Image Formation and Analysis
(Formation et Analyse d'Images)

James L. Crowley

ENSIMAG 3 - MMIS Option MIRV First Semester 2010/2011

Lesson 8 3 Jan 2011

Scale Invariant Image Description

Lesson Outline:

1 Image Scale Space

Continuous Case.

Let $P(x, y)$ be the image.
Let $G(x, y, \sigma)$ by a Gaussian filter of "scale" σ

Image Scale space is a 3D continuous space $P(x,y,s)$

$P(x, y, s) = P(x,y) * G(x, y, 2^{s/2})$

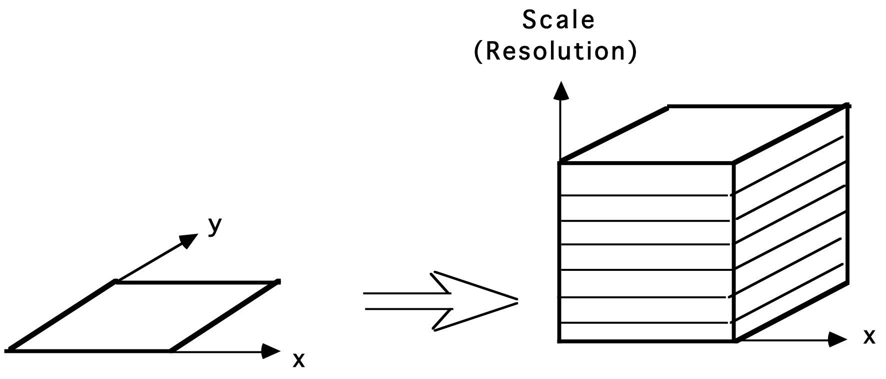
Note that the scale axis (s) is logarithmic. $s = 2 \text{Log}_2(\sigma) = \text{Log}_2(2^{s/2})$

Scale invariance: If a shape in an image is made larger by $D = 2^d$

$p(x,y) \to p(x2^d, y2^d)$

Then the scale space projection of appearance is shifted by s

$P(x,y,s+d) = p(x2^d, y2^d) * G(x2^d, y2^d, 2^d)$



The appearance of a pattern in the image results in a unique structure in $P(x, y, s)$.
This structure is "equivariant" in position, scale and rotation.
Translate the pattern by Δx, Δy and the structure translates by Δx, Δy in $P(x, y, s)$.

Rotate by θ in x, y and the structure rotates by θ in $P(x, y, s)$.

Scale by a factor of 2^s, and the structure translates by s in $P(x, y, s)$.

Scale space :
- Separates global structure from fine detail.
- Provides context for recognition.
- Provides a description that is invariant to position, orientation and scale.

Discrete Scale Space - The Gaussian Pyramid

Let P(i,j) be a discrete representation for $P(x,y) = P(i\Delta x, j\Delta x)$
Suppose P(i,j) is an image array of size M x M pixels.

We propose to sample scale with a step size of $\Delta\sigma = 2^{1/2}$ so that $\sigma_k=2^{k/2}$

$$P(x, y, s) = p(i\ \Delta x_k, j\ \Delta x_k, k) \quad \text{such that } \Delta x_k = 2^{(k-1)/2}$$

For a Gaussian Kernel filter $G(i,j,k) = G(x, y, \sigma_k=2^{k/2})$

As the Gaussian impulse response dilates, the sample density also dilates so that the impulse response remains invariant.

Scale space is presented by a pyramid:

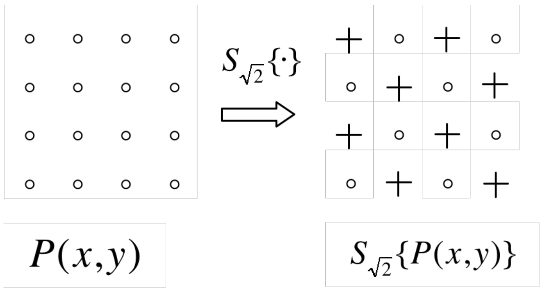
$$P(i, j, k) = P(x/2^{(k-1)/2}, y/2^{(k-1)/2}, 2^{k/2}) = P(x, y) * G(x, y, 2^{k/2})$$

Diagonal, Square root of two Sampling

Problem : How can we sample an image for for odd k? $\Delta x = 2^{k/2} = 2^{(k-1)/2}\sqrt{2}$

for k odd, $\Delta x_k = \{1, 2, 4, 8\ldots\}$
for k even, $\Delta x_k = \{\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, \ldots\}$

How ? with the diagonal sampling operator $S_{\sqrt{2}}\{\}$

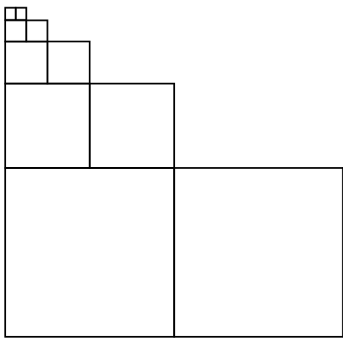


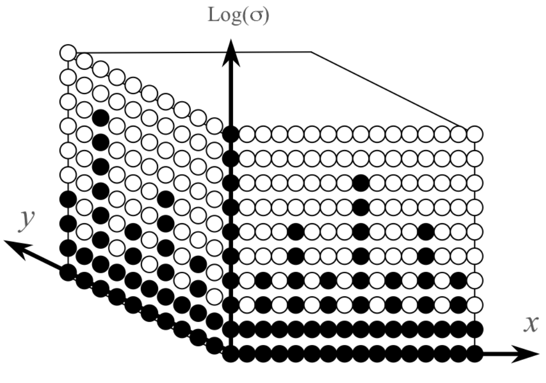
For k even, the $\sqrt{2}$ resampling operator, $S_{\sqrt{2}}{}^{k}\{\}$, selects even columns of even rows and odd columns of odd rows.

For k odd, diagonal sample operator eliminates every second column (starting with even columns on even rows and odd columns on odd rows). For k odd, resampling eliminates every second row (odd rows).

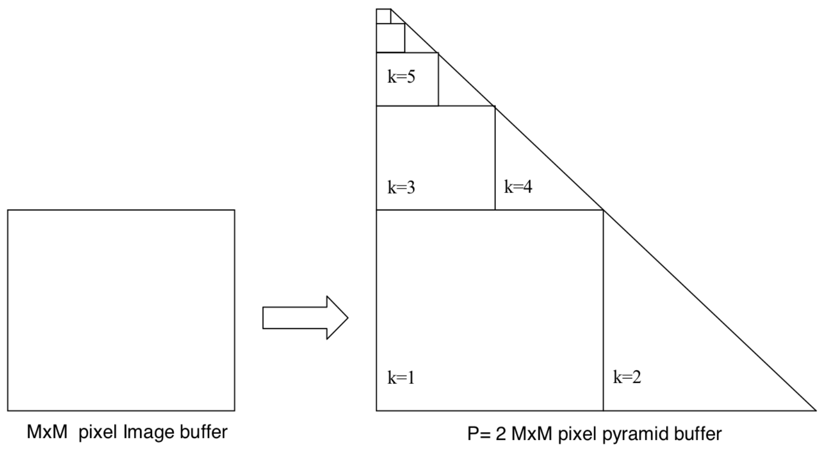
$$S_{\sqrt{2}^k}\{P\{x,y)\} = \begin{cases} P(x,y) & \text{if } (x+y)^2 \text{ Mod } 2^{k-1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

Data Structure





The even numbered images are diagonally sampled, eliminating half the pixels.



For an image of size MxM, number of pixels is

$$P = MxM \text{ x } (1 + ½ + ¼ + \ldots) = 2M^2$$

2 Scale Invariant Pyramid Algorithm

Cost of computing p(i,j,k) is

$$C= O(M^2((N_0+1)^2+(N_1+1)^2+(N_2+1)^2+\ldots+(N_{M-4}+1)^2))$$

if we use "seperable" convolution:

$$P(i,j) * G(i,j, 2^{k/2}) = P(i,j) * G(i, 2^{k/2}) * G(j, 2^{k/2})$$

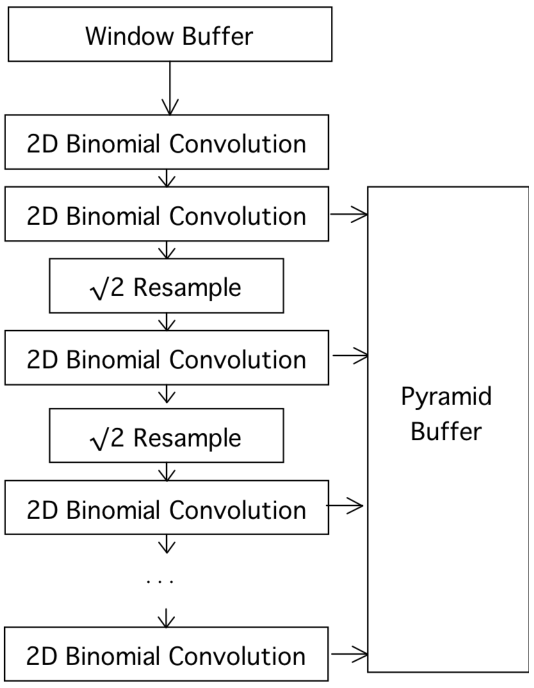
then

$$C= O(M^2\cdot 2(N_0+N_1+N_2+N_3+\ldots+N_{M-4}+M-4+1)$$

$$C= O(M^2\cdot 2(8+16+32+64+\ldots+ N_{M-4})+6).$$

Practically, the computational cost is exorbitant.

We can use Cascade Convolution Methods to reduce cost to O(N)

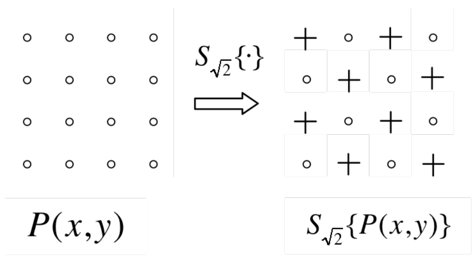


Each binomial convolution provides convolution with G(x,y,1).

Cumulative Variance is 1 2, 4, 8, 16, 32, 64,…
Cumulative standard deviation (σ) is $1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8, 8\sqrt{2}, 16, \ldots$

The cumulative Variance, Scale, and sample rates are:

k	$\sigma_k^2 = 2^k$	$\sigma_k = 2^{k/2}$	$\Delta x = 2^{\frac{k-1}{2}}$
1	2	$\sqrt{2}$	1
2	4	2	$\sqrt{2}$
3	8	$2\sqrt{2}$	2
4	16	4	$2\sqrt{2}$
5	32	$4\sqrt{2}$	4
6	64	8	$4\sqrt{2}$
7	128	$8\sqrt{2}$	8
8	256	16	$8\sqrt{2}$
9	512	$16\sqrt{2}$	16
10	1024	32	$16\sqrt{2}$
11	2048	$32\sqrt{2}$	32
12	4096	64	$32\sqrt{2}$
13	8192	$64\sqrt{2}$	64
14	16384	128	$64\sqrt{2}$
15	32768	$128\sqrt{2}$	128
16	65536	256	$128\sqrt{2}$

Note that odd levels are resampled on a $\sqrt{2}$ grid: 

Using the Pyramid to compute image derivatives

For an image P(i, j), the derivatives can be approximated by convolution with Derivatives of Gaussians

$$P_x(i,j) \approx P * G_x(i,j,\sigma)$$
$$P_y(i,j) \approx P * G_y(i,j,\sigma)$$
$$P_{xx}(i,j) \approx P * G_{xx}(i,j,\sigma)$$
$$P_{xy}(i,j) \approx P * G_{xy}(i,j,\sigma)$$
$$P_{yy}(i,j) \approx P * G_{yy}(i,j,\sigma)$$

Note: it is NECESSARY to specify σ. Small σ is not necessarily best.

With the Pyramid, derivatives can be obtained directly by sum and difference.

$$P_x(i,j,k) = < P(i,j), G_x(i,j,2^{k/2}) \approx P(i+1,j,k) - P(i-1,j,k)$$
$$P_y(i,j,k) = < P(i,j), G_x(i,j,2^{k/2}) \approx P(i,j+1,k) - P(i,j-1,k)$$
$$P_{xx}(i,j,k) = < P(i,j), G_{xx}(i,j,2^{k/2}) \approx P(i+1,j,k) - 2P(i,j,k) + P(i-1,j,k)$$
$$P_{yy}(i,j,k) = < P(i,j), G_{yy}(i,j,2^{k/2}) \approx P(i,j+1,k) - 2P(i,j,k) + P(i,j-1,k)$$

$$P_{xy}(i,j,k) = < P(i,j), G_{xy}(i,j,2^{k/2}) >$$
$$\approx P(i+1,j+1,k) - P(i-1,j+1,k) - P(i+1,j-1,k) + P(i-1,j-1,k)$$

The Gradient $\vec{\nabla} P(i,j)$ is calculated by $P(i,j) * \vec{\nabla} G(i,j,\sigma)$

where $\quad \vec{\nabla} G(i,j,\sigma) = \begin{pmatrix} G_x(i,j,\sigma) \\ G_y(i,j,\sigma) \end{pmatrix}$

Gradient: $\quad \vec{\nabla} P(i,j) = \begin{pmatrix} P_x(i,j) \\ P_y(i,j) \end{pmatrix} \approx \vec{\nabla}(P * G(i,j,\sigma)) = P * \vec{\nabla} G(i,j,\sigma) = \begin{pmatrix} P * G_x(i,j,\sigma) \\ P * G_y(i,j,\sigma) \end{pmatrix}$

Laplacien: $\nabla^2 P(i,j) = P * \nabla^2 G(i,j,\sigma) = P_{xx}(i,j) + P_{yy}(i,j) \approx P * G_{xx}(i,j,\sigma\ ) + P * G_{yy}(i,j,\sigma\ )$

For a pyramid

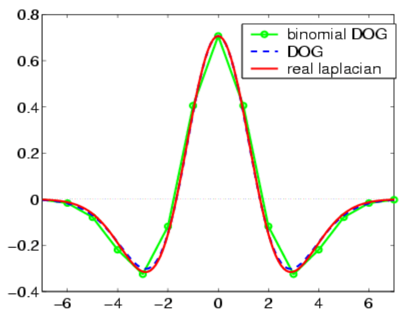
Diffusion Equation: $\quad \nabla^2 G_x(i,j,\sigma) = G_{xx}(i,j,\sigma) + G_{yy}(i,j,\sigma) = \dfrac{\partial G(i,j,\sigma)}{\partial \sigma}$

As a consequence: $\quad \nabla^2 G(i,j,\sigma) \approx G(i,j,\sigma_1) - G(i,j,\sigma_2)$

This typically requires $\quad \sigma_1 \geq \sqrt{2}\ \sigma_2$

Thus it is common to use:

$$\nabla^2 P(i,j,k) = <p(i,j), \nabla^2 G(i,j,\sigma_k)> \approx P(i,j,k) - P(i,j,k-1)$$



Oriented Derivatives

Gaussian Derivatives are Steerable:

$$G_1^{\theta}(x,y,\sigma) = \cos(\theta)\cdot G_x(x,y,\sigma) + \sin(\theta)\cdot G_y(x,y,\sigma)$$

Thus in the pyramid

1st order $\quad P_1^{\theta}(i,j,k) = Cos(\theta)P_x(i,j,k) + Sin(\theta)P_y(i,j,k)$

2nd order $\quad P_1^{\theta}(i,j,k) = Cos(\theta)^2 P_{xx}(i,j,k) + Sin(\theta)^2 P_{yy}(i,j,k) + 2Cos(\theta)Sin(\theta)P_{xy}(i,j,k)$

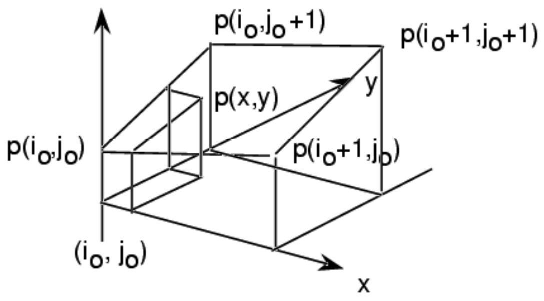
3rd order

$$P_3^{\theta}(i,j,k) = Cos(\theta)^3 P_{xxx}(i,j,k) + Cos(\theta)^2 Sin(\theta)P_{xxy}(i,j,k) + Cos(\theta)Sin(\theta)^2 P_{xyy}(i,j,k) + Sin(\theta)^3 P_{yyy}(i,j,k)$$

By steering the derivatives to the local orientation, we obtain an "invariant" measure of local contrast. We can also "steer" in scale to obtain invariance to size.

Note, we can NOT steer the mixed derivatives, i.e $P_{xy}(i,j,k)$

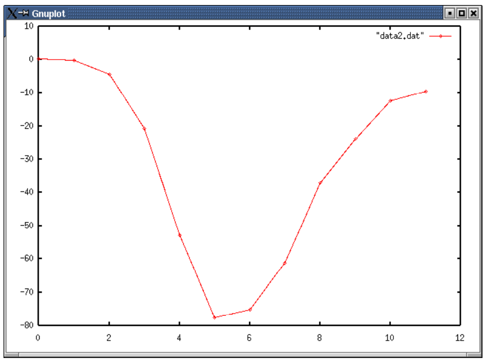
Interpolation



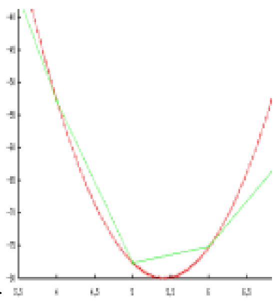
$$a = p(i_0+1, j_0) - p(i_0, j_0)$$
$$b = p(i_0, j_0+1) - p(i_0, j_0)$$
$$c = a + p(i_0, j_0+1) - p(i_0+1, j_0+1)$$

$$p(x,y) = a(x-i_0) + b(y-j_0) + c(x-i_0)(y-j_0) + p(i_0, j_0)$$

Scale Inerpolation



=>



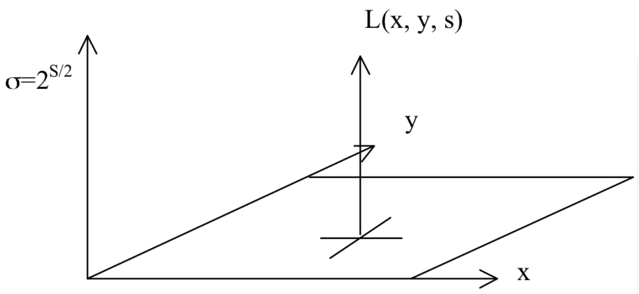
$$x_{\max} = x_2 + \frac{f(x_1) - f(x_3)}{2(f(x_1) + f(x_3) - 2f(x_2))}$$

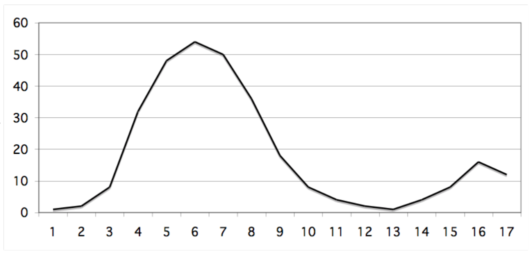
Laplacian Profile

At an every image point, the Laplacian profile the Laplacian of the image computed over a continuous (exponential) range of scales.

$$L(x, y, s) = P(x, y) * \nabla^2 G(x, y, 2^{s/2})$$

The Laplacian profile is invariant to rotation and equivariant to changes in distance.





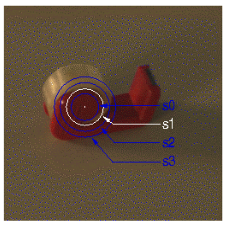
A change in viewing distance at x, y shifts the function L(x,y,s) in s.
The function remains the same. Thus the maximum is a local invariant.

The "intrinsic" scale at a point (x, y) is $\sigma_i = 2^{s_i/2}$

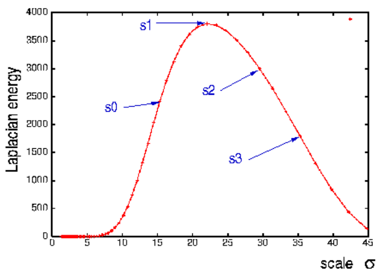
where:

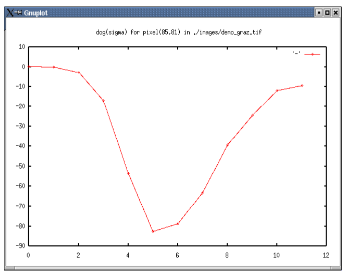
$$s_i = \arg-\max_{s}\{L(x,y,s)\}$$

Examples:



zero crossing of Laplacian at si



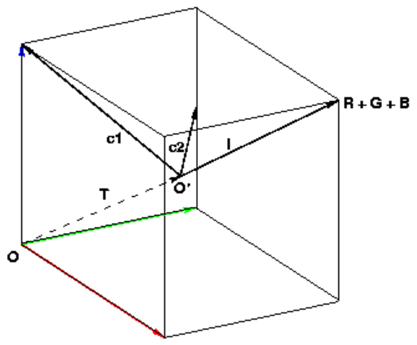


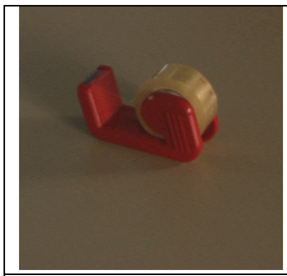
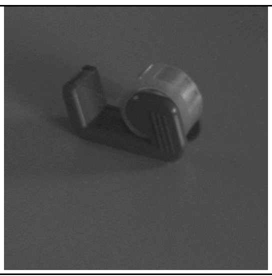
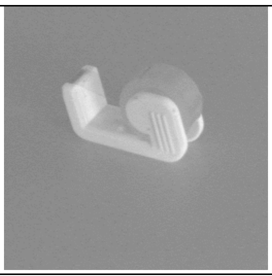
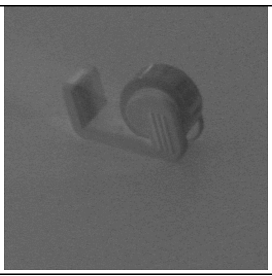
The scale of the maximal Laplacian is an invariant at ALL image points.

Color Opponent Scale Space

$$(R, G, B) \Rightarrow (L, C_1, C_2) \qquad \begin{pmatrix} L \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 0.33 & 0.33 & 0.33 \\ -0.5 & -0.5 & 1 \\ 0.5 & -0.5 & 0 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

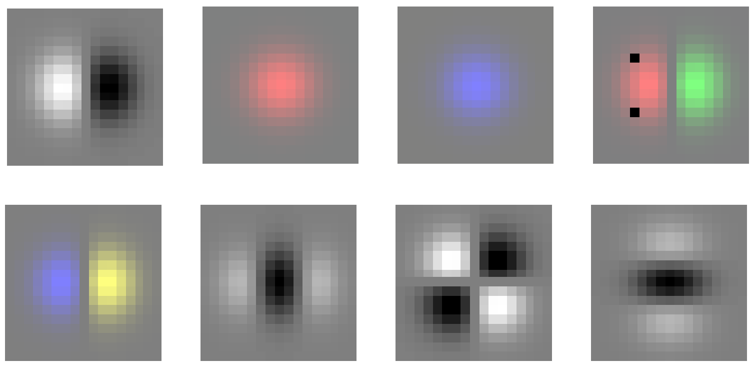
This representation separates luminance and chrominance.



				
RGB		B-W	R-G	R+G-B

This makes it possible to "steer" the chrominance to an illumination color

$$\begin{pmatrix} L \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 0.33 & 0.33 & 0.33 \\ -0.5 & -0.5 & 1 \\ 0.5 & -0.5 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 R \\ \alpha_2 G \\ \alpha_3 B \end{pmatrix}$$



We then compute 3 pyramids : L(i,j,k), C_1(i,j,k), and C_2(i,j,k),

This gives us a feature vector for appearance:

$$\vec{\varphi} = \begin{bmatrix} \varphi_0 \\ \vdots \\ \varphi_k \end{bmatrix} = \begin{bmatrix} G_x^L \\ G^{C_1} \\ G^{C_2} \\ G_x^{C_1} \\ G_x^{C_2} \\ G_{xx}^L \\ G_{xy}^L \\ G_{yy}^L \end{bmatrix}$$