Adaptive Wavelet Lifting for Image Retrieval

Patrick Oonincx and Paul de Zeeuw

Center for Mathematics & Computer Science (CWI),
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

We build a feature vector that can be used for content-based image retrieval of grayscale images of objects against a background of texture. The feature vector is based on moment invariants of detail coefficients produced by the lifting scheme. The prediction filters in this scheme are chosen adaptively: low order (small stencils) near edges and high order elsewhere. The aim is to retrieve similar images of an object irrespective of translation, rotation, reflection or resizing of the object, light conditions and the background texture. We present preliminary results.

Keywords: Image retrieval, lifting scheme, moment invariants, adaptive scheme, affine transformations

1. INTRODUCTION

Content-based image retrieval (CBIR) is a widely used term to indicate the process of retrieving desired images from a large collection on the basis of features. The extraction process should be automatic (i.e. no human interference) and the features used for retrieval can be either primitive (colour, shape, texture) or semantic (involving identity and meaning). In this paper we confine ourselves to grayscale images of objects against a background of texture. This class of images occurs in various databases created for the combat of crime: stolen objects,[1] tyre tracks and shoe sole impressions.[2] Given an image of an object (a so-called query) we want to identify all images in the database which contain the same object irrespective of translation, rotation or resizing of the object, light conditions and the background texture. A classic approach to the problem of recognition of similar images is by the use of Hu's moment invariants.[3] In Do et al.[4] the wavelet transform modulus maxima is employed. To measure the (dis)similarity between images, moments of the set of maxima points are determined (per scale) and subsequently Hu's invariants are computed. Thus, each image is indexed by a vector in the wavelet maxima moment space. By its construction, this feature vector is predominantly a measure for shapes.

We propose to bring in adaptivity by using different wavelet filters for smooth and unsmooth parts of the image. The filters are used in the context of the (redundant) lifting scheme.[5] The degree of "smoothness" is determined by measuring the *relative local variance* (RLV). Near edges low order prediction filters are activated which lead to large lifting detail coefficients along thin curves. At texture-like backgrounds high order prediction filters are activated which lead to negligible detail coefficients. Moments and subsequently moment invariants are computed with respect to these wavelet detail coefficients. With the computation of the detail coefficients a certain preprocessing is required to make the method robust for shifts over a non-integer number of gridpoints. Further we introduce *the homogeneity condition* which means that we demand a homogeneous change in the elements of a feature vector if the image seen as a density distribution is multiplied by a scalar. We present preliminary results that demonstrate that the above adaptivity leads to improvement of retrieval rates and discriminative power.

2. THE LIFTING SCHEME

The lifting scheme as introduced by Sweldens[5] in 1997 is a method for constructing wavelet transforms that are not necessarily based on dilates and translates of one function. The construction does not rely on the Fourier transform at all which makes it also suitable for functions on irregular grids. The transform allows for a fully in-place calculation, which means that no auxiliary memory is needed for the computations.

The idea of lifting is based on splitting a given set of data into two subsets. In the one-dimensional case this can mean that starting with a signal $x \in l^2(\mathbb{Z})$ the even and odd samples are collected into two new signals, i.e., $x = x_e + x_o$, where $x_e(n) = x(2n)$ and $x_o(n) = x(2n+1)$, for all $n \in \mathbb{Z}$. The next step of the lifting scheme is to predict the value of $x_e(n)$ given the sequence x_o. This prediction uses a prediction operator $\mathcal{P}$ acting on x_o. The

Correspondence: E-mail: {Patrick.Oonincx,Paul.de.Zeeuw}@cwi.nl

Wavelets: Applications in Signal and Image Processing IX, Andrew F. Laine, Michael A. Unser, Akram Aldroubi, Editors, Proceedings of SPIE Vol. 4478 (2001) © 2001 SPIE · 0277-786X/01/$15.00

predicted value $(\mathcal{P}x_o)(n)$ is subtracted from $x_e(n)$ yielding a 'detail' signal d. An update of the odd samples x_o is needed to avoid aliasing problems. This update is performed by adding $\mathcal{U}d$ to the sequence x_o, with $\mathcal{U}$ the update operator. The lifting procedure can also be seen as a 2-band filter bank. This idea has been depicted in Figure 1: T is the splitting operator (the so-called lazy wavelet transform), generally T can be any bijective transform from $l^2(\mathbb{Z})$ to $l^2(\mathbb{Z}, \mathbb{R}^2)$. The inverse lifting scheme can immediately be found by undoing the prediction and update operators.

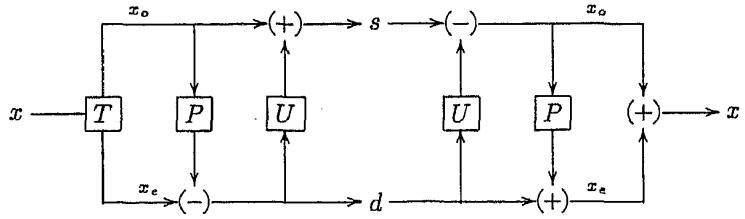


Figure 1. The lifting scheme: splitting, predicting, updating.

In Figure 1 this boils down to simply changing each + into a − and vice versa.

The lifting scheme can also be used for higher dimensional signals.[6] In two space dimensions we can split our dataset into two quincunx grids, see Figure 2. This division is also called "checkerboard" or "red-black" division. The pixels on the red spots (○) are used to predict the samples on the black spots (●), while updating of the red

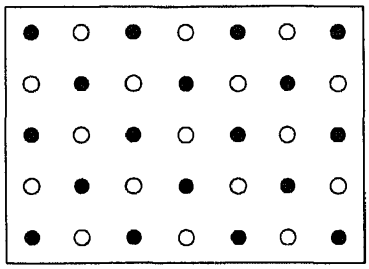


Figure 2. Two quincunx grids constituting a rectangular grid.

spots is performed by using the detailed data on the black spots. An example of a lifting transform with second order prediction and update filters is given by

$$\begin{aligned}(\mathcal{P}x)(i,j) &= [x(i-1,j)+x(i,j-1)+x(i+1,j)+x(i,j+1)]/4,\ i \bmod 2 \neq j \bmod 2,\\ (\mathcal{U}x)(i,j) &= [x(i-1,j)+x(i,j-1)+x(i+1,j)+x(i,j+1)]/8,\ i \bmod 2 = j \bmod 2.\end{aligned}$$

The algorithm using the quincunx lattice is also known as the red-black wavelet transform by Uytterhoeven and Bultheel.[7] In general $\mathcal{P}$ can be written as

$$(\mathcal{P}x)(i,j) = \sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m)\, x(i+n, j+m),\ i \bmod 2 \neq j \bmod 2, \tag{1}$$

with $S_{\tilde{N}}$ a subset of $\{(n,m) \in \mathbb{Z}^2 \mid (n+m) \bmod 2 = 1\}$ and $a_{\tilde{N}}(s)$, $s \in S_{\tilde{N}}$, a set of coefficients in $\mathbb{R}$. In this case a general formula for $\mathcal{U}$ reads

$$(\mathcal{U}x)(i,j) = \sum_{(n,m)\in S_N} a_N(n,m)\, x(i+n, j+m)/2,\ i \bmod 2 = j \bmod 2, \tag{2}$$

with S_N depending on the number of required primal vanishing moments N. For several elements in S_N the coefficients $a_N(s)$ attain the same values. Therefore we take these elements together in subsets of S_N, i.e.,

$$\begin{aligned}V_1 &= \{(\pm1,0),(0,\pm1)\},\ V_2 = \{(\pm1,\pm2),(\pm2,\pm1)\},\ V_3 = \{(\pm3,0),(0,\pm3)\},\\ V_4 &= \{(\pm2,\pm3),(\pm3,\pm2)\},\ V_5 = \{(\pm1,\pm4),(\pm4,\pm1)\},\ V_6 = \{(\pm5,0),(0,\pm5)\}, \qquad (3)\\ V_7 &= \{(\pm3,\pm4),(\pm4,\pm3)\} \qquad (4)\end{aligned}$$

order N	V_1	V_2	V_3	V_4	V_5	V_6	V_7
2	$1/4$	0	0	0	0	0	0
4	$10/32$	$-1/32$	0	0	0	0	0
6	$87/2^8$	$-27/2^9$	2^{-8}	$3/2^9$	0	0	0
8	$5825/2^{14}$	$-2235/2^{15}$	$625/2^{16}$	$425/2^{15}$	$-75/2^{16}$	$9/2^{16}$	$-5/2^{12}$

Table 1. Quincunx Neville filter coefficients

Table 1 indicates the values of all $a_N(s)$, $s \in V_k$ for different values of N (2 through 8) when using quincunx Neville filters,[6] which are the filters we use in our approach. We observe that $S_8 = V_1 + \cdots + V_7$ and so a 44 taps filter is used as prediction/update if the required filter order is 8. For an illustration of the Neville filter of order 4 see Figure 3. Here the numbers 1, 2 correspond to the values of the filtercoefficients as given in V_1 and V_2 respectively at that position.

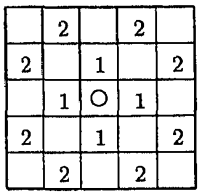


Figure 3. Neville filter of order 4

3. AFFINE INVARIANCES AND LIFTING COEFFICIENTS

Traditional wavelet analysis and the lifting scheme yield detail and approximation coefficients that are localised in scale and space. Unfortunately, they are not translation invariant, which causes the coefficients to attain values in the same range of the original values (after translation), but still different. For the classical wavelet transform a means to translation invariance is given by the *redundant* wavelet transform,[8] which is a non-decimated wavelet transform (at all scales). As a consequence the number of data in all subbands is the same as the number of input data of the transform. Not only more memory is used by the redundant transform, also the computational complexity of the fast transform increases. For the non-decimated transform the computional complexity is $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N)$ for the fast wavelet transform.

The redundant wavelet transform has its analogue in the redundant lifting scheme. In one dimension this works out as follows. Instead of partitioning a signal $x \in l^2(\mathbb{Z})$ into x_e and x_o we copy x to both x_e and x_o. The next step of the lifting scheme is to predict x_e by

$$x_e = x_e - \mathcal{P}^{(j)} x_o. \tag{5}$$

The prediction filter $\mathcal{P}^{(j)}$ is the same filter as used for the non-redundant case, but now it depends on the resolution level, since at each level zero padding is applied to $\mathcal{P}$. This holds also for the update filters $\mathcal{U}^{(j)}$. Hence, the update step reads

$$x_o = x_o + \mathcal{U}^{(j)} x_e. \tag{6}$$

In two space dimensions we use the red-black division instead of the even-odd division. We use zero padding for the filters at each resolution level. Whether the said redundant transform is invariant under reflections and rotations as well, depends on the filters (wavelets) themselves. Symmetry of the filters is a necessary condition to guarantee certain rotation and reflection invariances. This is a condition that is not satisfied by many well-known wavelet filters. Also a necessary condition for these invariances is that the values of the image on the grid points are not affected by a rotation or reflection. In practice, this means that only reflections in the horizontal, the vertical and the diagonal axis and rotations over multiples of $\pi/2$ can lead to the same lifting coefficients.

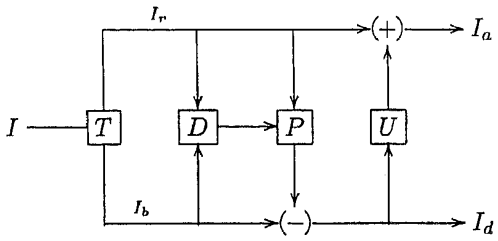


Figure 4. Generating coefficients via adaptive lifting

We observe that the Neville filters we use in our approach have the required symmetry properties, see (3). So with these filters the redundant scheme does not only guarantee translation invariance, but also invariance under rotations over multiples of $\pi/2$ or reflections in the horizontal, vertical and diagonal axis is assured too. Therefore we choose in our approach to use a redundant lifting scheme. In our application we also allow for rotations that are not multiples of $\pi/2$. Similitude invariance is discussed in Section 5.2.

4. ADAPTIVE LIFTING

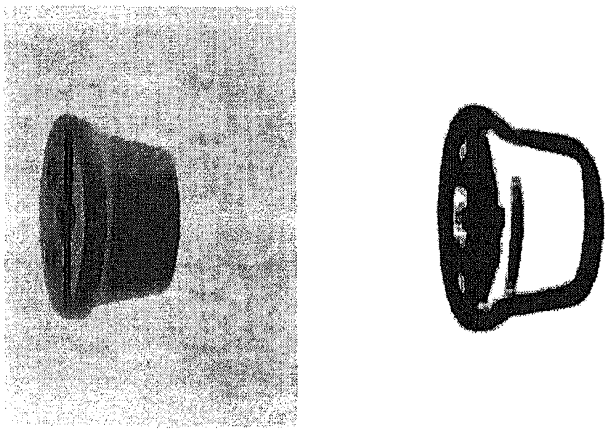
When using the lifting scheme or a classical wavelet approach, the prediction/update filters or wavelet/scaling functions are chosen in a fixed fashion. Generally they can be chosen in such way that a signal is approximated with very high accuracy using only a limited number of coefficients. Discontinuities mostly give rise to large detail coefficients which is unfortunate for applications like compression. For our purpose large detail coefficients near edges in an images are desirable, since they can be identified with the shape of objects we want to detect. However, they are undesirable if such large coefficients are related to the background of the image. This situation occurs if a small filter is used on a texture-like background that contains irregularities locally. In this case a large smoothing filter gives rise to small coefficients for the background. These considerations lead to the idea of using different prediction filters for different parts of the signal. The signal itself should indicate (for example by means of local behavior information) whether a high or low order prediction filter should be used. Such an approach is commonly referred to as an adaptive approach. Many of these adaptive approaches have been described already thoroughly in the literature.[9–13] In this paper we follow the approach proposed by Baraniuk et al.[14] called the space-adaptive approach. This approach follows the scheme as shown in Figure 4. After splitting all pixels of a given image I into two complementary groups I_r and I_b (red/black), the pixels in I_r are used to predict the values in I_b. This is done by means of a prediction filter acting on I_r, i.e., $\mathcal{P}(I_r)$. In the adaptive lifting case this prediction filter depends on local information of the image pixels I_r. Choices for $\mathcal{P}$ may vary from high to low order filters, depending on the regularity of the image locally. For the update operator, we choose the update filter that corresponds to the prediction filter with lowest order from all possible to be chosen $\mathcal{P}$. Baraniuk et al.[14] choose to start the lifting scheme with an update operator $\mathcal{U}$ followed by an adaptively chosen prediction operator. The reason for interchanging the prediction and update operator is that this is convenient for compression. In particular they took for the filters of $\mathcal{U}$ and $\mathcal{P}$ the $(1, N)$ branch of the Cohen-Daubechies-Feauveau (CDF) filter family.[15] The order of the prediction filter N was chosen to be 1, 3, 5 or 7, depending on the local behavior of the signal.

Relative local variance In our approach we use a second order Neville filter for the update step and an Nth order Neville filter for the prediction step, where $N \in \{2, 4, 6, 8\}$. We cause the prediction filter to depend on the *relative local variance* of an image. This relative local variance (RLV) of an image I is given by

$$\mathrm{rlv}[I](i,j) = \sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} (I(k,l) - \overline{\mu_{i,j}})^2 / \mathrm{var}\,(I), \tag{7}$$

with

$$\overline{\mu_{i,j}} = \sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} I(k,l)/(2T+1)^2. \tag{8}$$



a) original image b) decision map (RLV)

Figure 5. An object on a wooden background and its rel. local variance (decision map): white=8th order, black=2nd order.

For the window size we take $T = 5$, since with this choice all $I(k,l)$ that are used for the prediction of $I(i,j)$ contribute to the RLV for (i,j), even for the 8th order Neville filter. When the RLV is used at higher resolution levels we first have to downsample the image I appropriately.

The first time the prediction filter is applied (to the upper left pixel) we use the 8th order Neville filter on the quincunx lattice as given in Table 1. For all other subsequent pixels (i,j) to be predicted, we first compute $\mathrm{rlv}[I](i,j)$. Then quantizing the values of the RLV yields a decisionmap indicating which prediction filter should be used at which positions. Values above the highest quantizing level induce a 2nd order Neville filter, while values below the lowest quantizing levels induce an 8th order Neville filter. For the quantizing levels we take multiples of the mean of the RLV. Test results have shown that $[\mu(\mathrm{rlv})\ \ 1.5\,\mu(\mathrm{rlv})\ \ 2\,\mu(\mathrm{rlv})]$ are quantizing levels that yield a good performance in our application. In Figure 5 we have depicted an image (left) and its decision map based on the RLV (right).

5. MOMENT INVARIANTS

5.1. Introduction and recapitulation

To construct a feature vector from the obtained wavelet coefficients at several scales we use invariants based on moments of the coefficients up to third order. Traditionally, these features have been widely used in pattern recognition applications to recognize the geometrical shapes of different objects.[3]

We regard an image as a density distribution function $f \in S(\mathbb{R}^2)$, the Schwartz class. In order to obtain translation invariant statistics of such f we use central moments of f for our features. The $(p+q)th$ order central moment $\mu_{pq}(f)$ of f is given by

$$\mu_{pq}(f) = \int_{\mathbb{R}}\int_{\mathbb{R}} (x - x_c)^p (y - y_c)^q f(x,y)\, d(x - x_c)\, d(y - y_c), \tag{9}$$

with the center of mass

$$x_c = \frac{\int_{\mathbb{R}}\int_{\mathbb{R}} x f(x,y)\, dx\, dy}{\int_{\mathbb{R}}\int_{\mathbb{R}} f(x,y)\, dx\, dy} \quad \text{and} \quad y_c = \frac{\int_{\mathbb{R}}\int_{\mathbb{R}} y f(x,y)\, dx\, dy}{\int_{\mathbb{R}}\int_{\mathbb{R}} f(x,y)\, dx\, dy}. \tag{10}$$

Computing the centers of mass x'_c and y'_c of $g(x,y) = f(x-a, y-b)$ yields $x'_c = x_c - a$, $y'_c = y_c - b$. Combining this with (9) shows that $\mu_{pq}(f) = \mu_{pq}(g)$, i.e., the central moments are translation invariant.

We also require that the features should be invariant under rotations, reflections and scalar multiplications. Hu showed in his paper[3] that from the introduced central moments μ_{pq} expressions can be derived that are invariant under both rotations and reflections. These invariants are

$$\begin{aligned}
&I_1 = \mu_{20} + \mu_{02}, \quad I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \\
&I_3 = p_1^2 + p_2^2, \quad I_4 = p_3^2 + p_4^2, \quad I_5 = 2\left(p_1 p_3^3 - 3p_1 p_3 p_4^2 + 3p_1 p_3^2 p_4 - p_1 p_4^3\right), \\
&I_6 = 2p_5(p_3^2 - p_4^2) + 8\mu_{11} p_3 p_4,
\end{aligned}$$

with

$$p_1 = \mu_{30} - 3\mu_{12},\ p_2 = \mu_{03} - 3\mu_{21},\ p_3 = \mu_{30} + \mu_{12},\ p_4 = \mu_{03} + \mu_{21}.$$

To these 6 invariants we can add a seventh one, which is only invariant under rotations and changes sign under reflections. It is given by

$$I_7 = 2\left(p_1 p_4^3 + 3p_2 p_3 p_4^2 - 3p_1 p_3^2 p_4 - p_2 p_3^3\right).$$

Since we want to include reflections as well in our set of invariant transformations we will use $|I_7|$ instead of I_7 in our approach. From now on, we will identify $|I_7|$ with I_7. We observe that all possible linear combinations of these invariants are also invariant under proper orthogonal transformations and translations. Therefore we can call these seven invariants also invariant generators.

5.2. Normalization

Here we increase the number of circumstances under which images should be considered invariant. Firstly, if an image consists of a crisp object against a neutral background, a change in the dimensions of the object should still lead to the conclusion that the object remains the same. Invariance under this type of scaling, the similitude invariance, can be obtained by normalizing the moments μ_{pq}. Secondly, differences in luminosity also affect images of the same object. We will allow for a moderate variance with varying luminosity in the construction of a feature vector, see Section 5.2.2.

5.2.1. Similitude invariance

Uniform dilations (by a scalar $\alpha > 0$) of the whole image or objects in an image against a neutral background will result in new central moments given by[3]

$$\mu'_{pq} = \alpha^{p+q+2}\mu_{pq}. \tag{11}$$

It follows in particular that $\mu'_{00} = \alpha^2 \mu_{00}$, and also $\mu'_{20} + \mu'_{02} = \alpha^4(\mu_{20} + \mu_{02})$. Combining this result with (11) yields

$$\frac{\mu'_{pq}}{(\mu'_{00})^{\frac{p+q+2}{2}}} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q+2}{2}}} \quad \text{and} \quad \frac{\mu'_{pq}}{(\mu'_{20}+\mu'_{02})^{\frac{p+q+2}{4}}} = \frac{\mu_{pq}}{(\mu_{20}+\mu_{02})^{\frac{p+q+2}{4}}}$$

respectively. As we recall that both μ_{00} and $\mu_{20} + \mu_{02}$ are invariants w.r.t. rotation and reflection this shows how to normalize the moments to achieve invariance under dilation. The first choice leads to the following new set of invariant generators

$$I'_1 = I_1/\mu_{00}^2,\ I'_2 = I_2/\mu_{00}^4,\ I'_3 = I_3/\mu_{00}^5,\ I'_4 = I_4/\mu_{00}^5,\ I'_5 = I_5/\mu_{00}^{10},\ I'_6 = I_6/\mu_{00}^7,\ I'_7 = I_7/\mu_{00}^{10}. \tag{12}$$

The second choice leads to a different but similar result. It may be more suitable (as a starting point) in case the density distribution is given by wavelet detail coefficients (Section 5.5).

5.2.2. A feature vector of invariants

We introduce a feature vector, derived from the invariants discussed in the previous section. Distances between feature vectors, measured by the Euclidean norm, are supposed to indicate the difference between, or resemblance of, images. Obviously, the specific choice of the separate elements, e.g. the weights assigned to these (or alternatively employed within a norm), may lead to highly different results. If, in a first and naive approach we compose the following feature vector $I \in \mathbb{R}^7$:

$$I \equiv (I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5 \quad I_6 \quad I_7)^T \tag{13}$$

with the I_k as defined in Section 5.1, then, indeed, hereby obtained results turn out useless as the various elements appear to operate in different orders of magnitude. We remedy the arbitrariness of the definition of the vector by introducing an additional condition: *the homogeneity condition*, to be described and explained below.

Consider the simple model that for a change in luminosity of an object the distribution function f is mapped onto a different f' by means of an affine transformation. For grayscale images this means that the grayvalue of all pixels is multiplied by a scalar $\lambda > 0$ and shifted along a distance b (the offset). A scalar multiplication of the distribution function f does not affect the center of mass (10). From (9) it follows directly that

$$\mu_{pq}(\lambda f) = \lambda \mu_{pq}(f), \text{ for all } \lambda \neq 0.$$

The homogeneity condition means that we demand a homogeneous change in the elements of a feature vector if the density distribution f is multiplied by the said scalar. We observe that neither the vector I (13) nor the vector I' derived from (12), satisfies the homogeneity condition, as multiplication of f by $\lambda \neq 0$ leads to the transforms:

$$I \longrightarrow (\lambda I_1 \quad \lambda^2 I_2 \quad \lambda^2 I_3 \quad \lambda^2 I_4 \quad \lambda^4 I_5 \quad \lambda^3 I_6 \quad \lambda^4 I_7 \quad)^T$$

and

$$I' \longrightarrow (\lambda^{-1} I'_1 \quad \lambda^{-2} I'_2 \quad \lambda^{-3} I'_3 \quad \lambda^{-3} I'_4 \quad \lambda^{-6} I'_5 \quad \lambda^{-4} I'_6 \quad \lambda^{-6} I'_7 \quad)^T.$$

The following operator

$$\mathcal{R}_p(u) = \text{sign}(u)|u|^{1/p}, \text{ for } p \in \mathbb{N} \text{ and } u \in \mathbb{R} \tag{14}$$

when applied to an invariant I_k produces again an invariant. It is a "legal" operation that invariants can be subjected to, i.e., neither their invariance properties nor their discriminative power are lost. The feature vectors

$$\tilde{I} = (I_1 \quad \mathcal{R}_2(I_2) \quad \mathcal{R}_2(I_3) \quad \mathcal{R}_2(I_4) \quad \mathcal{R}_4(I_5) \quad \mathcal{R}_3(I_6) \quad \mathcal{R}_4(I_7))^T, \tag{15}$$

$$\tilde{I}' = (I'_1 \quad \mathcal{R}_2(I'_2) \quad \mathcal{R}_3(I'_3) \quad \mathcal{R}_3(I'_4) \quad \mathcal{R}_6(I'_5) \quad \mathcal{R}_4(I'_6) \quad \mathcal{R}_6(I'_7))^T \tag{16}$$

now satisfy the homogeneity condition as can be easily verified. By numerical experiments it is shown that hereby the vector elements remain in comparable range. At this point we might consider to introduce a distance measure between feature vectors F and F' which vanishes altogether if $F = \lambda F'$, $\lambda \in \mathbb{R}$. This is a matter for future investigation.

We conclude this section by the remark that the detail coefficients produced by the lifting scheme (at all scales) are invariant with respect to the offset in the above mentioned affine transform. We discuss this point in more detail in the next section.

5.3. Lifting & Moment Invariants

Firstly we elaborate briefly on the numerical computation of moments. Secondly we discuss feature vectors applied to selections of lifting (detail) coefficients.

5.3.1. Computation of moments

Using the values of the image pixels (or later on lifting coefficients) we construct an interpolating function based on piecewise constant approximation. This interpolating function is normalized in such way that the shortest side of the interpolated image has size 1. The central moments are now computed using these interpolating functions. These interpolating functions are not in the Schwartz class but since they are measurable and have compact support it is possible to perform the integration in (9)–(10) and compute the moments.

5.3.2. Feature vectors revisited

To create feature vectors from a given image, using a wavelet lifting approach, we first compute the detail coefficients d_1 that appear in the lifting scheme for this image. The coarse scale approximation data s_1 is used again in the lifting scheme to obtain d_2 and s_2. After K recursive lifting steps we can use the detail coefficients $d_1, \ldots, d_K$ for computing feature vectors, as follows. After each step we select a relatively small set of large detail coefficients. Particularly, we take the L_j largest coefficients in d_j (in modulus). The set of these coefficients is given by W_j. All other detail coefficients $d_j(n,m)$, $(n,m) \notin W_j$ are put to zero. Of this newly constructed 'image' d_j the moments μ^j_{pq} are computed. However, before calculating the invariants at this scale j we have to take into account that the number of coefficients L_j may not be a constant as a function of j. This is due to the fact that at lower resolution levels details in the picture that cause high-valued detail coefficients may be lost. Therefore we choose to use moments ν_{pq} that are normalized to the number of coefficients as well, i.e., $\nu^j_{pq} = \mu^j_{pq}/L_j$.

After K lifting steps we have at our disposal $\vec{I}^{(j)}$, $j = 1, \ldots, K$. Combining them into one $7K$-dimensional vectors yields the feature vector. We compute the distance between images as the variance weighted Euclidean distance between two feature vectors. The weight factors are given by the inverse variances for each vector entry, computed over all corresponding entries in the feature vectors resulting from the database we use. We take $K = 4$, a common choice with other wavelet approaches.[4,16]

Offset in the affine transform As already mentioned before the detail coefficients obtained in this way are invariant to the offset b in the affine transform of our luminosity model. To show this we construct a new image $y \in l^2(\mathbb{Z}^2)$, given an image $x \in l^2(\mathbb{Z}^2)$, by

$$y(i,j) = x(i,j) + b,$$

for all $i, j \in \mathbb{Z}$ and for a given $b \in \mathbb{R}$. According to (1) the detail coefficients of y are given by

$$\begin{aligned} d_y(i,j) &= y(i,j) - (\mathcal{P}y)(i,j) \\ &= x(i,j) + b - \sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m)\,(x(i+n, j+m) + b) \\ &= x(i,j) + b - b - \sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m)\,x(i+n, j+m) \\ &= d_x(i,j), \end{aligned}$$

if and only if

$$\sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m) = 1,$$

for the given prediction filter. This necessary and sufficient condition is satisfied for the Neville filters we use in our approach, independent of the filter order. Generally, the detail coefficients are invariant to offsets in the range of an image if and only if the sum of the prediction filter coefficient is equal to 1. Also in classical wavelet analysis this is a well-known and commonly used condition.

As discussed in the previous chapters, our approach is based on the computation of statistics of detail coefficients. The magnitude of these coefficients depend on both the sharpness of transitions (e.g. edges) in the image and the order of the prediction filter P. Since our database consists of images of single objects, detail coefficients are expected to give a good representation of the contours around and within these objects. Prediction filters of a low order yield a more crisp representation of these contours than high order prediction filters. Obviously, low order filters are bound to match with highly irregular functions while high order filters are related to regular functions/smooth surfaces. As mentioned already before, a prediction filter of low order is very undesirable if the single object is placed on a texture-like background and we want to identify similar objects at different backgrounds. Texture can consist of a smooth surface with sharp transitions superposed on it. When using a low order filter all these transitions are translated into detail coefficients with high magnitudes. A high order filter takes a large region around these transitions into account, see Table 1. Therefore the magnitudes of the coefficients related to the edges in the texture drop.

Our approach combines both ingredients. We want to obtain a small set of large detail coefficients related to an object photographed in an image and a large set of small detail coefficients related to the background in the same

image. By doing this our image retrieval system becomes less dependent on the background, which will improve retrieval rates. Following these observations we modify the algorithm as described in Section 5.3.2. Instead of using a given prediction filter we choose a filter related to the content of the input image locally. For the update filter we just use a 2nd order Neville filter. This lifting procedure is repeated over 4 levels to obtain a 28-dimensional feature vector. As described before this feature vector is the result of computing moment invariants of some selected lifting coefficients (W_j at level j). In our approach we take a threshold value T_j to select the set of coefficients W_j, namely

$$d_j(n,m) \in W_j \iff |d_j(n,m)| \geq T_j.$$

The threshold value we propose is $T_j = 2\,\mu_j$, with μ_j the mean magnitude of all detail coefficients at level j. We observe that due to the adaptive character in our approach magnitudes of the detail coefficients related to the background are likely to be less than the proposed threshold, since they let the mean of the magnitude fall. On the other hand the most interesting coefficients will increase the mean and therefore they are likely to be picked out and put into the set W_j.

Finally, in this application we do not bother about reconstructing images from their lifting coefficients. We only use the detail coefficients for computing statistics of an image. Since we do not need to know afterwards at which place in the spatial domain which filter has been used, we do not have to keep track of the decisions we have made during the prediction procedure. For coding for example this is necessary, which makes adaptive lifting not that straightforward. By not keeping track of the filter orders one has used, different images may yield the same detail coefficients. In our approach this is not really a problem, since all images in the database are of the same type and are thus very unlikely to yield similar detail coefficients along 4 levels at the same spatial positions.

The algorithm is not yet complete: we need to take additional precautions, see Sections 5.4 and 5.5.

5.4. Filtering and Moment invariants: Preprocessing

When using our algorithm we have to deal with the fact that in practice most affine transformations do not map an image on one (Cartesian) lattice onto an image defined on the same lattice. Generally, an interpolation filter is used to get a new image, which is defined on the original lattice. It can be shown that if the difference between the image I, that was not defined on the original lattice, and its interpolated version $\tilde{I}$, defined on the lattice, is small then also the difference in the moments μ_{pq} will be small. So the moments are continuous in the image functions in some set. More precise, we have

$$|f(x,y) - g(x,y)| < \delta\,|x|^{-\alpha}\,|y|^{-\beta} \implies |\mu_{pq}(f) - \mu_{pq}(g)| < \varepsilon,$$

for $0 \leq p < \alpha - 1$ and $0 \leq q < \beta - 1$.

Applying a high pass filter (prediction filter) on both the interpolated and the original image yield some other results. Although the detail coefficients of both images will only slightly differ from each other, they will induce large differences in the central moments μ_{pq}. This is due to the fact that the moments are computed out of the magnitudes of these coefficients. So, relatively small differences are not canceled out against each other, but they accumulate towards large differences in the moments. Simulation results have shown that they are highly influenced by this phenomenon.

As a remedy we propose to use a smoothing filter on all images in the database at each resolution level. In this way small blurring filters are used at the lowest resolution level and large blurring filters are used at the highest levels. In our setup a (5×5) 2D cubic spline filter has been used, however also other smoothing filters (Gaussian) gave a good performance. A rigorous and detailed mathematical treatment of the above can be found in a forthcoming paper.[17]

5.5. Filtering and Moment invariants: Similitude invariance

In Section 5.2.1 we have already seen that uniform dilations (by $\alpha > 0$) result in a multiplication of Hu's moments by $\mu'_{pq} = \alpha^{p+q+2}\mu_{pq}$. However, if the object in an image can just be replaced by an orbit on a neutral background, e.g., a circle, then a uniform dilation yields $\mu'_{pq} = \alpha^{p+q+1}\mu_{pq}$. Assuming that the selected detail coefficients also induce orbits instead of regions, this change in moments also holds for the moments computed from these coefficients. To increase stability in the lifting approach all moments were also divided by L_j yielding new moments ν^j_{pq}. For these moments we have

$$\nu^j_{pq} = \alpha^{p+q}\nu^j_{pq}.$$

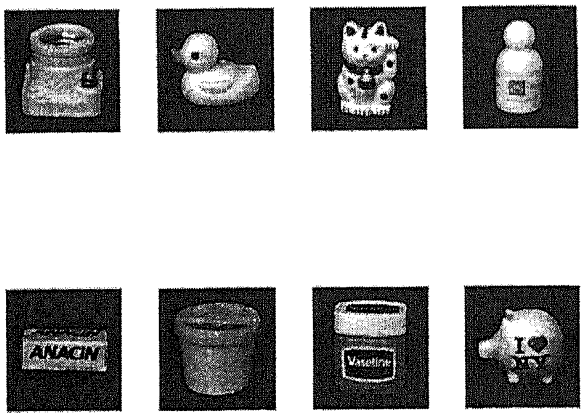


Figure 6. Object library of 8 images of size 128 × 128.

Following the computations in Section 5.2.1 invariance of the wavelet moments ν_{pq}^j is achieved by dividing them by $(\nu_{20}^j + \nu_{02}^j)^{\frac{p+q}{2}}$, yielding at level j the new moments

$$\xi_{pq}^j = \frac{\nu_{pq}^j}{(\nu_{20}^j + \nu_{02}^j)^{\frac{p+q}{2}}} = \frac{\mu_{pq}^j \, L_j^{\frac{p+q}{2}-1}}{(\mu_{20}^j + \mu_{02}^j)^{\frac{p+q}{2}}}.$$

We observe that invariant I_1 is now always equal to 1 due to this normalization. Therefore we replace this invariant by $I_1 = \mu_{00}$.

Of course the assumption that the detail coefficients are only grouped into orbits and not into regions is not always justified. We expect that a small correction of this normalization factor will give a better performance. However, testing this invariance on large test sets with several normalization factors has to confirm our expectations. This will be reported in an upcoming paper.

6. SIMULATION

To give a 'proof of principle' of our proposed adaptive lifting approach, we constructed a synthetic database of 64 images. These images can be divided into 8 classes, each of one consisting of images of one of the objects depicted in Figure 6, translated, rotated and reflected over various distances/angles and finally pasted on an arbitrary chosen (out of 4) wooden texture background (256 × 256). For the simulation each image was used as a query to retrieve the other 7 relevant ones. So far we have not really used the similitude transform. As mentioned before we want to investigate this invariant in combination with high pass filters in the near future.

The effectiveness of our approach (solid line) is shown in Figure 7 with both the ideal case (crosses) and the case in which the lifting scheme with a fixed prediction filter was used (dotted line). In this figure the performance using an 8th order filter has been depicted since it performed slightly better than lifting with low order filters. The average number of retrieved images of the same class of the query image (vertical axis) has been plotted against different number of allowed top retrievals. As we can see, retrieval rates increase by 5-10% by using an adaptive approach in our test case.

Another result is the following. We sort the weighted Euclidean distances between one query image and all 63 other images in increasing order and collect them into one vector. This has been done for all 64 query images. The mean of each entry in these 64 vectors has been plotted in Figure 8. In the left picture we see the mean mutual distances for both the adaptive (solid line) and the non-adaptive (dotted line). In the middle and right picture we depict respectively the lowest and highest mean distances. Obviously, distances amongst images in a similar class (approximately first 8) are much smaller in the adaptive case than when using the classical lifting scheme. On the other hand we see the distances become much larger in our approach if the query image belongs to different similarity classes. This gives reason to believe that our approach is usable for other types of databases with background textures quite different from the ones we have used so far.

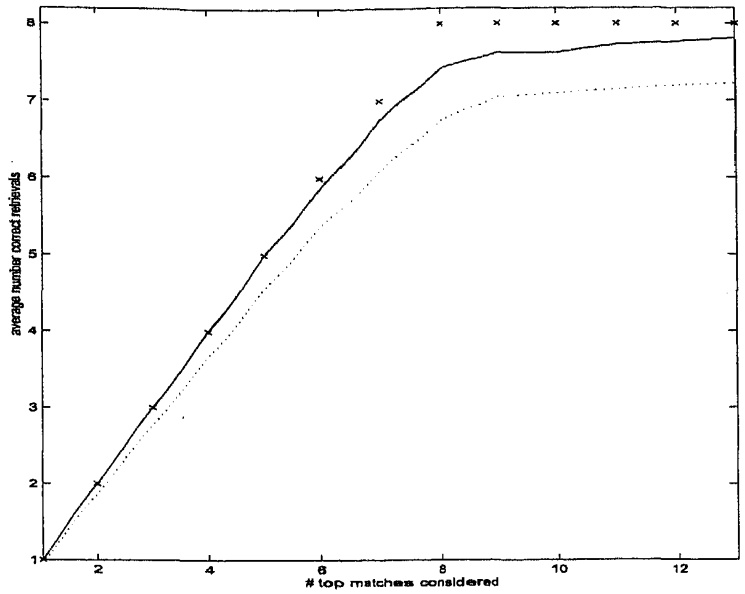


Figure 7. Retrieval performance of adapted (solid) and non-adapted (dotted) approaches.

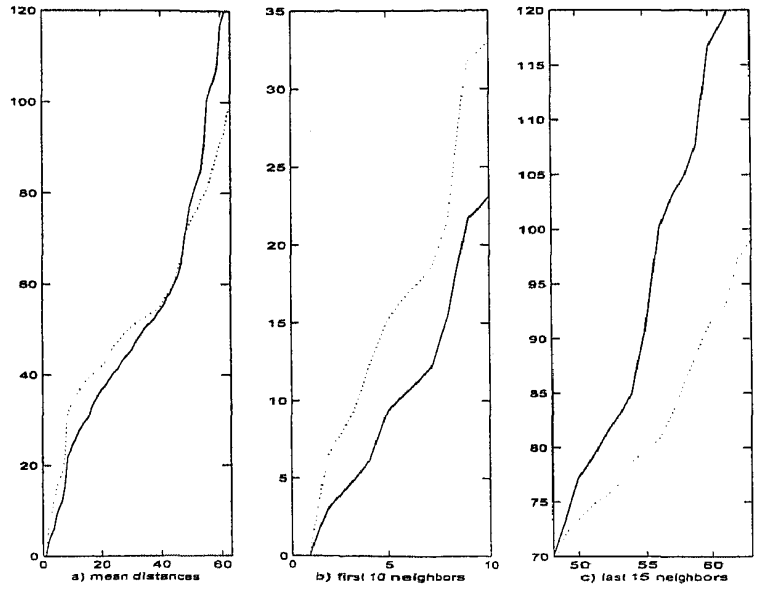


Figure 8. Mean mutual distances in object database.

7. CONCLUSIONS/FUTURE RESEARCH

In this paper we described how adaptive lifting can be used for certain types of content based image retrieval. Test results show that the adaptive approach performs better than non-adaptive approaches both in a qualitative and a quantitative manner. The combination of adaptive lifting coefficients and moment invariants yields an improved retrieval system based on shape related information. Presumably the adaptive scheme can also be used in combination with other methods to extract other types of features in an image. Besides, following quantitative results we may expect that our approach can also work out for database of objects against many other types of texture-like backgrounds.

Furthermore, we have seen that the classical method of computing moment based invariants encounters some serious difficulties when applying them on images, that have been filtered by a high pass filter (detail coefficients). Both the computations of the moments themselves as well as certain invariants have to deal with these problems. In this paper we suggested solutions to these problems, but further research has to be carried out for a better modeling of these problems. Rigorous mathematical descriptions and solutions for these problems will be dealt with in an upcoming paper.[17]

ACKNOWLEDGMENTS

This research was supported financially by the Technology Foundation (STW), project no. CWI4616.

REFERENCES

1. A. Wegner, website: "Global Retrieval, Access and information System for Property items", 1996. `www.arttic.com/grasp/`
2. W. Ashley, "What shoe was that? The use of computerised image database to assist in identification", *Forensic Science International*, **82**, 7–20, 1996.
3. M. Hu, "Visual pattern recognition by moment invariants", *IRE Trans. Inf. Th.*, IT-8, 179–187, 1962.
4. M. Do, S. Ayer and M. Vetterli, "Invariant image retrieval using wavelet maxima moment", *Proc. Visual '99*, Amsterdam, 1999.
5. W. Sweldens, "The lifting scheme: A construction of second generation wavelets", *SIAM J. Math. Anal.*, 29(2), 511–546, 1997.
6. J. Kovacevic and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions", *IEEE Trans. Imag. Proc.*, 9(3), 480–496, 2000.
7. G. Uytterhoeven and A. Bultheel, "The red-black wavelet transform", TW Report 271, Dept. Comp. Sc., Katholieke Universiteit Leuven, Leuven, 1997.
8. G. Nason and B. Silverman, "The stationary wavelet transform and some statistical applications", in *Wavelets and Statistics* (Antoniadis and Oppenheim eds.), 281–299, 1995.
9. E. Candes and D. Donoho, "Curvelets, Multiresolution Representation, and Scaling Laws", *Proc. SPIE 4119, Wavelet Applications in Signal and Image Processing VIII*, 1–12, 2000.
10. T. Chan and H. Zhou, "Adaptive ENO-wavelet transforms for discontinuous functions", Tech. Rep. 21, Comp. and Appl. Mathematics, UCLA, Los Angeles, 1999.
11. D. Donoho, "Wedgelets: nearly minimax estimation of edges", Tech. rep. Statistics Dept. Stanford University, 1997.
12. S. Mallat and Z. Zhang, "Matching Pursuits with Time Frequency Dictionaries", *IEEE Trans. Signal Processing*, 41(12), 3397–3415, 1993.
13. W. Trappe and K. Liu, "Adaptivity in the lifting scheme", *Proc. 33th Conf. on Inf. Sciences and Systems*, Baltimore, 950–955, 1999.
14. R. Baraniuk, R. Claypoole, G. Davis and W. Sweldens, "Nonlinear wavelet transforms for image compression via lifting", *to appear in IEEE Trans. Imag. Proc.*
15. A. Cohen, I. Daubechies and J. Feauveau, "Bi-orthogonal bases of compactly supported wavelets", *Comm. Pure Appl. Math.*, 45, 485–560, 1992.
16. B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data", *IEEE Trans. PAMI*, 18(8), 837–842, 1996.
17. P. Oonincx and P.M. de Zeeuw, "On the use of filters and Hu's moment invariants", in preparation.