Retrieving Images by Appearance[*]

S. Ravela R. Manmatha
Multimedia Indexing and Retrieval Group
University of Massachusetts, Amherst
{ravela,manmatha}@cs.umass.edu

Abstract

A system to retrieve images using a description of the image intensity surface is presented. Gaussian derivative filters at several scales are applied to the image and low order 2D differential invariants are computed. The resulting multi-scale representation is indexed for rapid retrieval. Queries are designed by the users from an example image by selecting appropriate regions. The invariant vectors corresponding to these regions are matched with the database counterparts both in feature and coordinate space. This yields a match score per image. Images are sorted by the match score and displayed. Experiments conducted with over 1500 images of objects embedded in arbitrary backgrounds are described. It is observed that images similar in appearance and whose viewpoint is within small view variations of the query can be retrieved with an average precision[1] of 56%.

1 Introduction

The goal of image retrieval systems is to operate on collections of images and, in response to visual queries, extract relevant images. There are several issues that must be understood before image retrieval can be successful. Foremost among these is an understanding of what 'retrieval of relevant images' means. Relevance, for users of a retrieval system, is most likely associated with semantics. Encoding semantic information into a general image retrieval system entails solving such problems as feature extraction, segmentation and, object and context recognition. These are extremely hard problems that are as yet unsolved. However, in many situations attributes associated with an image, when used together with some level of user input, correlate well with the kind of semantics that are desirable. Consequently, recent work has focused directly on retrieval using attributes such as color, texture, shape, and combinations thereof.

In this paper, images are retrieved by a characterization of the visual appearance of objects. The focus is on retrieving 'similar' objects. For example, when a face is presented as a query it is expected that the system should not only retrieve the same person's face but rank other faces before it ranks, cars, trains or apes. Similarly if a car is a query(see Figure 1) then it is expected that cars be ranked before faces or trains(see Figure 2). Intuitively, an object's visual appearance in an image depends not only on its three-dimensional geometric shape, but also on its albedo, its surface texture, the view point from which it is imaged, among other factors. It is non-trivial to separate the different factors that constitute an object's visual appearance. However, we posit that the shape of an imaged object's intensity surface closely relates to its visual appearance. Here a local characterization of the intensity surface is constructed and images are retrieved using a measure of similarity for this representation. The experiments conducted in this paper verify the association that objects that appear to be visually similar can be retrieved by a characterization of the shape of the intensity surface.

Different representations of appearance have been used in object recognition [8]. Other representations have been used for specific types of retrieval such as face recognition. The approach taken here does not rely on image segmentation (manual or automatic) or binary feature extraction. Unlike some of the previously mentioned methods, no training is required. Since the representation is local, objects can be embedded in different backgrounds. Using an *example image and user interaction to construct queries*,

[*] This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation under grant number IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

[1] precision is the proportion of retrieved images that are relevant

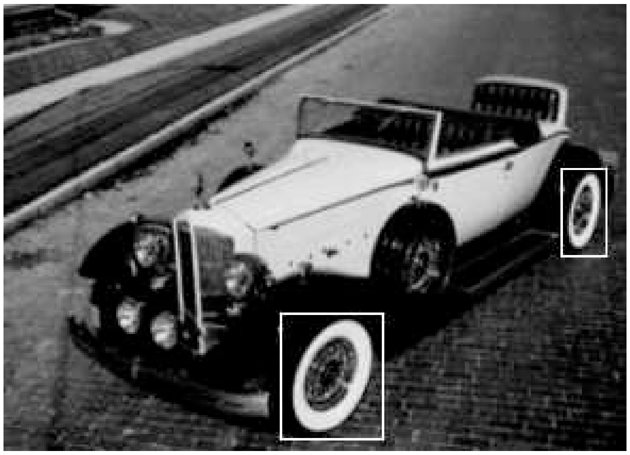

Figure 1: Allowing the user to construct queries by selecting the boxes shown

Synapse retrieves similar images within small view and size variation in the order of their *similarity in syntactic appearance to a query.*

The claim is that, up to a certain order, the *local appearance* of the intensity surface (around some point) can be represented as responses to a set of scale parameterized Gaussian derivative filters (see Section 3). The family of Gaussian filters are unique in their ability to describe the *scale-space* or *deep structure* [2, 4, 1] of a function and are well suited for representing appearance.

In this paper an indexable strategy for image retrieval is then developed using feature vectors constructed from combinations of the derivative filter outputs. These combinations yield a set of differential invariants [1] that are invariant to two-dimensional rigid transformations. Retrieval is achieved in two computational steps. During the off-line computation phase each image in the database is first filtered at sampled locations and then filter responses across the entire database are indexed(see Section 4). The run-time computation of the system begins with the user selecting an example image and marking a set of salient regions within the image. The responses corresponding to these regions are matched with those of the database and a measure of fitness per image in the database is computed in both feature space and coordinate space (see Section 4.2). Finally, images are displayed to the user in the order of fitness (or match score) to the query (see Section 6).

2 Related Work

A number of different techniques have been used to retrieve images using color, shape, texture and appearance. Due to lack of space, the reader is referred to [5] for a review of the literature.

Gaussian derivative representations have also been used in the context of recognition. Indexed differential invariants have recently been used [8] for object recognition. We also index using differential invariants but there are several differences compared with [8]. First, the invariants corresponding to the low two order derivatives are used as opposed to nine invariants as in [8]. This is because we observed that nine invariants at a single scale are much more discriminating and do not work as well when retrieving similar images. The low two orders perform better for this task. Second, their indexing algorithm depends on interest point detection and is, therefore, limited by the stability of the interest operator. We on the other hand sample the image. Third, the authors do not incorporate multiple scales into a single vector whereas here three different scales are chosen. In addition the index structure and spatial checking algorithms differ.

3 Characterization of Appearance

This section shows how appearance may be represented using a multi-scale feature vector constructed by filtering an image with a set of Gaussian derivative filters. The multi-scale feature vector are transformed so that the elements within this vector are invariant to 2D rigid transformations. This transformed feature vector is called the multi-scale invariant vector.

The local N-jet of $I(\mathbf{x})$ at scale σ and order N is defined as the set [3]:

$$J^N[I](\mathbf{x},\sigma) = \{I_{i_1\ldots i_n,\sigma} | n = 0\ldots N\} \qquad (1)$$

where

$$I_{i_1\ldots i_n,\sigma}(\mathbf{x}) = (I \star G_{i_1\ldots i_n})(\mathbf{x},\sigma)$$

the Gaussian derivatives are specified by

$$G_{i_1\ldots i_n} = \frac{\delta^n}{\delta_{i_1}\ldots\delta_{i_n}} G$$

, $i_k = x_1 \ldots x_D, k = 1\ldots n$. and G is the Gaussian.

The set $\lim_{N\to\infty} J^N[I](\mathbf{x},\sigma)$ specifies the Taylor expansion of I_σ up to derivatives of order N. Thus, for any order N, the local N-jet at scale σ contains all the information required to reconstruct I at the scale of observation σ up to order N. That is, up to any order the derivatives locally characterize the shape of the intensity surface, i.e. appearance, to that order. From the experiments shown in this paper it is also observed that this representation can be used to retrieve images that appear visually similar.

The choice of the Gaussian as the smooth test function, as opposed to others, is motivated by the fact that it is unique in describing the scale-space or deep structure of an arbitrary function. For a review of scale space, the reader is referred to [9, 2, 1, 4]. Scale-space has an important physical interpretation in that it models the change in appearance of an imaged object as it moves away from a camera. An argument is therefore made for a *multi-scale feature vector* which describes the intensity surface locally at several scales. A *multi-scale feature vector* at a point p in an image I is given by the vector:

$$\left\{ J^N \left[I\right] \left(\mathbf{p}, \sigma_1\right), J^N \left[I\right] \left(\mathbf{p}, \sigma_2\right) \ldots J^N \left[I\right] \left(\mathbf{p}, \sigma_k\right) \right\} \quad (2)$$

for some order N and a set of scales $\sigma_1 \ldots \sigma_k$. In practice the zeroth order terms are dropped to achieve invariance to constant intensity changes. Multi-scale vectors represent appearance more robustly than a single-scale vector. This can be viewed from several different perspectives. Since, multi-scale vectors are values computed at several different kernel sizes, therefore, they contain more information than fixed window operators. Equivalently, multi-scale vectors contain information at several different bandwidths and with the choice of a Gaussian accurately represent the intensity surface at different depths from the camera. From a practical standpoint this means that mis-matches due to an accidental similarity at a single scale can be reduced.

4 Indexable Retrieval Strategy

In earlier work [7, 6], the appropriateness of the derivative vector representation is evaluated using the well known correlation metric. A measure of similarity between two feature vectors can be obtained by correlating them or computing the distance between the vectors. The feature vector used is the local 2-jet (without the zeroth order term), computed at a fixed scale [7, 6] i.e., $\langle I_x, I_y, I_{xx}, I_{xy}, I_{yy} \rangle_\sigma$, computed at scale σ, Using this representation in conjunction with correlation, we verify that [7, 6], at any scale a reasonable retrieval of visually similar images is possible. Typically, in-plane rotations of up to 20^o and out-of-plane rotations of up to 30^o can be tolerated. Second, a range of size variations, determined a priori, can be handled by searching across the scale parameter of the Gaussian. In particular similar objects within size changes of $\frac{1}{4} \ldots 4$ could be retrieved [7, 6].

There are several limitations to the correlation approach. First, correlation is computationally expensive. Second, using the derivatives directly in a feature vector restricts tolerance to rotations. Third, the use of vectors at a fixed scale can lead to mismatches due to accidental similarity solely as a result of the fixed scale of observation. These issues are partially addressed below. First, the derivative feature vector is transformed so that it is invariant to 2D rigid transformations. Second, correlation is replaced with an indexable strategy that results in an order of magnitude of speed increase and third vectors at multiple scales are used simultaneously to improve robustness. In this paper the multi-scale vector is computed at three different scales placed half an octave apart.

4.1 Indexing Multi-Scale Invariant Vectors

Given the derivatives of an image I, *irreducible differential invariants* (invariant under the group of displacements) can be computed in a systematic manner [1]. The value of these entities is independent of the choice of coordinate frame (up to rotations). The irreducible set of invariants up to order two of an image I are:

$$\begin{array}{lll} d_0 & = I & \text{Intensity} \\ d_1 & = I_x^2 + I_y^2 & \text{Magnitude} \\ d_2 & = I_{xx} + I_{yy} & \text{Laplacian} \\ d_3 & = I_{xx} I_x I_x + 2 I_{xy} I_x I_y + I_{yy} I_y I_y & \\ d_4 & = I_{xx}^2 + 2 I_{xy}^2 + I_{yy}^2 & \end{array}$$

Here, the vector, $\Delta_\sigma = \langle d_1, \ldots d_4 \rangle_\sigma$ is computed at three different scales. The element d_0 is not used since it is sensitive to gray-level shifts. The resulting multi-scale invariant vector has at most twelve elements. Computationally, each image in the database is filtered with the first five partial derivatives of the Gaussian (i.e. to order 2) at three different scales at uniformly sampled locations. Then the multi-scale invariant vector $D = \langle \Delta_{\sigma_1}, \Delta_{\sigma_2}, \Delta_{\sigma_3} \rangle$ is computed at those locations.

A location across the entire database can be identified by the *generalized coordinates*, defined as, $c = (i, x, y)$ where i is the image number and (x, y) a coordinate within this image. The computation described above generates an association between generalized coordinates and invariant vectors. This association can be viewed as a table $M : (i, x, y, D)$ with $3+k$ columns(k is the number of fields in an invariant vector) and number of rows, R, equal to the total number of locations (across all images) where invariant vectors are computed.

To index the database by fields of the invariant vector, the table M is split into k smaller tables $M_1^{'} \ldots M_k^{'}$, one for each of the k fields of the invariant vector. Each of the smaller tables $M_p^{'}, p = 1 \cdots k$ contains the four columns $(D(p), i, x, y)$. At this stage any given row across all the smaller tables contains the same generalized coordinate entries as in M. Then, each $M_p^{\prime}$ is sorted and a binary tree is used to represent the sorted keys. As a result, the entire database is indexed. A given invariant value can, therefore, be located in $\log(R)$ time (R = number of rows).

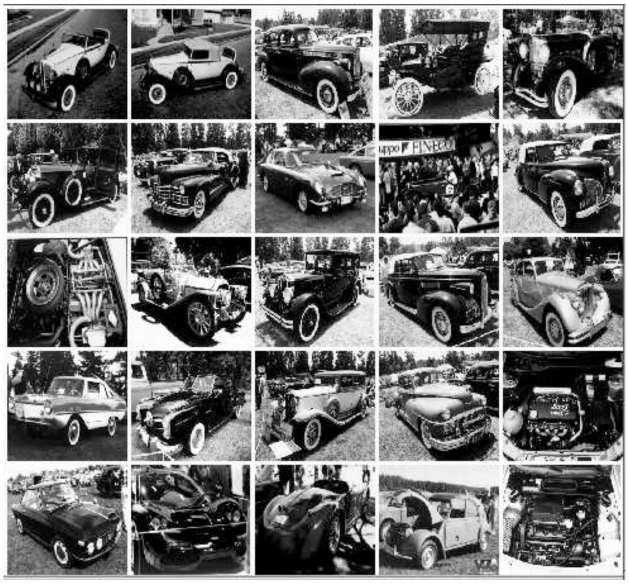

Figure 2: The results of the car query

4.2 Matching Invariant Vectors

Run-time computation begins with the user marking selected regions in an example image. At sampled locations within these regions, invariant vectors are computed and submitted as a query. The search for matching images is performed in two stages. In the first stage each query invariant is supplied to the 'find-by-value' algorithm and a list of matching generalized coordinates is obtained. In the second stage a spatial check is performed on a per image basis, in order to verify that the matched locations in an image are in spatial coherence with the corresponding query points. In this section the 'find-by-value' and spatial checking components are discussed.

The multi-scale invariant vectors at sampled locations within regions of a query image can be treated as a list. The n^{th} element in this list contains the information $Q_n = (D_n, x_n, y_n)$, that is, the invariant vector and the corresponding coordinates. In order to find-by-invariant-value, for any query entry Q_n, the database must contain vectors that are within a threshold $t = (t_1 \ldots t_k) > 0$. The coordinates of these matching vectors are then returned. This can be represented as follows. Let p be any invariant vector stored in the database. Then p matches the query invariant entry D_n only if $D_n - t < p < D_n + t$. This can be rewritten as

$$\&_{j=1}^{k} \left[D_n\left(j\right) - t\left(j\right) < p\left(i\right) < D_n\left(j\right) - t\left(j\right) \right]$$

where & is the logical *and* operator and k is the number of fields in the invariant vector. To implement the comparison operation two searches can be performed on each field. The first is a search for the lower bound, that is the smallest entry larger than $D_n(j) - t(j)$ and then a search for the upper-bound i.e. the largest entry smaller than $D_n(j)+t(j)$. The block of entries between these two bounds are those that match the field j. In the inverted file the generalized coordinates are stored along with the individual field values and the block of matching generalized coordinates are copied from disk. To implement the logical-and part, an intersection of all the returned block of generalized coordinates is performed. The generalized coordinates common to all the k fields are the ones that match query entry Q_n. The find by value routine is executed for each Q_n and as a result each query entry is associated with a list of generalized coordinates that it matches.

The association between a query entry Q_n and the list of f generalized coordinates that match it by value can be written as

$$\begin{aligned} A_n &= \left\langle x_n, y_n, c_{n_1}, c_{n_2} \ldots c_{n_f} \right\rangle \\ &= \left\langle x_n, y_n, \left(i_{n_1}, x_{n_1}, y_{n_1}\right) \ldots \left(i_{n_f}, x_{n_f}, y_{n_f}\right) \right\rangle \end{aligned}$$

Here x_n, y_n are the coordinates of the query entry Q_n and $c_{n_1} \ldots c_{n_f}$ are the f matching generalized coordinates. The notation c_{n_f} implies that the generalized coordinate c matches n and is the f^{th} entry in the list. Once these associations are available, a spatial fit on a per image basis can be performed. In order to describe the fitness measure, two definitions are needed. First, define the distance between the coordinates of two query entries m and n as $\delta_{m,n}$. Second, define the distance between any two generalized coordinates c_{m_j} and c_{n_j} that are associated with two query entries m, n as $\delta_{c_{m_j}, c_{n_k}}$

Any image u that contains two points (locations) which match some query entry m and n respectively are coherent with the query entries m and n only if the distance between these two points is the same as the distance between the query entries that they match. Using this as a basis, a binary fitness measure can be defined as

$$\mathcal{F}_{m,n}\left(u\right) = \begin{cases} 1 & \text{if } \exists j \exists k \mid \left| \delta_{m,n} - \delta_{c_{m_j}, c_{n_k}} \right| \leq T \\ & i_{m_j} = i_{n_k} = u, m \neq n \\ & \\ 0 & \text{otherwise} \end{cases}$$

That is, if the distance between two matched points in an image is close to the distance between the query points that they are associated with, then these points are spatially coherent (with the query). Using this fitness measure a match score for each image can be determined. This match score is simply the maximum number of points that together are spatially coherent (with the query). Define the match score by:

$$score\,(u) \equiv \overset{max}{m}\; S_m\,(u) \tag{3}$$

where, $S_m\,(u) = \sum_{n=1}^{f} \mathcal{F}\,(u)_{m,n}$. The computation of $score(u)$ is at worst quadratic in the total number of query points. The array of scores for all images is sorted and the images are displayed in the order of their score. T used in $\mathcal{F}$ is a threshold and is typically 25% of $\delta_{m,n}$. Note that this measure not only will admit points that are rotated but will also tolerate other deformations as permitted by the threshold. The value of the threshold is selected to reflect the rationale that similar images will have similar responses but not necessarily under a rigid deformation of the query points.

5 Query Construction

The ability for the user to construct queries by selecting regions is an important distinction between the approach presented here and elsewhere. Users can be expected to employ their considerable semantic knowledge about the world to construct a query. Such semantic information is difficult to incorporate in a system. An example of query construction is shown in Figure 1, where the user has decided to find cars similar to the one shown and decides that the most salient part are 'wheels'[2]. It is clear that providing such interaction removes the necessity for automatic determination of saliency. In the car example, the user provides the context to search the database by marking the wheels and retrieved images mostly contain wheels. The association of wheels to cars is not known to the system, rather it is one that the user decides is meaningful. Several other approaches in the literature take the entire feature set or some global representation over the entire image (see [5] for examples). While this may be reasonable for certain types of retrieval, it cannot necessarily be used for general purpose retrieval. Letting the user design queries eliminates the need for detecting the salient portions of an object, and the retrieval can be customized so as to remove unwanted portions of the image. Based on the feedback provided by the results of a query, the user can quickly adapt and modify the query to improve performance.

6 Experiments

The database used in this paper has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different background(s) and a small number of other miscellaneous objects such as houses. 1561 images were obtained from the Internet and the Corel photo-cd collection to construct this database. These photographs were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry. Also, the objects of interest are embedded in natural scenes such as car shows, railroad stations, country sides and so on.

A measure of the performance of the retrieval engine can be obtained by examining the recall/precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved and precision is the proportion of retrieved material that is relevant.

Consider as an example the query described in Figure 1. Here the user wishes to retrieve 'white wheel cars' similar to the ones outlined and submits the query. The top 25 results ranked in text book fashion are shown in Figure 2. Note that although there are several valid matches as far as the algorithm is concerned (for example image 11 a tire), they are not considered valid retrievals as stated by the user and are not used in measuring the recall/precision. This is inherently a conservative estimate of the performance of the system. The average precision (over recall intervals of 10[3]) is 57%.

One of the important parameters in constructing indices is the sample rate. The performance of the system was evaluated under sample rates of three pixels and five pixels respectively. The case where every pixel is used could not be implemented due to prohibitive disk requirements and lack of resources to do so. It is observed that there is a dramatic improvement in scores, and in most cases a substantial improvement in average precision.

Six other queries that were also submitted are depicted in Table 1. The recall/precision table over all seven queries is in Table 2. The third column of table shows the average precision for each query with a database sampling of 5 pixels and the fourth column shows with 3 pixels. The average precision and precision at recall intervals of 10, over all the queries for both samplings is shown in Table 2. This compares well with text retrieval where some of the best systems have an average precision of 50%[4]. The average precision over the same seven queries with both three and five pixel sampling is 56.2% for the five pixel case and 61.7% in the three pixel case. However, while the increase in sampling improves the precision it results in an increased storage requirement.

Unsatisfactory retrieval occurs for several reasons.

[2] see Figure 2 for the results

[3] The value $n(=10)$ is simply the retrievals up to recall n.

[4] Based on personal communication with Bruce Croft

Table 1: Queries submitted to the system and expected retrieval

Given(User Input)	Find	Precision (5)	Precision (3)
Face	All Faces	74.7%	61.5%
Face see	Same Person's Face	61.7%	75.5%
Monkey's coat	Dark Textured Apes	57.5%	57%
Both wheels, see Figure 1	White Wheeled Cars, see Figure 2	57.0%	63.7%
Coca Logo	All Coca Cola Logos	49.3%	74.9%
Wheel	White Wheeled Cars	48.6%	54.4%
Patas Monkey Face	All Visible Patas Monkey Faces	44.5%	47.1%

Table 2: Precision at standard recall points for seven Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision(5) %	100	95.8	90.3	80.1	67.3	48.9	39.9	34.2	31.1	18.2	12.4
Precision(3) %	100	100	90.4	80.9	75.7	55.9	49.4	47.6	40.6	20.7	17.1

average(5)	56.2%
average(3)	61.7%

First, it may possible that the query is poorly designed. In this case the user can design a new query and re-submit or refine the query by using a result of a previous search as a query. A second source of error is in matching generalized coordinates by value. The choice of scales in the experiments carried out in this case are $\frac{3}{\sqrt{2}}, 3, 3\sqrt{2}$. It is possible that locally the intensity surface may have a very close value, so as to lie within the chosen threshold and thus introduce an incorrect point. By adding more scales or derivatives such errors can be reduced, but at the cost of increased discrimination and decreased generalization. Many of these 'false matches' are eliminated in the spatial checking phase. Errors can also occur in the spatial checking phase because it admits much more than a rotational transformation of points with respect to the query configuration. Overall the performance to date has been very satisfactory and we believe that by experimentally evaluating each phase the system can be further improved.

The time it takes to retrieve images is dependent linearly on the number of query points. On a Pentium Pro-200 Mhz Linux machine, typical queries execute in between one and six minutes.

7 Conclusions

Within small view variations, images that are similar to a query are retrieved. These images are also observed to be visually similar and we posit that this method has good potential for image retrieval.

While retrieval of objects across different sizes has been implemented elsewhere [7] using correlation, in this paper, the multi-scale invariant vector was used only to robustly characterize appearance. The next immediate step is to explicitly incorporate matching across size variations akin to the correlation approach.

A second important question is, what types of invariants should constitute a feature vector ? This is an open research issue. Finally, although the current system is some what slow, it is yet a remarkable improvement over our previous work. We believe that by examining the spatial checking and sampling components further increases in speed are possible.

References

[1] Ludvicus Maria Jozef Florack. *The Syntactic Structure of Scalar Images.* PhD thesis, University of Utrecht, 1993.

[2] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[3] J. J Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[4] Tony Lindeberg. *Scale-Space Theroy in Computer Vision.* Kluwer Academic Publishers, 1994.

[5] S. Ravela and R. Manmatha. Image retrieval by appearance. In *Proc. of SIGIR*, Philadelphia, July 1997.

[6] S. Ravela and R. Manmatha. Image retrieval by similarity of visual appearance. In *Workshop on Content based Access of Image and Video Databases.* IEEE, June 1997.

[7] S. Ravela, R. Manmatha, and E. M. Riseman. Image retrieval using scale-space matching. In *Computer Vision - ECCV '96*, pages 273–282, Cambridge, U.K., April 1996. 4th European Conf. Computer Vision, Springer.

[8] Cordelia Schmid and Roger Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition.* IEEE, June 1996.

[9] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.