Generalized Fractal Image Coding Using Projections Onto Convex Sets

Mehran Ebrahimi and Edward R. Vrscay
Department of Applied Mathematics
University of Waterloo

University of Waterloo

General Motivation

Today we talk about image approximations using long-range self-similarities, by some so-called fractal-based method!

An image is approximated by the fixed point of a contractive operator.

$$u \cong \bar{u} = T\bar{u}$$

In a complete metric space, Banach's Contraction Mapping Principle (CMP) guarantees the convergence of the iterative scheme $u_{n+1} = Tu_n$ hence, $u_n \longrightarrow \bar{u}$.

IFSM: A model that explains how to construct such contractive operator for images in practice.

$$u(x,y) \cong (Tu)(x,y) = \sum_i \phi_i(u(w_i^{-1}(x,y)))$$

$$\phi_i(t) = \alpha_i t + \beta_i$$

Questions

- Question 1.
  Is Banach's CMP the only recipe that guarantees the convergence in modeling the fractal-type self-similarities?

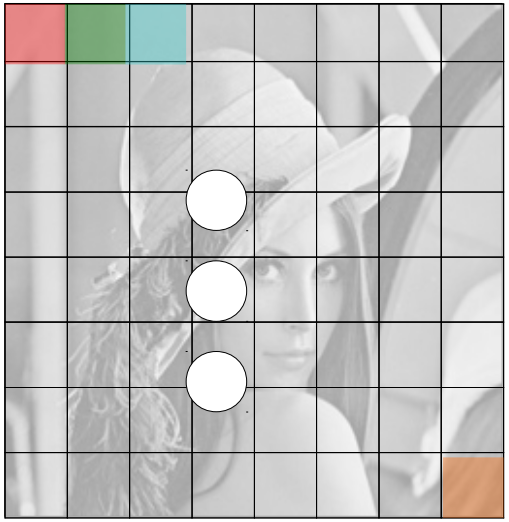
- Question 2.
  Is there a model that can also handle inexact fractal-type self-similarities?

- Question 3.
  Is there a model to "easily" incorporate additional knowledge about the image along with the fractal code?

Agenda

- Fractal Imaging - Brief Review
- Projections onto Convex-sets
  - How does it relate to fractal coding?
- Our Recent Experiments
  - Missing code

Fractal encoding example



Range blocks

Domain blocks

Range	Domain	Scale	Offset
1	J_1	α_1	β_1
2	J_2	α_2	β_2
3	J_3	α_3	β_3
⋮			
n	J_n	α_n	β_n

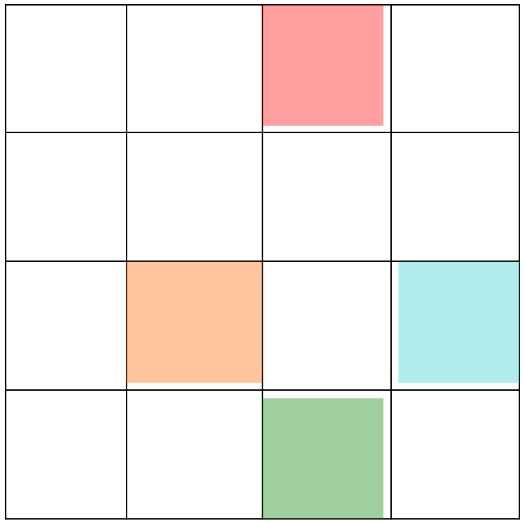
Given each range block R_i find a corresponding domain-block D_{J_i} and α_i, β_i

such that $\left\| R_i - [\alpha_i \; Downsampled \; (D_{J_i}) + \beta_i] \right\|$ is minimzed.

This is performed by searching and the least squares technique.

Fractal decoding example

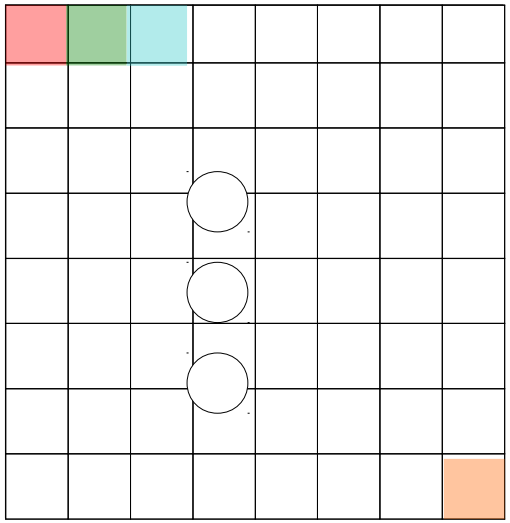
Range	Domain	Scale	Offset
1	J_1	α_1	β_1
2	J_2	α_2	β_2
3	J_3	α_3	β_3
	⋮		
n	J_n	α_n	β_n



Domain blocks

X_0

Iterate

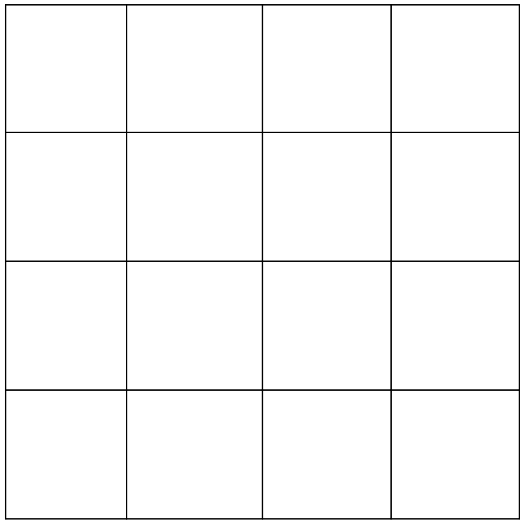


Range blocks

$X_1 = T(X_0)$

Fractal decoding example

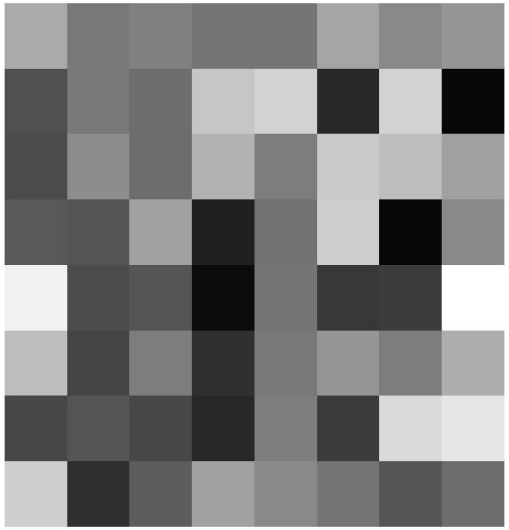
Range	Domain	Scale	Offset
1	J_1	α_1	β_1
2	J_2	α_2	β_2
3	J_3	α_3	β_3
	⋮		
n	J_n	α_n	β_n



Domain blocks

X_0

Iterate

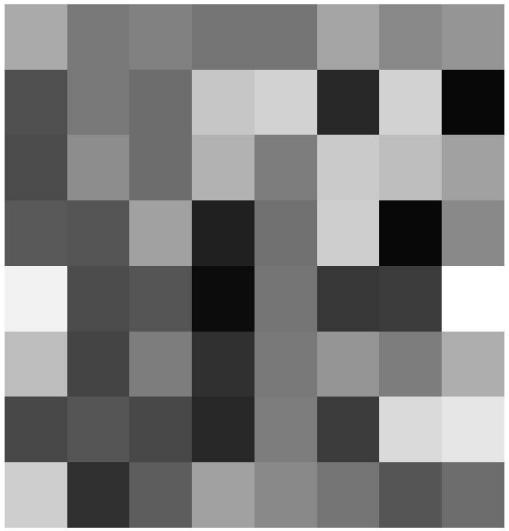


Range blocks

$X_1 = T(X_0)$

Fractal decoding example

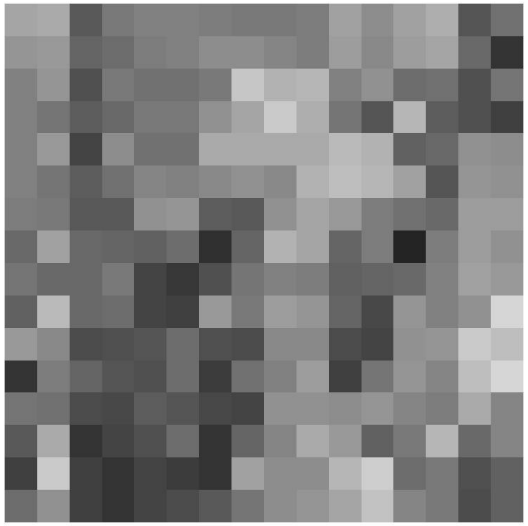
Range	Domain	Scale	Offset
1	J_1	α_1	β_1
2	J_2	α_2	β_2
3	J_3	α_3	β_3
	⋮		
n	J_n	α_n	β_n



Domain blocks

X_1

Iterate

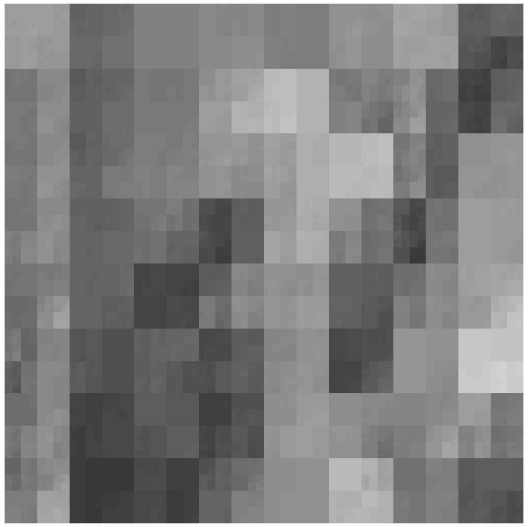


Range blocks

$X_2 = T(X_1)$

Fractal decoding example

Range	Domain	Scale	Offset
1	J_1	α_1	β_1
2	J_2	α_2	β_2
3	J_3	α_3	β_3
	⋮		
n	J_n	α_n	β_n



Fractal decoding example

Finer partition
Larger code size

16x16 range block grid
each block of size 16x16

32x32 range block grid
each block of size 8x8

64x64 range block grid
each block of size 4x4

Some Improvements in Fractal Coding

- Different partitionings
  - Quadtree, HV (Y. Fisher, 1994)
- Faster encoding
  - Complexity reduction using classification techniques (Oien, 1993, R. Hamzaoui, D. Saupe, 1998)
- Redundant representation
  - Multiple-domain (W. Xu, D. Fussell, 1996, S. Alexander 2005)
  - Lapped approximations with overlapping range blocks (E. Reusens, 1994, H.L. Ho, W.K. Cham, 1997)
- Multiscale representations
  - IFSW (B. Forte, E.R. Vrscay, 1995)
- ...

Agenda

- Fractal Imaging - Brief Review
- Projections Onto Convex-sets
  - How does it relate to fractal coding?
- Our Recent Experiments
  - Missing code

Projections Onto Convex Sets (POCS)

- Serial (Periodic) Projections

  If the sets $C_i,\ i = 1, 2, \ldots, m$ are closed and convex with non-empty intersection the sequence

  $$X^{(n+1)} = P_m \ldots P_2 P_1 X^{(n)}$$

  converges to a point in their intersection, where P_i is the projection onto each C_i . [D. C. Youla, H. Webb, 1982]

- Parallel Projections Method (PPM)

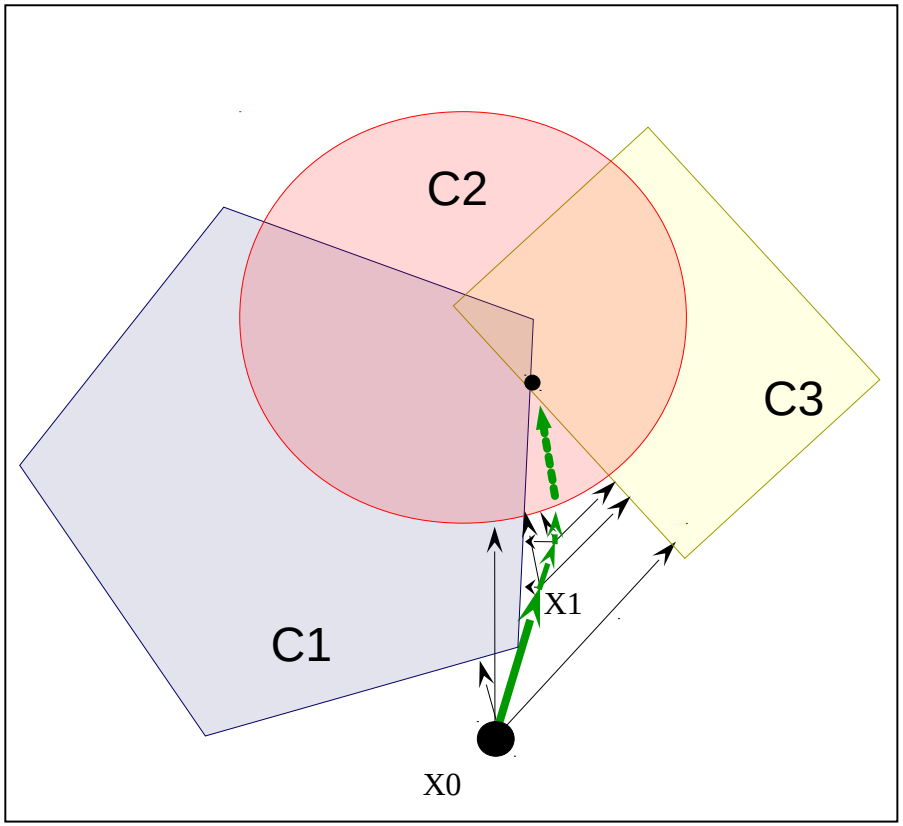
  Suppose that at least one of the C_i's is bounded and $G \neq \emptyset$ is the set of minimizers of the proximity function

  $$\Psi(X) = \frac{1}{2}\sum_{i=1}^{m} w_i d(X, C_i)^2 \quad \text{where,} \quad \sum_{i=1}^{m} w_i = 1 \text{ and } w_i > 0 \ .$$

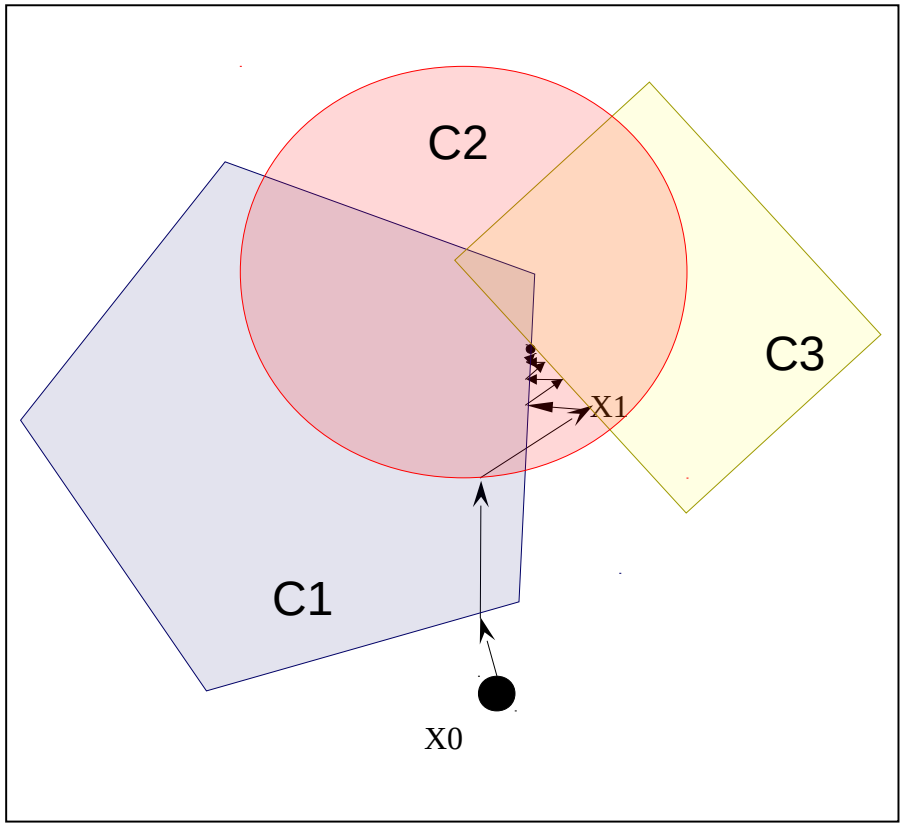
  Then, $$X^{(n+1)} = X^{(n)} + \lambda_n \Big(\sum_{i=1}^{m} w_i P_i(X^{(n)}) - X^{(n)} \Big) \ , \ \{\lambda_n\} \subset [\epsilon, 2-\epsilon] \ , \ 0 < \epsilon < 1 \ .$$

  converges to an element of G. [P. L. Combettes 1994]

Projections Onto Convex Sets (POCS)



Parallel



Serial

Collage Errors and Associated Sets

$$u(x,y) \cong (Tu)(x,y) = \sum_i \phi_i(u(w_i^{-1}(x,y)))$$

$$\phi_i(t) = \alpha_i t + \beta_i$$

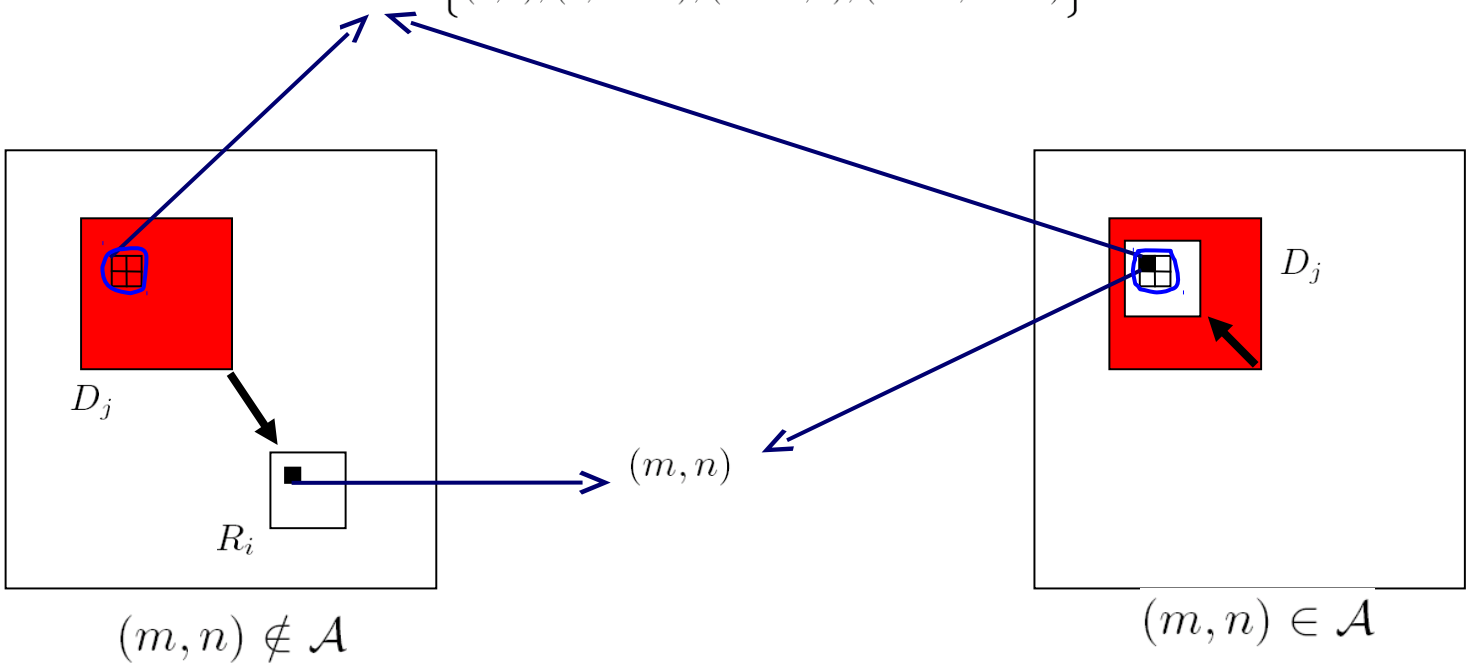
Type	Collage distance	Associated set
Global	$\Delta(u) = \left\|(Tu) - u\right\|_p$	$\Psi = \left\{u \in L^p(X) : \Delta(u) \leq \delta\right\}$
Block-wise	$\Delta^{(k)}_{(i,j)}(u) = \left\|\phi_i\left(u\left(w^{-1}_{i,j,(k)}(R_i)\right)\right) - u(R_i)\right\|_p$	$\Psi^{(k)}_{(i,j)} = \left\{u \in L^p(X) : \Delta^{(k)}_{(i,j)}(u) \leq \delta^{(k)}_{(i,j)}\right\}$
Point-wise	$\Delta^{(k)}_{(i,j)(x,y)}(u) = \left\|\phi\left(u\left(w^{-1}_{i,j,(k)}(x,y)\right)\right) - u(x,y)\right\|$	$\Psi^{(k)}_{(i,j)(x,y)} = \left\{u \in L^p(X) : \Delta^{(k)}_{(i,j)(x,y)}(u) \leq \delta^{(k)}_{(i,j)(x,y)}\right\}$

The idea of using these types of constraints in coding was mentioned by H. Puh and P. L. Combettes, “Operator theoretic image coding”, 1996.

Discrete Point-wise Collage Constraints

$$\Psi^{(k)}_{(i,j)(m,n)} = \left\{ U \in l^2(X) : \left| \tfrac{\alpha}{4}\{U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\} + \beta - U(m,n) \right| \leq \delta^{(k)}_{(i,j)(m,n)} \right\}$$

$$\mathcal{A} = \left\{ (s,t), (s,t+1), (s+1,t), (s+1,t+1) \right\}$$



Coding and Decoding

$$\Psi^{(k)}_{(i,j)(m,n)} = \left\{ U \in l^2(X) : \left| \tfrac{\alpha}{4} \{ U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1) \} + \beta - U(m,n) \right| \leq \delta^{(k)}_{(i,j)(m,n)} \right\}$$

- **Code Ingredients**
  - Some or all of the above constraints.
  - Any additional suitable convex and closed constraint.
  - Assumption: The projections on the additional constraints are easy to compute.

- **Decoding**
  - If the intersection of all of the constraints is non-empty, use the serial POCS.
  - Otherwise, use a suitable parallel POCS algorithm.

Coding and Decoding

$$\Psi^{(k)}_{(i,j)(m,n)} = \Big\{ U \in l^2(X) : \Big| \tfrac{\alpha}{4} \{ U(s,t) + U(s,t \quad + 1) \} + \quad (m,n) \Big| \le \delta^{(k)}_{(i,j)(m,n)} \Big\}$$

- **Code Ingred**
  -
  - A easy to compute.
- **Deco**
  - s is non-empty, use the serial POCS.
  - Otherwis ilabl rallel POCS algorithm.

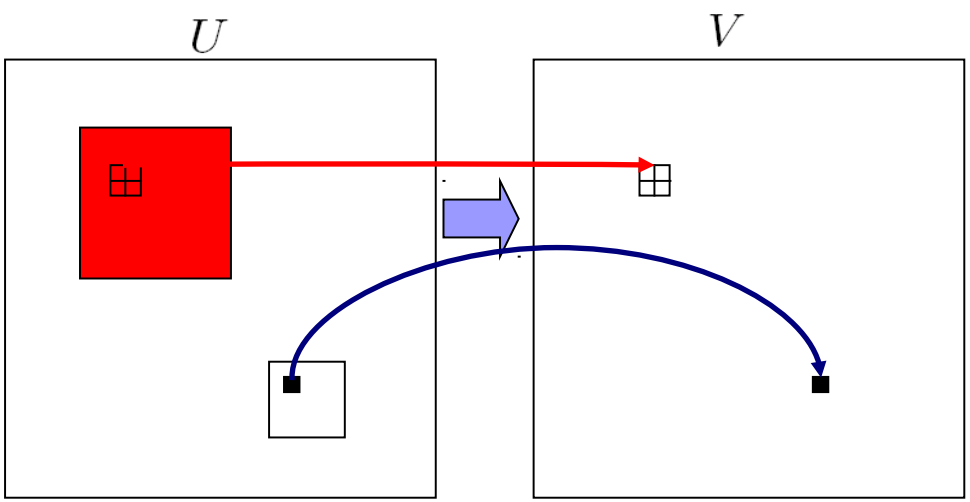
A fractal code is just a set of "constraints" with a unique intersection!

Computing Projections: Case $(m,n) \notin \mathcal{A}$

$$\Psi^{(k)}_{(i,j)(m,n)} = \left\{ U \in l^2(X) : \left| \tfrac{\alpha}{4}\{U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\} + \beta - U(m,n) \right| \leq \delta^{(k)}_{(i,j)(m,n)} \right\}$$

$$r = \frac{\alpha}{4}\Big[U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\Big] + \beta - U(m,n)$$

$$\delta = \delta^{(k)}_{(i,j)(m,n)} \geq 0$$



Projection of U on $\Psi^{(k)}_{(i,j)(m,n)}$

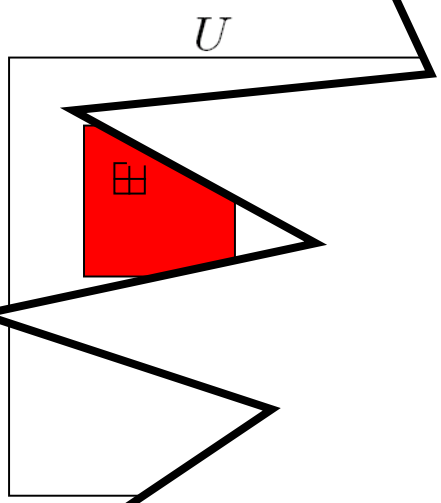
$$V(p,q) = U(p,q) + \begin{cases} \text{Case } (p,q) \in \mathcal{A} \\ \frac{-\alpha/4}{1+4(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{-\alpha/4}{1+4(\alpha/4)^2}(r+\delta), & r < -\delta \\ \text{Case } (p,q) = (m,n) \\ \frac{1}{1+4(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{1}{1+4(\alpha/4)^2}(r+\delta), & r < -\delta \\ \text{Case } (p,q) \notin \mathcal{A} \cup \{(m,n)\} \\ 0. \end{cases}$$

Computing Projections: Case $(m,n) \notin \mathcal{A}$

$$\Psi^{(k)}_{(i,j)(m,n)} = \left\{ U \in l^2(X) : \left| \tfrac{\alpha}{4}\{U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\} + \beta - U(m,n) \right| \le \delta^{(k)}_{(i,j)(m,n)} \right\}$$

$$r = \frac{\alpha}{4}\Big[U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\Big] + \beta - U(m$$

$$\delta = \delta^{(k)}_{(i,j)(m,n)} \ge 0$$



Pixels in the range and domain blocks are alternating simultaneously !

$(p,q) \in \mathcal{A}$

$\frac{\alpha/4}{1+4(\alpha/4)^2}(r-\delta), \quad r > \delta$

$0, \quad |r| \le \delta$

$\frac{\alpha/4}{(\alpha/4)^2}(r+\delta), \quad r < -\delta$

Case $(p,q) = (m,n)$

$\frac{1}{(\alpha/4)^2}(r-\delta), \quad r > \delta$

$|r| \le \delta$

$\frac{1}{1+4(\alpha/4)^2}(r+\delta), \quad r < -\delta$

Case $(p,q) \notin \mathcal{A} \cup \{(m,n)\}$

$0.$

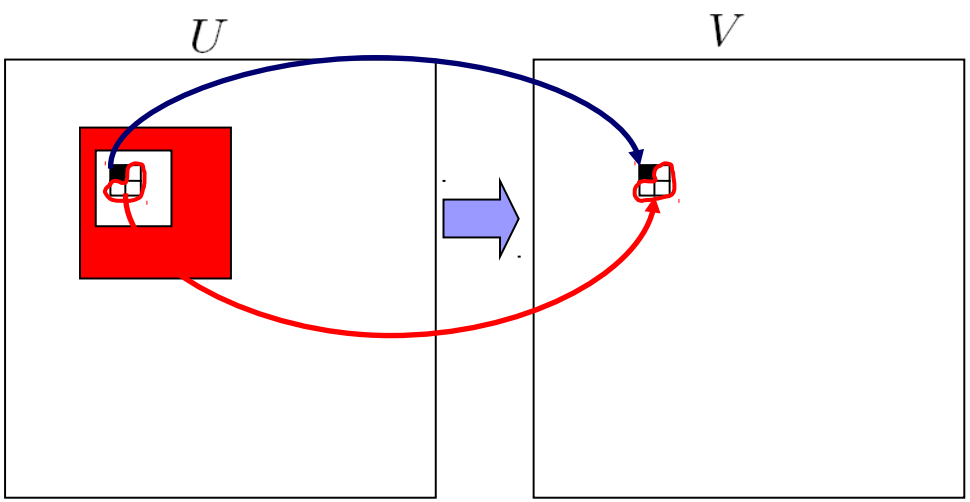
ction o on $\Psi^{(k)}_{(i,j)(m,n)}$

Computing Projections: Case $(m,n) \in \mathcal{A}$

$$\Psi^{(k)}_{(i,j)(m,n)} = \left\{ U \in l^2(X) : \left| \tfrac{\alpha}{4}\{U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\} + \beta - U(m,n) \right| \le \delta^{(k)}_{(i,j)(m,n)} \right\}$$

$$r = \frac{\alpha}{4}\Big[U(s,t) + U(s,t+1) + U(s+1,t) + U(s+1,t+1)\Big] + \beta - U(m,n)$$

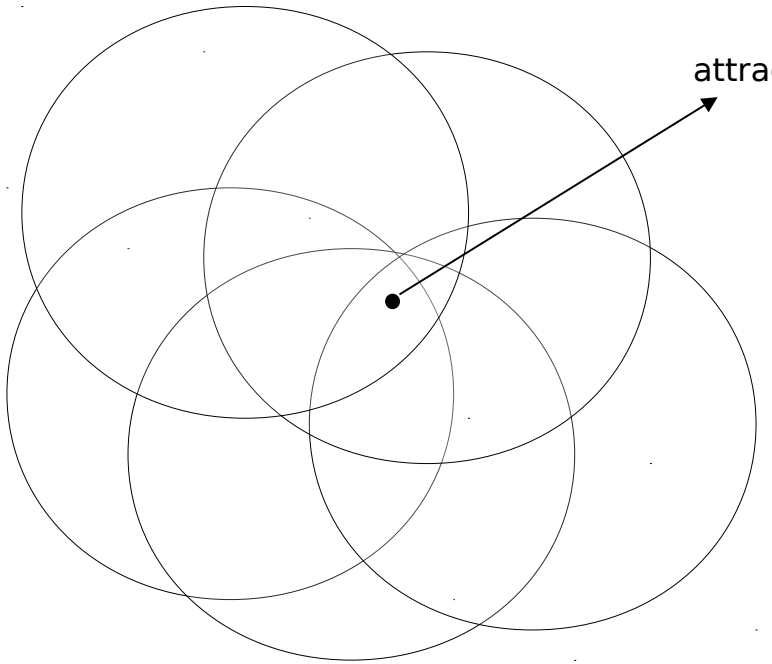
$$\delta = \delta^{(k)}_{(i,j)(m,n)} \ge 0$$



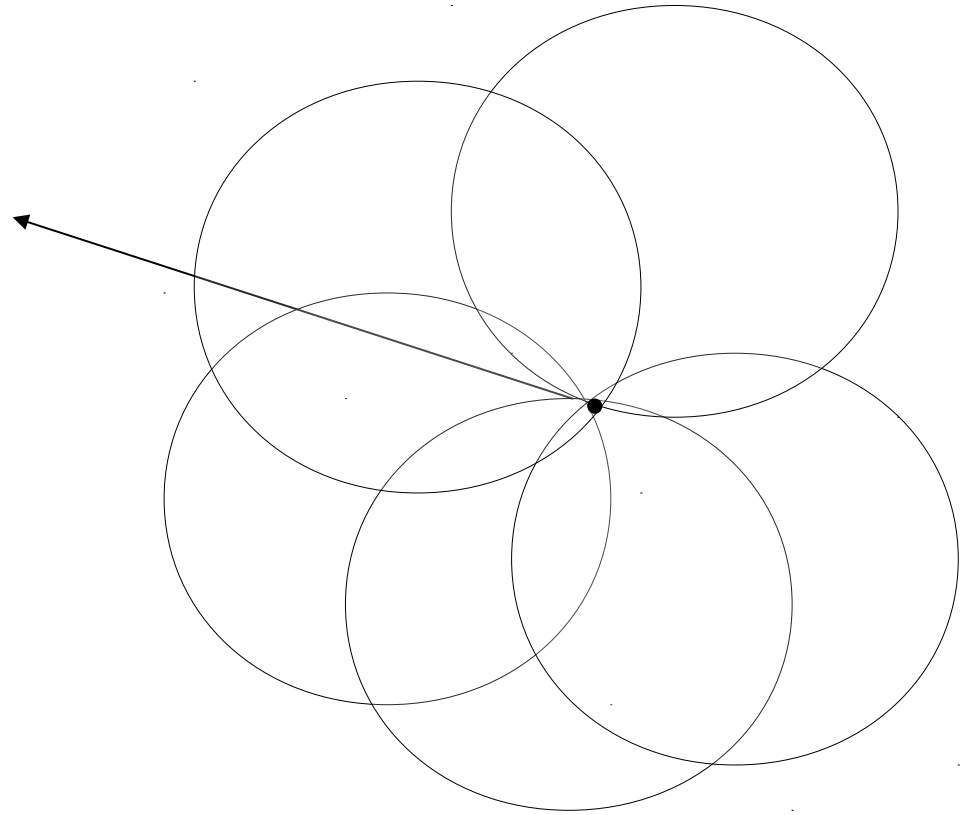
Projection of U on $\Psi^{(k)}_{(i,j)(m,n)}$

$$V(p,q) = U(p,q) + \begin{cases} \text{Case } (p,q) \in \mathcal{A} \setminus \{(m,n)\} \\ \frac{-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \le \delta \\ \frac{-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r+\delta), & r < -\delta \\ \text{Case } (p,q) = (m,n) \\ \frac{1-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \le \delta \\ \frac{1-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r+\delta), & r < -\delta \\ \text{Case } (p,q) \notin \mathcal{A} \\ 0. \end{cases}$$

A schematic view of constraints and the IFS attractor



In case of positive deltas, the intersection of all of the constraints is not unique, the attractor is a point in this intersection.

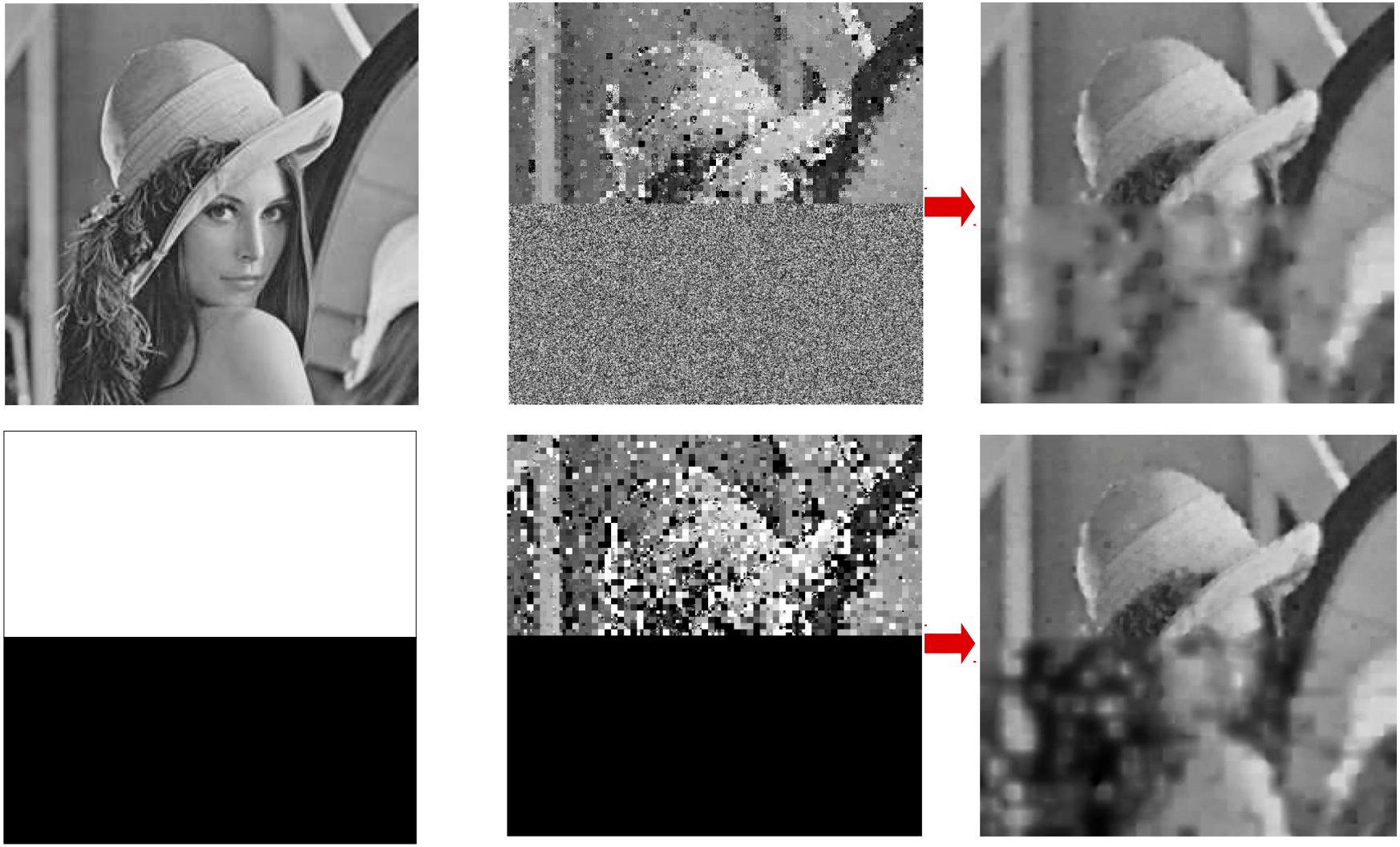


If all deltas are zeros, the intersection of the constraints is unique, and is the attractor of the IFS.

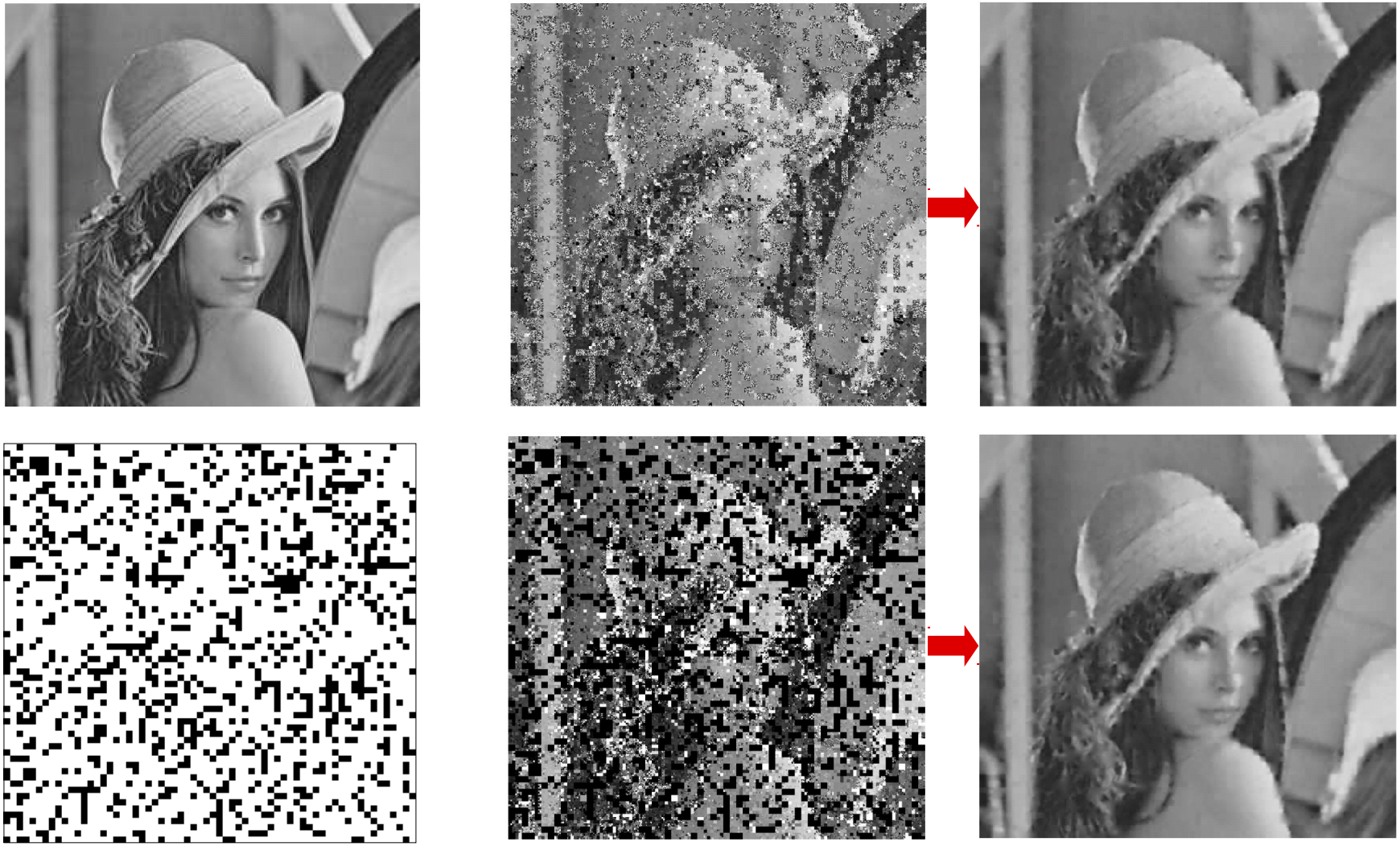
Agenda

- Fractal Imaging - Brief Review
- Projections Onto Convex-sets
  - How does it relate to fractal coding?
- Our Recent Experiments
  - Missing code

Incomplete Fractal Code 1



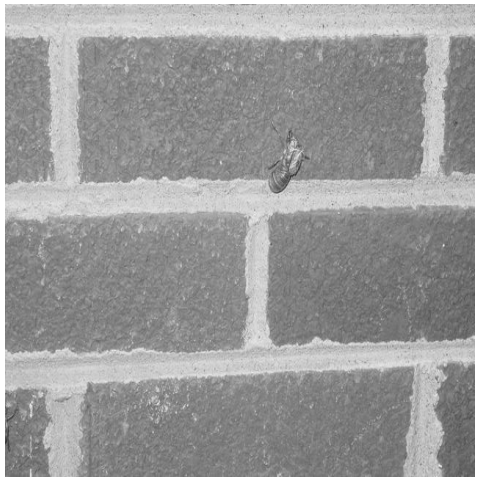
Incomplete Fractal Code 2



Other Potentials of the Method

De-noising?

Zooming?



Reducing blocking artifacts simultaneous with decoding?

Conclusions

- Question 1.
  Is Banach's CMP the only recipe that guarantees the convergence in modeling the fractal-type self-similarities?
  Answer: NO

- Question 2.
  Is there a model that can also handle inexact fractal-type self-similarities?
  Answer: YES

- Question 3.
  Is there a model to "easily" incorporate additional knowledge about the image along with the fractal code?
  Answer: YES

A primitive attempt was presented to introduce a more "flexible" view of the traditional fractal image coding!