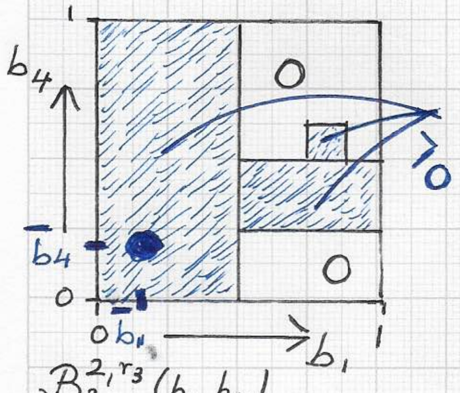
▪ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... In the following, we briefly sketch concepts and ideas that relate to the methods described. They could be explored in more depth, to further optimize the recognition of an object/material class and to improve computational efficiency.

• Use of the reduced function $B_j^{n_j, r_j}$. We consider the use of the class-3-identifying function $B_3(b_1, \dots, b_n)$, reduced to $B_3^{2, r_3}(b_1, b_4)$.



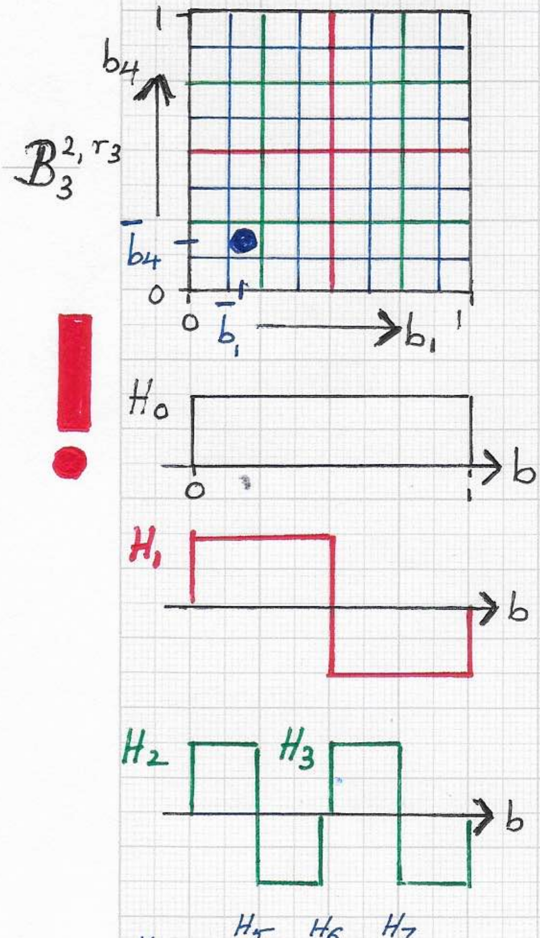
$B_3^{2, r_3}(b_1, b_4)$

The left figure shows B_3^{2, r_3} over its dimensionally reduced subspace domain $[0,1]^2$, considering only variables b_1 and b_4. The point • $= (\bar{b}_1, \bar{b}_4)$ represents the specific value of a similarity analysis of a new, to-be-classified object/material. The figure indicates that the tuple $(\bar{b}_1, \bar{b}_4)$ lies inside a region of the domain of the reduced Haar wavelet expansion B_3^{2, r_3} where the value of B_3^{2, r_3} is larger than 0. The value of $B_3^{2, r_3}(\bar{b}_1, \bar{b}_4)$ is interpreted as a (real) probability value; assuming one has computed a data-based optimal threshold value t_3, one can use this condition: $B_3^{2, r_3}(\bar{b}_1, \bar{b}_4) > t_3 \Rightarrow$ call 3. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

• Use of coefficients of (reduced) Haar wavelet expansion $B_j^{n_j, r_j}$. In some situations it can be advantageous to consider the coefficients of specific tensor product Haar wavelet basis functions for classification when expanding the class-identifying function with n_j terms and using r_j resolutions.



The left figure illustrates the example used on the previous page in more detail. The expansion B_3^{2,r_3} uses only two "b-variables" and uses detail levels 0, 1, 2 and 3, i.e., B_3^{2,r_3} is given as $B_3^{2,r_3}(b_1, b_4) = \sum_{j=0}^{7} \sum_{i=0}^{7} c_{i,j} H_i(b_1) H_j(b_4)$.

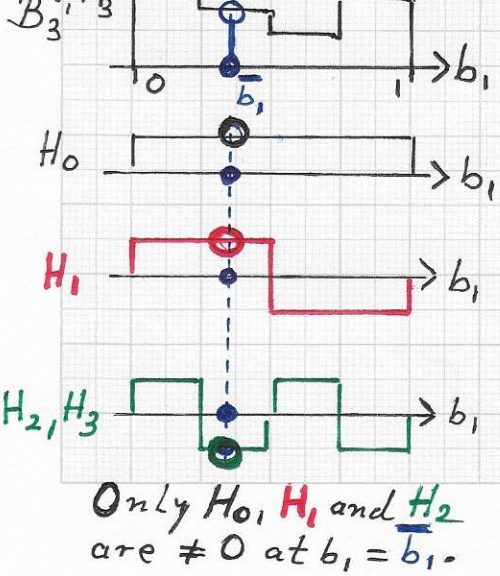
In addition to the requirement $B_3^{2,r_3}(\bar{b}_1, \bar{b}_4) > t_3$ to be satisfied for identifying class 3, one can, IN ADDITION, require that the coefficient $c_{i,j}$ in the expansion satisfy requirements as well, e.g., Here, there are $8 \times 8 = 64$ coefficients $c_{i,j}$, $i,j = 0 \ldots 7$. For example, one could add the requirements that $|c_{i,j}| > t_{i,j}^3$ for the recognition of a class-3 object/material.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• <u>Laplacian eigenfunctions and neural networks:...</u> In this case, where each test "$|c_{i,j}| > t_{i,j}^3$?" can have FALSE or TRUE as result, there **2^{64} possible Boolean result tuples**, i.e., 64-tuples for the alphabet $\{F, T\} = \{FALSE, TRUE\}$, defining the 64-tuples from **$(F, F, \ldots, F)$** to **$(T, T, \ldots, T)$**. Since the number of possible Boolean result tuples grows rapidly with the number of tests "$|c_{i,j}| > t_{i,j}^3$?", one would limit these tests to a "small and computationally acceptable number." If one considered such "coefficient test," one could, again, use the number of samples in the sample database as a "base measure" for determining a corresponding number of potential coefficient tests.

• **Use of compact, finite support of class-identifying functions in Haar wavelet representation.** The left figure is an abstract, simplified sketch of the fact that many terms in a tensor product Haar wavelet expansion are 0 as a consequence of products of the univariate basis functions have value 0 for a specific tuple $(\bar{b}_\cdot, \ldots, \bar{b}_\cdot)$, $\bar{b}_1$ in the shown example.

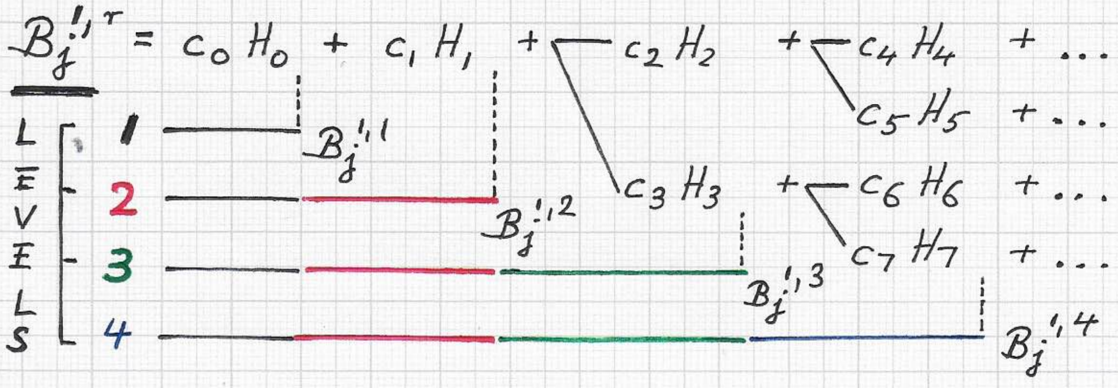


Only H_0, H_1 and $\underline{H_2}$ are $\neq 0$ at $b_1 = \bar{b}_1$.

...

OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

- Laplacian eigenfunctions and neural networks: ... This insight makes it possible to accelerate the evaluation of all class-identifying functions $B_j^{n_j, r_j}$ substantially: Given a specific tuple $(\bar{b}_\cdot, \ldots, \bar{b}_\cdot)$ in the (reduced) n_j-dimensional domain, one must determine those tensor product basis functions that have a value $\neq 0$ for this tuple and subsequently use only these basis functions for the calculation of $B_j^{n_j, r_j}(\bar{b}_\cdot, \ldots, \bar{b}_\cdot)$. Thus, one needs to design a simple method that determines the set of indices of basis functions with values $\neq 0$ for a given domain tuple $(\bar{b}_\cdot, \ldots, \bar{b}_\cdot)$.



- **Use of dyadic multi-resolution nature of class-identifying function $B_j^{n_j, r_j}$.** The above figure illustrates the inherent multi-resolution characteristic of a univariate Haar wavelet expansion $B_j^{1,r}$, using 1, 2, 3 and 4 levels of resolution. For example, $B_j^{1,3}$ is a univariate function defined by (the first) three levels of resolution of $B_j^{1,r}$. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:… Breaking down this function $B_j^{1,r}$ into its summands shows that only one summand must be used when adding the non-zero summand for a specific level of resolution — for evaluation for the domain tuple $(\bar{b}_1)$, in the univariate case.

$$\underline{B_j^{1,r}(\bar{b}_1)} = c_0 H_0 + c_1 H_1 + c_{l_2} H_{l_2} + c_{l_3} H_{l_3} + \ldots =$$
$$= T_0 + T_1 + T_2 + T_3 + \ldots,$$

where $l_2 \in \{2,3\}$, $l_3 \in \{4,5,6,7\}$ etc.

Thus, each term T_i represents the contribution of one level-of-detail of the expansion $B_j^{1,r}(\bar{b}_1)$. Since the value of a term T_i can be negative, zero or positive, the value of $B_j^{1,r}(\bar{b}_1)$ fluctuates when adding more terms for evaluation.

p_0	p_1	p_2	p_3	P
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

0 = FALSE
1 = TRUE

One can use this fact for determining a number for the resolution levels needed to obtain the required classification performance: The left table shows Boolean variables p_i with truth value $p_i = (B_j^{1,i} > t_j)$ and $B_j^{1,1} = T_0$, $B_j^{1,2} = T_0 + T_1$, $B_j^{1,3} = T_0 + T_1 + T_2$ etc. The result Boolean variable P indicates for which tuples (p_0, p_1, p_2, p_3) class j is recognized or not. …