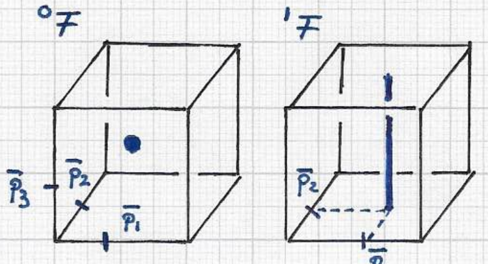
Bernd Hamann 7/5/2022

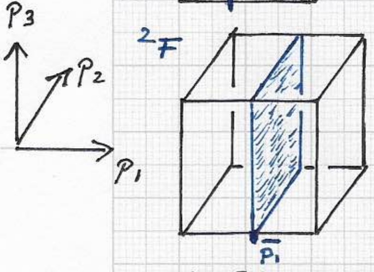
Stratovan

▪ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Other possible integral characteristics, quantities of multi-variate decider functions:





For example, in a **3D** decider function domain integral quantities (averages) can be computed for **3D**, **2D**, **1D** or **OD** subdomains. Integral quantities for **d**-dimensional subdomains are denoted by ${}^{d}\mathcal{F}$ in the figure. Coordinate system-aligned quantities for a trivariate decider function are the following:

${}^{0}\mathcal{F}$: $\mathcal{F}(\bar{p}_1, \bar{p}_2, \bar{p}_2)$

${}^{1}\mathcal{F}$: $\int_{p_1} \mathcal{F}(p_1, \bar{p}_2, \bar{p}_3)\, dp_1$

$\int_{p_2} \mathcal{F}(\bar{p}_1, p_2, \bar{p}_3)\, dp_2$

$\int_{p_3} \mathcal{F}(\bar{p}_1, \bar{p}_2, p_3)\, dp_3$

${}^{2}\mathcal{F}$: $\iint_{p_1,p_2} \mathcal{F}(p_1, p_2, \bar{p}_3)\, dp_1\, dp_2$

$\iint_{p_1,p_3} \mathcal{F}(p_1, \bar{p}_2, p_3)\, dp_1\, dp_3$

$\iint_{p_2,p_3} \mathcal{F}(\bar{p}_1, p_2, p_3)\, dp_2\, dp_3$

${}^{3}\mathcal{F}$: $\iiint_{p_1,p_2,p_3} \mathcal{F}(p_1, p_2, p_3)\, dp_1\, dp_2\, dp_3$

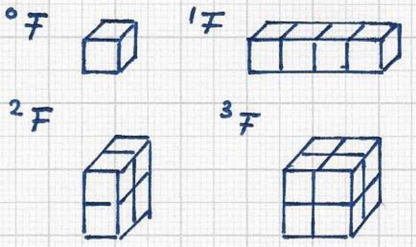
• **Note.** The discussed examples used a specific method for making the given functions ($\mathcal{F}_1$, $\mathcal{F}_2$, IDEAL) compatible with respect to integral properties: the average integral value Φ was used to enforce a unique and thus consistent integral value. **One can use other scaling values to ensure compatibility. Further, the behavior and properties of some of these given functions could be - relatively - more important.** The functions $\mathcal{F}_1(p_1)$, $\mathcal{F}_2(p_2)$, ... of a multivariate decider function $\mathcal{F}(p_1, p_2, \ldots)$ represent integral/average behavior, e.g., $\mathcal{F}_1(p_1) = \int_{p_2} \mathcal{F}(p_1, p_2)\, dp_2$ defines integral/average behavior in p_2-direction of the **2D** domain. **One can consider the use of various other integral/average quantities in the higher-dimensional setting**, e.g., consider $\mathcal{F} = \mathcal{F}(p_1, p_2, p_3)$ and $\mathcal{F}_{1,2}(p_1, p_2) = \int_{p_3} \mathcal{F}(p_1, p_2, p_3)\, dp_3$. (See left figure.)

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

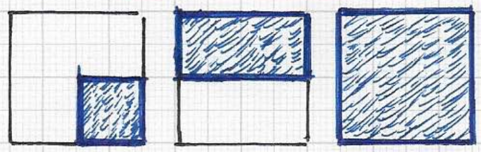
• Laplacian eigenfunctions and neural networks:...

Discrete approximations of integral/average characteristics via sets of uniform voxels representing 0D, 1D, 2D and 3D subdomains in a unit cube:

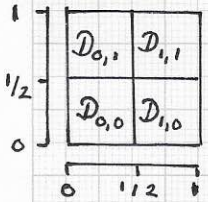


In the discrete setting, the definition of such integral/average quantities reduces to the computation of simple integrals (sums) of box function-based constant spline functions.

^0F, ^1F and ^2F data for a simple bivariate example:



^0F data ^1F data ^2F data



The unit square consists of subdomains $D_{i,j}$ with unknown values of associated $c_{i,j}$ coefficients. Given are "F data" of type ^0F, ^1F, ^2F:

^0F: $\phi(c_{1,0}) \Rightarrow \int_{D_{1,0}} = \frac{1}{4}\, c_{1,0}$

^1F: $\phi(c_{0,1}, c_{1,1}) \Rightarrow \int_{D_{0,1} \cup D_{1,1}} = \frac{1}{4}(c_{0,1} + c_{1,1})$

^2F: $\phi(c_{0,0}, c_{1,0}, c_{0,1}, c_{1,1}) \Rightarrow$
$\int_{D_{0,0} \cup D_{1,0} \cup D_{0,1} \cup D_{1,1}} = \frac{1}{4}(c_{0,0} + \ldots + c_{1,1})$

In the previous discussion, univariate decider functions were considered. A univariate decider function has as its value an integral/average of an unknown multi-variate decider function to be constructed: $F_i(p_i)$ has as its value – for a fixed value of p_i – the integral/average of all the other variables p_j $(j \neq i)$; for example, $F_1(p_1) = \int\int \ldots \int_{p_2, p_3, \ldots, p_H} \ldots$
$\ldots F(p_1, p_2, p_3, \ldots, p_H)\, dp_2\, dp_3 \ldots dp_H$. The figure on the previous page illustrates the generalization of the univariate decider function (for the case of smooth functions defined on a continuous domain). **If such general integral/average characteristics are known or provided "via experiments," then it will be desirable to use this data in the construction process of the unknown multi-variate decider function $F(p_1, \ldots, p_H)$.** A priori knowledge, conditions always to be satisfied or material samples can be the basis for such characteristics.

...

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... The thoughts expressed on the previous pages and related aspects of material classification allow us to summarize and point out important considerations at a higher level. The following points represent a selection:

1) When defining the (under- or overdetermined) linear system to determine the coefficients of a box function-based representation of a multivariate decider function $F(p_1, p_2, ...)$, one should include all available known integral/average properties of the unknown function F: 0D, 1D, 2D, 3D,... integral properties of the introduced types ${}^0F$, ${}^1F$, ${}^2F$, ${}^3F$,... The values of such properties can be "learned" via the use of available training data or are a priori known and user-specified "constraint data" that always must be satisfied.

2) Boolean and decider functions are very similar and closely related. A Boolean function has arguments with discrete, binary values, e.g., 0 and 1, and it generates a binary value — to determine whether "something IS or IS NOT the case (TRUE, FALSE). A decider function has real-valued arguments,

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... in principle, and it generates a real value, in principle. (Of course, in practical implementations, real numbers and real-number ranges are quantized/discretized.) A decider function is "more general" than a Boolean function as it represents the set {0,1} with the real interval [0,1]; in fact, it replaces the discrete set {0,1} with the continuous range [0,1]. The reason for this fact is simple: The described decider functions are based on probability; a truth value should be a value between 0 and 1. A decider function serves the purpose of making a DECISION; considering one, some or all available scale-specific similarity values between a classified and to-be-classified material, the function DECIDES whether the two materials belong to the same class or not – indicated via a probability value in the interval [0,1]. This relationship should make it possible, in principle, to map a complex Boolean logic circuit to a corresponding (approximating) decider function design – and vice versa, keeping in mind the binary-value constraint of Boolean functions.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... (In various bivariate decider functions $F(p_1, p_2)$ discussed, F was referred to as a Boolean **AND** or **OR** function, already high-lighting this similarity between decider and Boolean functions.)

3) It is important to recall the **MEANING** of decider functions at this point. For example, consider the function $F_1(p_1)$, where $p_1 = .8$ and $F_1(.8) = .9$. This is the intended and assumed meaning: "A to-be-classified image of some material class has a degree-of-similarity of .8 when compared at texture-scale 1 against an already classified image of a known material class. For this value of the degree-of-similarity, .8, the probability that the to-be-classified image belongs to the class it is compared against is .9." It is crucial to recognize that this classification probability value (.9) results when using only one texture-scale, scale 1, in the classification process. Thus, in some or many classifications of to-be-classified images, it is likely that "high" classification probability values could result when considering only a small number of texture-scales.

...