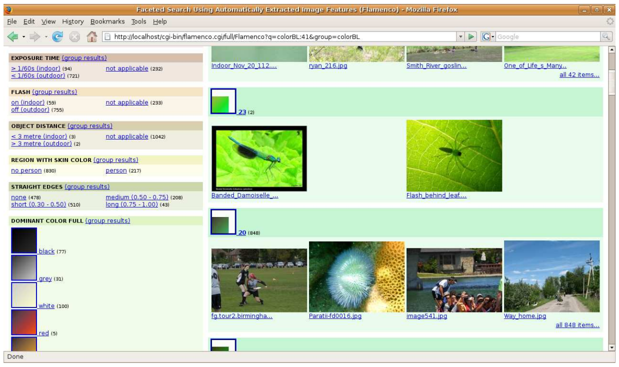

Fig. 4. View after having chosen green as the most frequent color on the lower left. Note that this already yields some sports images.

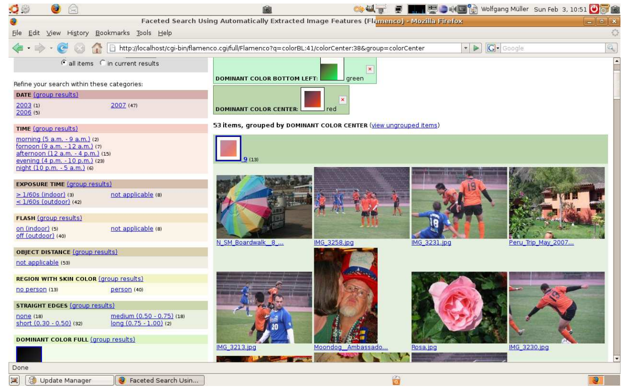

Fig. 5. View after having chosen green as the most frequent color in the lower left and red as the most frequent color in the center.

Directionality vertical: This feature originally proposed in [10] seeks to characterize images by the main *direction* of lines in the image. Images with a strong vertical directionality contain many items such as flagposts and the like.

Directionality horizontal: The same as the previously described feature. Strong horizontal directionality usually means that the image does contain a clearly delimited horizon.

4. A REAL-WORLD SEARCH EXAMPLE

In our opinion (and this point appears to be shared by people to whom we demonstrated the system), the strong point of VFS is providing an intuition about the features used in the system. A real-world search example gives an impression about this strength.

We want to find closeup images about people playing soccer. How do we go about it? We first consider a typical closeup of people playing sports on a lawn. The ground will probably be green. So, we ask VisualFlamenco to show only images that have a shade of green as the dominant color in the lower left. The result is shown in Fig. 4. Please note that all of these screenshots only show *partial results.*

At the next step, we make use of the idea that usually football players wear clothes in vivid colors, say red. The result is shown in Fig. 5.

Thirdly, we show what happens after also asking for images that are *coarse* in their center favouring a few people on the image over crowds or forests. Fig. 6 shows the result. In fact, here the new facet does not provide us with any noticeable improvement. So we remove it from the query and we restart at the state of Fig. 5.

We try imposing coarseness not only in the center, but in the full image. This does what we expect, shown in Fig. 7.

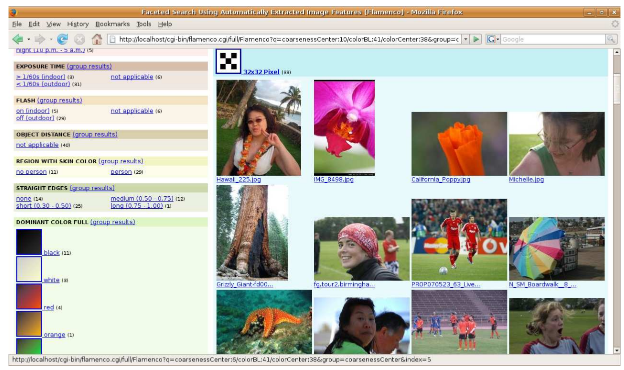

Fig. 6. View after having chosen a high coarseness level in the center in addition to the color facets selected in Fig. 5.

We further increase the fraction of soccer photos in our query results by asking for green as the dominant color also in the lower right of the image, yielding Fig. 8.

Not *all* resulting images are actually soccer photos, but a large fraction. The user can go on from there by *e.g.* finding out the time at which some good images were taken and trying to get other images from the same series. Another possibility would be to enter these images into a QbvE system, thus effectively combining explorative search via facets with lookup search via QbvE.

5. USER EXPERIMENTS

In addition to continuous “self-tests” of this system, we did user experiments. Our subjects were six students, one student in psychology (female), four students in business information systems (male), and one student of politics (male). Except for one student, no-one had any experience in content-based retrieval. In the beginning of each experiment, textual

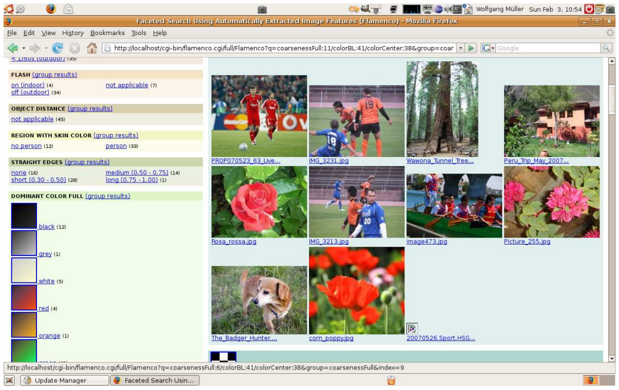

Fig. 7. View after having chosen a high coarseness level in the *full image* in addition to the color facets selected in Fig. 5.

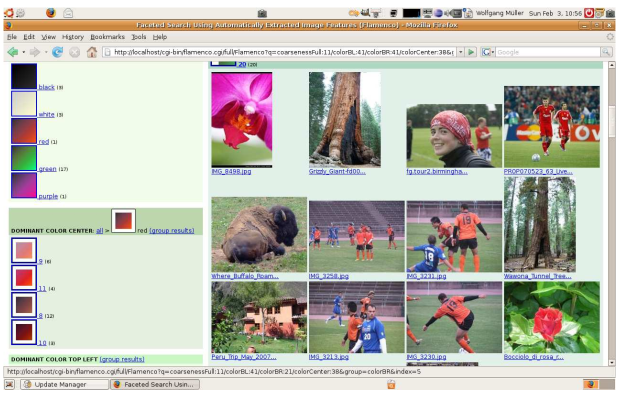

Fig. 8. We now have chosen green as the most frequent color in the lower left *and* the lower right quarter of the image, red as most frequent color in the center and high coarseness in the full image.

faceted search was described and demonstrated, then the features used for VFS were described.

Afterwards, tests were performed using a collection of 11'000 flickr images. First, participants were asked to get to know the system in 20 minutes of open-ended browsing. Then, 3 category searches were performed (among the categories *sunset, cityscape, outdoor sports, winter landscapes, images from Egypt, submarine, forests and trees, lakes*). In each search, one class was given, and the users had to find 3 images for that class within 10 minutes. Thus, 3 such experiments were completed in 30 minutes. Following category search, users had to search for two *target images* that were shown to the users (10 minutes per target). Where needed, users were helped to choose color facets by a tool that translated HSV values into color bins. Comments of the users were noted, queries were logged.

Even though we initially considered some of the categories as hard, 17 out of 18 category searches were successful. Experimental data support that VFS enabled the users to reach their goals using widely differing queries, *i.e.* the system gives the user browsing freedom.

9 of 12 target searches were successful. One target was not found by anyone, while one other target was only found by one participant. The target not found pinpoints one difficulty of the method: in some cases where there are many gradients in an image, the dominant color is hard to guess, *creating a mismatch between expectation and result*.

In a post-experiment questionnaire, all except one of the participants considered our approach as very useful (grade A in a scale from A to E) or useful (grade B). The dominant color was overall considered the most useful facet (grade A-B). It appears that coarseness facets are *very* useful to some users (A), and completely useless for others (E)[1]. More detailed results can be found in [11].

6. CONSIDERING EFFICIENCY IN VISUALFLAMENCO

As outlined in the previous sections, we do have reached a useful state of our system. On the other hand, there are still a couple of interesting, challenging, open problems. We feel that some facets are useful but not adopted by the users. And there is the issue of speed. The speed of the system (some queries take more than 5 seconds to answer) is the strongest point of critique by the users in the after-experiment questionnaire.

In fact, the two problems are strongly related: In a slow system, the user will concentrate on the features he has the strongest confidence in. Trying features, and learning features is punished by long runtimes. So we have looked more strongly into the issue of what creates the long running times.

The Flamenco software that does the actual query processing (we just enter the facet information and changed the code to be able to use visual facets) is provided as open-source (free, libre) software by Marti Hearst's group at UC Berkeley. It consists of Python (*http://www.python.org*) code that builds on a web framework (*http://www.webwareforpython.org*). Indexing data are persisted in a MySQL database (*http://www.mysql.com*).

The Flamenco software sends three types of queries to the database:

Lookup: These queries are performed to find out which user interface elements are to be generated, and what they are to look like. An example is looking up the textual/visual description of a facet, as well as the background color in which the facet is to be shown.

[1] We note that there was no correlation visible between occupation and the type of features used by the participant. For example the person using the coarseness feature most often was the psychologist who had neither a vision nor a computer science background.

Fig. 9. Preview in VisualFlamenco: the user knows the result size he will get with the next click.

Set intersection with result list: This is the actual query processor that finds out which images match a query. This is a `SELECT...GROUP BY` query in MySQL that finds out all items that match a given combination of fact IDs. Items are grouped by facet values for the next level in the same facet hierarchy or (alternatively) for the next level in a facet chosen by the user.

Set intersection with count: These queries are very costly, and actually they are practically hidden from the user. In fact, Flamenco counts for each click possibility, how many items will match this click possibility. This is done by a series of `SELECT COUNT(*) FROM...` queries. The numbers are shown next to the facet visualizations (see Fig. 9).

When seeking to optimize Flamenco, we see two main approaches:

Stateful query processing: So far, Flamenco processes queries by intersecting the result sets for each facet selected by the user. Let us assume that the facets $1, 3, 5$ have been selected and the result sets being named S_1, S_3, S_5, then the overall query result would be $S_1 \cap S_3 \cap S_5$. Instead, it would be more efficient to memorize at each step the result set. After one click our result R_1 would be $R_1 = S_1$. After the second click we would obtain $R_2 = S_1 \cap S_3 = R_1 \cap S_3$ and then $R_3 = S_1 \cap S_3 \cap S_5 = R_2 \cap S_3$, thus effectively reducing the number of records of the database that have to be considered.

Suppressing the result size preview: While the result size preview is useful in giving the user an idea of what to expect (thus curing one of the main weaknesses of the Boolean retrieval model), this comes at a price. However, for small collections (up to, say, 10'000 images), processing a facet query amounts to intersecting two lists of size $< 10'000$, and displaying the corresponding photos. Clearly, displaying photos is the most costly step of a query, telling the user that a query does generate an empty result can be obtained in far less than 0.1s.

So we suggest to drop the result size preview in favour of snappier query performance, arguing that the preview becomes obsolete if what it tries to tell becomes visible in less than 0.1s. In our next design iteration we seek to verify the validity of this view. A compromise would be to *first* show the result images and processing the preview queries asynchronously.

7. LESSONS LEARNED AND FUTURE WORK

In this paper we describe our second design iteration. Our own experience, as well as the described user experiments, so far strongly support that *VisualFlamenco is a useful idea* for finding images with desired visual properties. We feel that this is a *useful complement* to known metadata-based approaches such as timelines and tagging (all the more as all can be easily combined in one interface), and *we feel that this is easier to use and easier to understand than query by sketch.*

The main lesson to be learned for us was that *features that are useful for query by visual example are not necessarily useful for VisualFlamenco*. In VisualFlamenco it is mainly important to use features that can be visualized in a way that gives a clear message to the user how to use the feature. The general idea here is to use mainly winner-takes-all features, such as *the most frequent color* instead of seeking to express multi-modal distributions.

However, there is still much work to be done. In self-experiments we got the impression that visual features call for other query expansion functionality than that provided by the (initially text-facet-based) Flamenco software. Another point of critique is that we are currently using too much screen real-estate.

Thirdly, we are in the process of re-designing the backend towards faster response times with respect to our Flamenco-based prototype.

8. REFERENCES

[1] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain, “Content-based image retrieval at the end of the early years,” *IEEE TPAMI*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee, “Finding the flow in web site search,” *Commun. ACM*, vol. 45, no. 9, pp. 42–49, 2002.

[3] Gary Marchionini, “Exploratory search: from finding to understanding.,” *Commun. ACM*, vol. 49, no. 4, pp. 41–46, 2006.

[4] Susanne Boll, Phillip Sandhaus, Ansgar Scherp, and Sabine Thieme, “Metaxa – context- and content-driven metadata enhancement for personal photo books,” in *International MultiMedia Modeling Conference 2007*, pp. 332–343.

[5] Michael S. Lew, “Next-generation Web searches for visual content,” *IEEE Computer*, vol. 33, no. 11, pp. 46–53, 2000.

[6] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathon Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker, “Query by image and video content: The QBIC system,” *IEEE Computer*, vol. 28, no. 9, pp. 23–32, Sept. 1995.

[7] Wolfgang Müller, Markus Zech, and Andreas Henrich, “VisualFlamenco: Faceted Browsing for Visual Features,” in *IEEE International Symposium on Multimedia*, Demo, Taichung, Taiwan, December 2007, pp. 71–72.

[8] H. Tamura, S. Mori, and Y. Yammawaki, “Textural features corresponding to visual perception,” *IEEE Transactions on Systems, Man, Cybernetics*, vol. 8, no. 6, pp. 460–473, June 1978.

[9] Matthew R. Boutell and Jiebo Luo, “Beyond pixels: Exploiting camera metadata for photo classification,” *Pattern Recognition*, vol. 38, no. 6, pp. 935–946, 2005.

[10] S. Battiato, G. Gallo, and S. Nicotra, “Perceptive visual texture classification and retrieval,” in *ICIAP ’03: Proceedings of the 12th International Conference on Image Analysis and Processing*, Washington, DC, USA, 2003, pp. 524–529, IEEE Computer Society.

[11] Markus Zech, “Facettierte Suche auf automatisch extrahierten Bildfeatures,” M.S. thesis, University of Bamberg, Bamberg, Bavaria, Germany, September 2007.