Combinatorial Ricci Curvature for Image Processing

Emil Saucan
EE Department, Technion

Joint work with

Eli Appleboim and Gershon Wolansky and
Yehoshua Y. Zeevi.

New York

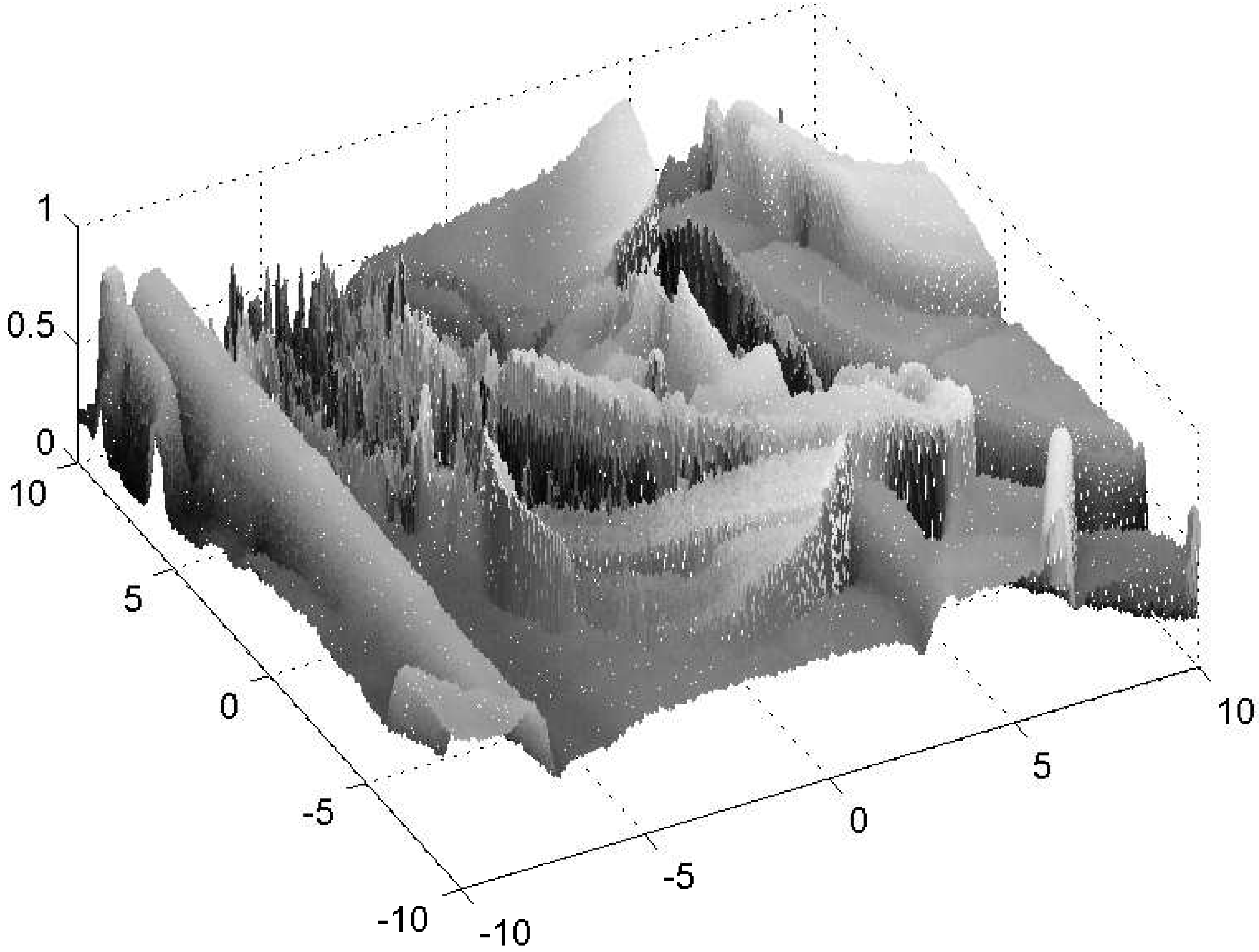
September 10, 2008.

Two major motivations for this work:

[A] The recent interest generated by G. Perelman's seminal work on the Ricci flow and its application in the proof of Thurston's Geometrization Conjecture and the application of thus flow to Computer Graphics, etc. (mainly the work of Gu et al.).

[B] The search for a curvature that can be applied (and computed) to the "very non-smooth" objects (perhaps high dimensional) of Image Processing, such as images produced using ultrasound imaging, MRI or CT.

Also, the need to work with square (and cubical grids) – as standard in Image Processing.


1
0.5
0
10
5
0
-5
-10
-10
-5
0
5
10

Note that:

• Ricci curvature measures the defect of the manifold from being locally Euclidean in various tangential directions.

• Ricci curvature represents an average of sectional curvatures.

• Ricci curvature behaves as the Laplacian of the metric.

• In dimension $n = 2$, Ricci curvature reduces to the classical Gaussian curvature.

Fortunately, a purely theoretically work allowing such a development exists: Robin Forman has developed a Combinatorial Ricci Curvature for weighted cell complexes – objects that generalize such classical and common notions of polygonal meshes and of weighted graphs.

Moreover, it produces generalized Laplacians in any dimension, thus allowing for diffusion techniques in higher dimensions.

As an extra bonus, it is **singularly fitted for cubical grids**.

To generalize the notion of Ricci curvature, in a manner that would include weighted cell complexes, one starts from the following form of the *Bochner-Weitzenböck* formula for the *Riemann-Laplace operator* $\Box_p$:

$$\Box_p = dd^* + d^*d = \nabla_p^*\nabla_p + Curv(R)\,, \tag{1}$$

where $\nabla_p^*\nabla_p$ is the *Bochner* (or *rough*) *Laplacian* and where $Curv(R)$ is a complicated expression with linear coefficients of the *curvature tensor*. (Here ∇_p is the *covariant derivative*.)

Of course, for cell-complexes one cannot expect such differentiable operators. However, a *formal* differential exists.*

Forman proved that an analogue of the Bochner-Weitzenböck formula holds in this setting, i.e. that there exists a canonical decomposition of the form:

$$\Box_p = B_p + F_p \tag{2}$$

where B_p is a *non-negative operator* and F_p is a certain diagonal matrix. B_p and F_p are called, in analogy with the classical Bochner-Weitzenböck formula, the *combinatorial Laplacian* and *combinatorial curvature function*, respectively.

*in our combinatorial context (the operator) "d" being replaced by "∂" – the boundary operator of the cellular chain complex.

Moreover, if $\alpha = \alpha^p$ is a p-dimensional cell (or p-cell, for short), then we can define the *curvature functions*:

$$\mathcal{F}_p =< F_p(\alpha), \alpha >, \tag{3}$$

Where $< \alpha, \alpha >= w_\alpha$ – the *weight* of the cell α,
and

$$< \alpha, \beta >= 0 \text{ if } \alpha \neq \beta.$$

For dimension $p = 1$ we obtain, by analogy with classical case, the following definition of discrete (weighted) *Ricci curvature*:

Definition 1 *Let $\alpha = \alpha^1$ be a 1-cell (i.e. an edge). Then the* Ricci curvature *of α is defined as:*

$$\text{Ric}(\alpha) = \mathcal{F}_1(\alpha). \tag{4}$$

While general weights are possible, making the combinatorial Ricci curvature extremely versatile, it turns out that it is possible to restrict oneself only to so called *standard* * *weights*, i.e. such that:

Definition 2 *The set of weights* $\{w_\alpha\}$ *is called a* standard set of weights *iff there exist* $w_1, w_2 > 0$ *such that given a* p*-cell* α^p*, the following holds:*

$$w(\alpha^p) = w_1 \cdot w_2^p$$

(Note that the combinatorial weights $w_\alpha \equiv 1$ represent a set of standard weights, with $w_1 = w_2 = 1$.)

*or *geometric* – because they are proportional to the *geometric content* (s.a. *length* and *area*).

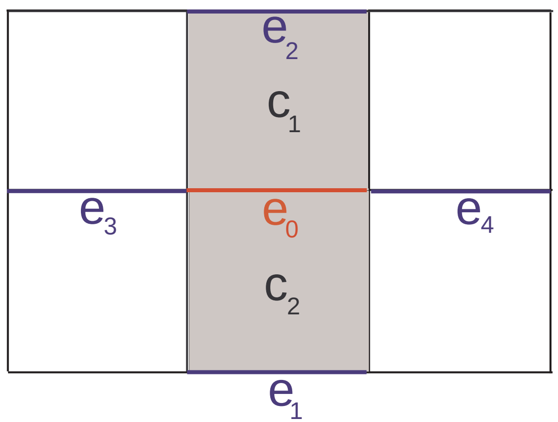
Using standard weights we obtain the following (admittedly horrendous!) formula for polyhedral (and in fact much more general) complexes:

$$\mathrm{Ric}(\alpha^p) = w(\alpha^p)\left[\left(\sum_{\beta^{p+1}>\alpha^p} \frac{w(\alpha^p)}{w(\beta^{p+1})} + \sum_{\gamma^{p-1}<e_2} \frac{w(\gamma^{p-1})}{w(\alpha^p)}\right)\right. \tag{5}$$

$$\left. - \sum_{\alpha_1^p \| \alpha^p, \alpha_1^p \neq \alpha^p} \left| \sum_{\beta^{p+1}>\alpha_1^p, \beta^{p+1}>\alpha^p} \frac{\sqrt{w(\alpha^p)w(\alpha_1^p)}}{w(\beta^{p+1})} - \sum_{\gamma^{p-1}<\alpha_1^p, \gamma^{p-1}<\alpha^p} \frac{w(\gamma^{p-1})}{\sqrt{w(\alpha^p)w(\alpha_1^p)}} \right| \right],$$

where $\alpha < \beta$ means that α is a face of β, and the notation $\alpha_1 \parallel \alpha_2$ signifies that the simplices α_1 and α_2 are *parallel*, where two p dimensional cells are parallel iff they either are common faces of a $p+1$ cell, or if they have a common $p-1$ face*, but not both simultaneously†

For example, e_1, e_2, e_3, e_4 are all the edges parallel to e_0.



*i.e. they have a common parent or a common child
†naturally!....

Together with the formula above, the (dual) formula for the combinatorial Laplacian is also obtained to be:

$$\Box_p(\alpha_1^p, \alpha_2^p) = \sum_{\beta^{p+1} > \alpha_1^p, \beta^{p+1} > \alpha_2^p} \epsilon_{\alpha_1,\alpha_2,\beta} \frac{\sqrt{w(\alpha_1^p) w(\alpha_2^p)}}{w(\beta^{p+1})} \tag{6}$$

$$+ \sum_{\gamma^{p-1} < \alpha_1^p, \gamma^{p-1} < \alpha_2^p} \epsilon_{\alpha_1,\alpha_2,\gamma} \frac{w(\gamma^{p-1})}{\sqrt{w(\alpha_1^p) w(\alpha_2^p)}},$$

where $\epsilon_{\alpha_1,\alpha_2,\beta}, \epsilon_{\alpha_1,\alpha_2,\gamma} \in \{-1, +1\}$ depend on the relative orientations of the cells.

What makes this formulae palatable is the fact that only **parallel** faces play a role.

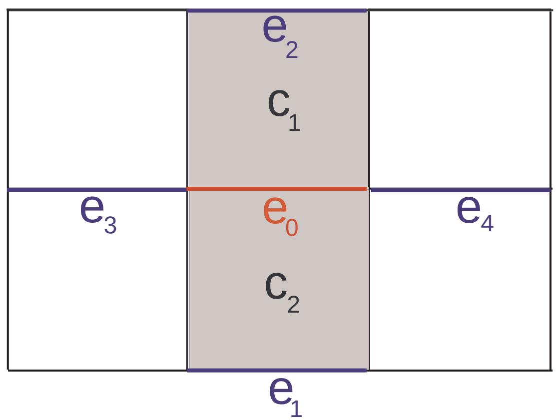
While this is a problem for triangular meshes (ubiquitous in Computer graphics, etc.) it is actually an **advantage** for the square (and cubical) grids considered in Image Processing, because of the “combinatorial clarity” regarding parallel faces.

Baring this in mind, and using standard weights (s.t. the weight of any vertex is $w(v) = 0$!) it is easy and straightforward to obtain the following simple formulae:

$$\text{Ric}(e_0) = w(e_0)\left[\left(\frac{w(e_0)}{w(c_1)} + \frac{w(e_0)}{w(c_2)}\right) - \left(\frac{\sqrt{w(e_0)w(e_1)}}{w(c_1)} + \frac{\sqrt{w(e_0)w(e_2)}}{w(c_2)}\right)\right]$$

and

$$\Box_1(e_0) = \Box_1(e_0, e_0) = \frac{w(e_0)}{w(c_1)} - \frac{w(e_0)}{w(c_2)}.$$



However, recall that for the Laplacian there exists more than one possible choice, depending upon the dimension p. The simplest, and operating on cells of the same dimensionality as the Discrete Ricci curvature, is $\Box_1$, computed above.

Instead of computing a Laplacian **along** the edge e_0, one can compute a Laplacian operating **across** the edge, namely $\Box_2(c_1, c_2)$:

$$\Box_2(c_1, c_2) = \frac{w(e_0)}{\sqrt{w(c_1)w(c_2)}}.$$

Before commencing any experiments with the combinatorial Ricci curvature in the context of images, we have to choose a set of weights for the 2-, 1- and 0-dimensional cells of the picture, that is for squares (pixels), their common edges and the vertices of the tilling of the image by the pixels.

Any such choice should be

- As natural and expressive as possible for Image Processing.

- The desire to employ solely standard weights.

In the beginning, it is natural to choose:

$w_1 = 1$ and $w_2 =$ length of a cell.*

A somewhat less arbitrary choice for the length (i.e. basic weight) of an edge, would be:

$$\text{Length(e)} = (\text{dimension of the picture})^{-1},$$

hence that for the area (i.e. basic weight) of a pixel α being:

$$\text{Area}(\alpha) = (\text{dimension of the picture})^{-2}.$$

*Remember also that that the weight of any vertex (0-cell) has to be 0.

The proper weight for a cell α should, however, take into account the gray-scale level (or height) h_α of the pixel in question, i.e.:

$$w_\alpha = h_\alpha \cdot \text{Area}(\alpha).$$

The natural weight for an edge e common to the pixels α and β is:

$$|h_\alpha - h_\beta|.$$

Remark 3 *A less "purely" combinatorial choice of cells and weights, is also considered, by passing to the dual cell complex. Convergence to the standard curvature is more easily proved in this case. Also, common convergence of both approaches holds.*

And now, for some experimental results...

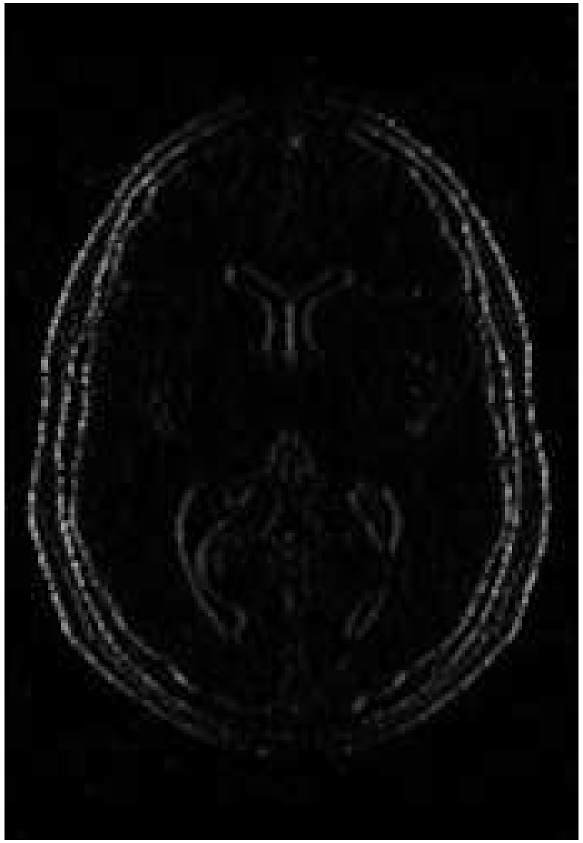
We start with the "compulsory" *Lenna*, and we compare Gaussian curvature (left) and Combinatorial Ricci curvature (right).

… and its Combinatorial Laplacian $\Box_1(e_0)$:

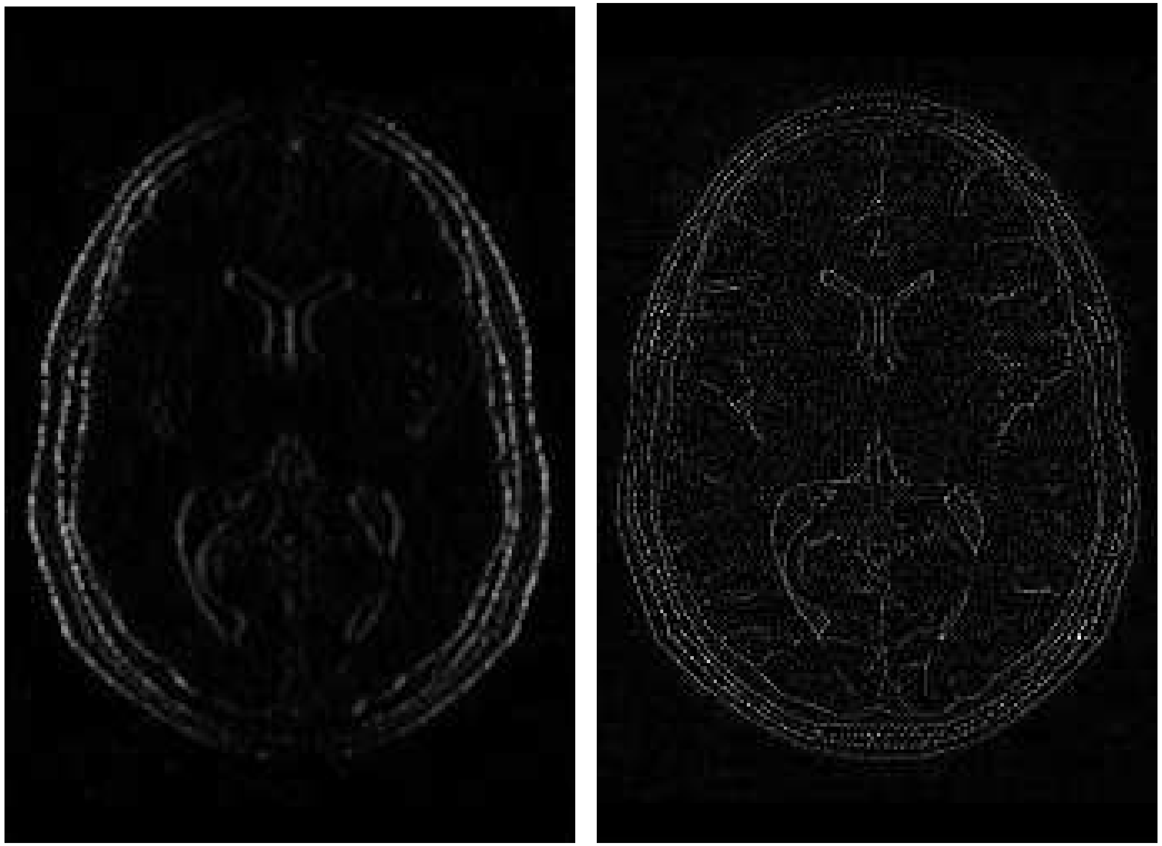
(Recall that $B_1 = \Box_1 - Ric$.)

And now, to some Medical Imaging related results:

First, an Axial Brain Scan (left) and its Combinatorial Ricci curvature (right)

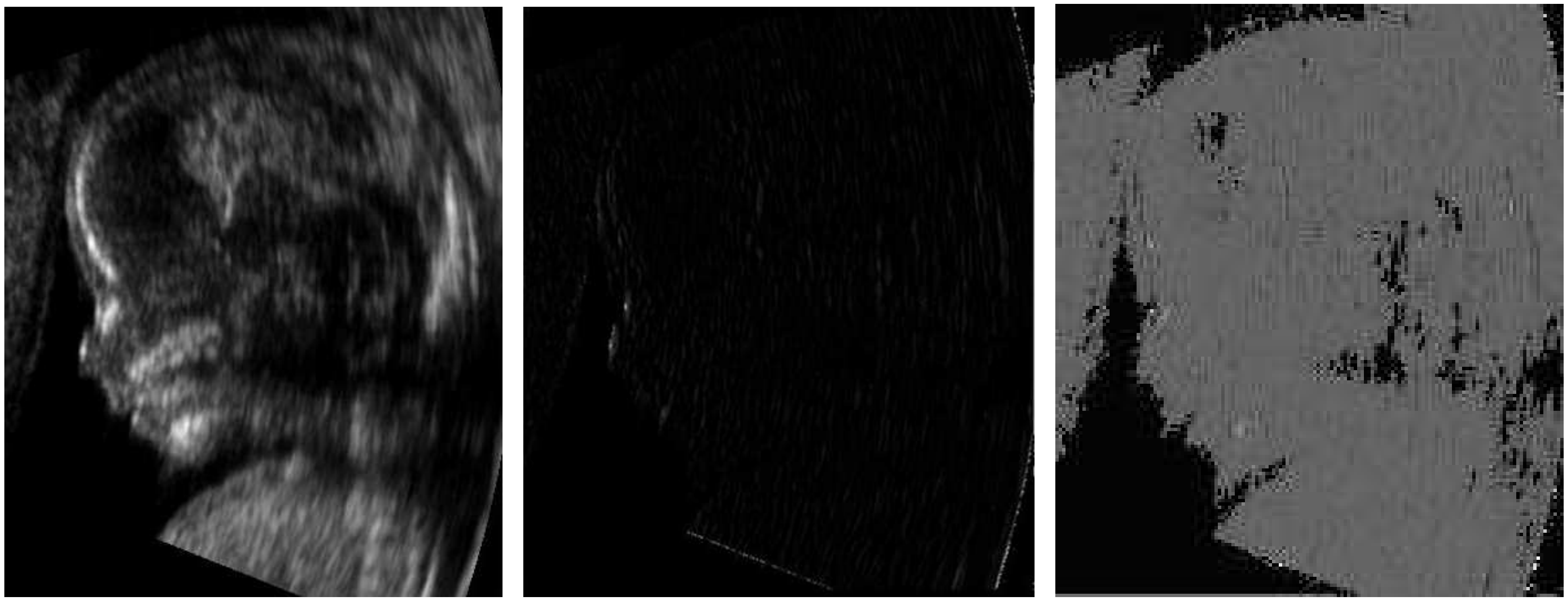


... and a comparison of its different Laplacians: Bochner (rough) Laplacian $B_1(e_0)$ (left) and "classical" Matlab (right).



... and some results for the challenging case of Ultrasound Images:

A 14 month old embryo (profile) (left); its Ricci curvature (middle) and its Bochner Laplacian (right).



Future work

- Experiment with **voxels**.

- Develop and experiment with a discrete version of the Ricci flow corresponding to the combinatorial Ricci curvature.*

- Determine, the optimal standard weights (for a given problem).

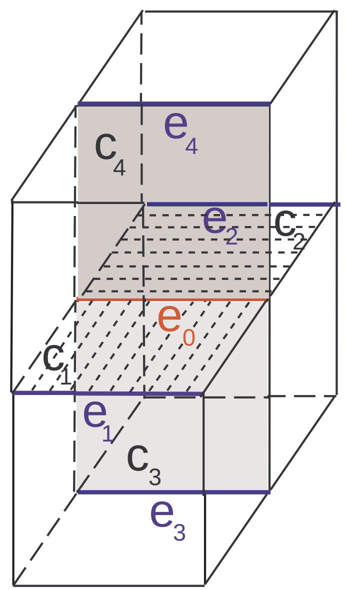
- The Combinatorial Laplacian is closely connected (by its very definition) to the cohomology groups – use it for computing cohomology groups (and by duality, of homology groups) of images.

*A *combinatorial flow* was already developed.

Appendix – Voxels

$$\mathrm{Ric}(e_0) = w(e_0)\left[\left(\sum_{c^2>e_0}\frac{w(e_0)}{w(c^2)} + \sum_{c^0<e_2}\frac{w(c^0)}{w(e_0)}\right)\right.$$

$$\left. - \sum_{e\|e_0, e\neq e_0}\left|\sum_{c^2>e, c^2>e_0}\frac{\sqrt{w(e_0)w(e)}}{w(c^2)} - \sum_{c^0<e, c^0<e_0}\frac{w(c^0)}{\sqrt{w(e_0)w(e)}}\right|\right].$$



And, since, as we have already noted, for digital images the vertex weights are always 0, we obtain the following expression for $\mathrm{Ric}(e_0)$:

$$\mathrm{Ric}(e_0) = w(e_0)\left[w(e_0)\left(\sum_1^4 \frac{1}{w(c_i)}\right) - \sqrt{w(e_0)}\left(\sum_1^4 \frac{\sqrt{w(e_i)}}{w(c_i)}\right)\right].$$