OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:... In other words, the Boolean function P depends on four Boolean variables $p_0, \dots, p_3$, each p_i-value indicating whether an approximation of $B_j^{1,T}(\bar{b}_1)$ using $(i+1)$ terms of its expansion yields FALSE (0) or TRUE (1) regarding the recognition of class j (in this univariate, 1D scenario). One can now compute "increasingly precise" values of a P-sequence: $P_0 = P(p_0)$, $P_{01} = P(p_0, p_1)$, $P_{012} = P(p_0, p_1, p_2)$, $P_{0123} = P(p_0, p_1, p_2, p_3)$ etc. The progression of this sequence makes it possible to establish a sufficient number of terms T_i to be used for classification. Further, one must keep in mind that the truth table (previous page) for variables $p_0, p_1, \dots, p_k$ has 2^{k+1} rows. Thus, it might be desirable to reduce the Boolean P-function via a K-map (Karnaugh or Karnaugh-Veitch diagram). Of the 16 possible Boolean value quadruples (p_0, p_1, p_2, p_3) in the truth table on the previous page, 7 lead to the P-value 1. The left diagram shows the 4×4 (p_0, p_1) and (p_2, p_3) tuple combinations of a K-map...

p_0,p_1 \ p_2,p_3	0,0	0,1	1,1	1,0
0,0	0	0	1	0
0,1	0	0	1	0
1,1	1	1	1	1
1,0	0	0	0	1

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... The truth table defines the Boolean function P as

$$P = (\neg p_0 \wedge \neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_0 \wedge p_1 \wedge p_2 \wedge p_3) \vee (p_0 \wedge \neg p_1 \wedge p_2 \wedge \neg p_3) \vee (p_0 \wedge p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (p_0 \wedge p_1 \wedge \neg p_2 \wedge p_3) \vee (p_0 \wedge p_1 \wedge p_2 \wedge \neg p_3) \vee (p_0 \wedge p_1 \wedge p_2 \wedge p_3).$$

Karnaugh's method uses the $(p_0, p_1)-(p_2, p_3)$ diagram on the previous page to identify "groups-of-16", "groups-of-8", "groups-of-4", "groups-of-2" and "groups-of-1" (allowable) patterns in the diagram for the minimization of the Boolean function P. The diagram shows three groups, defining the function P as

! $$P = (p_0 \wedge p_1) \vee (\neg p_0 \wedge (p_2 \wedge p_3)) \vee (p_0 \wedge \neg p_1 \wedge p_2 \wedge \neg p_3).$$

Considering the more precise notation of a P-function, this function is the function $P = P_{0123} = P(p_0, p_1, p_2, p_3)$ with its Boolean value depending on four Boolean variables p_i, $i = 0 \ldots 3$.

! When the number of variables p_i increases and becomes "Large," a basic K-map approach is no longer feasible and one must rely on other methods for minimizing P-functions (e.g., combinatorial optimization methods).

The same ideas apply to Haar wavelet expansions $B_j^{n_j, r_j}$ for $n_j > 1$, i.e., multivariate expansions: Only the needed T-terms ($\neq 0$) must be used.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... •Note. It is important to keep in mind that the number of tensor product Haar wavelet functions grows rapidly, with increasing numbers of resolution levels and (domain) dimensions. Thus, it is crucial for computational efficiency to use only those tensor product Haar wavelet functions that are non-zero – for a specific argument tuple/evaluation point in the multi-dimensional domain. For example, one can think of and group the Haar wavelet functions according to resolution levels L_i:

L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9
H_0	H_1	H_2	H_4	H_8	H_{16}	H_{32}	H_{64}	H_{128}	H_{256}
		H_3	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
			H_7	H_{15}	H_{31}	H_{63}	H_{127}	H_{255}	H_{511}

The left table lists the 10 groups of the univariate Haar wavelet basis functions obtained by dyadic domain interval subdivision. Only ONE of the Haar wavelet functions is NON-ZERO at level L_i. Here, the numbers of basis functions per resolution level are 1, 1, 2, 4, 8, 16, 32, 64, 128, 256. The total number of all these basis functions therefore is 512. When evaluating a univariate Haar wavelet expansion for these 10 levels, 10 functions are needed.

...

Stratovan

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... The important ratio to be considered in this context is the ratio defined by the number of functions needed for evaluation divided by the total number of (tensor product) basis functions in the expansion: **10/512** in the univariate 10-level example described on the previous page. For simplicity, we consider only a straightforward tensor product setting for the general multivariate, multiresolution Haar wavelet expansion: We assume that the multi-dimensional domain is N-dimensional, defined as $[0,1]^N$, and that L levels of resolution (levels $0, 1, \dots, L-1$) are used for the dyadic Haar basis functions. Thus, in the bivariate/two-dimensional case ($N=2$), the total number of tensor product basis functions is

$$(1+1+2^1+\dots+2^{L-2})\cdot(1+1+\dots+2^{L-2}) =$$
$$= 2^{L-1}\cdot 2^{L-1} = (2^{L-1})^2 =$$
$$= (2^{L-1})^N.$$

Of these $(2^{L-1})^2$ functions only $L \cdot L = L^2 = L^N$ functions are needed for evaluation. The left table shows ratios $(L/2^{L-1})^N$ for some L and N values. ...

$N \backslash L$	1	3	5	7	9	11
1	1	$3/4$	$5/16$	$7/64$	$9/256$	$11/1024$
3	1	$(3/4)^3$	$(5/16)^3$	$\dots$		$\left(\frac{11}{1024}\right)^3$
5	1	$(3/4)^5$				
7	1	$(3/4)^7$	$\vdots$			$\vdots$
9	1	$(3/4)^9$				
11	1	$(3/4)^{11}$	$(5/16)^{11}$	$\dots$		$\left(\frac{11}{1024}\right)^{11}$ *

* $(11/1024)^{11} = 2.2\cdot 10^{-22}$

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... The most important fact here is the fact that "only" L^N tensor product Haar wavelet basis functions must be used for evaluation, assuming that the number of resolution levels is the same (L) for each of the N dimensions. Further, since the originally given number of data/samples best-approximated by the tensor product wavelet expansion, S, the values of L and N should be related in a "meaningful way" to the value of S, e.g., $S = L^N$. Therefore, it is imperative to use, ideally, a "constant-time indexing scheme" that rapidly determines the L^N tensor product functions over the $[0,1]^N$ domain cube that are non-zero for a particular point in the domain cube.

• Computational feasibility. The methods and concepts described here and the previous pages are general and potentially very powerful for classification via multiresolution tensor product Haar wavelets. For a practially viable design of algorithms and data structures, efficiency considerations are most relevant. For example, an original function $B_j^{n,r}$ – identifying class-j materials/objects – is reduced in dimension to n_j and in resolution to r_j, which should be reduced further to $N_j < n_j$ and $L_j < r_j$ such that $S_j \approx L_j^{N_j}$.

...