POCS:

A Tool for Signal Synthesis & Analysis

Robert J. Marks II,
Distinguished Professor of Electrical and Computer Engineering
Baylor University
r.marks@ieee.org
http://RobertMarks.org/

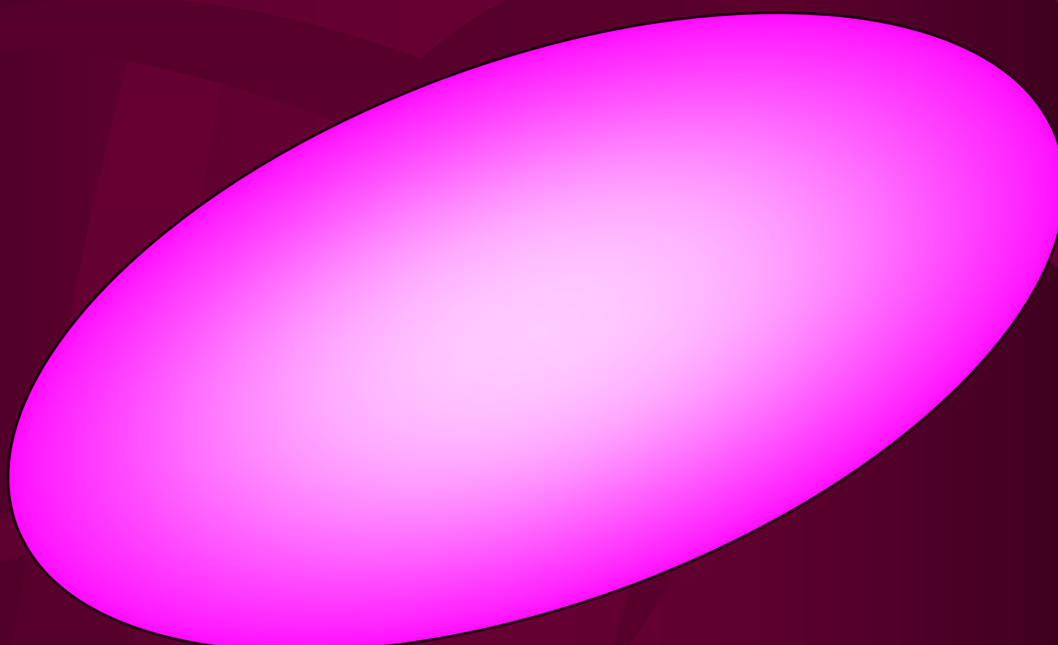
Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

POCS: What is a convex set?

In a vector space, a set C is convex iff $\forall\ \vec{x} \in C$ and $\vec{y} \in C$,

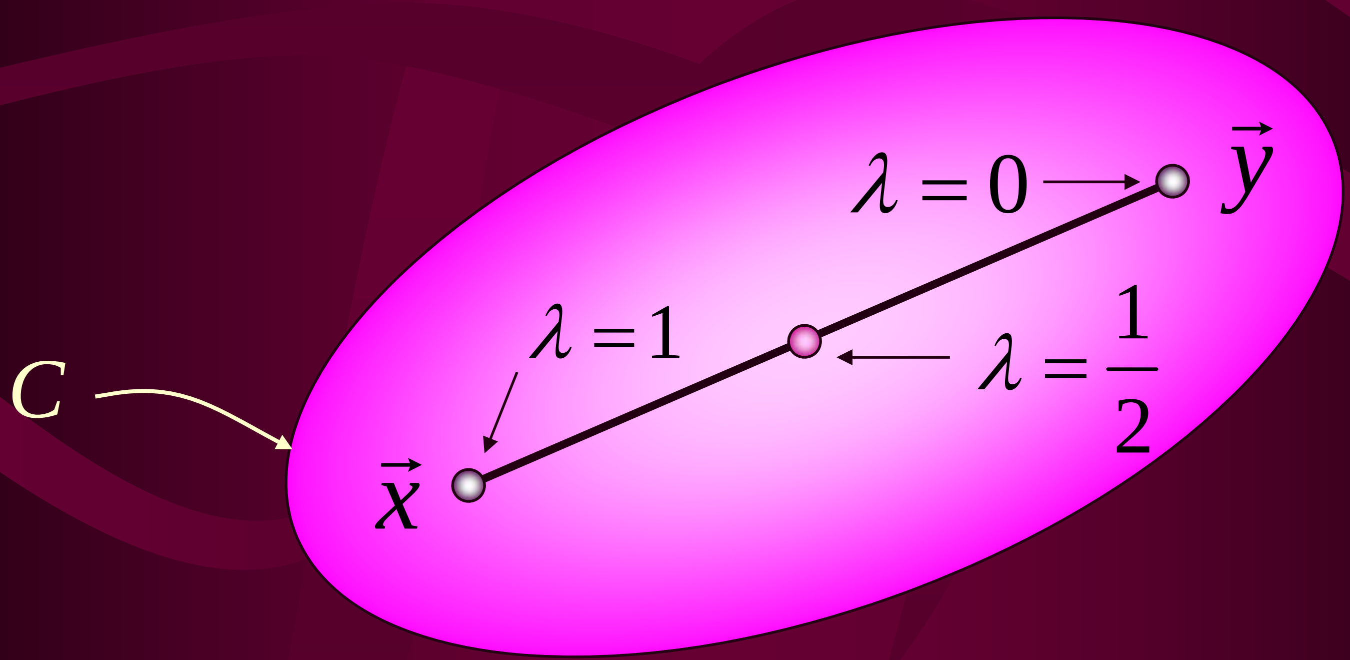
$$\Rightarrow \lambda\vec{x} + (1-\lambda)\vec{y} \in C \quad \forall \quad 0 \le \lambda \le 1$$



POCS: What is a convex set?

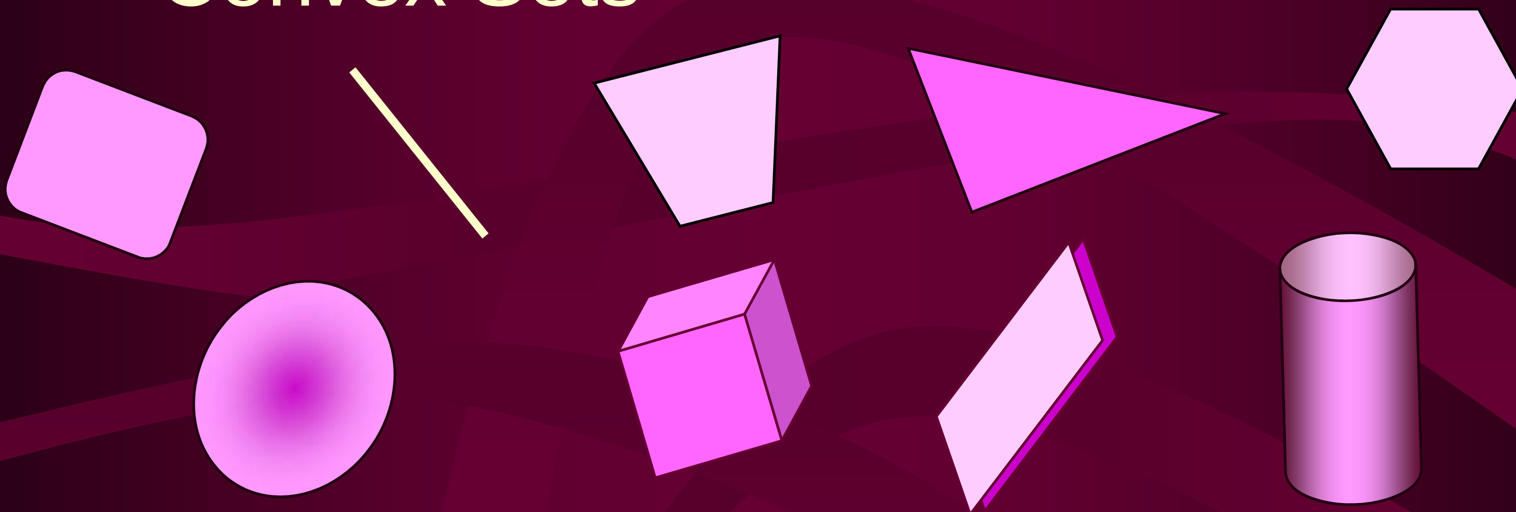
In a vector space, a set C is convex iff $\forall\ \vec{x} \in C$ and $\vec{y} \in C$,

$$\Rightarrow \lambda\vec{x} + (1-\lambda)\vec{y} \in C \quad \forall \quad 0 \le \lambda \le 1$$

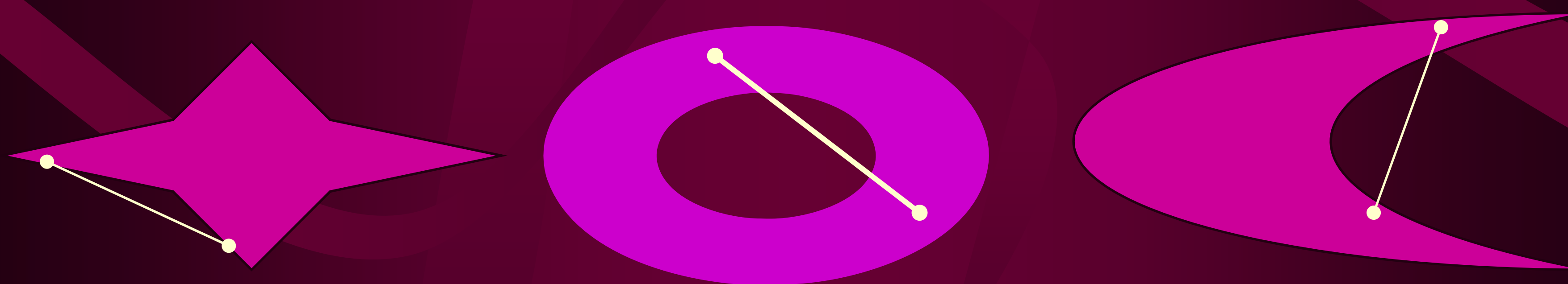


POCS: Example convex sets

- Convex Sets

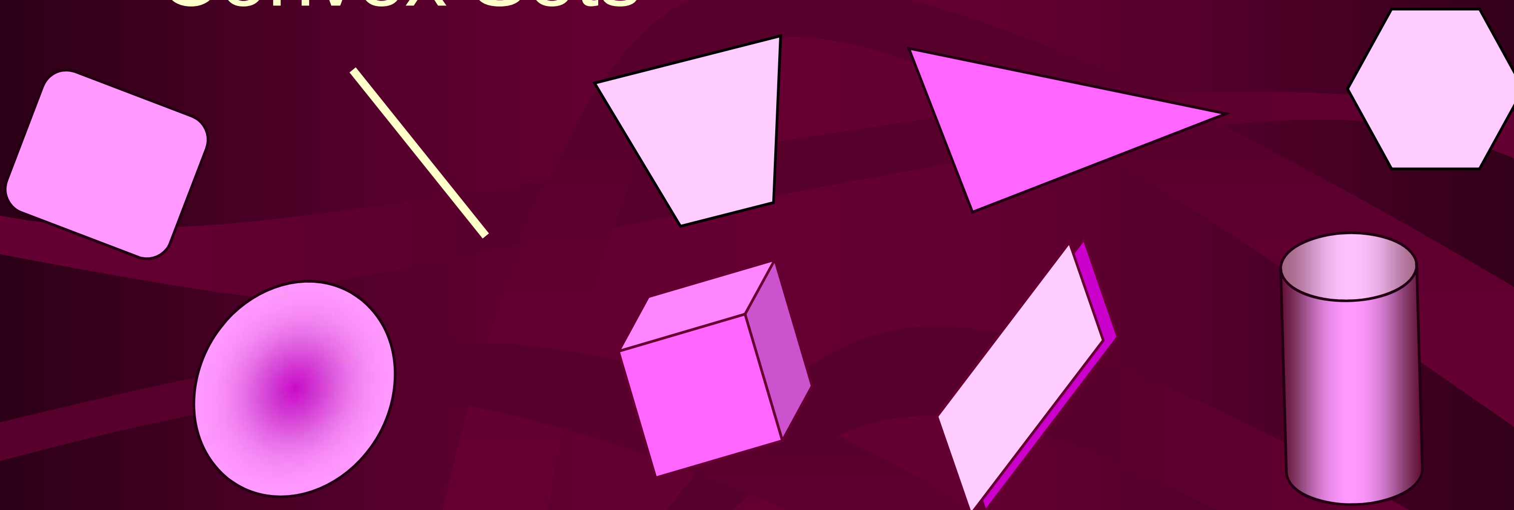


- Sets Not Convex

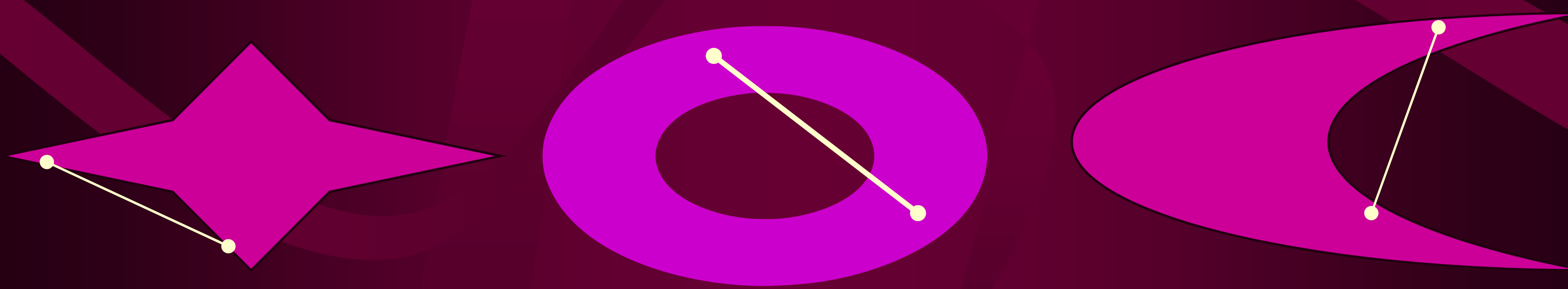


POCS: Example convex sets

- Convex Sets



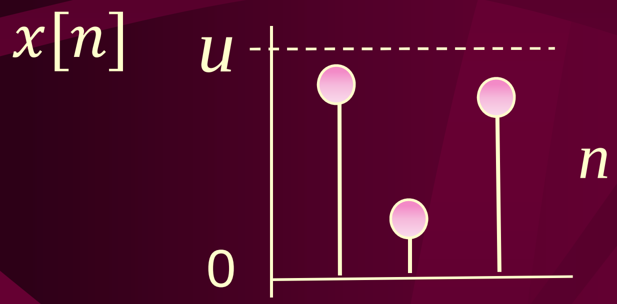
- Sets Not Convex



POCS: Example convex sets

- **Bounded Signals** – a box

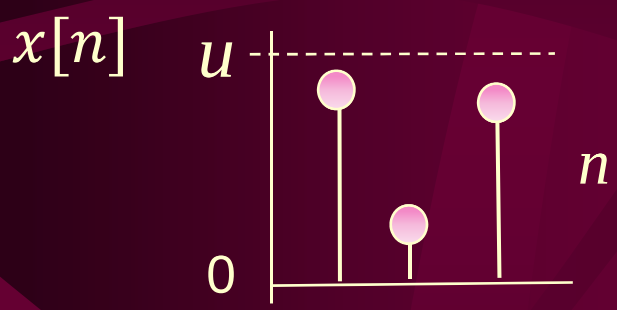
$$C = \{ x[n] \mid 0 \le x[n] \le u \} \quad ; n = 1,2,3$$

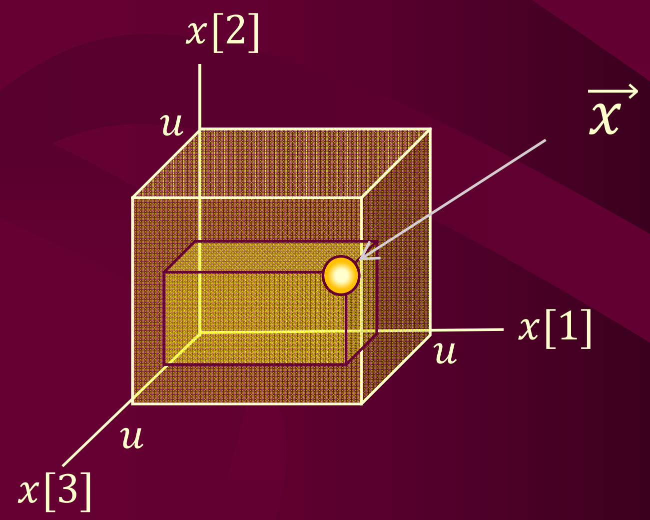


POCS: Example convex sets

- **Bounded Signals** – a box

$$C = \{x[n] \mid 0 \leq x[n] \leq u\} \quad ; n = 1,2,3$$

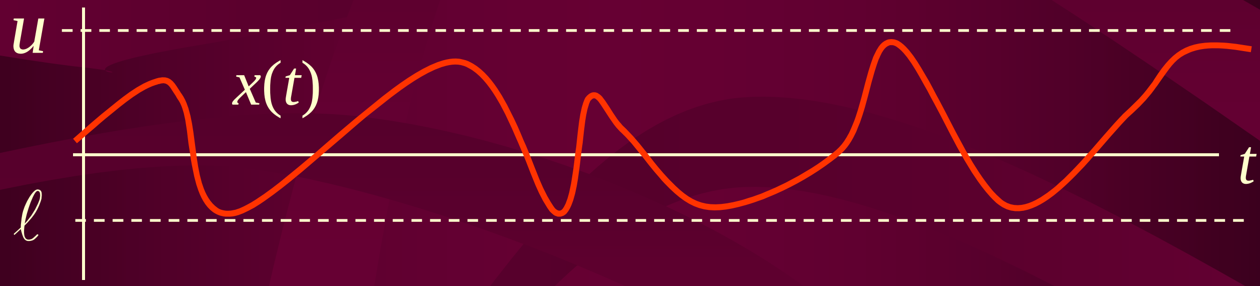




POCS: Example convex sets in a Hilbert space

- **Bounded Signals** – a box

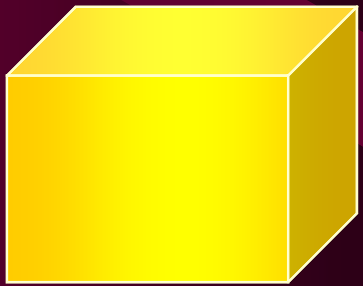
$$C = \{ x(t) \mid \ell \le x(t) \le u \}$$



- If $\ell \le x(t) \le u$ and $\ell \le y(t) \le u$,

  then, if $0 \le \lambda \le 1$

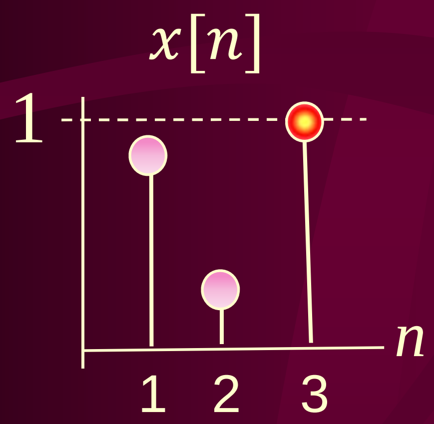
  $$\ell \le \lambda\, x(t) + (1 - \lambda)\, y(t) \le u.$$



POCS: Example convex sets in a Hilbert space

- **Identical Middles** – a plane

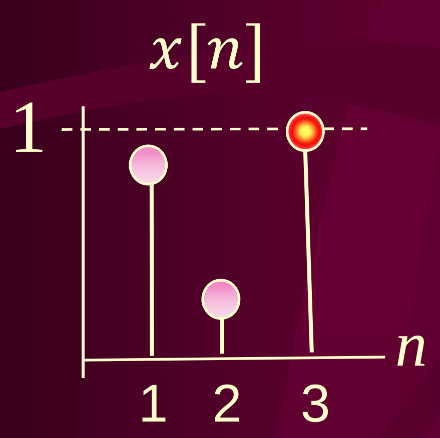
$$C = \left\{ x[n] \,\middle|\, x[3] \equiv 1 \right\},\ n = 1,2,3$$

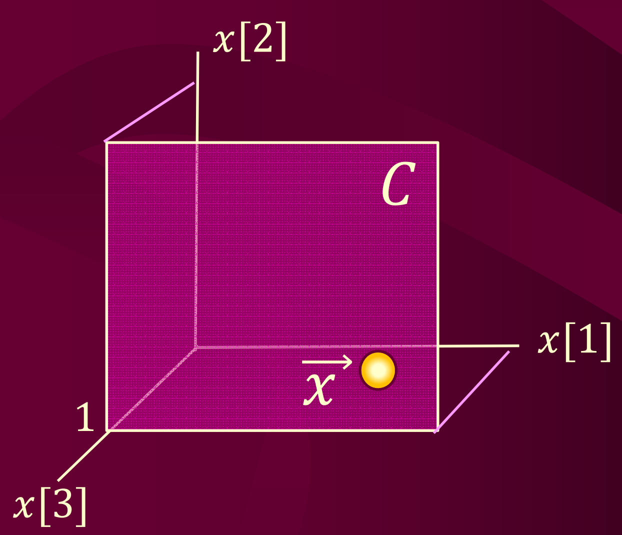


POCS: Example convex sets in a Hilbert space

- **Identical Middles** – a plane

$$C = \left\{ x[n] \,\middle|\, x[3] \equiv 1 \right\},\; n = 1,2,3$$

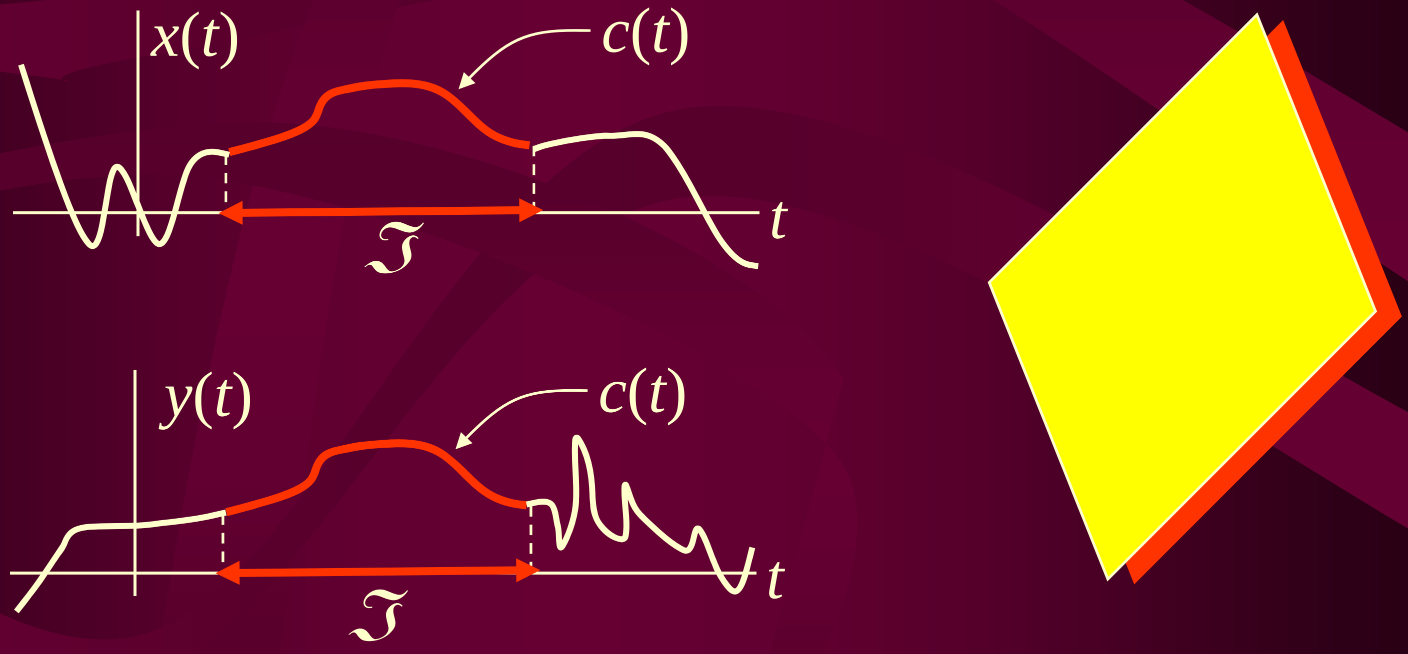




POCS: Example convex sets in a Hilbert space

- **Identical Middles** – a plane

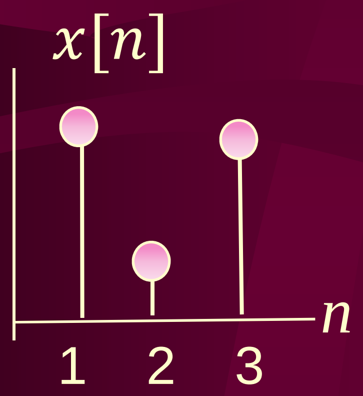
$$C = \left\{ x(t) \,\middle|\, x(t) \equiv c(t) \;;\; t \in \mathfrak{I} \right\}$$



POCS: Example convex sets in a Hilbert space

- **Fixed Area Signals** – a plane

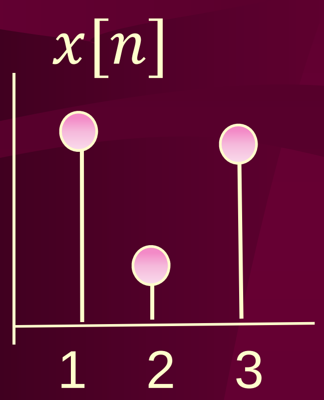
$$C = \left\{ x[n] \,\middle|\, x[1] + x[2] + x[3] = A \right\}$$

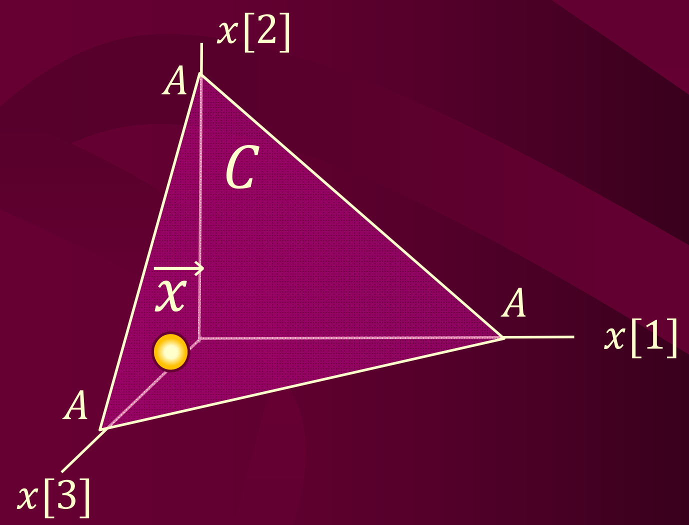


POCS: Example convex sets in a Hilbert space

- **Fixed Area Signals** – a plane

$$C = \left\{ x[n] \,\middle|\, x[1] + x[2] + x[3] = A \right\}$$





POCS: Example convex sets in a Hilbert space

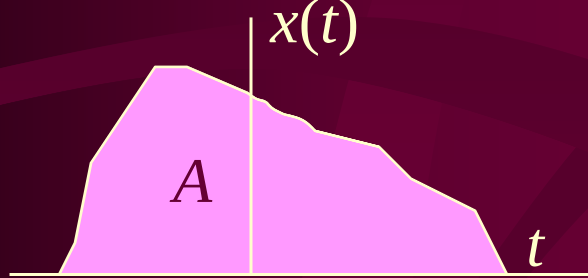
- **Fixed Area Signals** – a plane

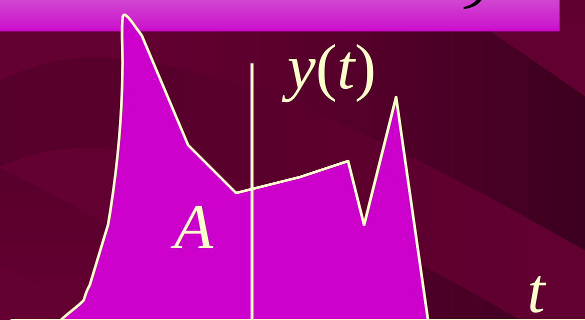
$$C = \left\{ x(t) \,\middle|\, \int_{-T}^{T} x(t) = A \right\}$$

POCS: Example convex sets in a Hilbert space

- **Fixed Area Signals** – a plane

$$C = \left\{ x(t) \,\middle|\, \int_{-T}^{T} x(t) = A \right\}$$





$$\int_{-T}^{T} \left[\lambda x(t) + (1-\lambda) y(t) \right] dt = \lambda A + (1-\lambda) A = A$$

POCS: Example convex sets in a Hilbert space

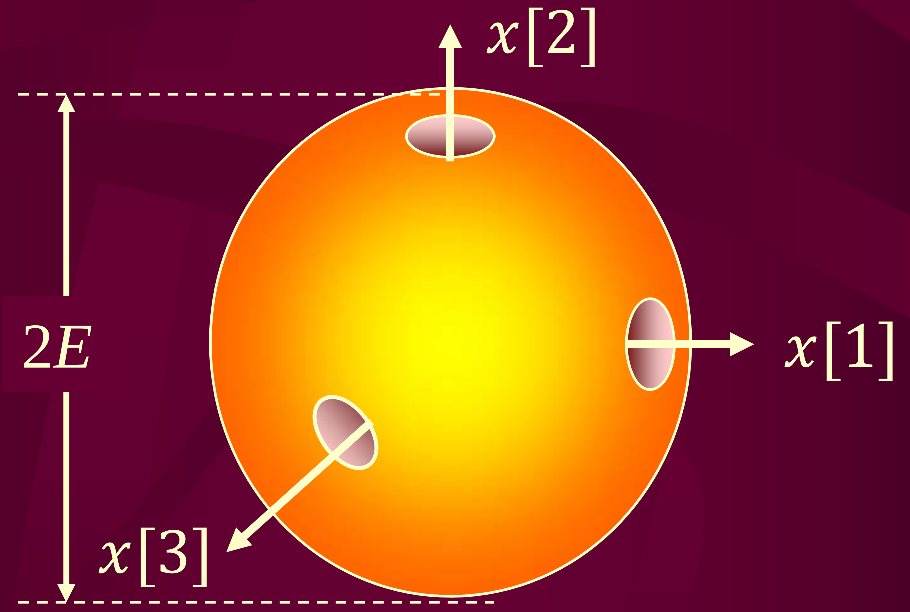
- **Bounded Energy Signals** – a ball

$$C = \left\{ x[n] \middle| x^2[1] + x^2[2] + x^2[3] \le E \right\}$$

POCS: Example convex sets in a Hilbert space

- **Bounded Energy Signals** – a ball

$$C = \left\{ x[n] \middle| x^2[1] + x^2[2] + x^2[3] \le E \right\}$$

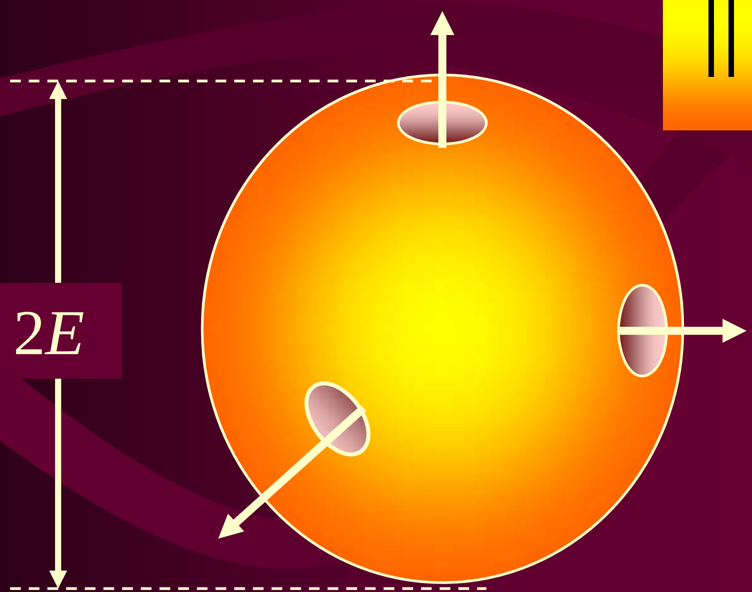


POCS: Example convex sets in a Hilbert space

- **Bounded Energy Signals** – a ball

$$C = \left\{ x(t) \,\middle|\, \| x(t) \|^2 \leq E \right\}$$

$$\| x(t) \|^2 = \int_{-\infty}^{\infty} \left| x(t) \right|^2 dt$$



POCS: Example convex sets in a Hilbert space

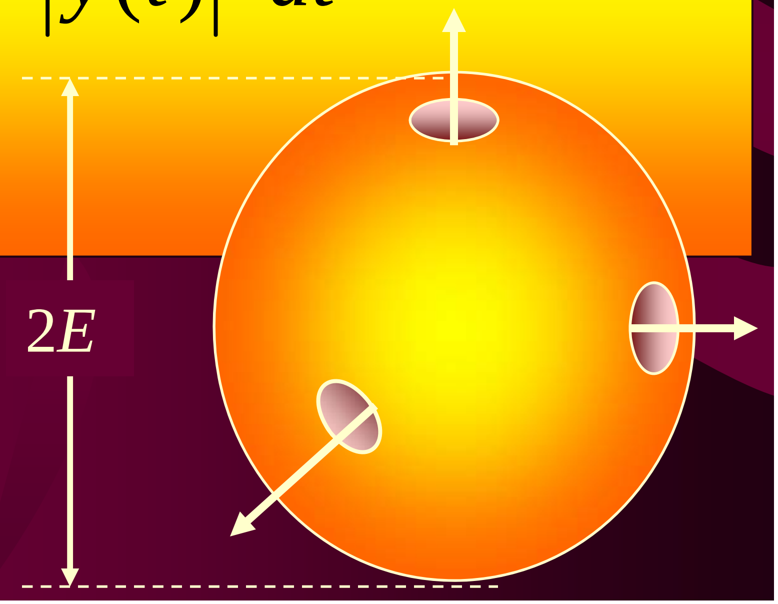
- **Bounded Energy Signals** – a ball

$$\| \lambda x(t) + (1-\lambda) y(t) \|^2 = \int_{-\infty}^{\infty} \left| \lambda x(t) + (1-\lambda) y(t) \right|^2 dt$$

Triangle Inequality

$$\leq \lambda^2 \int_{-\infty}^{\infty} |x(t)|^2 dt + (1-\lambda)^2 \int_{-\infty}^{\infty} |y(t)|^2 dt$$

$$\leq \left[\lambda^2 + (1-\lambda)^2 \right] E \leq E$$



POCS: Example convex sets in a Hilbert space

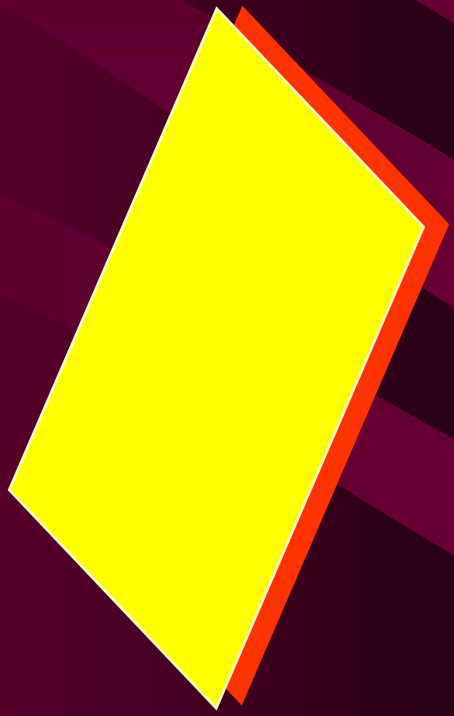
- **Bandlimited Signals** – a plane (subspace)

$$C = \left\{ x(t) \middle| X(u) \equiv 0 \; ; \; u > B \right\}$$

B = bandwidth

If $x(t)$ is bandlimited and $y(t)$ is bandlimited, then so is

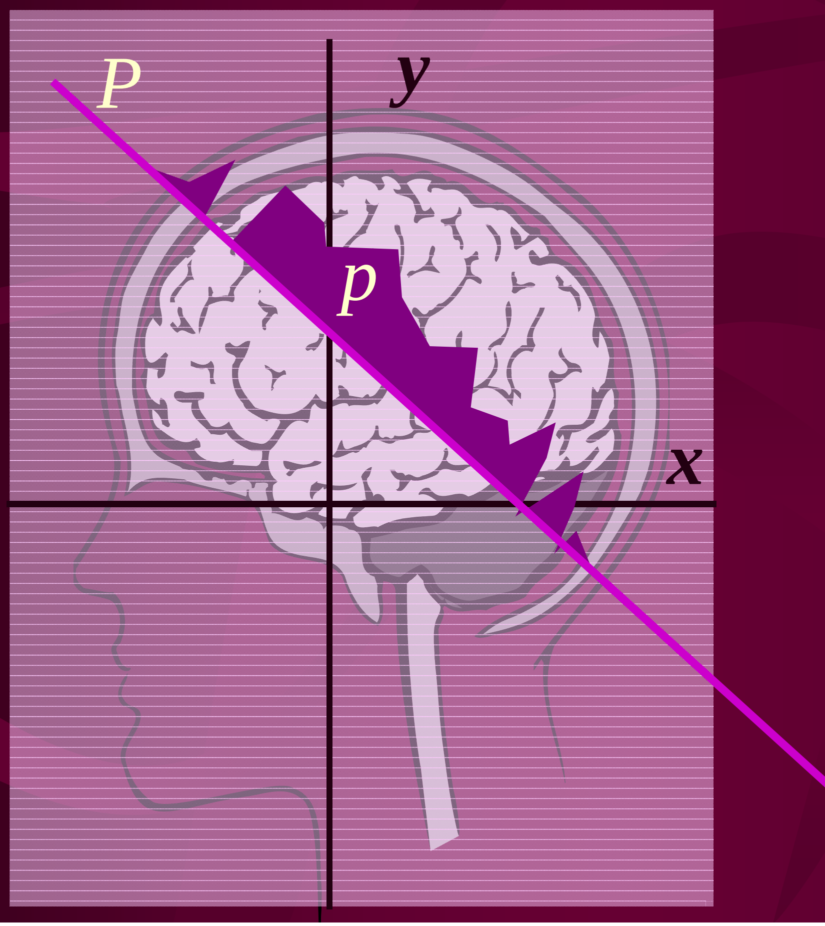
$$z(t) = \lambda\, x(t) + (1 - \lambda)\, y(t)$$



POCS: Example convex sets in a Hilbert space

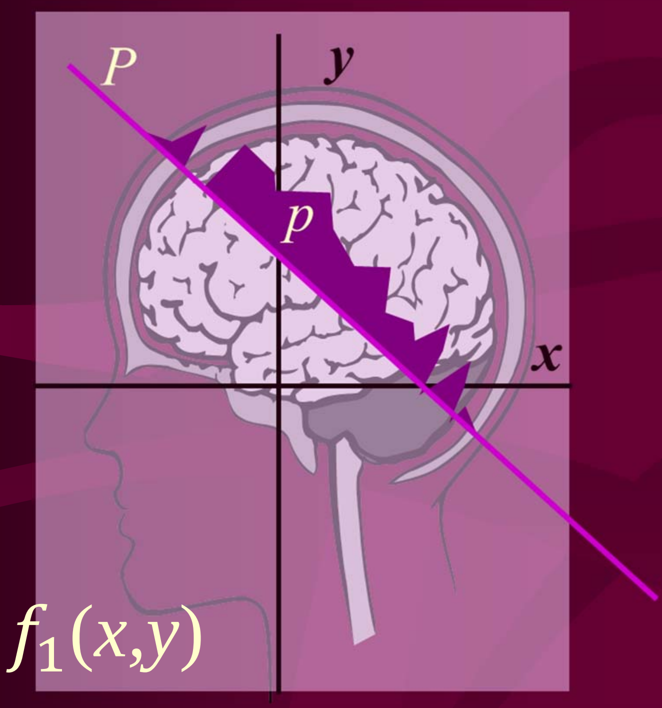
- **Tomographic Projections**

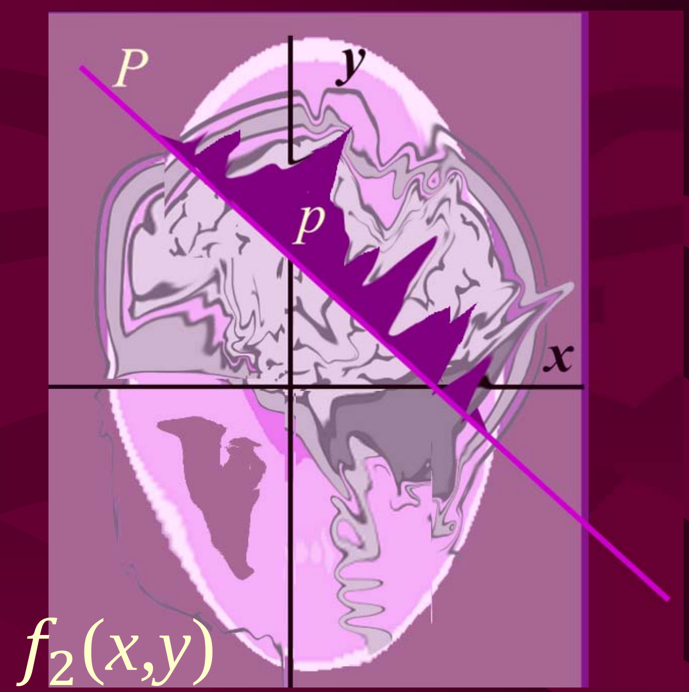
$$C_P = \left\{ f(x, y) \middle| \iint_P f(x, y)\, dx\, dy = p \right\}$$



POCS: Example convex sets in a Hilbert space

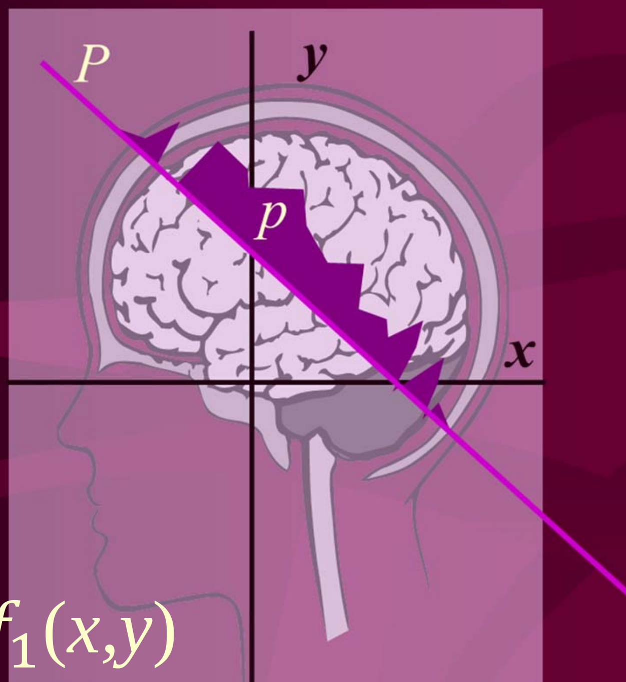
- **Identical Tomographic Projections** – a plane

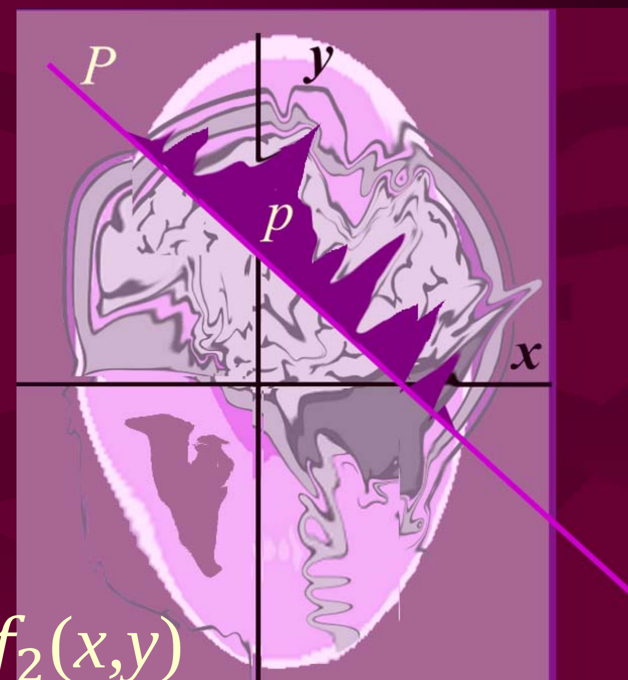




POCS: Example convex sets in a Hilbert space

- **Identical Tomographic Projections** – a plane





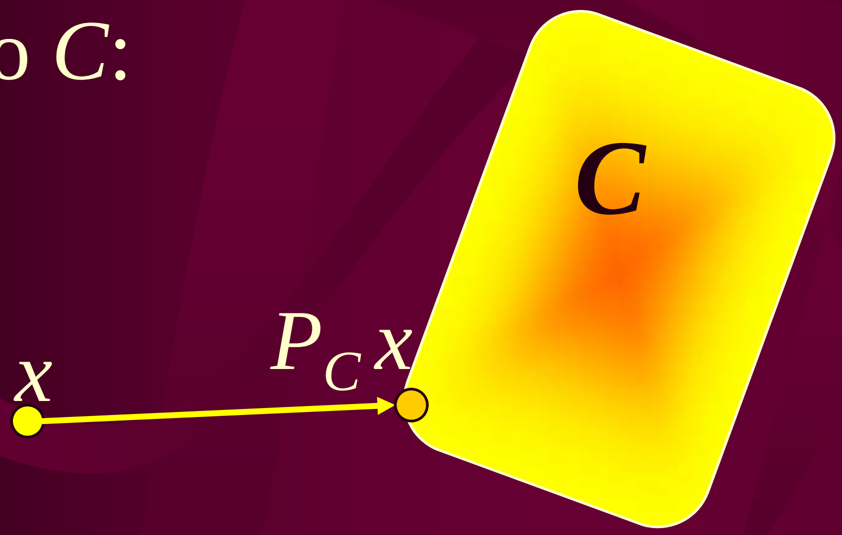
$$\iint_P \left[\lambda f_1(x,y) + (1-\lambda) f_2(x,y)\right] dxdy$$

$$= \lambda p + (1-\lambda) p = p$$

Same as constant area

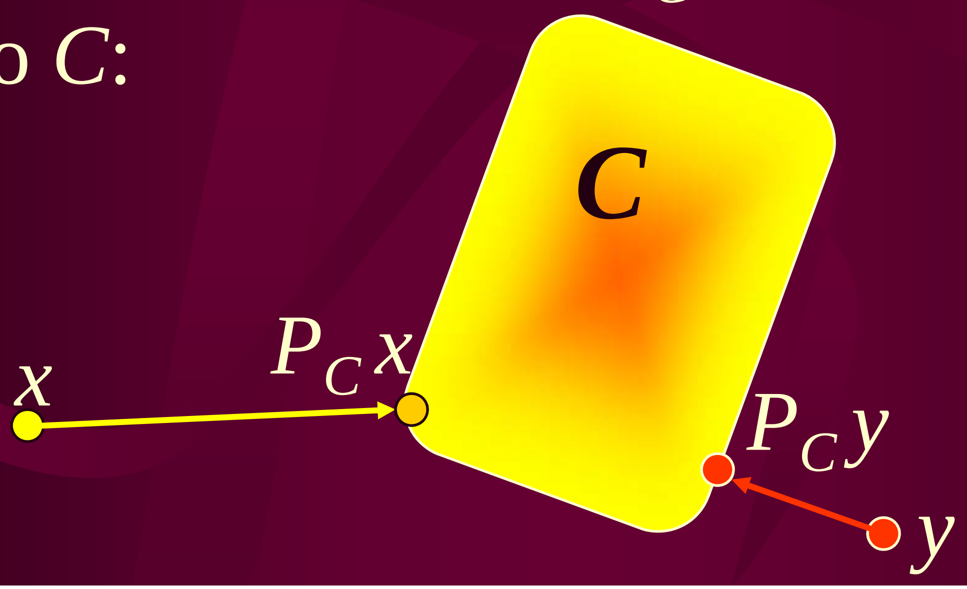
POCS: What is a projection onto a convex set?

- For every (closed) convex set, C, and every vector, x, in a Hilbert space, there is a *unique* vector in C, closest to x.
- This vector, denoted $P_C x$, is the ***projection*** of x onto C:



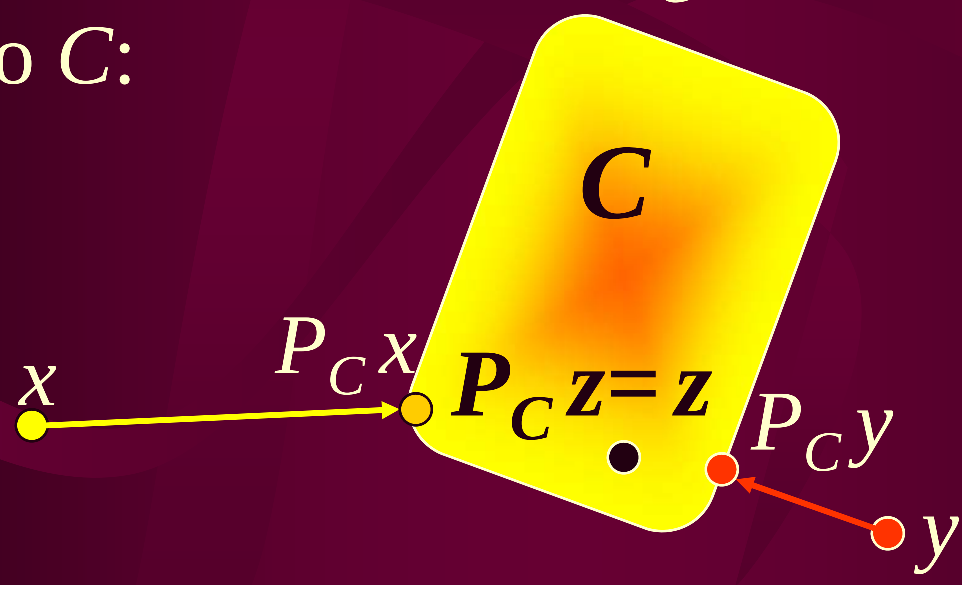
POCS: What is a projection onto a convex set?

- For every (closed) convex set, C, and every vector, x, in a Hilbert space, there is a *unique* vector in C, closest to x.
- This vector, denoted $P_C x$, is the ***projection*** of x onto C:



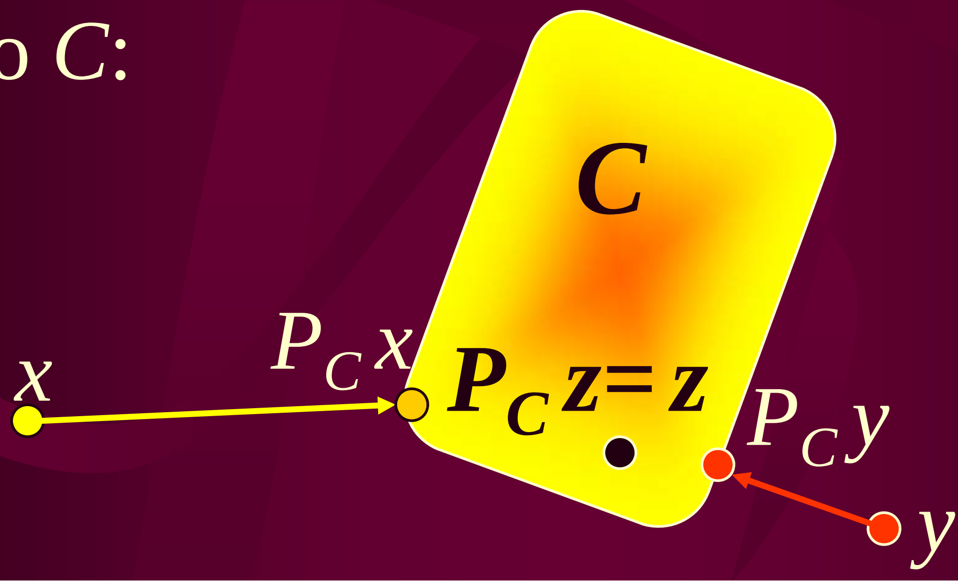
POCS: What is a projection onto a convex set?

- For every (closed) convex set, C, and every vector, x, in a Hilbert space, there is a *unique* vector in C, closest to x.
- This vector, denoted $P_C x$, is the ***projection*** of x onto C:



POCS: What is a projection onto a convex set?

- For every (closed) convex set, C, and every vector, x, in a Hilbert space, there is a *unique* vector in C, closest to x.
- This vector, denoted $P_C x$, is the ***projection*** of x onto C:

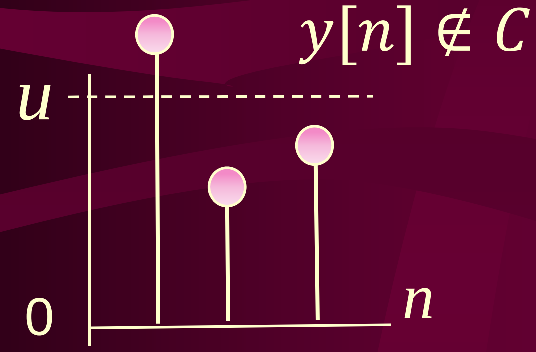


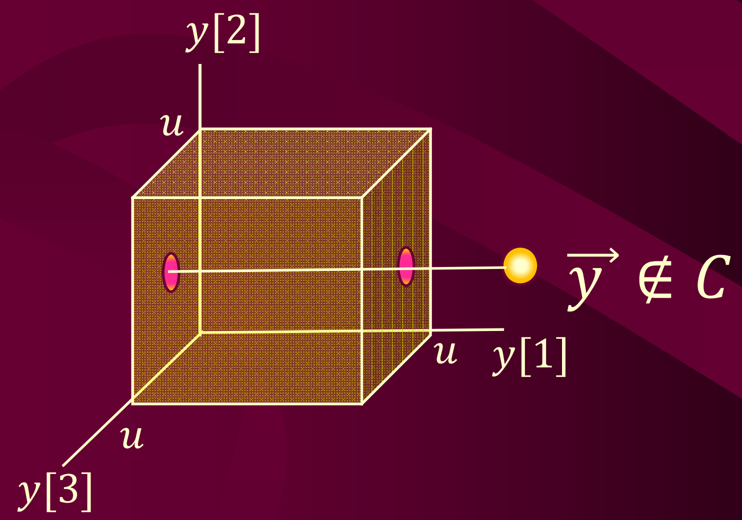
Projections are *idempotent.*

POCS: Example projections

- **Bounded Signals** – a box

$$C = \{x[n] \mid 0 \le x[n] \le u\} \quad ; n = 1,2,3$$

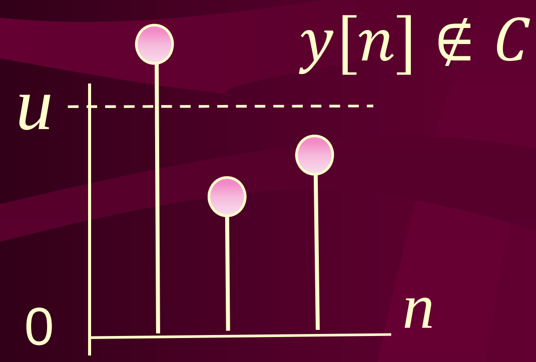


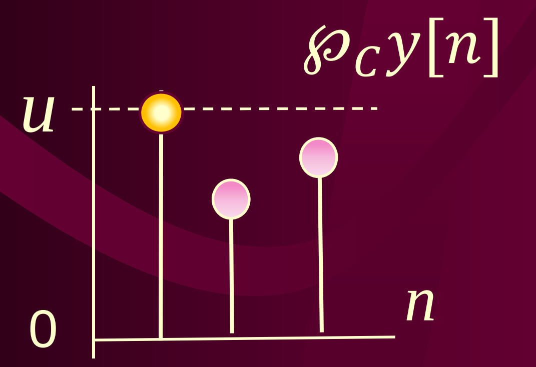


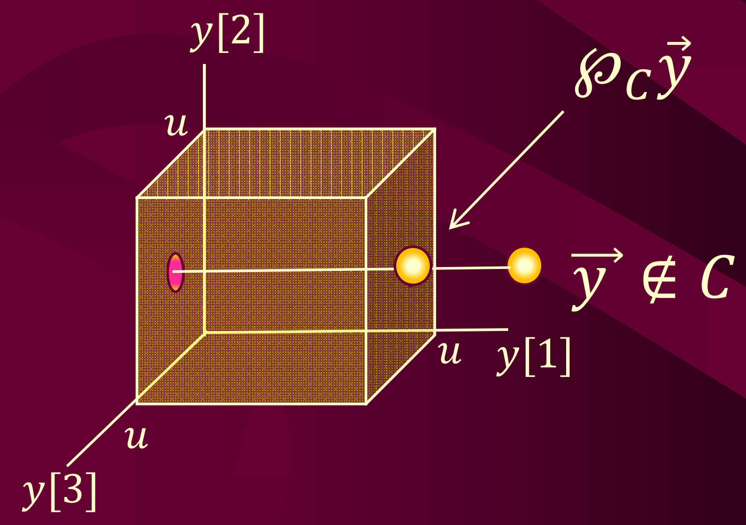
POCS: Example projections

- **Bounded Signals** – a box

$$C = \{x[n] \mid 0 \le x[n] \le u\} \quad ; n = 1,2,3$$



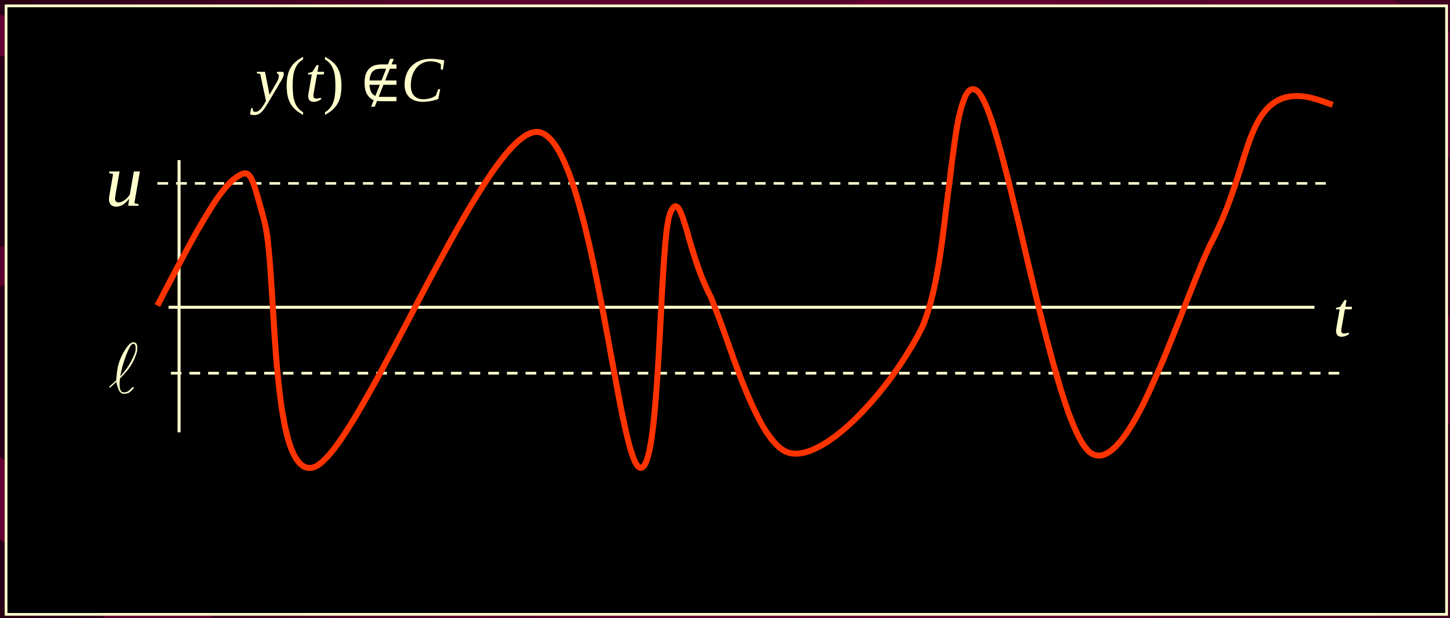




Example Projections

- **Bounded Signals** – a box

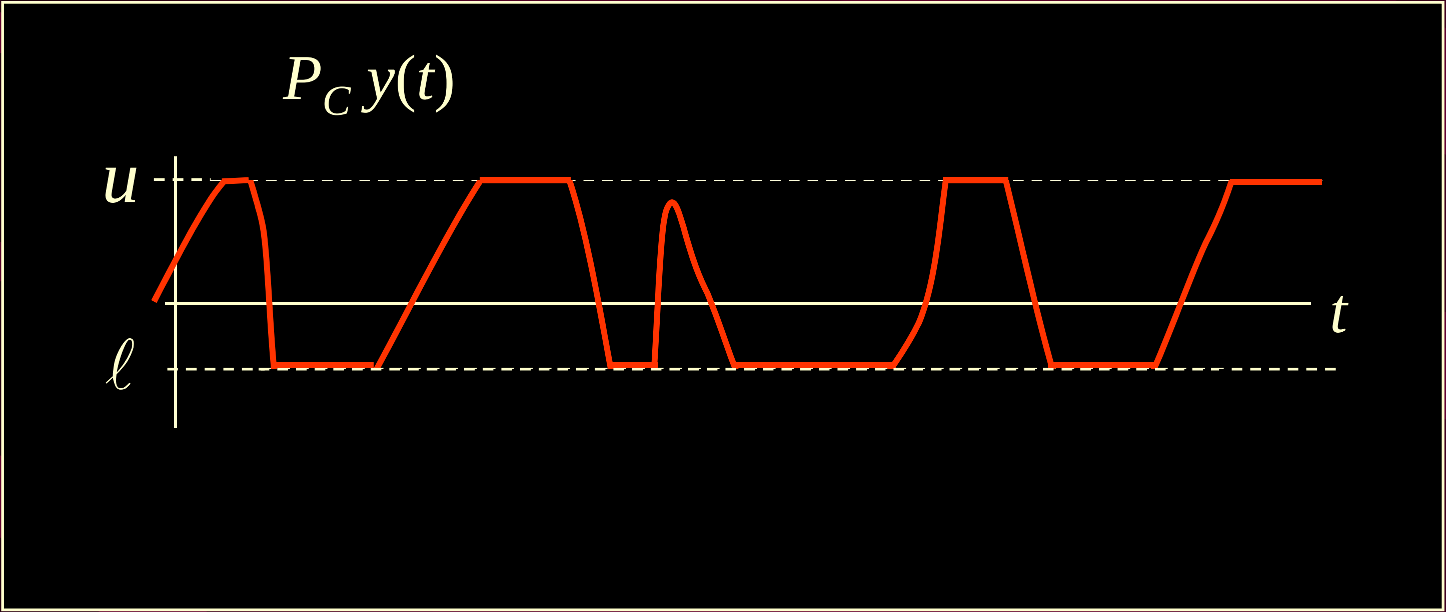
$$C = \left\{ x(t) \middle| \ell \le x(t) \le u \right\}$$



Example Projections

- **Bounded Signals** – a box

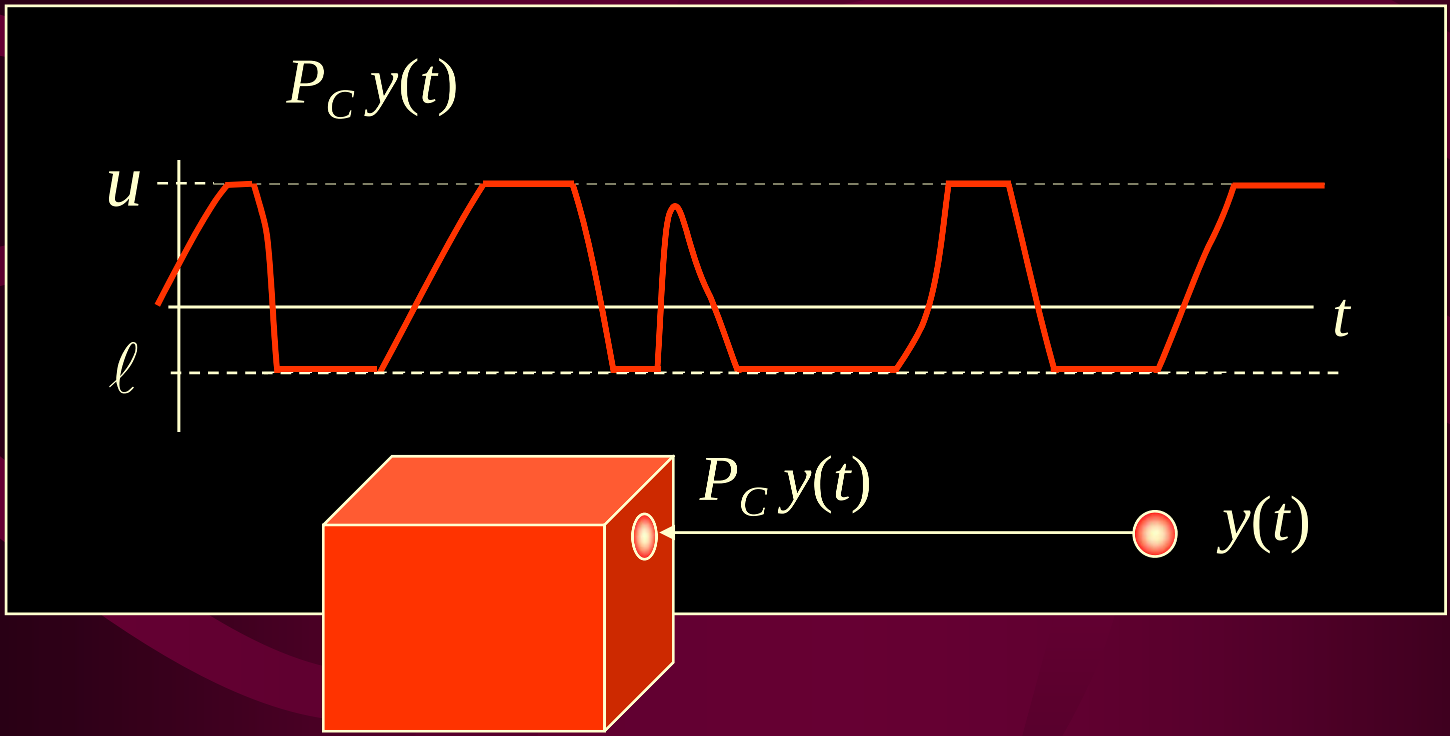
$$C = \left\{ x(t) \middle| \ell \le x(t) \le u \right\}$$



Example Projections

- **Bounded Signals** – a box

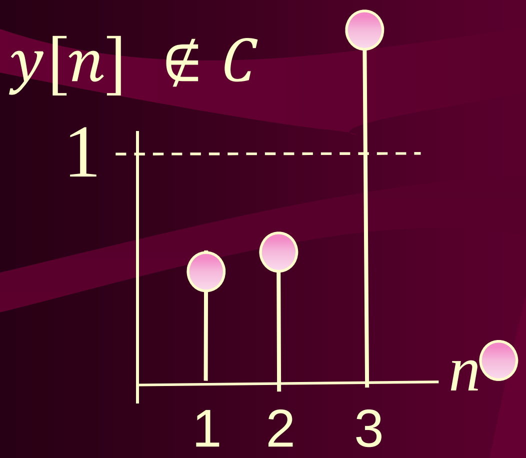
$$C = \left\{ x(t) \middle| \ell \le x(t) \le u \right\}$$

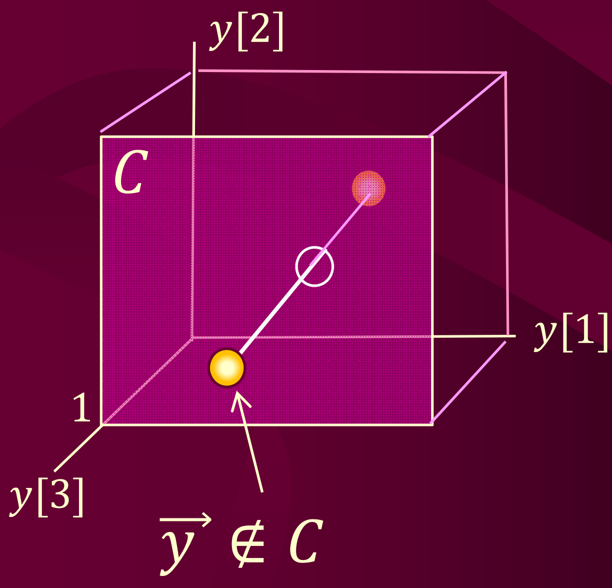


POCS: Example convex sets in a Hilbert space

- **Identical Middles** – a plane

$$C = \left\{ x[n] \,\middle|\, x[3] \equiv 1 \right\}, \; n = 1,2,3$$

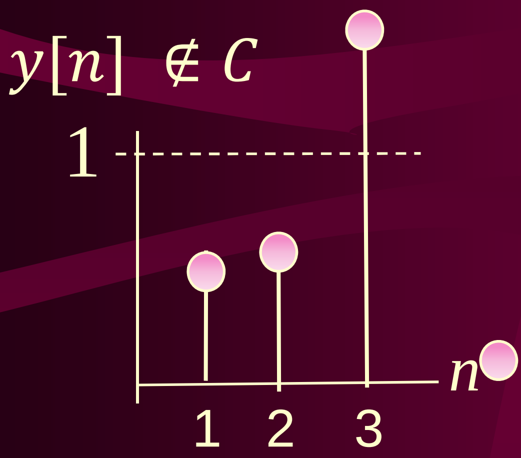


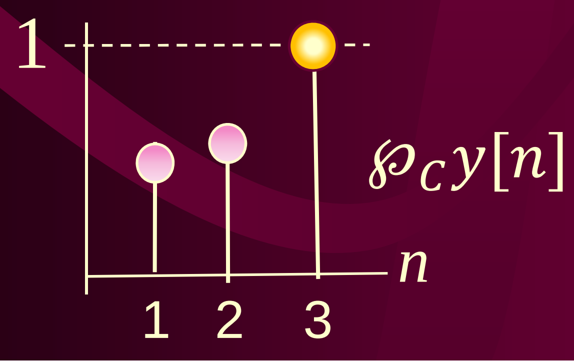


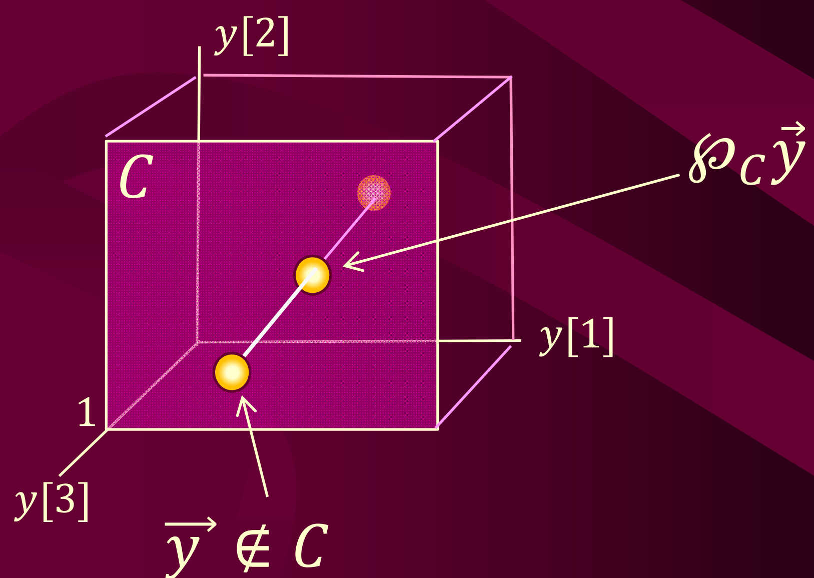
POCS: Example convex sets in a Hilbert space

- **Identical Middles** – a plane

$$C = \{x[n] \mid x[3] \equiv 1\},\ n = 1,2,3$$



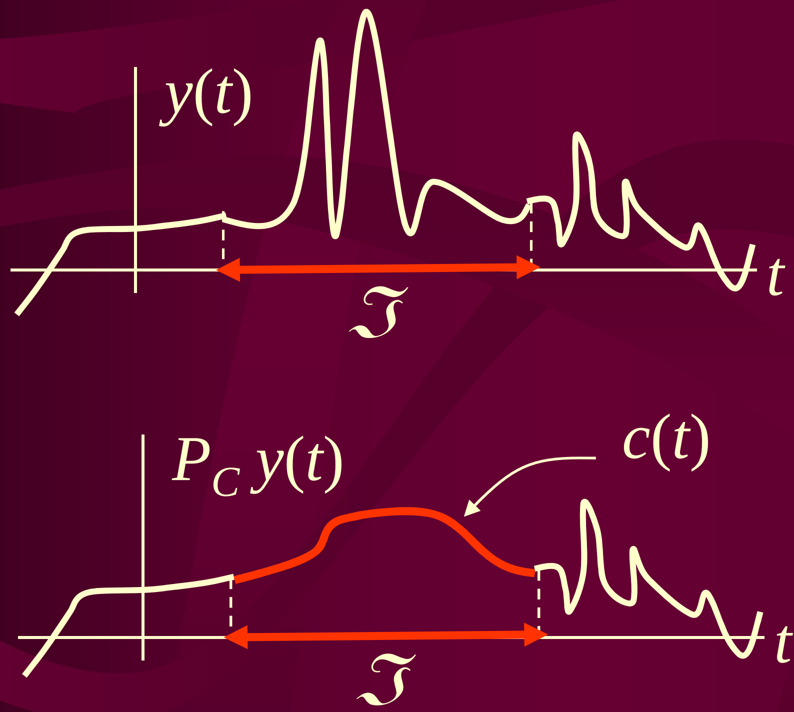


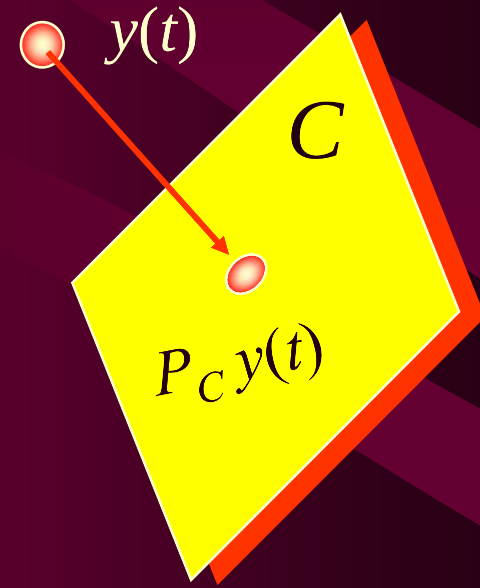


Example Projections

- **Identical Middles** – a plane

$$C = \left\{ x(t) \middle| x(t) \equiv c(t)\ ;\ t \in \mathfrak{I} \right\}$$

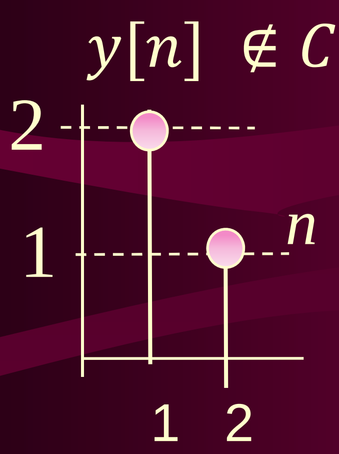


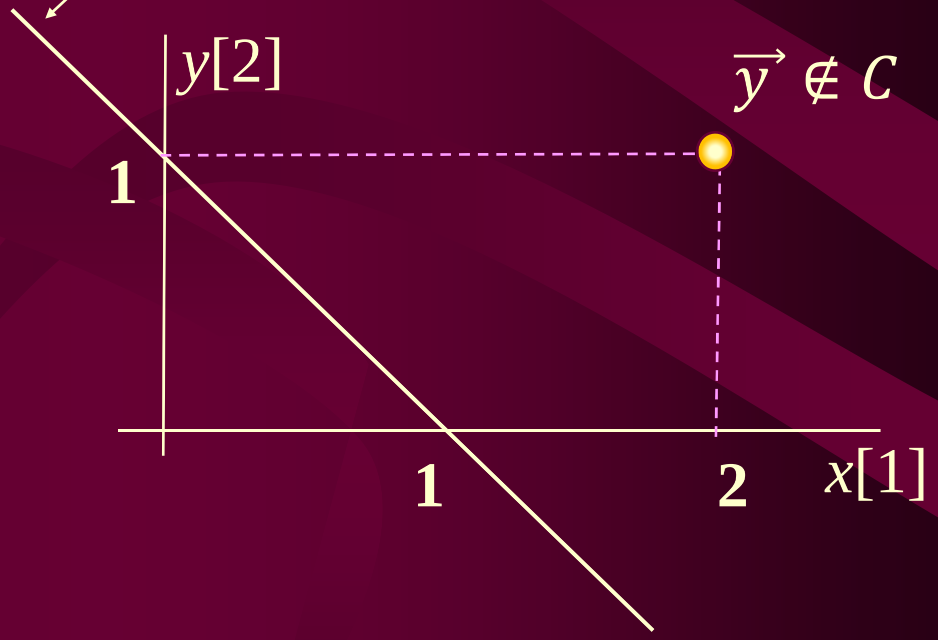


Example Projections

- **Constant Area Signals**
  - – a plane (linear variety)

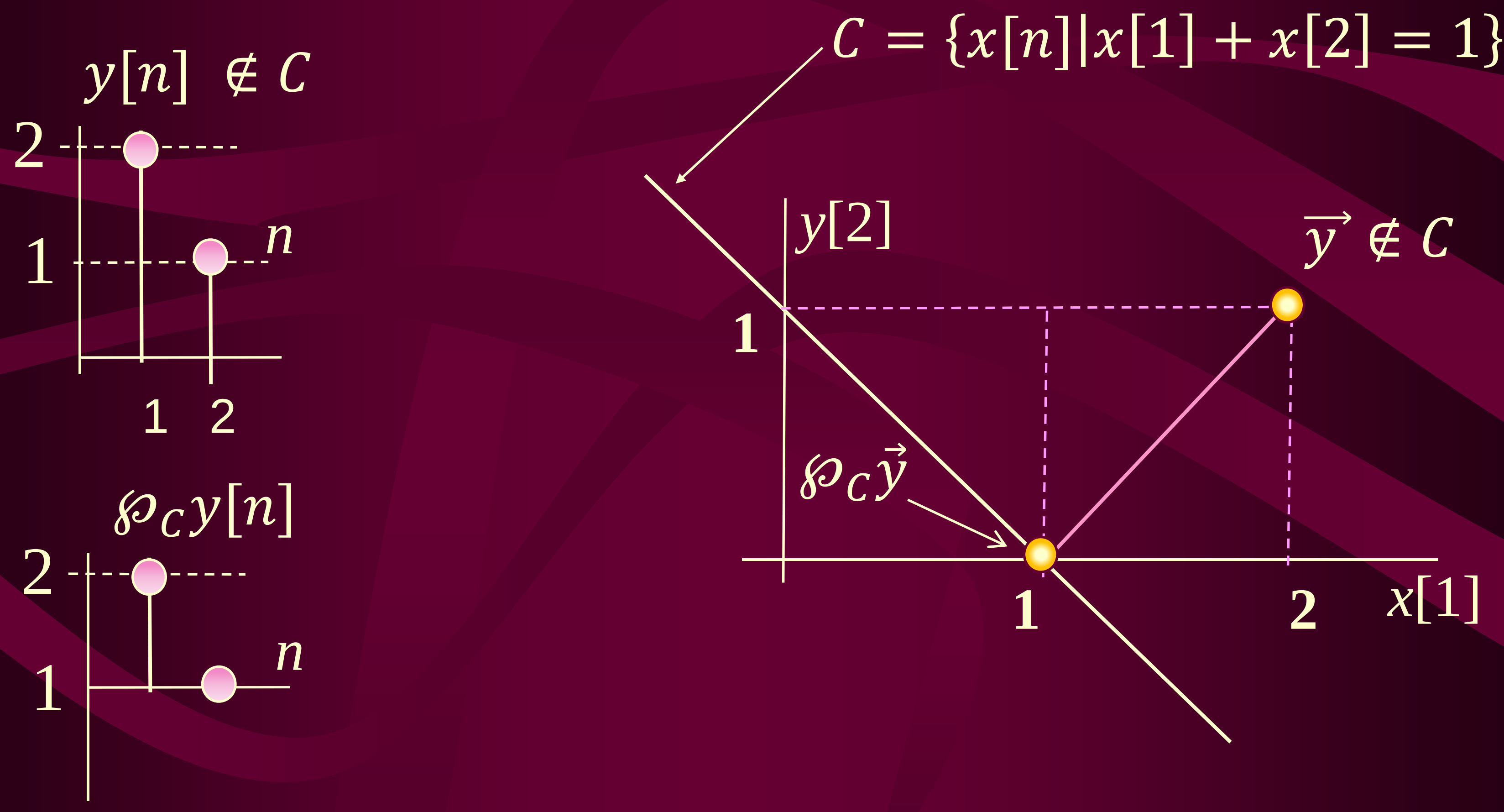
$C = \{x[n] | x[1] + x[2] = 1\}$





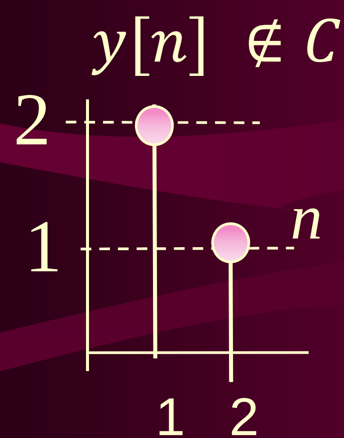
Example Projections

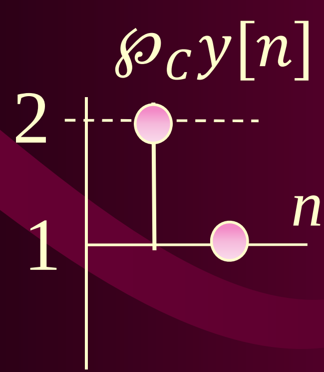
- **Constant Area Signals**
  – a plane (linear variety)

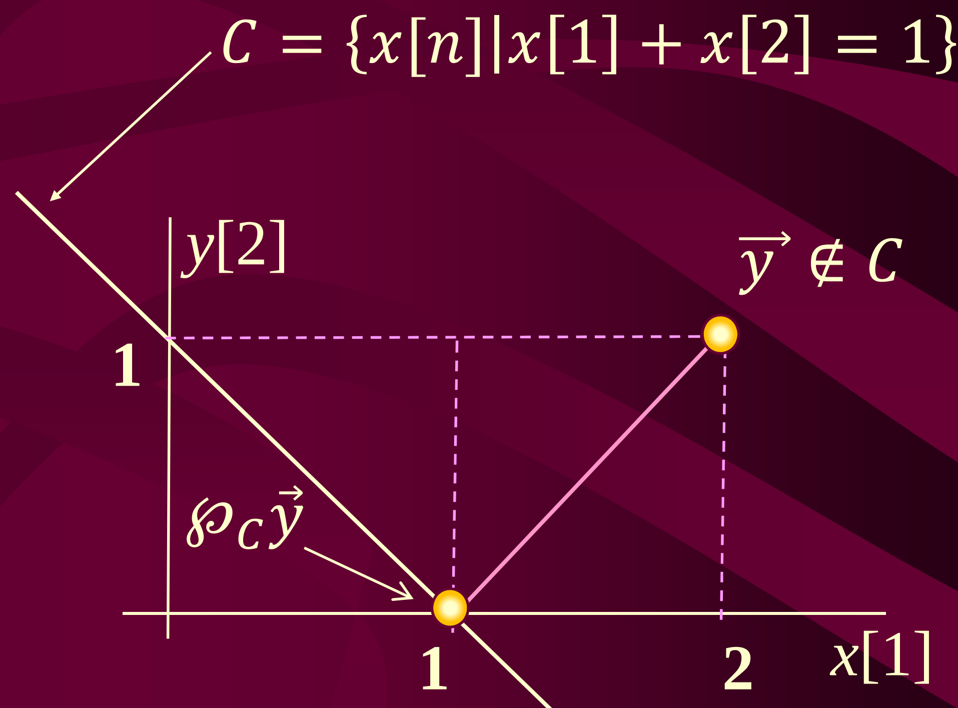


Example Projections

- **Constant Area Signals**
  - a plane (linear variety)





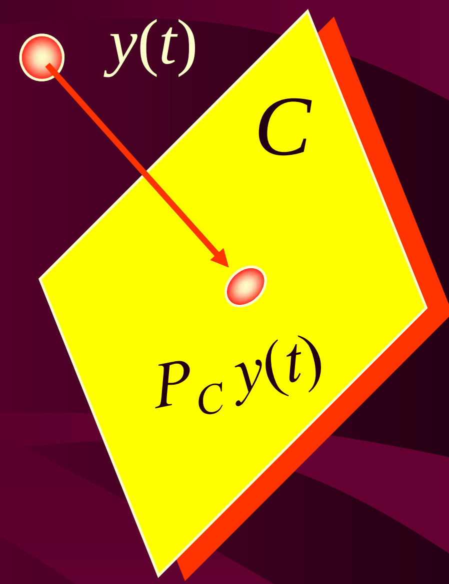


The same value (in this example, 1) is subtracted (or added) to every value.

Water Analogy

Example Projections

- **Constant Area Signals**
  - – a plane (linear variety)



Same value added to each element.

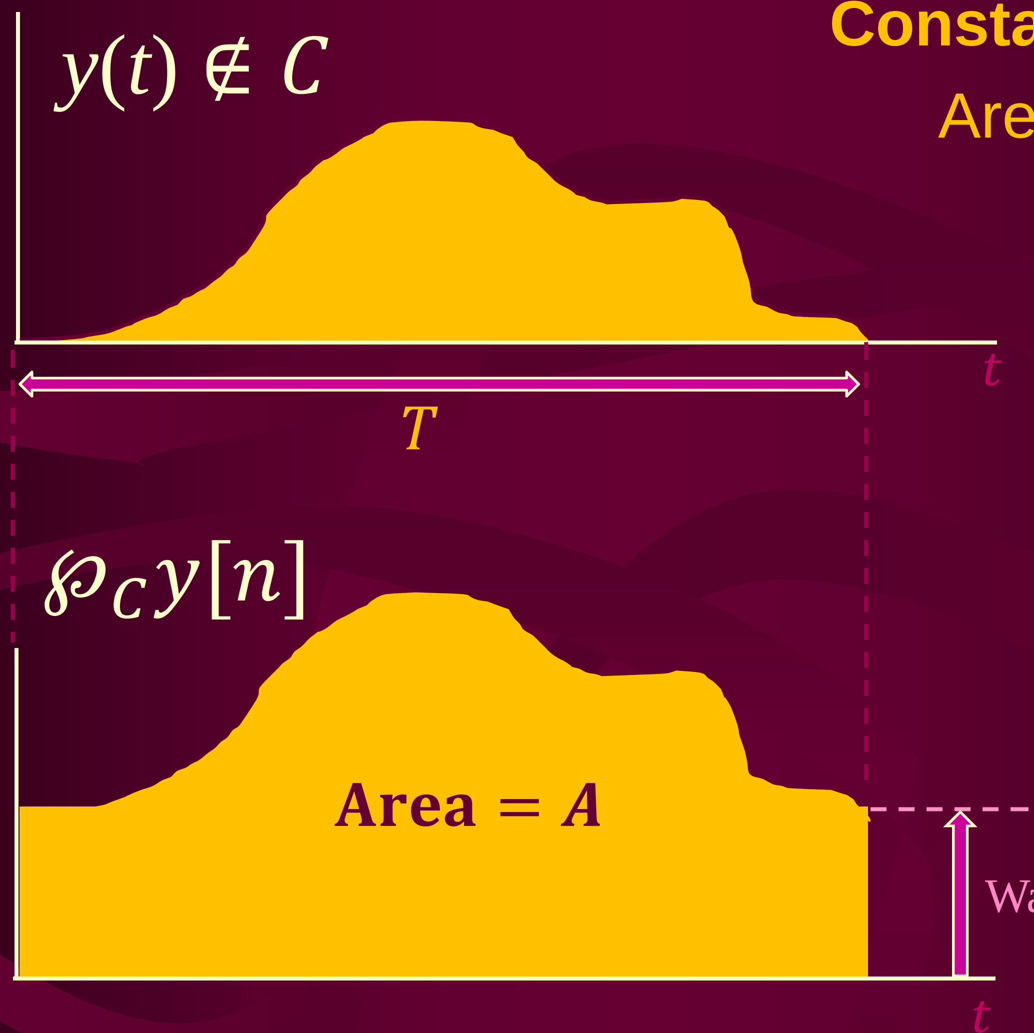
$$P_C \, \vec{y} = \vec{y} - \frac{A_{\vec{y}} - A}{\text{Interval}}$$

Water Analogy

Example Projections

Constant Area Signals

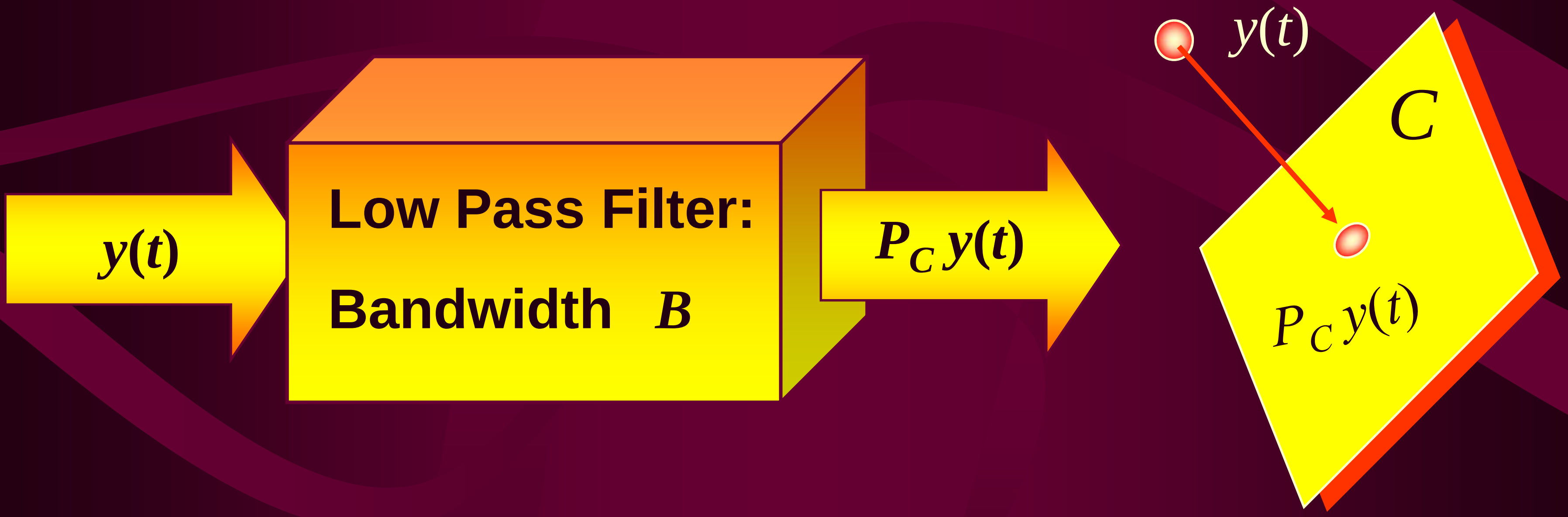
Area = A over an interval T



Example Projections

- **Bandlimited Signals** – a plane (subspace)

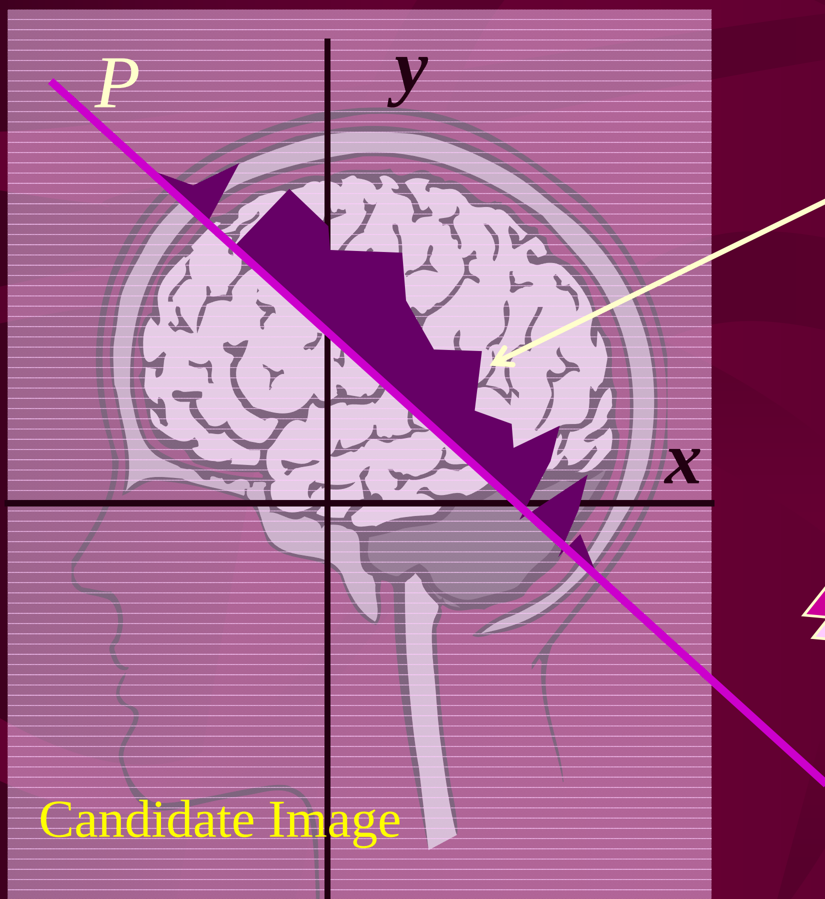
$$C = \left\{ x(t) \middle| X(u) \equiv 0 \;\; ; \;\; u > B \right\}$$



POCS: Example Projections

- **Tomographic Projections**

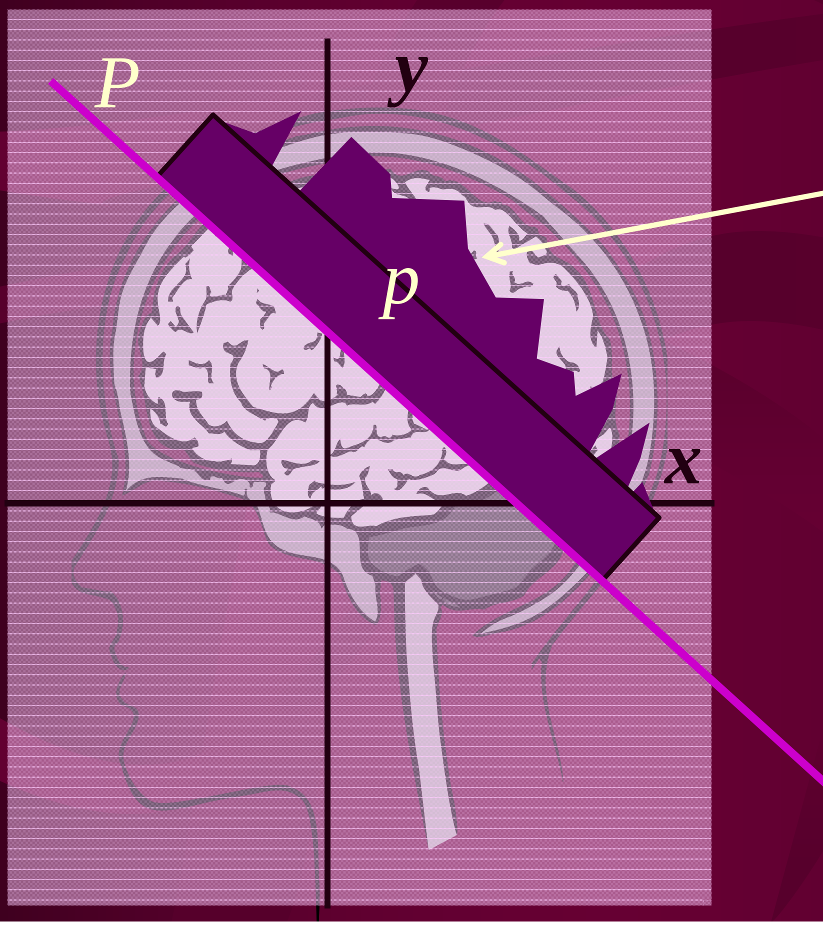
$$C_P = \left\{ f(x,y) \,\middle|\, \iint_P f(x,y)\,dxdy = p \right\}$$



POCS: Example Projections

- **Tomographic Projections**

$$C_P = \left\{ f(x,y) \middle| \iint_P f(x,y)\,dxdy = p \right\}$$

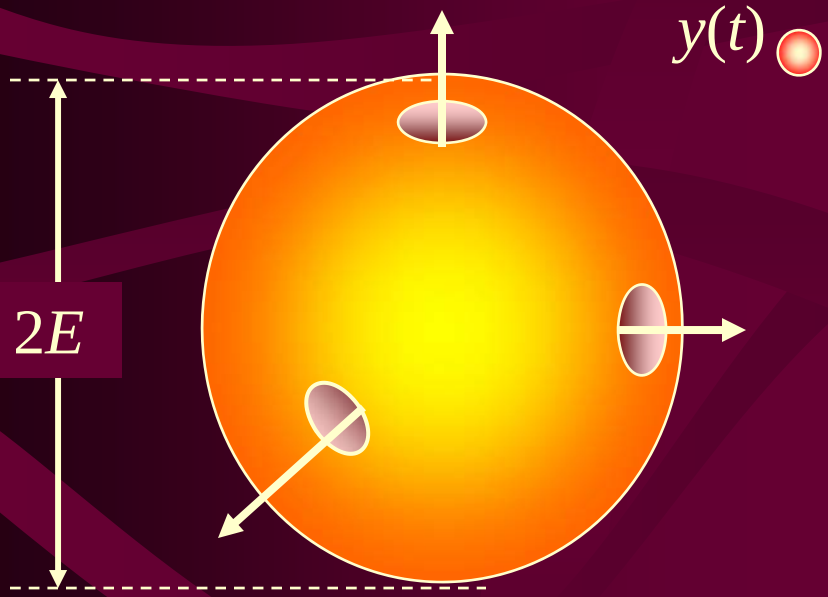


Raise (or lower) the water on the candidate image until the sum is p along the line P.

Example Projections

- **Bounded Energy Signals** – a ball

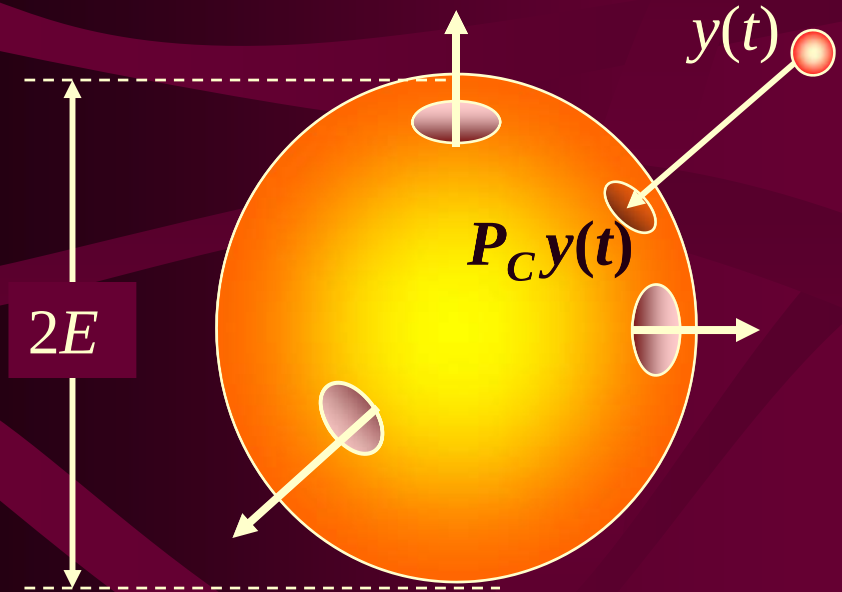
$$C = \left\{ x(t) \,\middle|\, \| x(t) \|^2 \le E \right\}$$



Example Projections

- **Bounded Energy Signals** – a ball

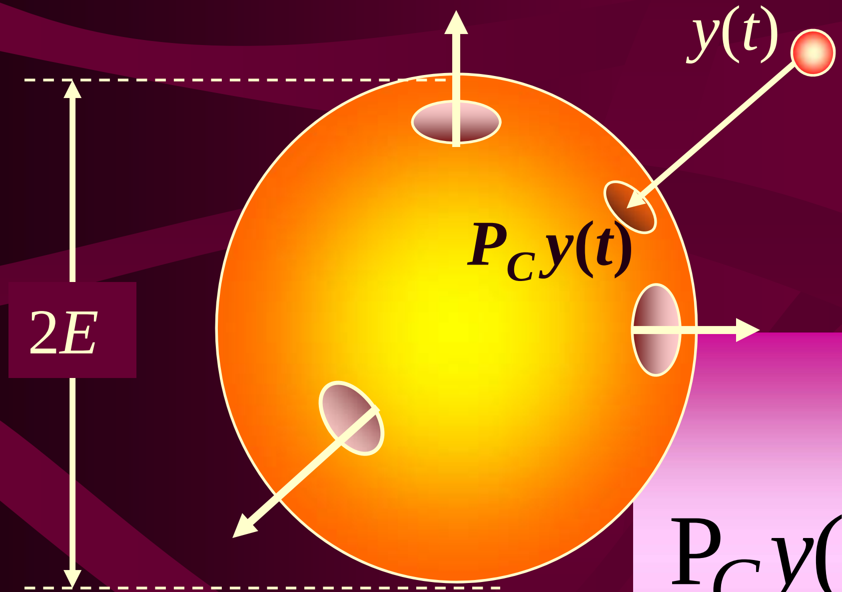
$$C = \left\{ x(t) \,\middle|\, \| x(t) \|^2 \le E \right\}$$



Example Projections

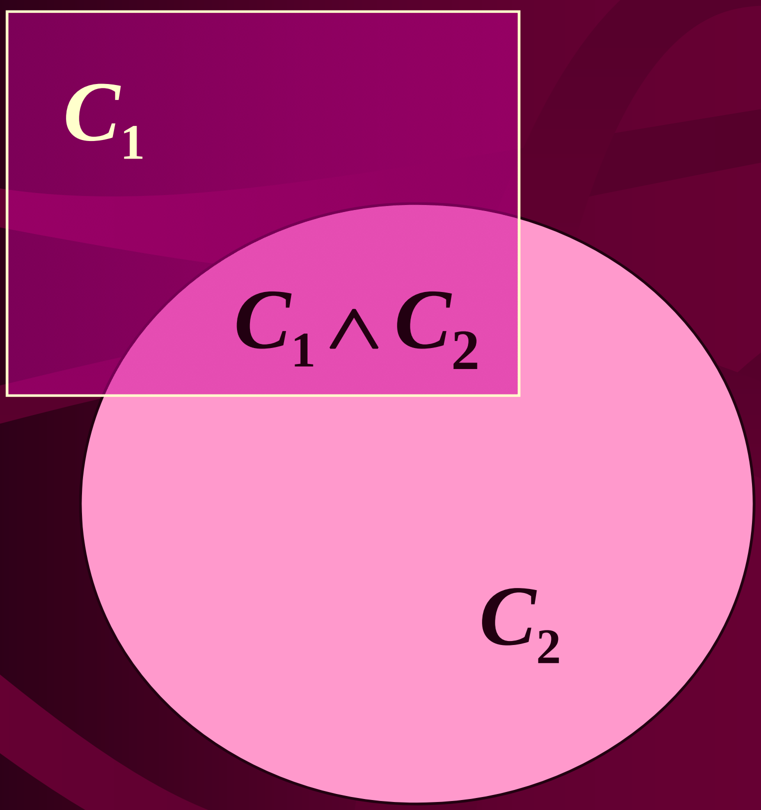
- **Bounded Energy Signals** – a ball

$$C = \left\{ x(t) \,\middle|\, \| x(t) \|^2 \le E \right\}$$

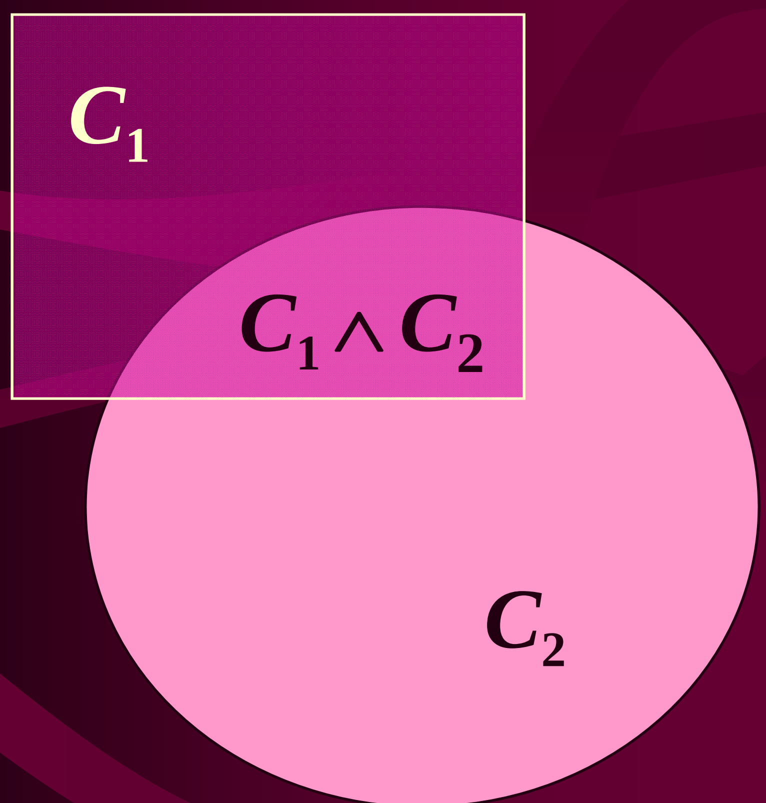


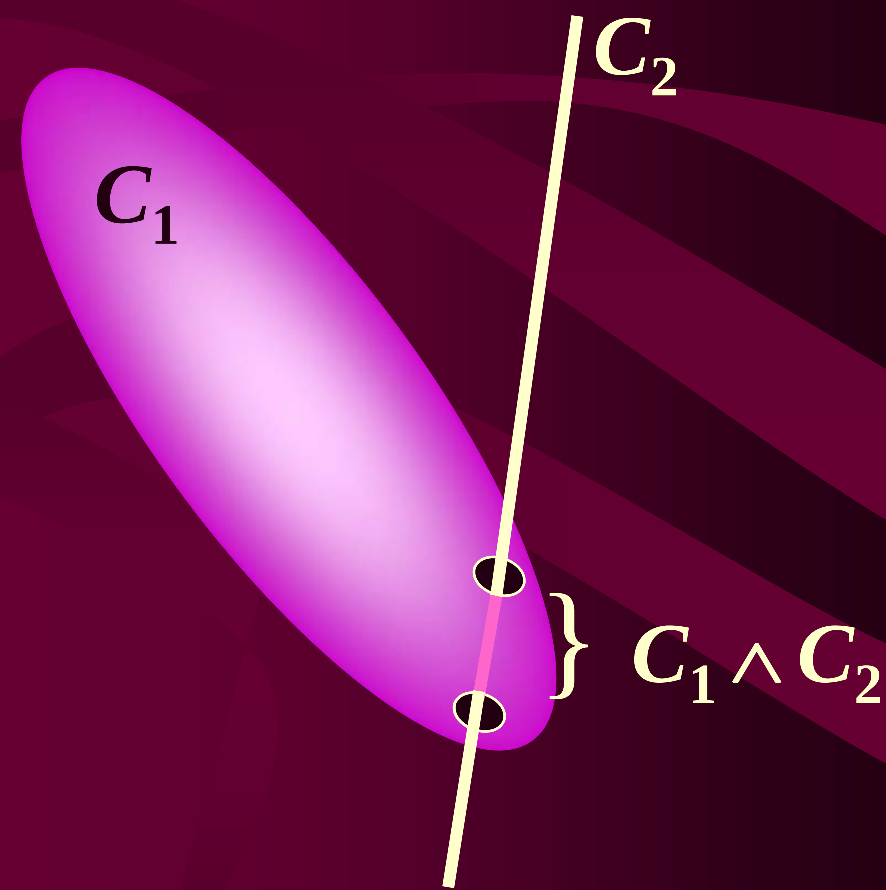
$$\mathrm{P}_C y(t) = \begin{cases} \sqrt{E}\,\dfrac{y(t)}{\| y(t) \|} & ; \| y(t) \| > E \\ y(t) & ; \| y(t) \| \le E \end{cases}$$

The intersection of convex sets is convex

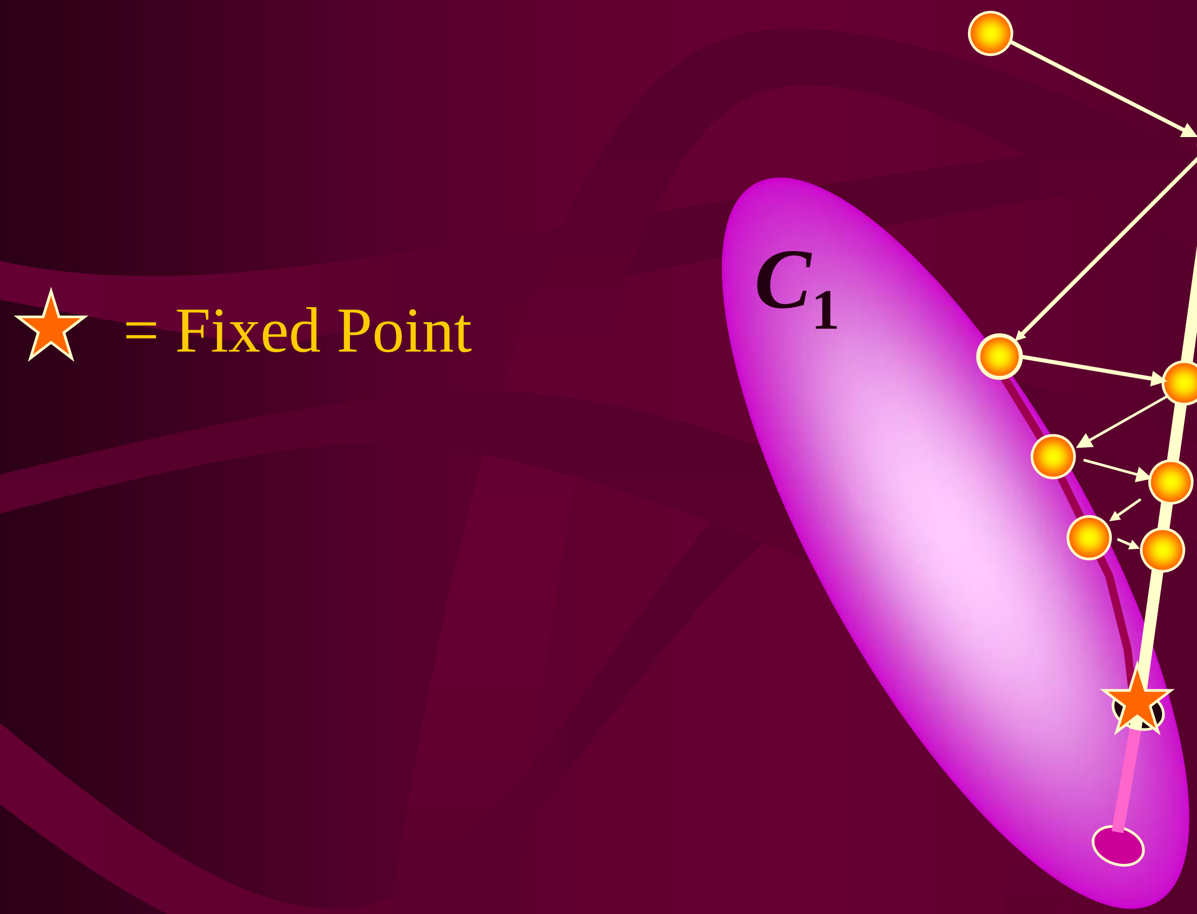


The intersection of convex sets is convex

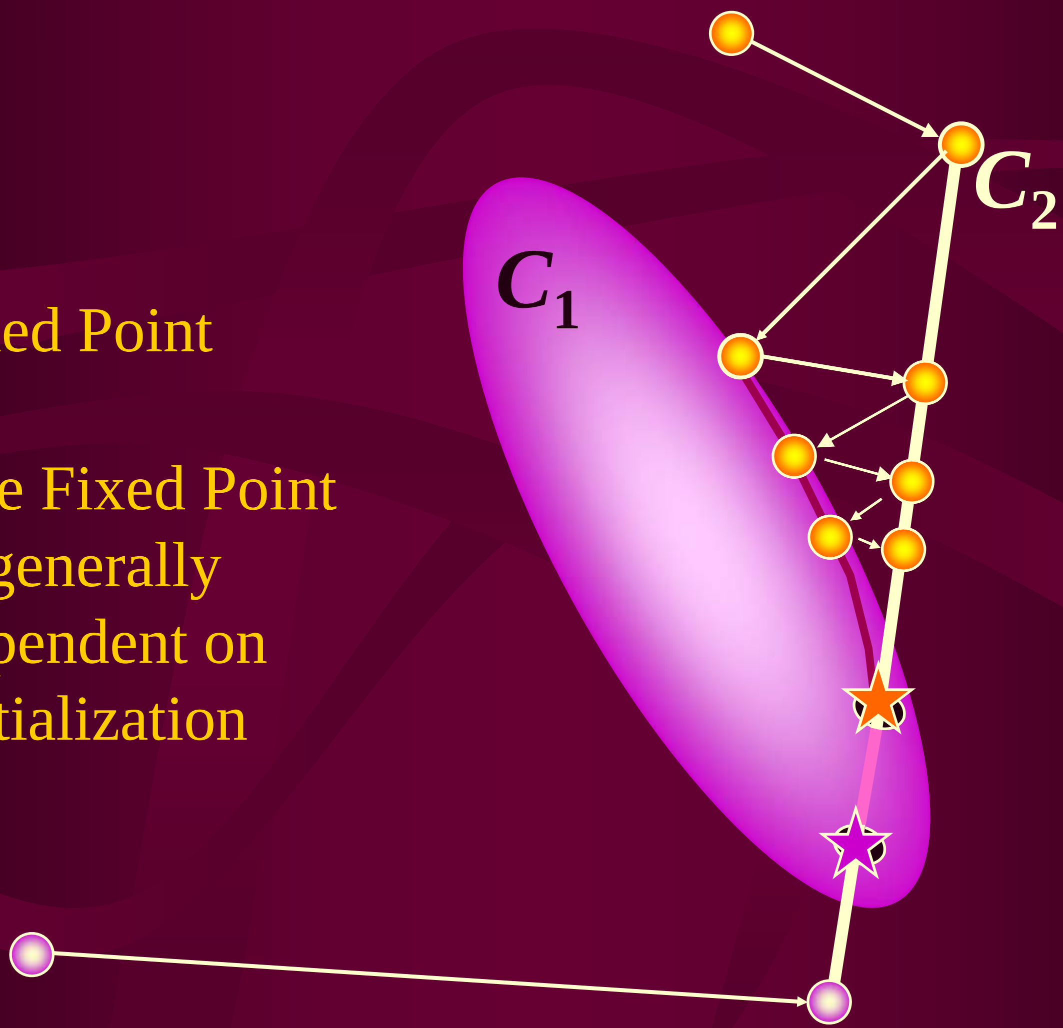




Alternating POCS will converge to a point common to both sets



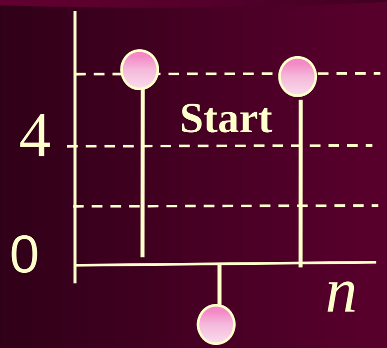
Alternating POCS will converge to a point common to both sets



☆ = Fixed Point

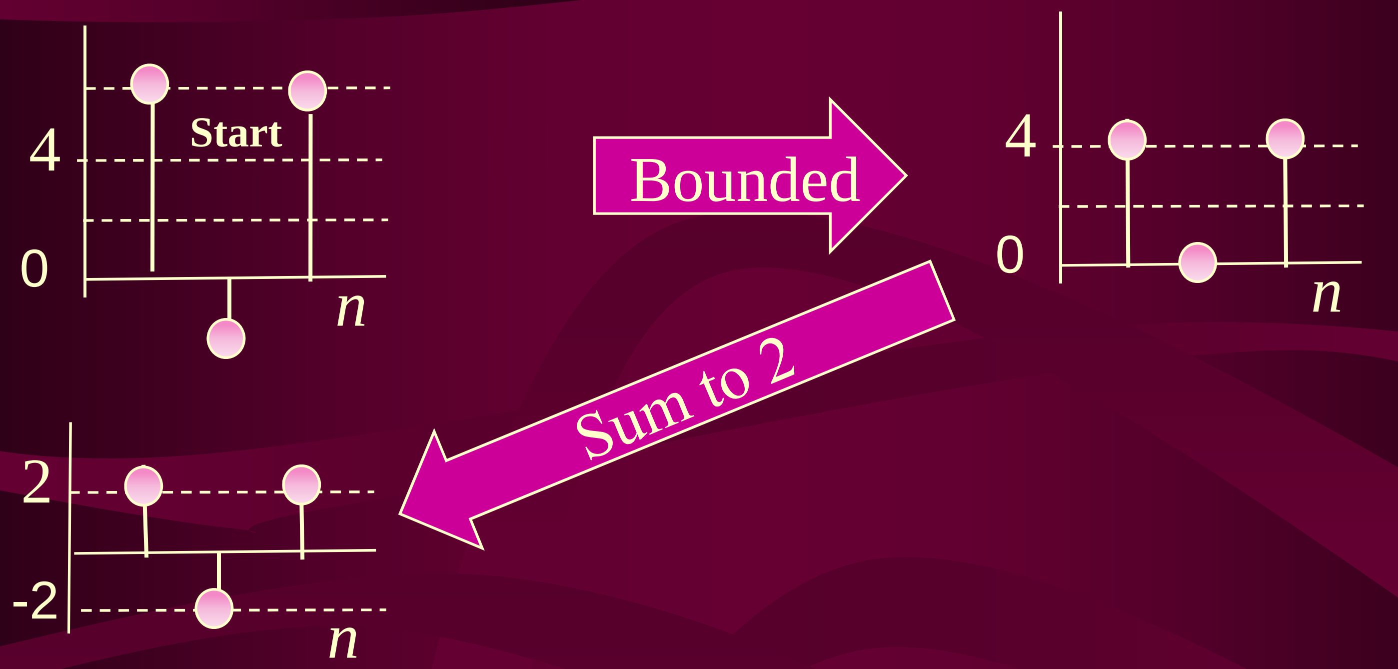
The Fixed Point is generally dependent on initialization

Bounded: $0 \leq x[n] \leq 4$ and Area: $x[1\} + x[2\} + x[3] = 2$

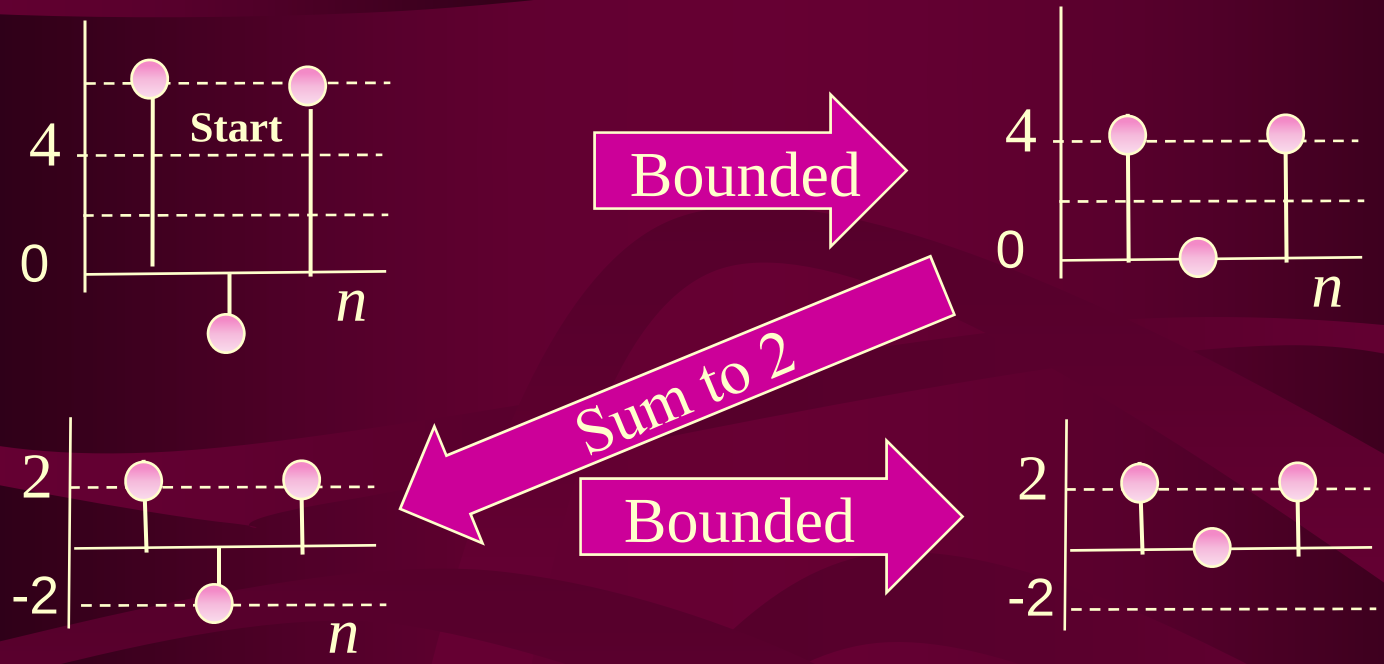


Bounded: $0 \le x[n] \le 4$ and Area: $x[1\} + x[2\} + x[3] = 2$

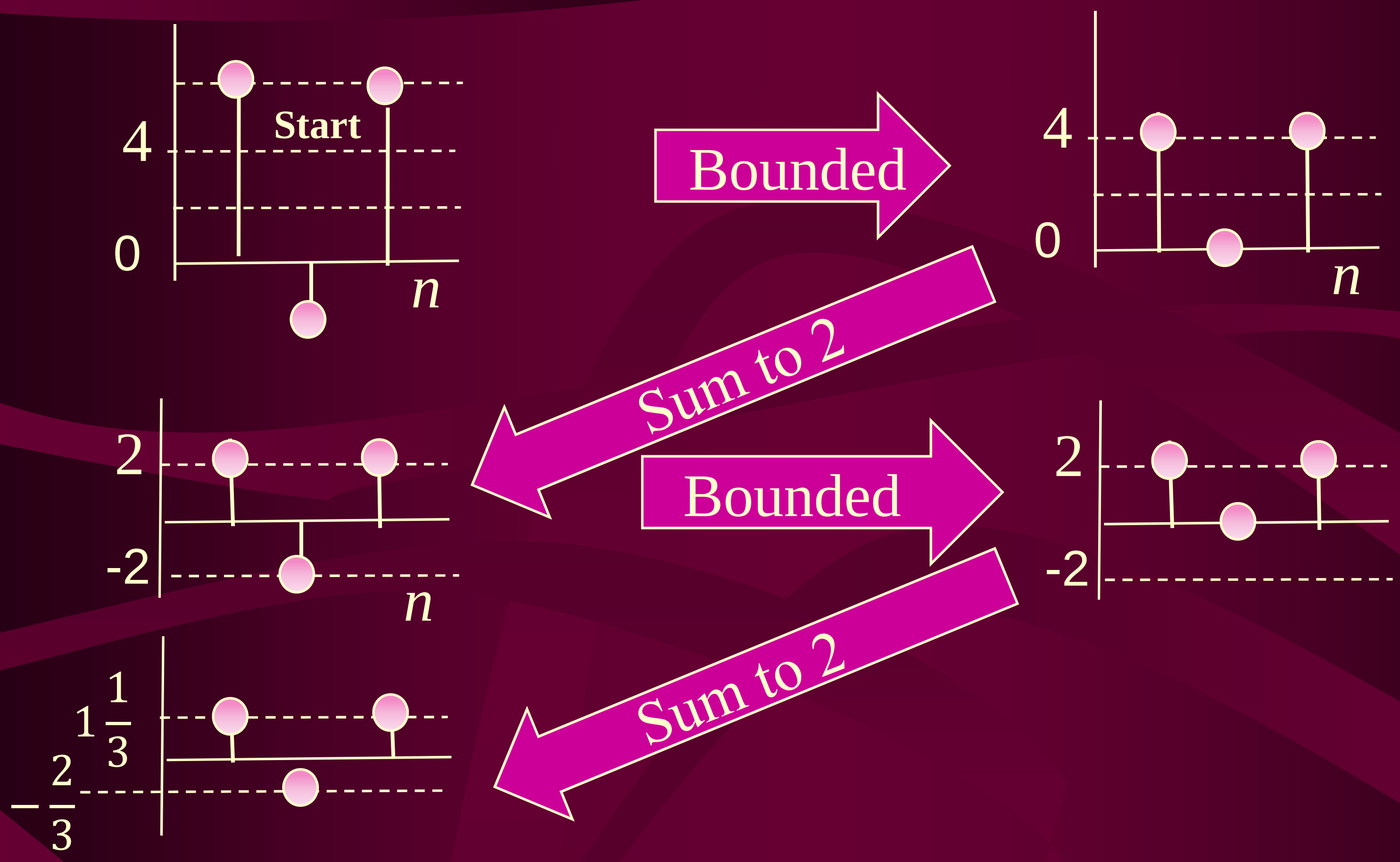
Bounded: $0 \le x[n] \le 4$ and Area: $x[1\} + x[2\} + x[3] = 2$



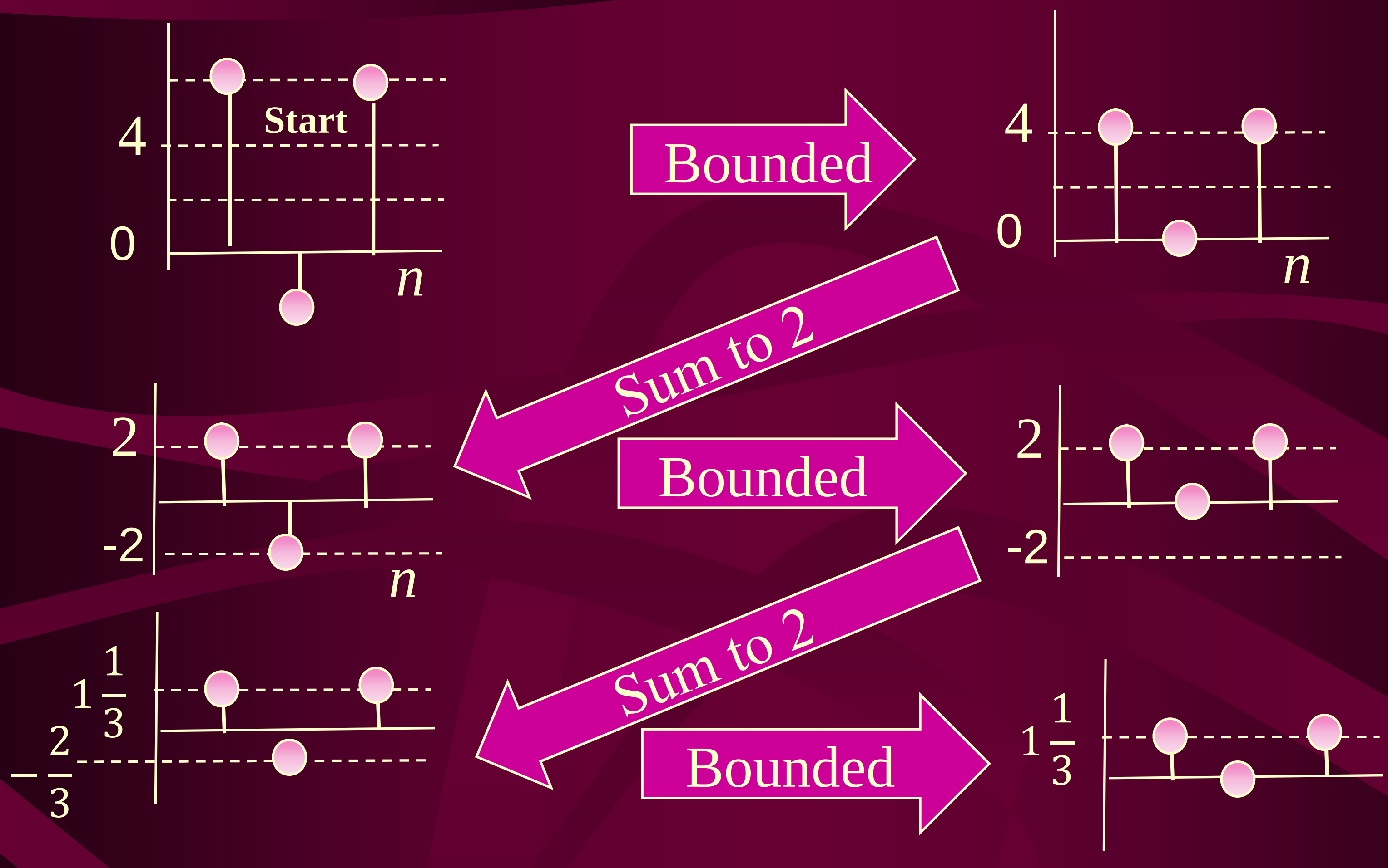
Bounded: $0 \le x[n] \le 4$ and Area: $x[1\} + x[2\} + x[3] = 2$



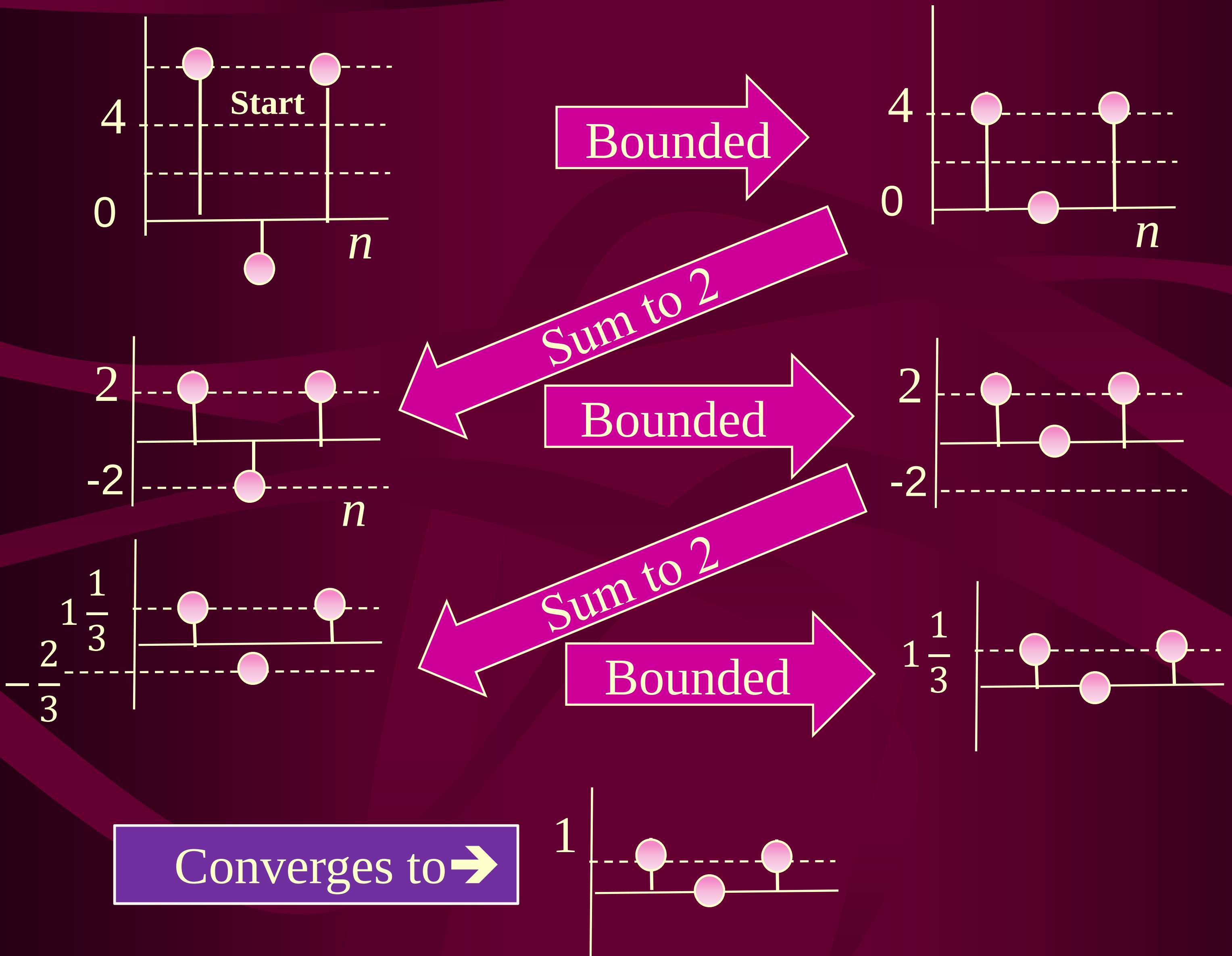
Bounded: $0 \leq x[n] \leq 4$ and Area: $x[1\} + x[2\} + x[3] = 2$



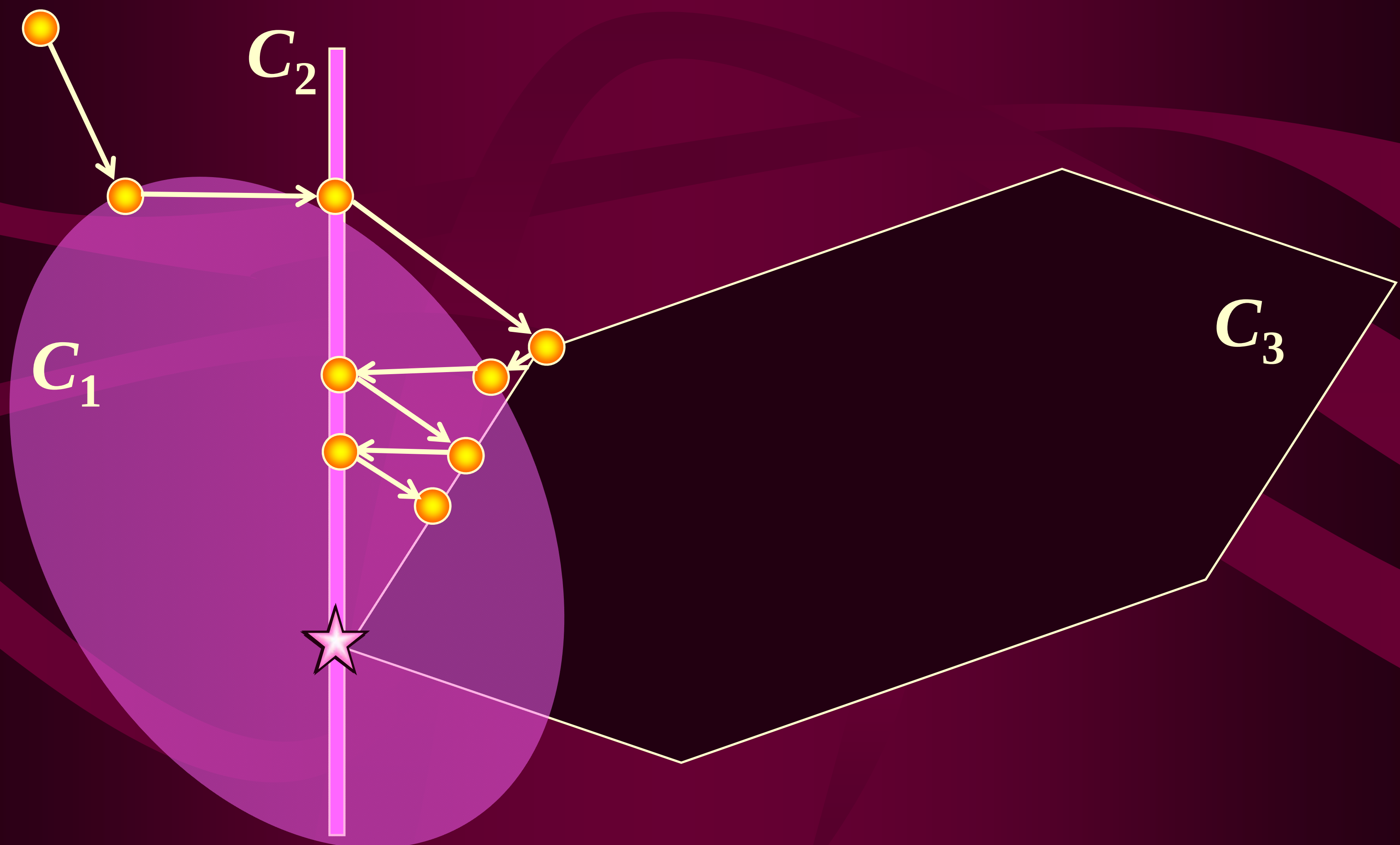
Bounded: $0 \le x[n] \le 4$ and Area: $x[1\} + x[2\} + x[3] = 2$



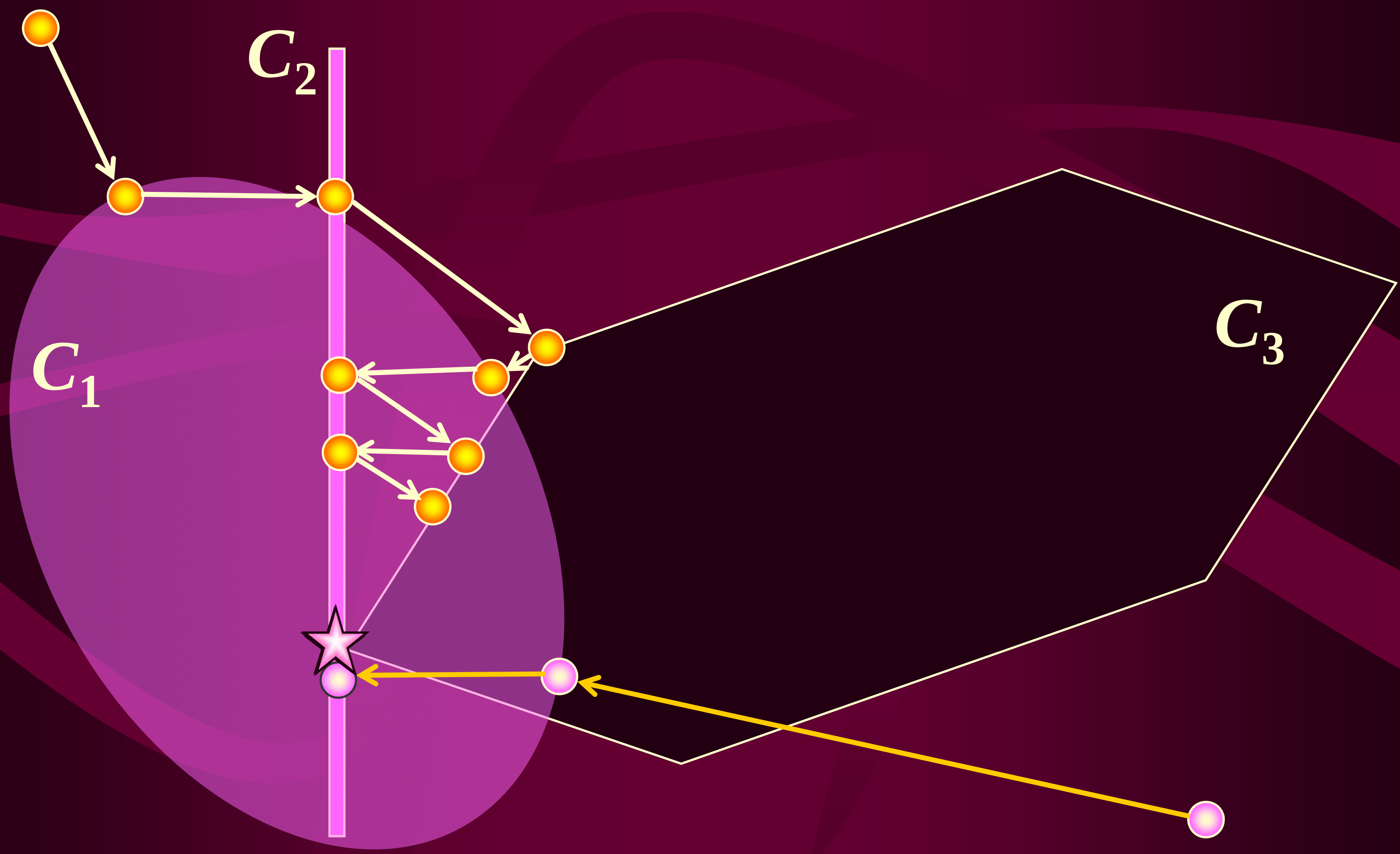
Bounded: $0 \leq x[n] \leq 4$ and Area: $x[1\} + x[2\} + x[3] = 2$



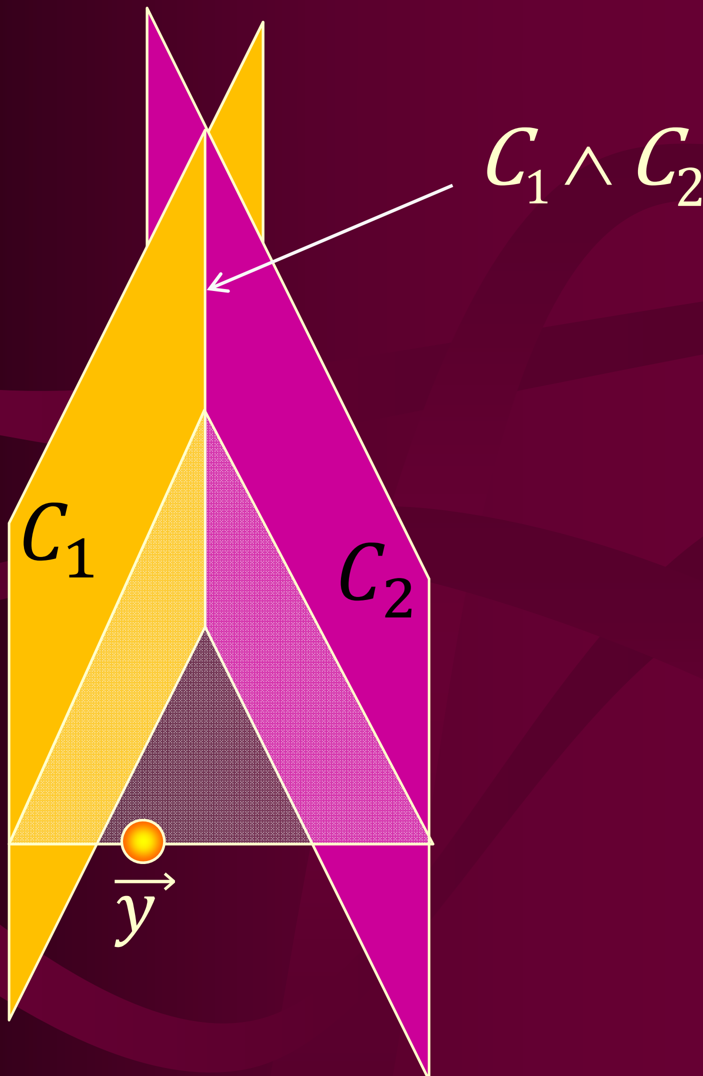
Lemma 1: Alternating POCS among N convex sets with a non-empty intersection will converge to a point common to all.



Lemma 1: Alternating POCS among N convex sets with a non-empty intersection will converge to a point common to all.

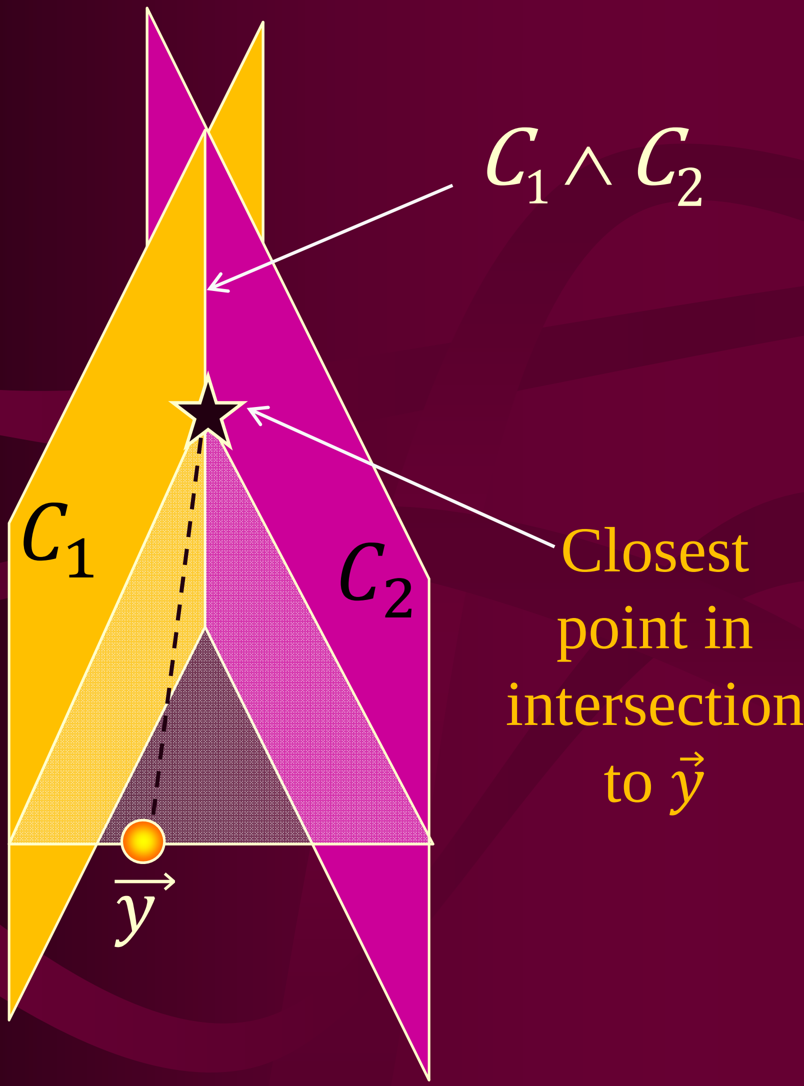


Von Neumann's Alternating Projection Algorithm



Consider two or more intersecting hyperplanes. (e.g. identical middles, constant area, bandlimited) and an external signal $\vec{y}$

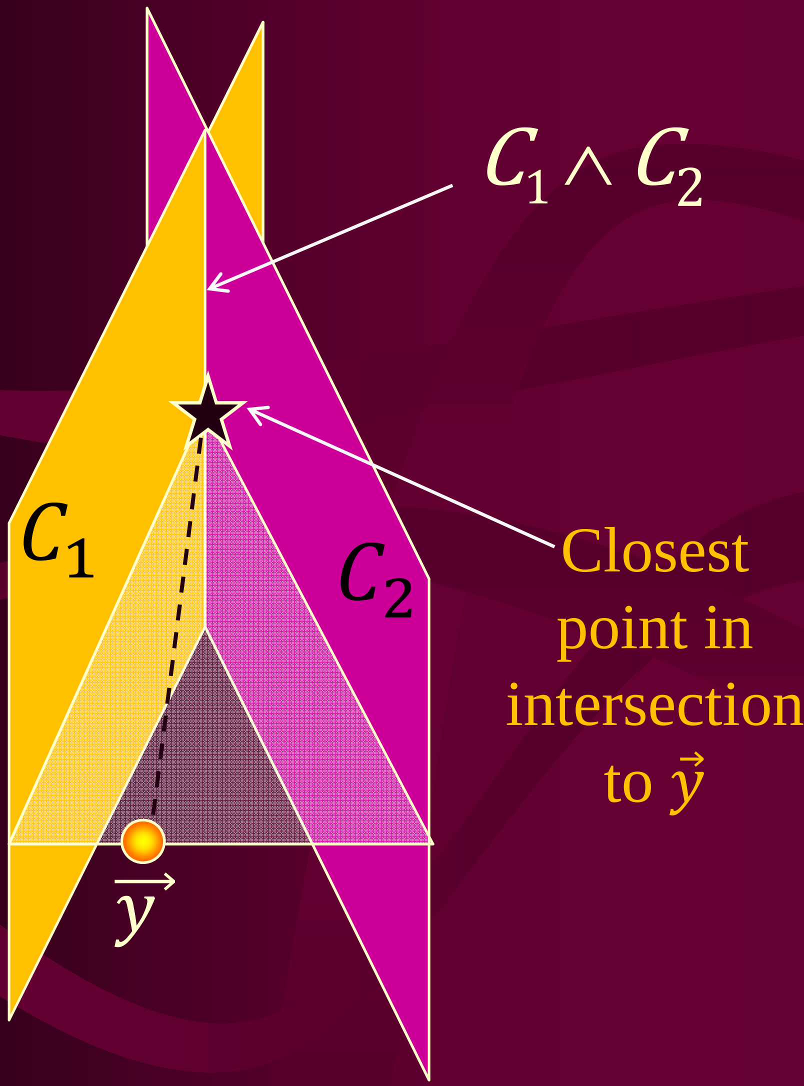
Von Neumann's Alternating Projection Algorithm



Consider two or more intersecting hyperplanes. (e.g. identical middles, constant area, bandlimited) and an external signal $\vec{y}$

There is a unique point in the intersection that is closest to $\vec{y}$

Von Neumann's Alternating Projection Algorithm

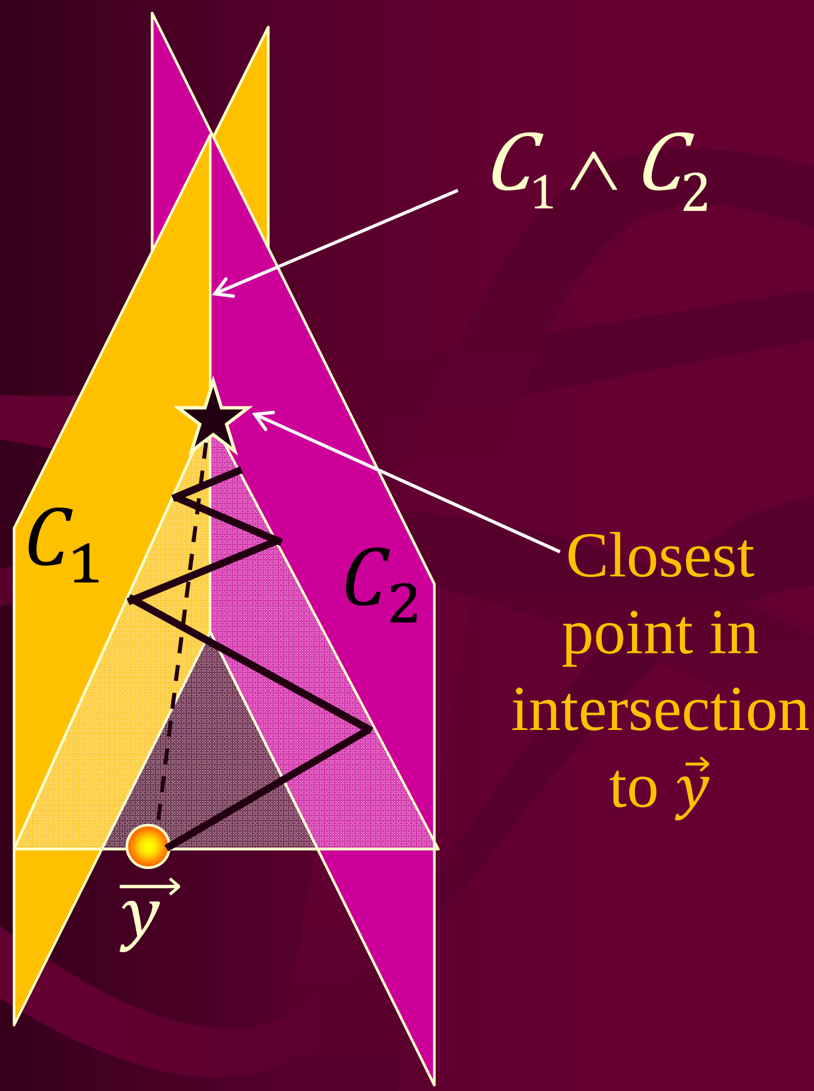


Consider two or more intersecting hyperplanes. (e.g. identical middles, constant area, bandlimited) and an external signal $\vec{y}$

There is a unique point in the intersection that is closest to $\vec{y}$

Alternating projections will converge to that point.

Von Neumann's Alternating Projection Algorithm

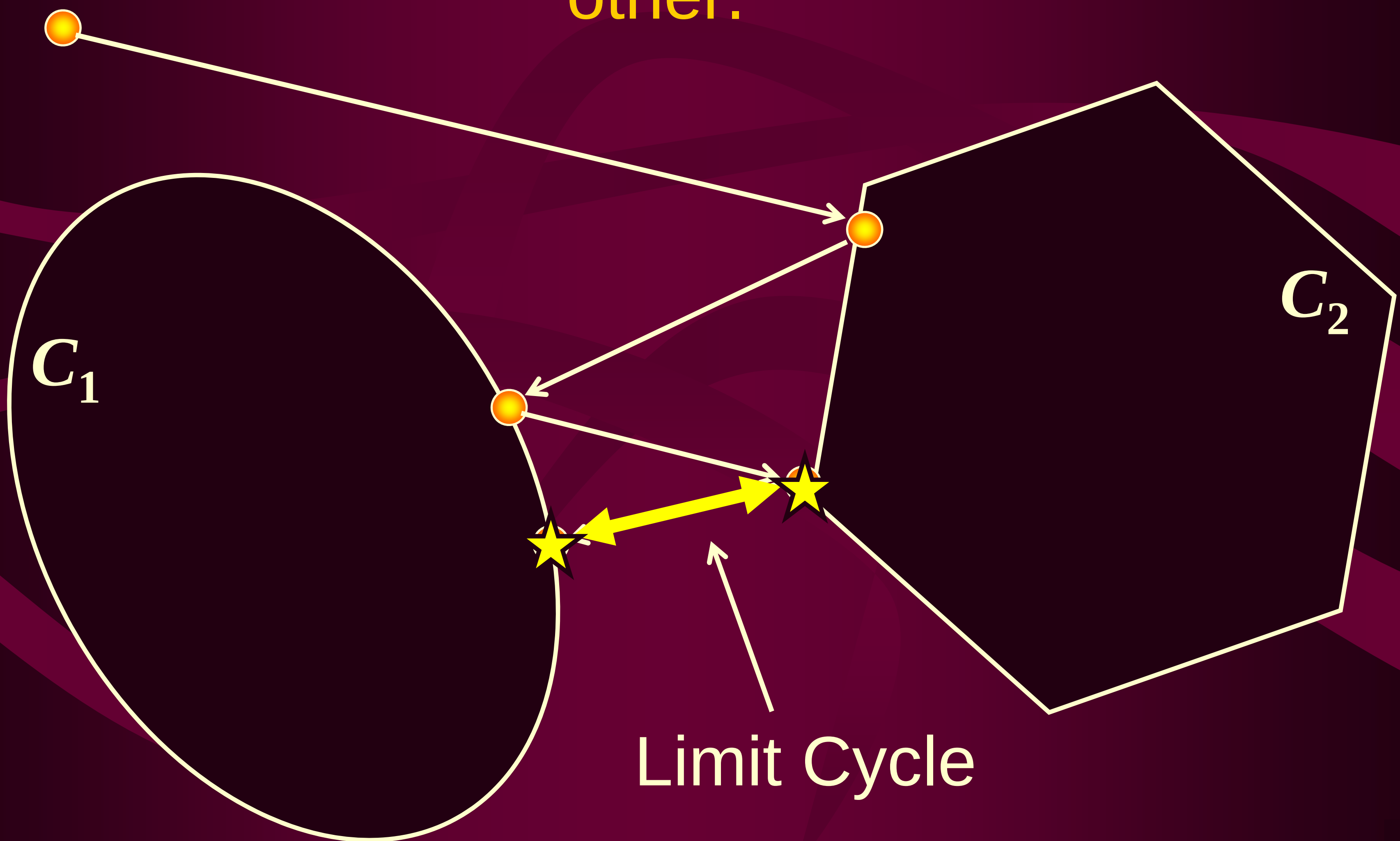


Consider two or more intersecting hyperplanes. (e.g. identical middles, constant area, bandlimited) and an external signal $\vec{y}$

There is a unique point in the intersection that is closest to $\vec{y}$

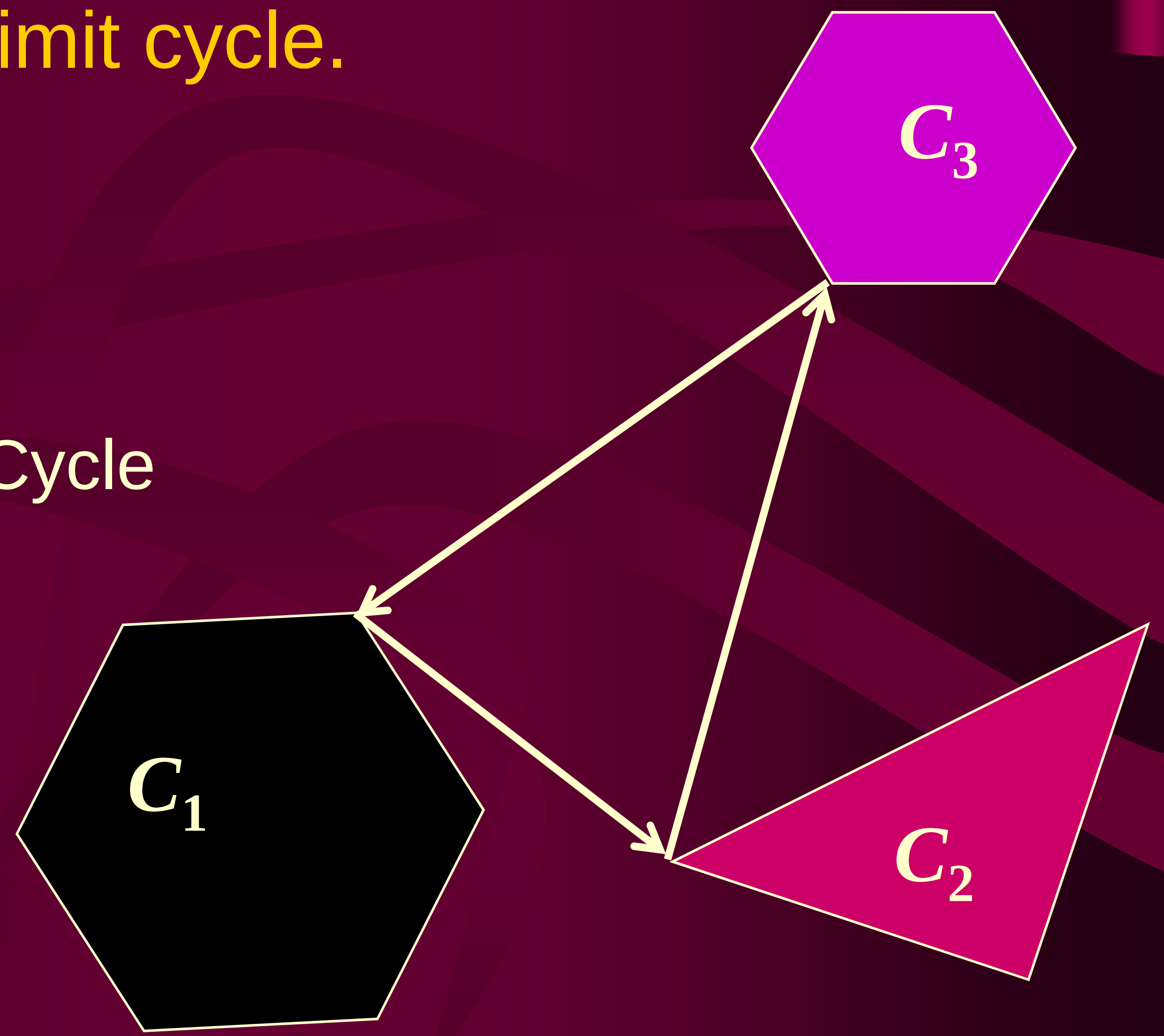
Alternating projections will converge to that point.

Lemma 2: Alternating POCS among 2 nonintersecting convex sets will converge to the point in each set closest to the other.



Lemma 3: Alternating POCS among 3 or more convex sets with an empty intersection will converge to a limit cycle.

1→ 2 → 3 Limit Cycle

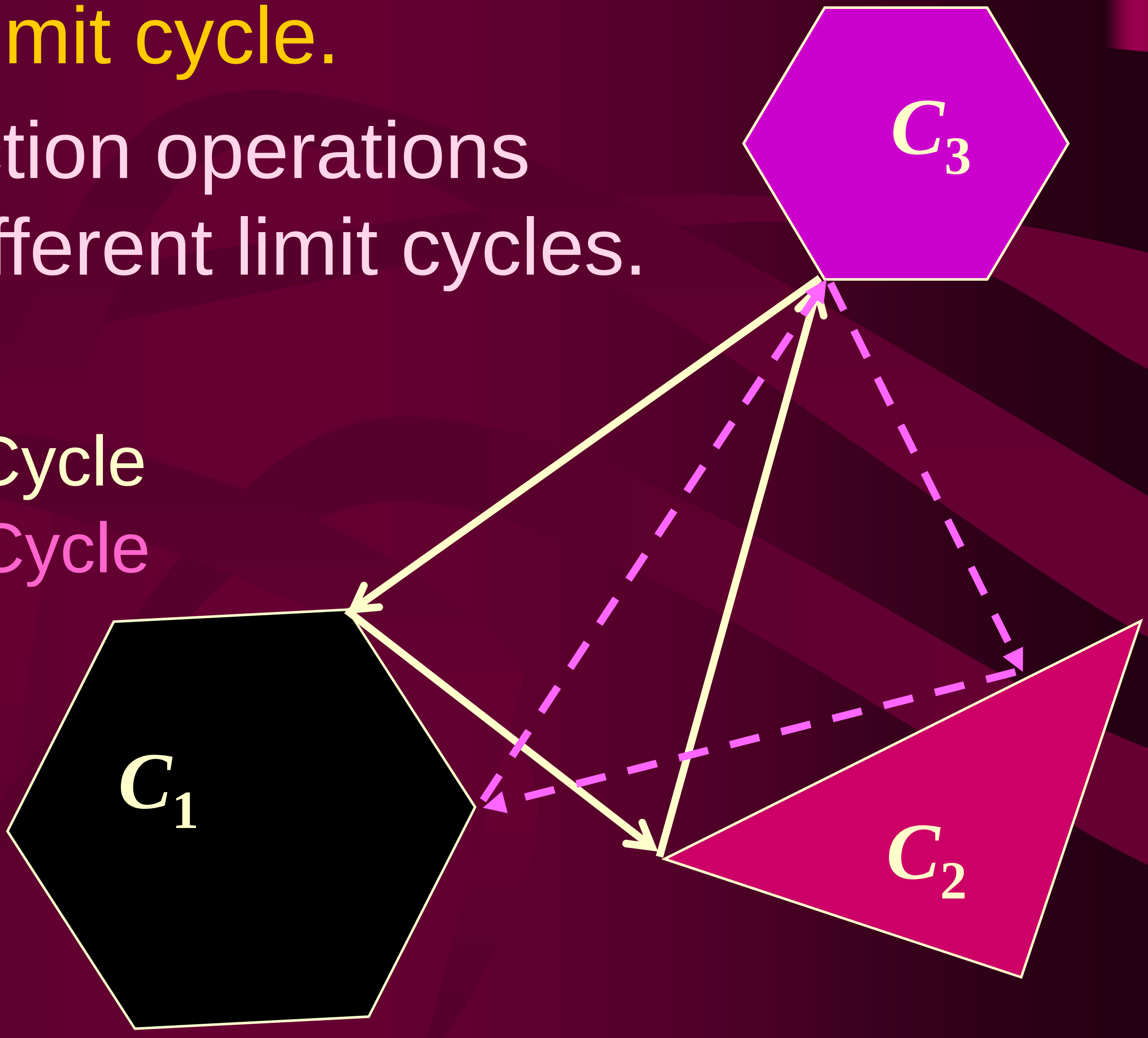


Lemma 3: Alternating POCS among 3 or more convex sets with an empty intersection will converge to a limit cycle.

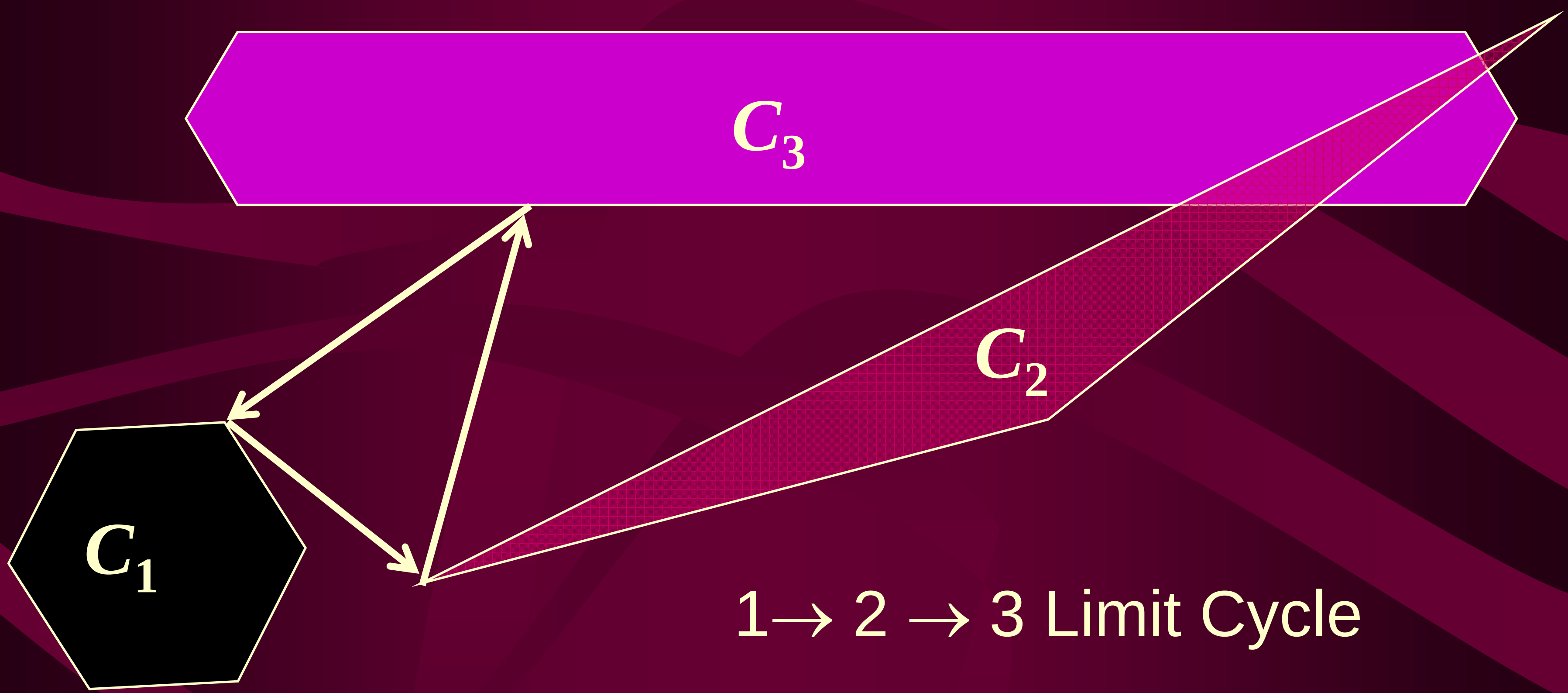
Different projection operations can produce different limit cycles.

$1 \rightarrow 2 \rightarrow 3$ Limit Cycle

$3 \rightarrow 2 \rightarrow 1$ Limit Cycle

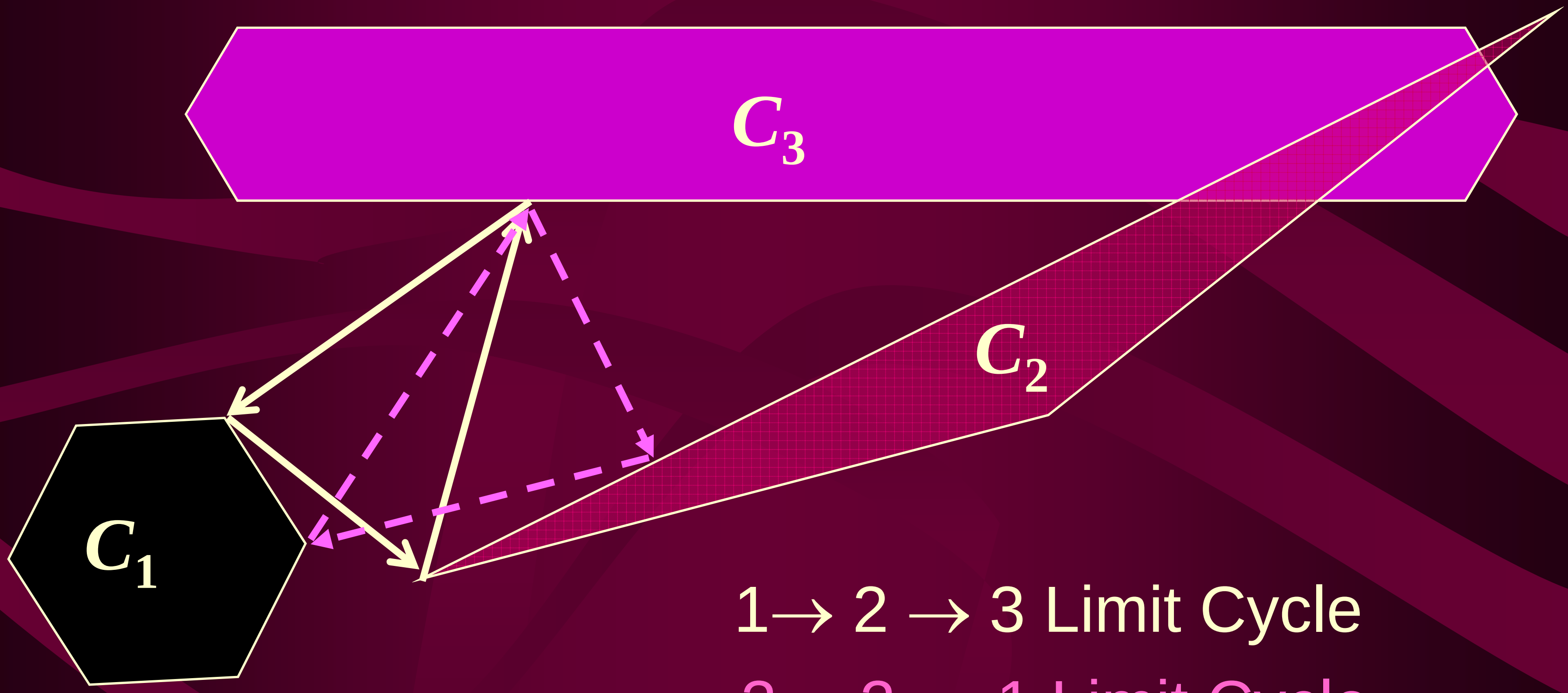


Lemma 3: (cont) POCS with non-empty intersection.



1→ 2 → 3 Limit Cycle

Lemma 3: (cont) POCS with non-empty intersection.

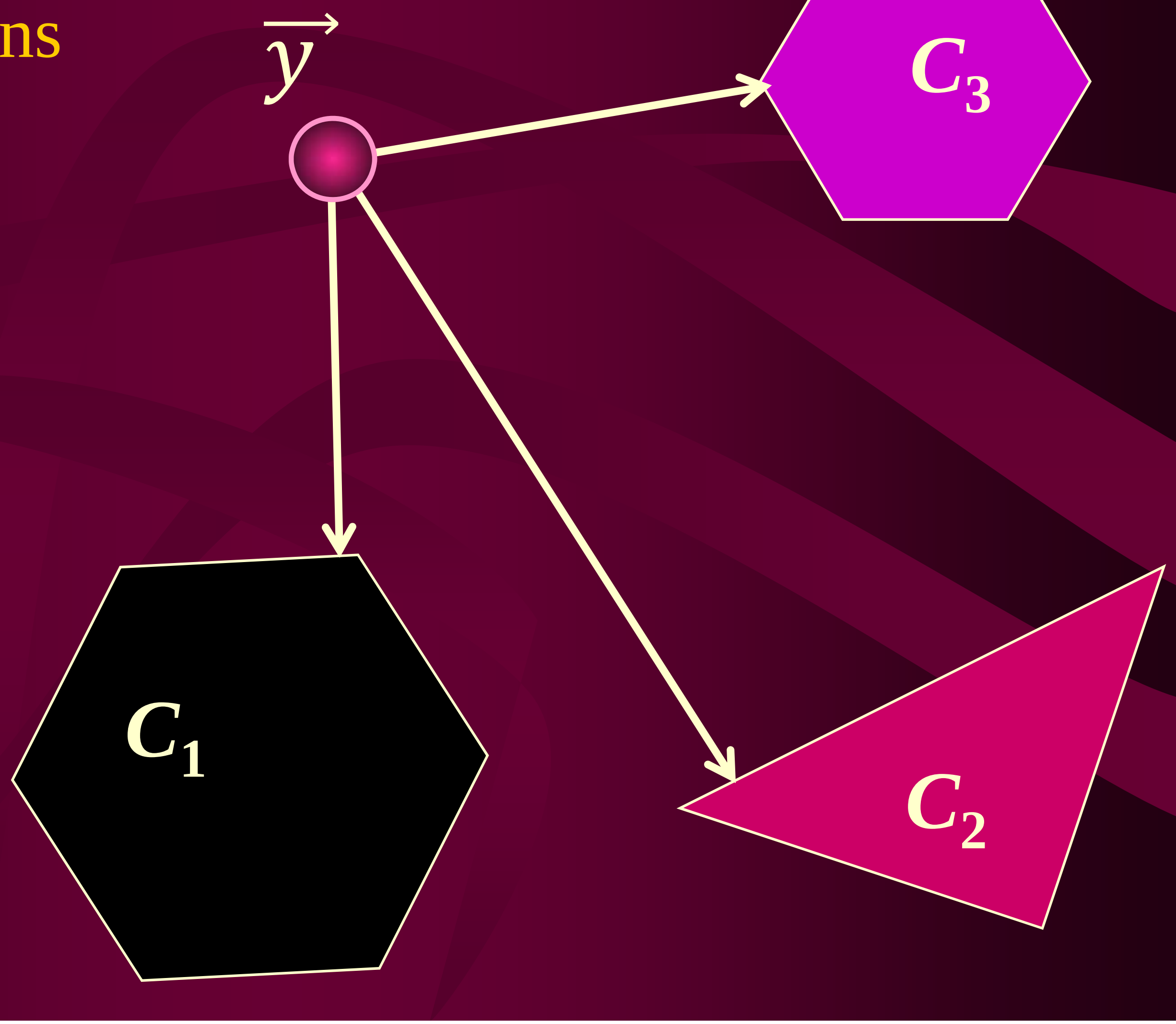


$1 \rightarrow 2 \rightarrow 3$ Limit Cycle

$3 \rightarrow 2 \rightarrow 1$ Limit Cycle

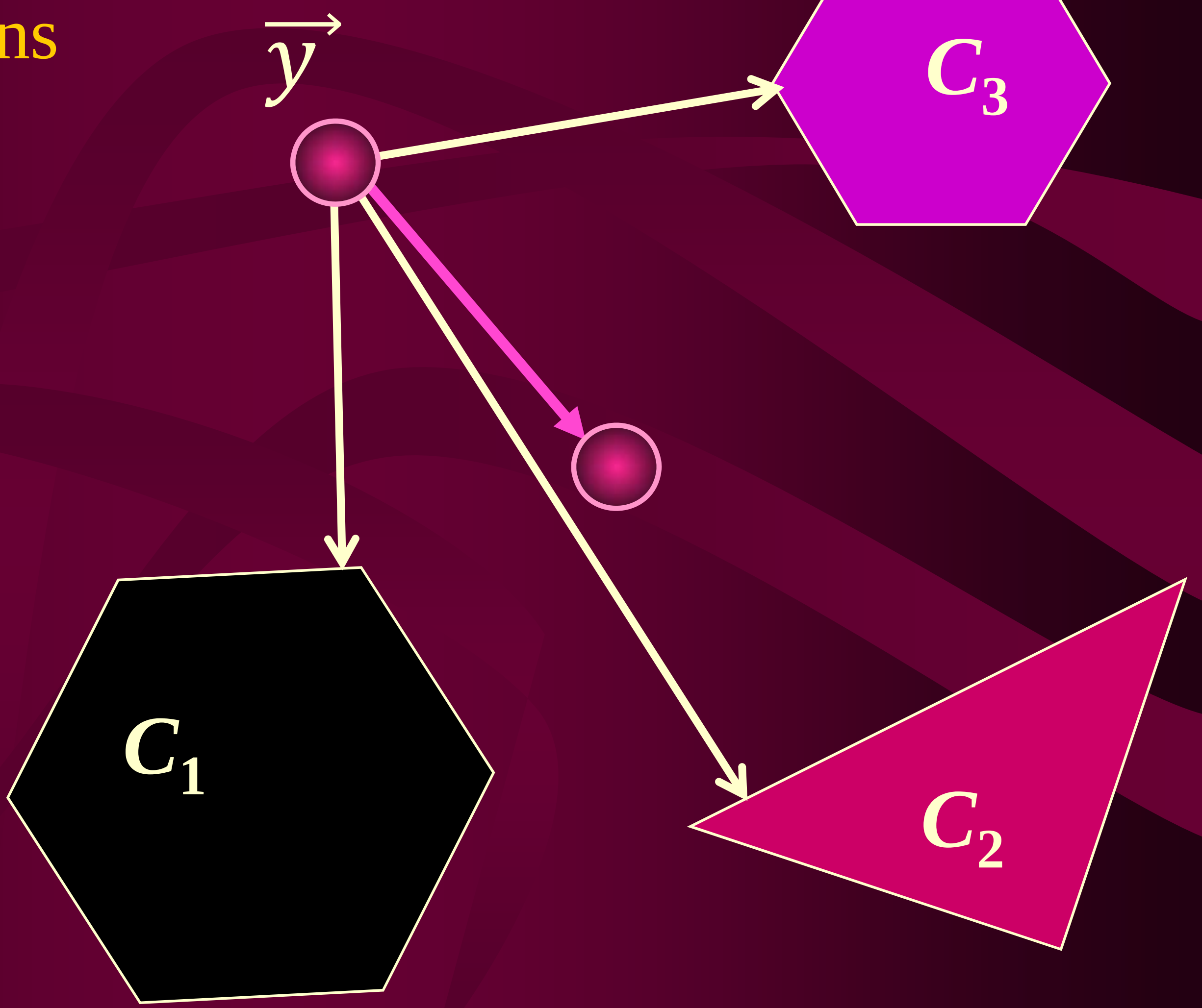
Simultaneous Weighted Projections

1. Take Projections
2.
3.



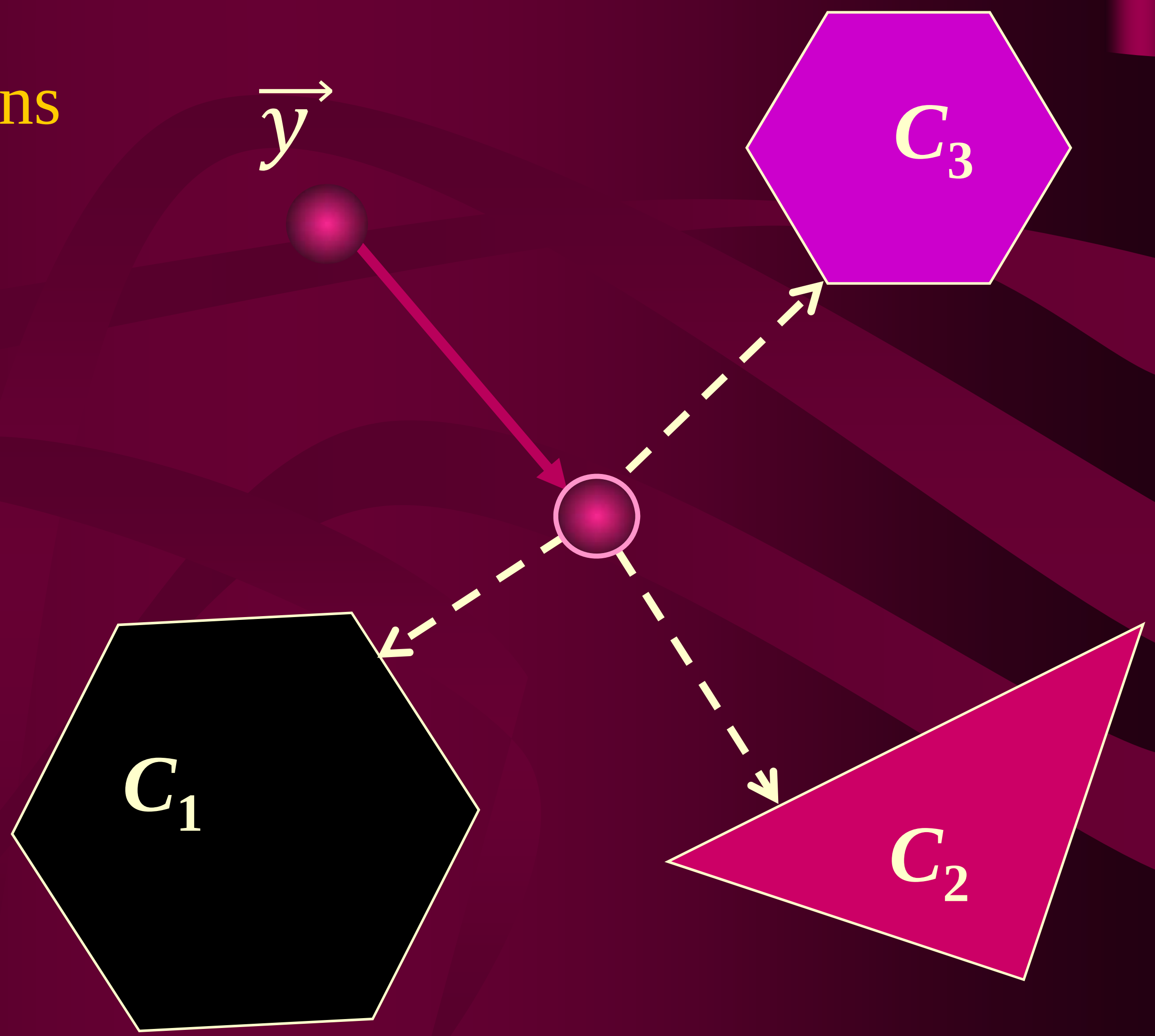
Simultaneous Weighted Projections

1. Take Projections
2. Average
3.



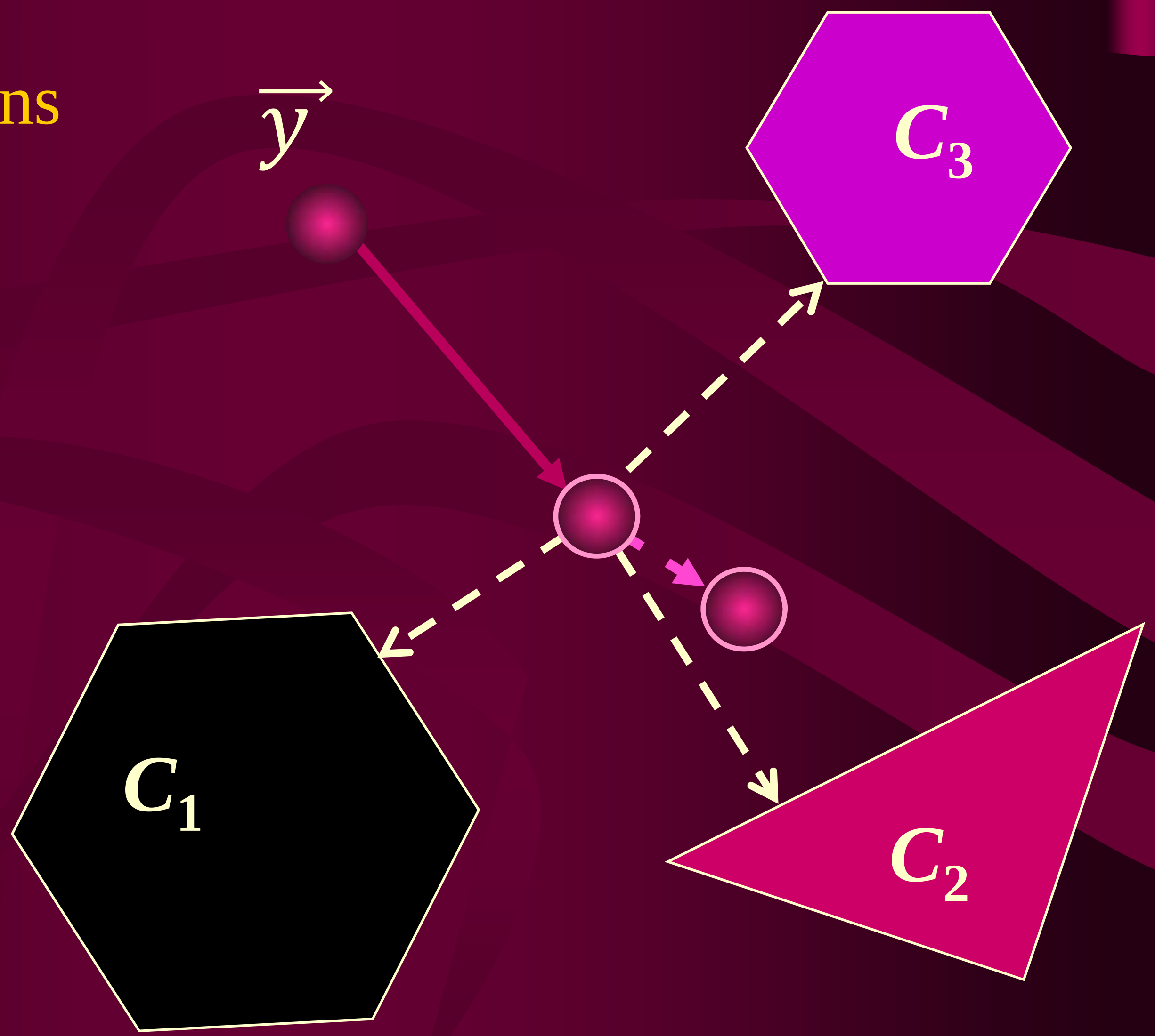
Simultaneous Weighted Projections

1. Take Projections
2. Average
3. Repeat



Simultaneous Weighted Projections

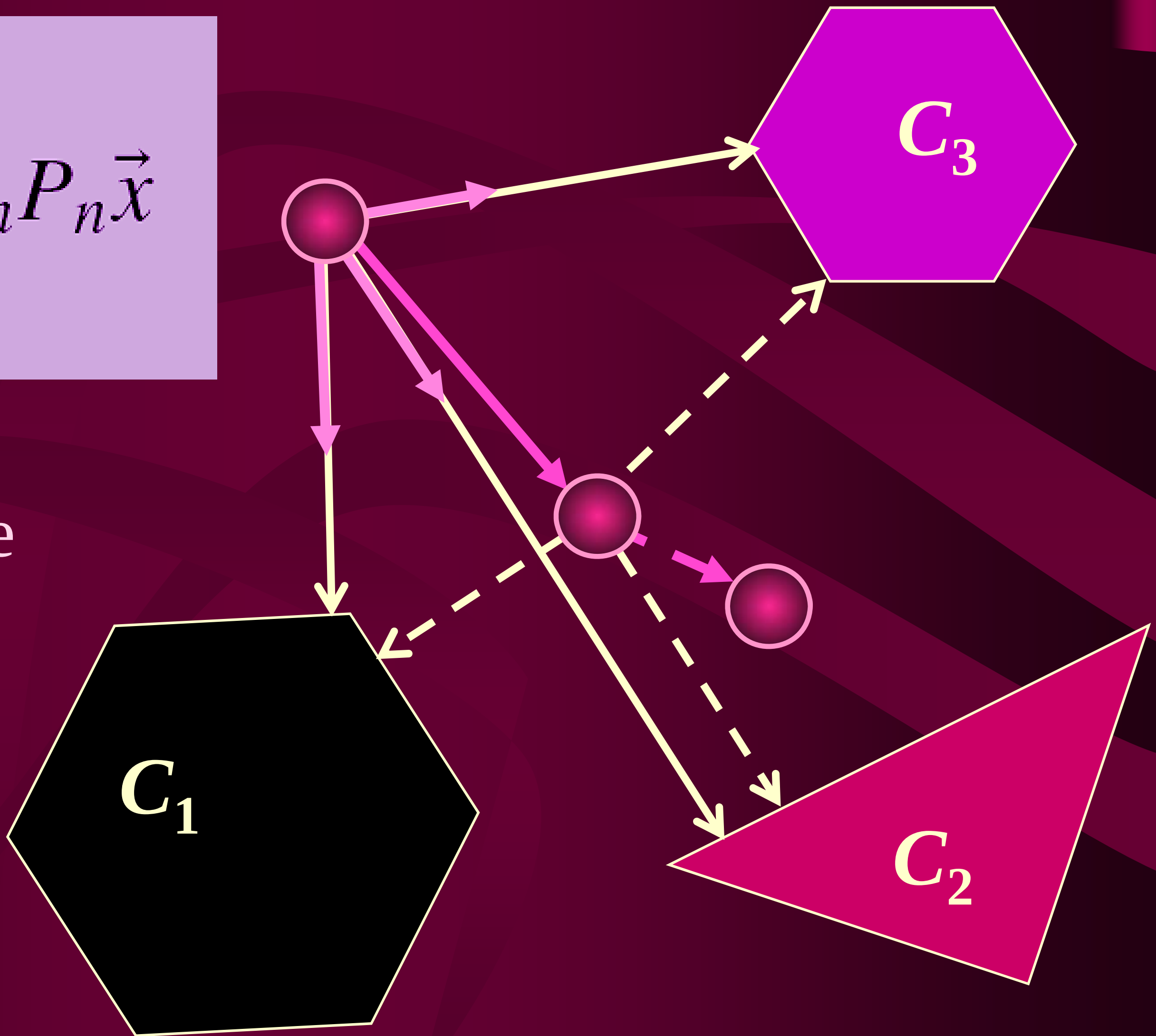
1. Take Projections
2. Average
3. Repeat



Simultaneous Weighted Projections

$$P_S \vec{x} := \sum_{n=1}^{N} w_n P_n \vec{x}$$

- Can also use weighted average
- Average uses $w_n = \frac{1}{N}$
- Gives Minimum Mean Square Error Solution



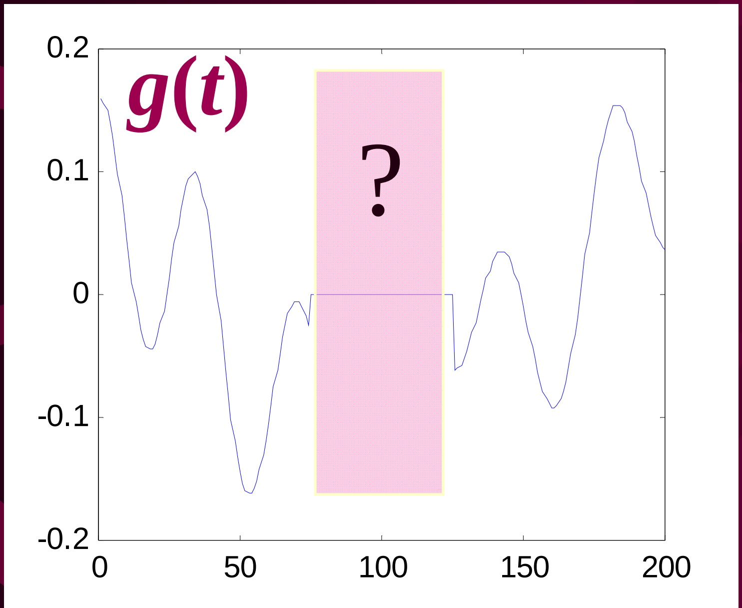
Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

Application:

The Papoulis-Gerchberg Algorithm

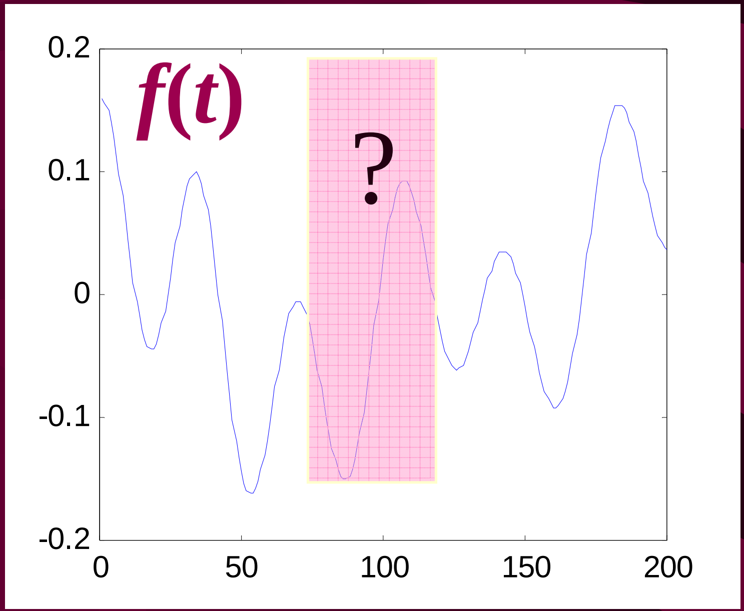
- Restoration of lost data in a band limited function:



BL

$\Rightarrow$

?

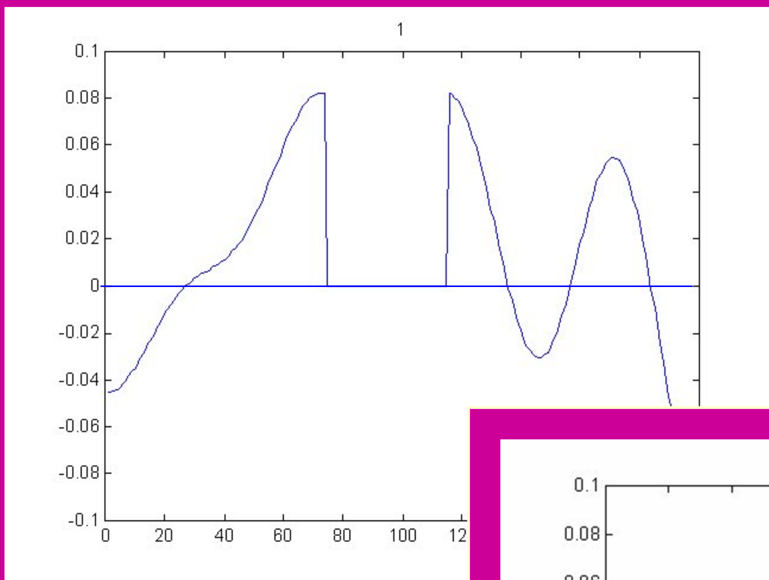


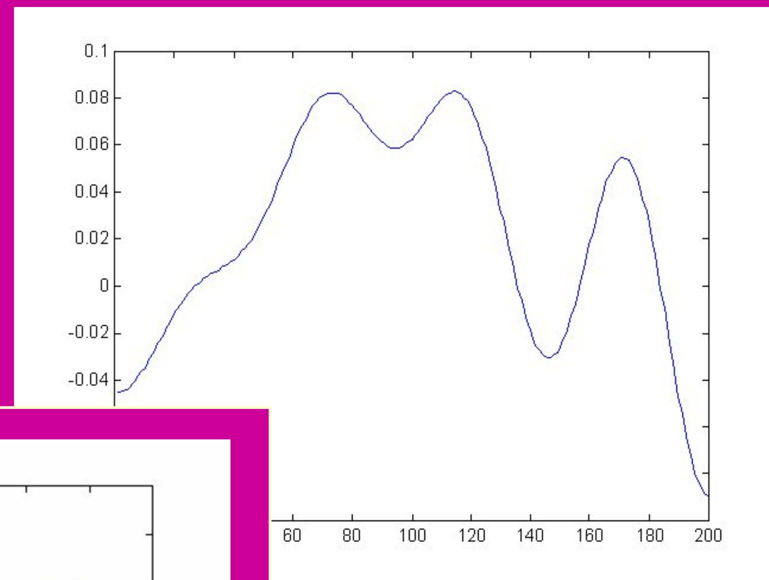
Convex Sets: (1) Identical Tails and (2) Bandlimited.

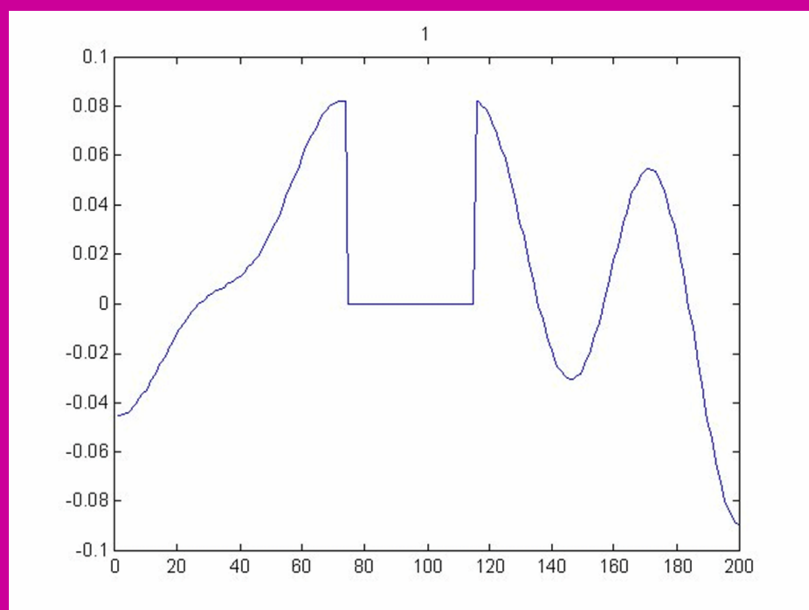
Application:

The Papoulis-Gerchberg Algorithm

Example

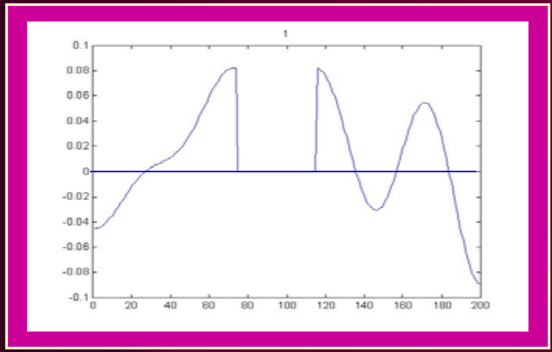






Application:

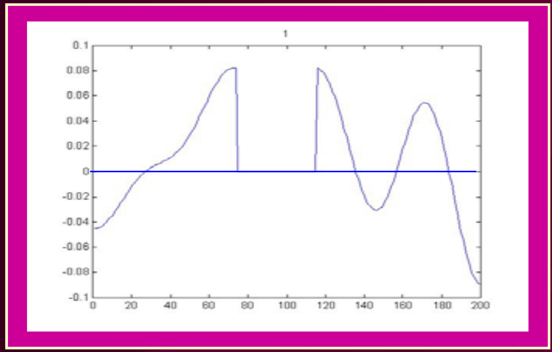
The Papoulis-Gerchberg Algorithm



Problem: Ill-posed.
Ill-conditioned

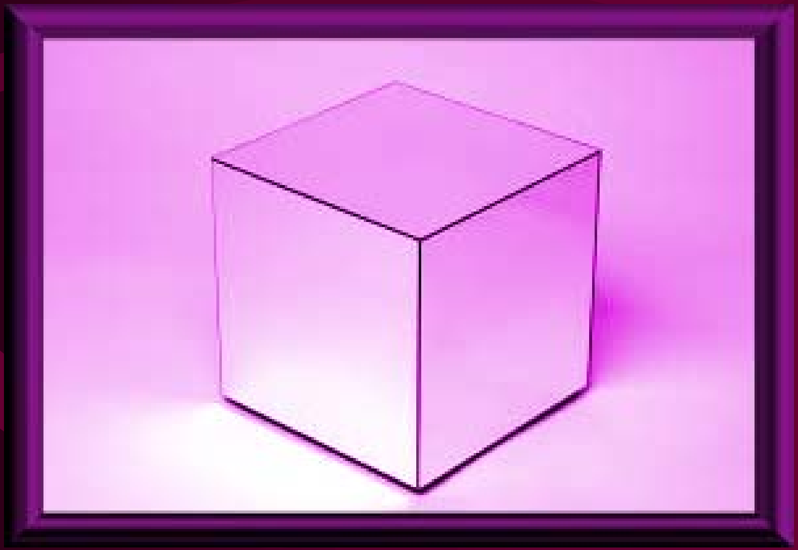
Application:

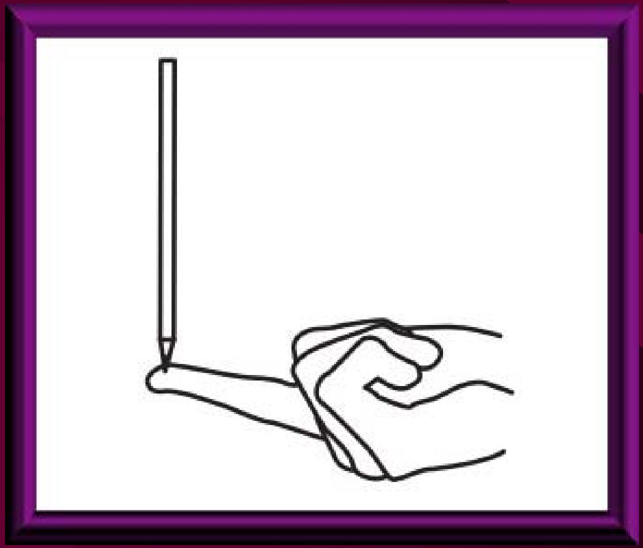
The Papoulis-Gerchberg Algorithm



Problem: Ill-posed.
Ill-conditioned

Example: Sum of forces =0. Therefore no motion (?)





Problem: noise

Ill-Conditioned Matrix

$$\mathbf{A} = \begin{bmatrix} 0.9562 & 0.6889 & 1.0230 & 0.4380 & 0.8832 \\ 0.6889 & 0.6044 & 0.9112 & 0.2860 & 0.7146 \\ 1.0230 & 0.9112 & 1.3752 & 0.4210 & 1.0710 \\ 0.4380 & 0.2860 & 0.4210 & 0.2087 & 0.3832 \\ 0.8832 & 0.7146 & 1.0710 & 0.3832 & 0.8725 \end{bmatrix}$$

$$\mathbf{A}^{-1} = 10^8 \times \begin{bmatrix} 0.8267 & 1.5639 & -0.5038 & -0.5633 & -1.2519 \\ 0.5816 & 1.1744 & -2.0727 & -2.4462 & 2.0681 \\ 0.0314 & -0.8723 & 0.9632 & 0.5381 & -0.7361 \\ -0.5853 & -2.1954 & 0.3746 & -0.3287 & 2.0751 \\ -1.0947 & -0.5101 & 0.8607 & 2.0575 & -0.4342 \end{bmatrix}.$$

close to singular

Ill-Conditioned Matrix

$$\mathbf{A}\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}$$

$$\mathbf{A}^{-1}\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}=\begin{bmatrix}1.0000\\1.0000\\1.0000\\1.0000\\1.0000\end{bmatrix}.$$

Ill-Conditioned Matrix

$$\mathbf{A}\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}$$

$$\mathbf{A}^{-1}\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}=\begin{bmatrix}1.0000\\1.0000\\1.0000\\1.0000\\1.0000\end{bmatrix}.$$

$$\mathbf{A}^{-1}\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9246\end{bmatrix}=\begin{bmatrix}-14,318\\23,657\\-8419\\23,736\\-4966\end{bmatrix}.$$

Ill-Conditioned Matrix

$$\mathbf{A}\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}$$

$$\mathbf{A}^{-1}\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9245\end{bmatrix}=\begin{bmatrix}1.0000\\1.0000\\1.0000\\1.0000\\1.0000\end{bmatrix}.$$

$$\mathbf{A}^{-1}\begin{bmatrix}3.9893\\3.2052\\4.8014\\1.7368\\3.9246\end{bmatrix}=\begin{bmatrix}-14,318\\23,657\\-8419\\23,736\\-4966\end{bmatrix}.$$

$$\text{cond}\,(\mathbf{A})=\frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

$$=1.8874\times 10^{9}$$

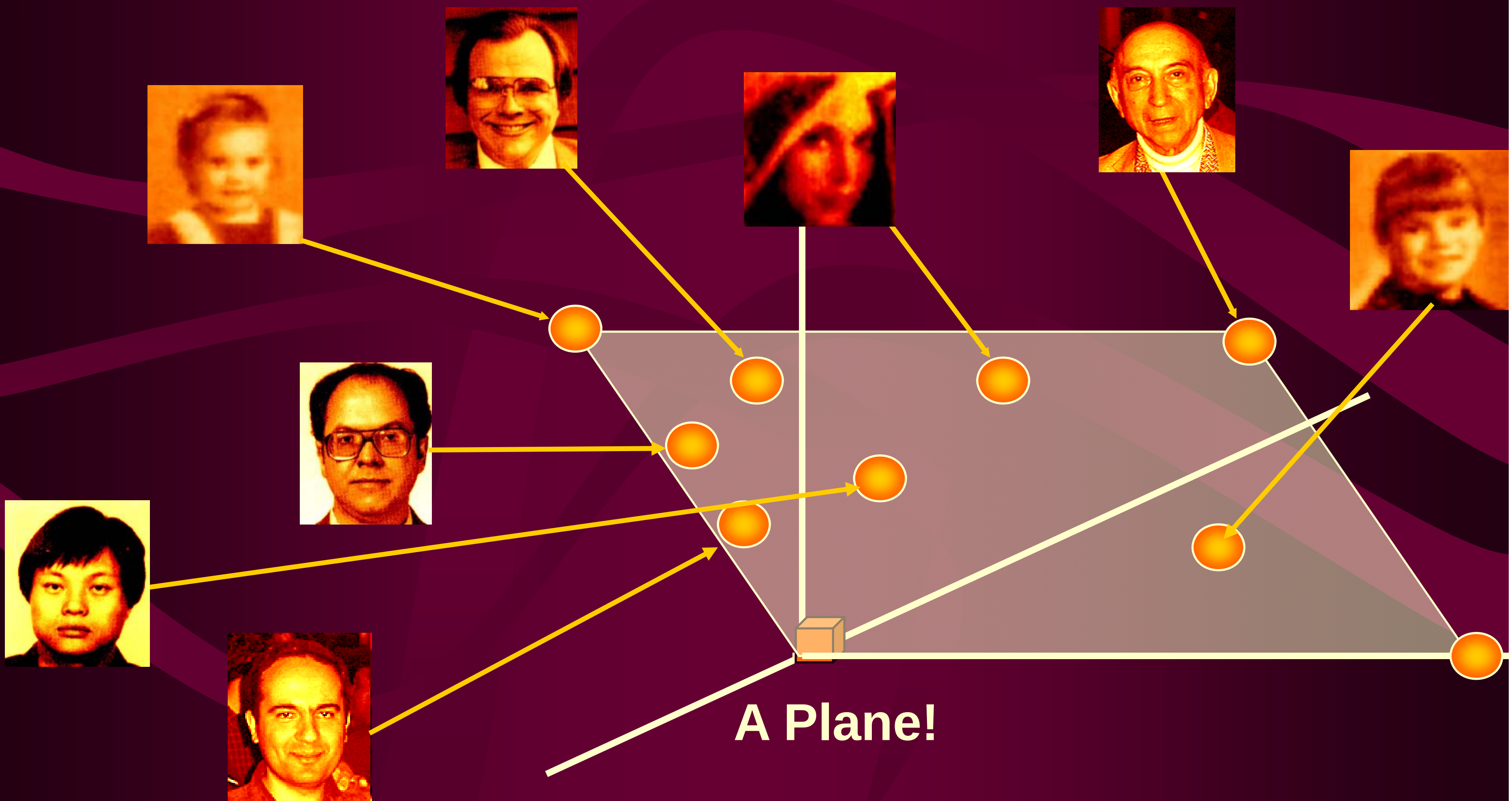
Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

Application:

Neural Network Associative Memory

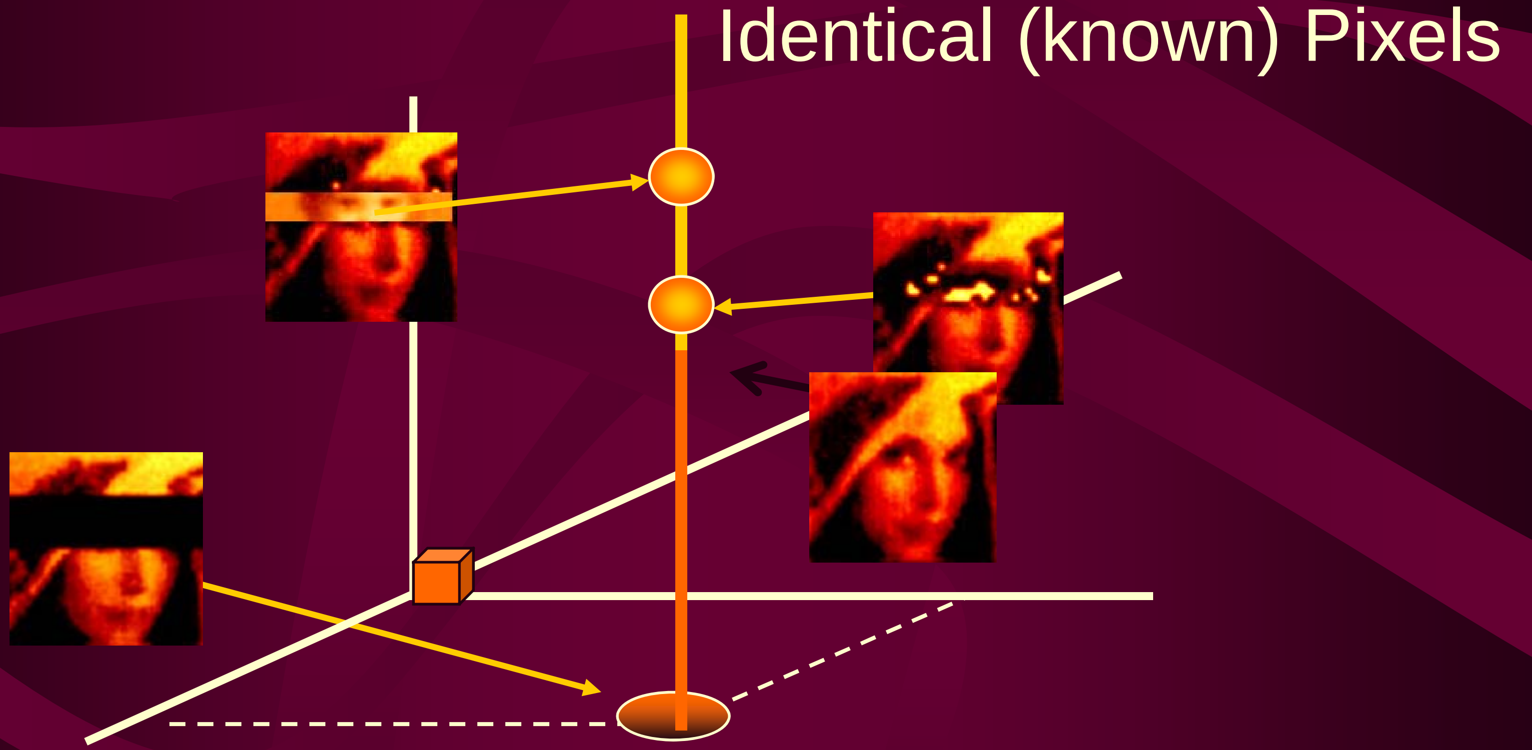
Placing the Library in Hilbert Space



Application:

Neural Network Associative Memory

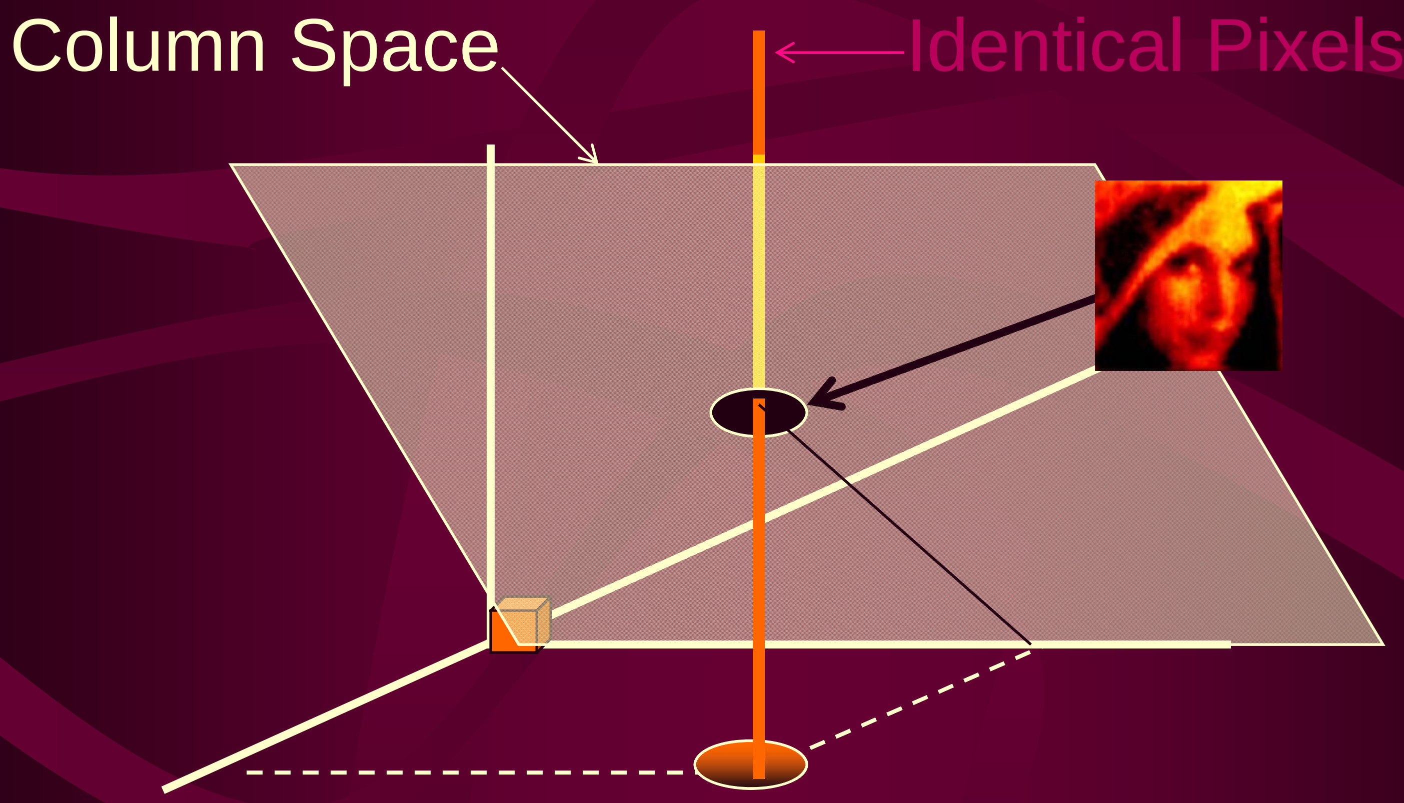
Associative Recall



Application:

Neural Network Associative Memory

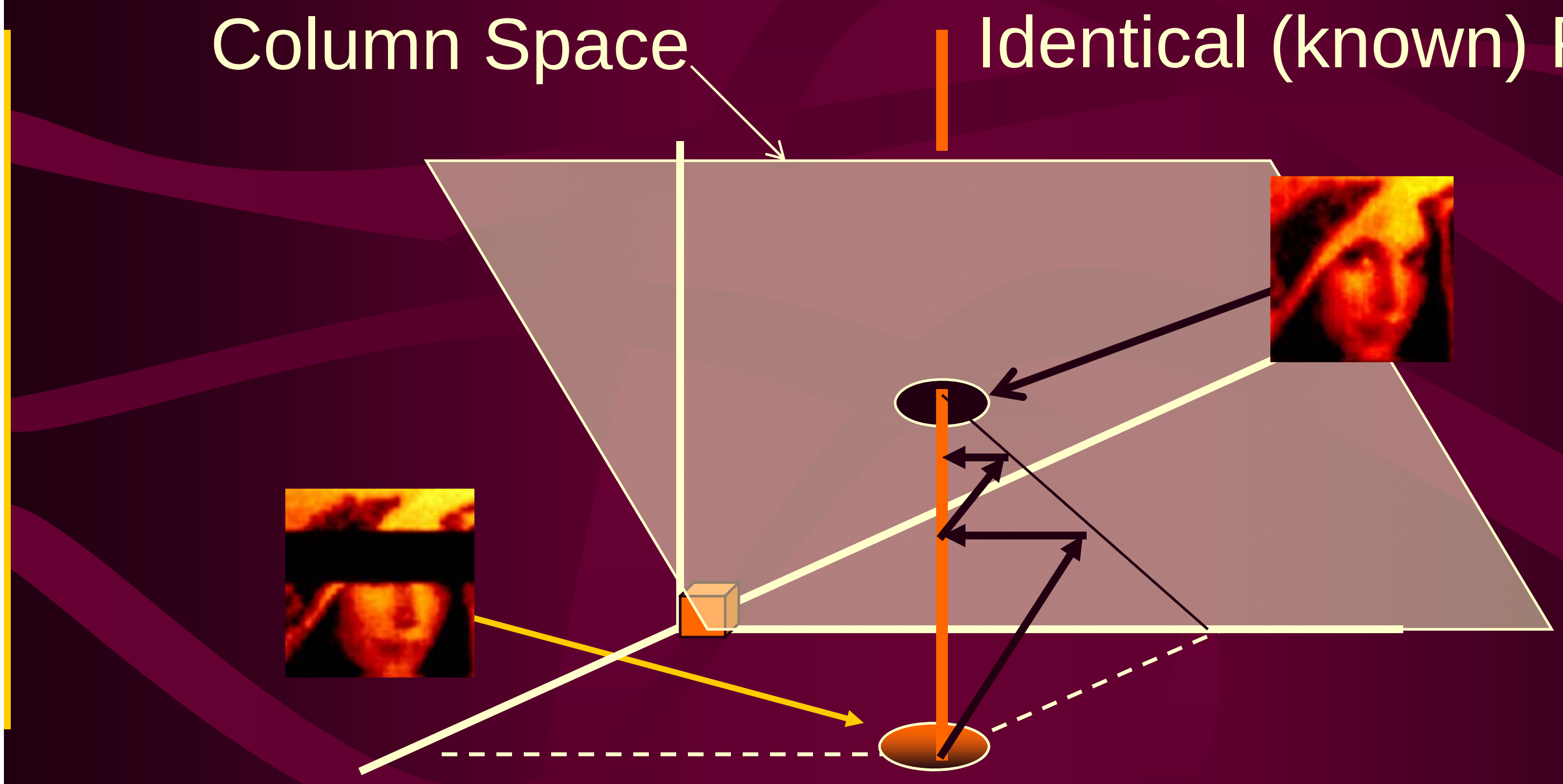
Associative Recall by POCS



Application:

Neural Network Associative Memory

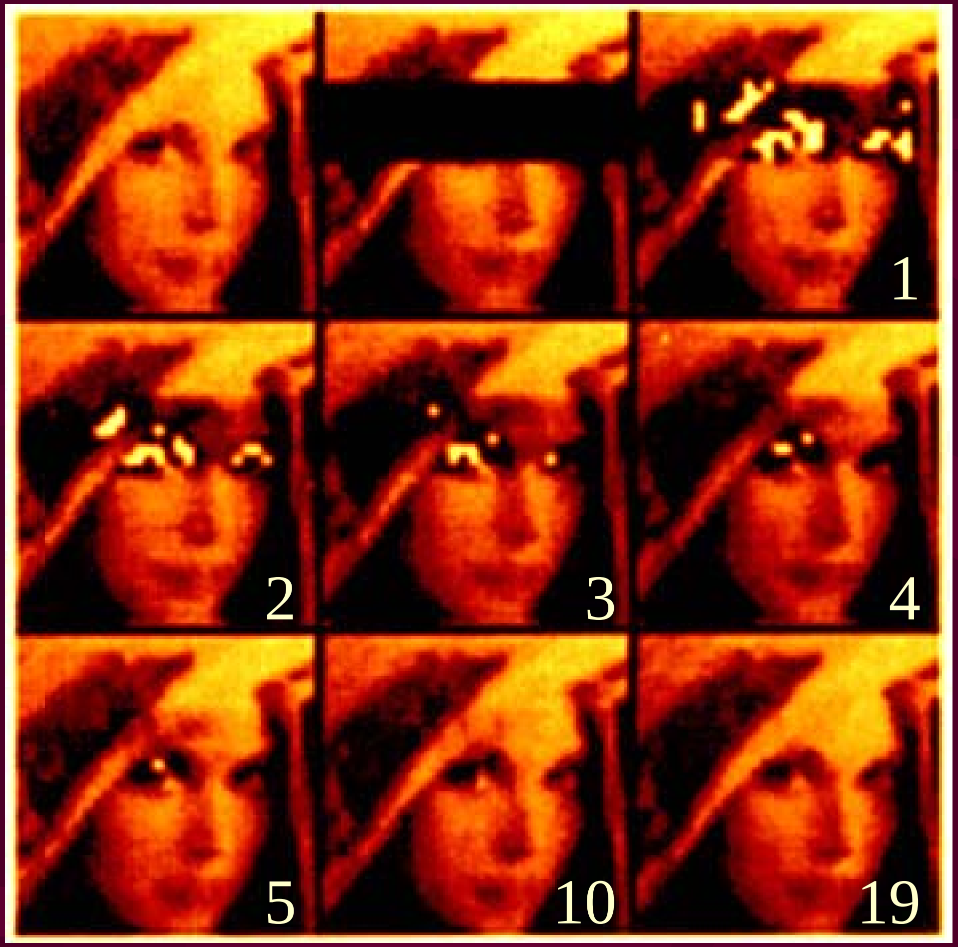
Associative Recall by POCS



Application:

Neural Network Associative Memory

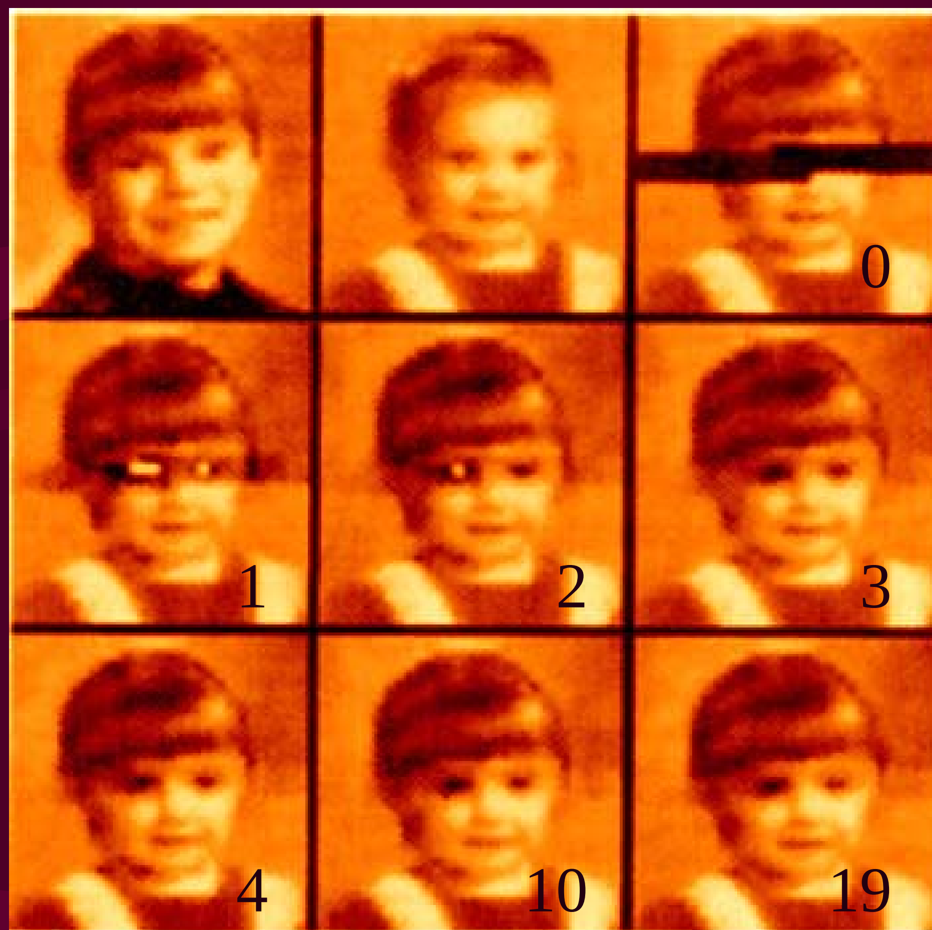
Example



Application:

Neural Network Associative Memory

Example



A Library of Mathematicians

A Library of Mathematicians

What does the Average Mathematician Look Like?

Convergence As a Function of Percent of Known Image

Convergence As a Function of Library Size

Convergence As a Function of Noise Level

Combining Euler & Hermite

Combining Euler & Hermite

Not Of This Library

Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

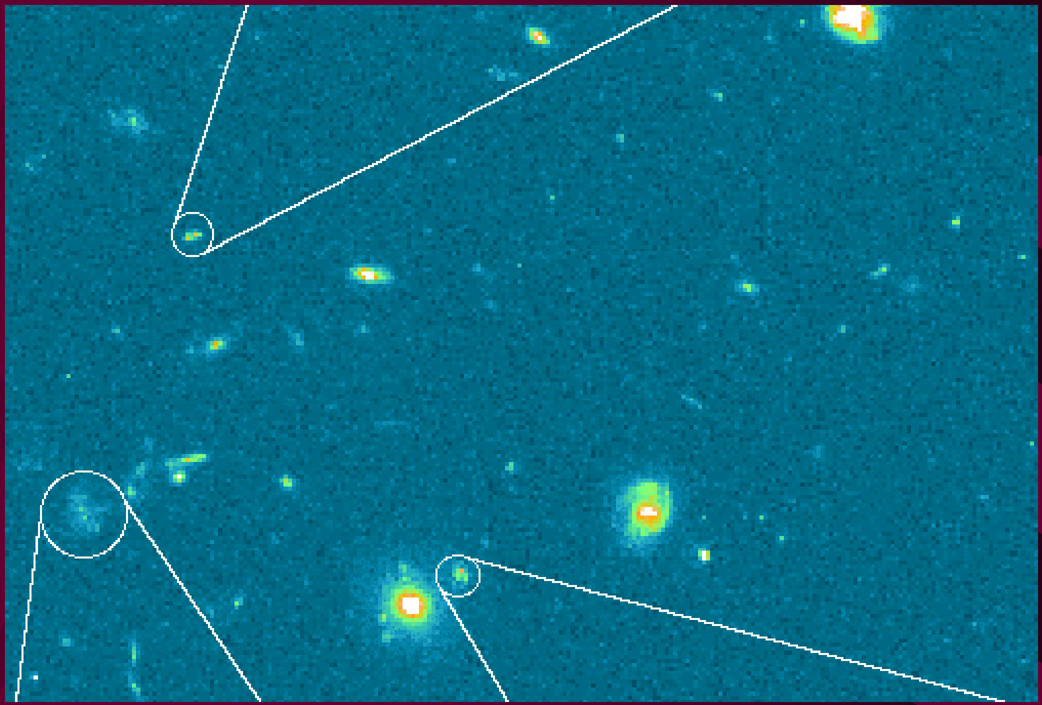
Application:

Combining Low Resolution Sub-Pixel Shifted Images

http://www.stecf.org/newsletter/stecf-nl-22/adorf/adorf.html

Hans-Martin Adorf

Problem: Multiple Images of Galazy Clusters with subpixel shifts.



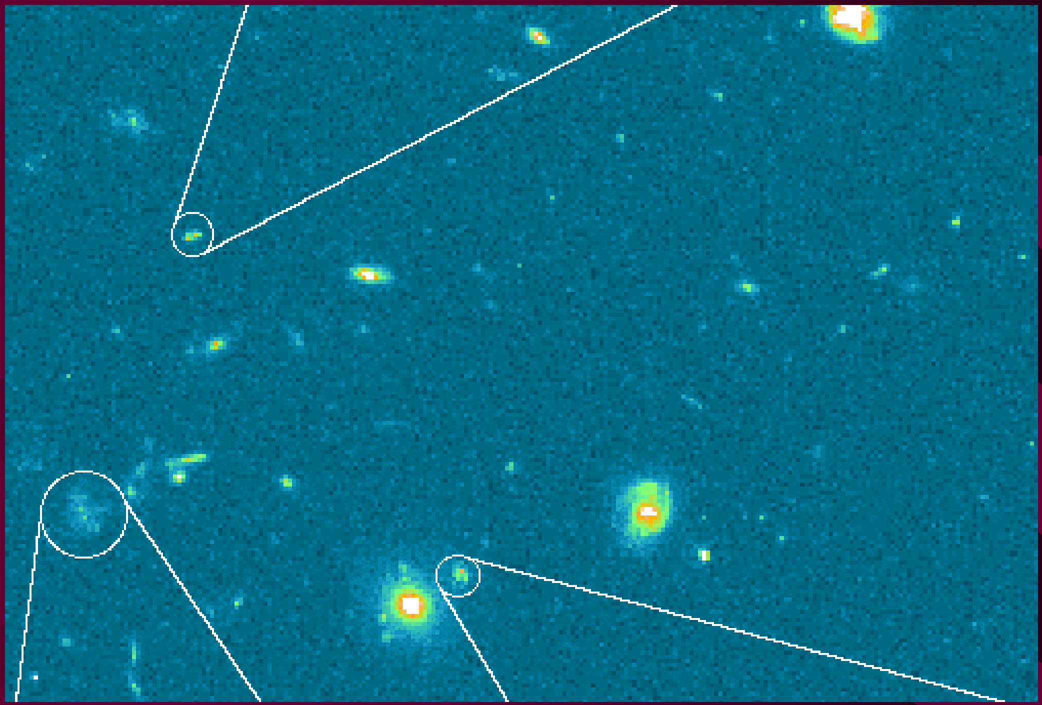
Application:

Combining Low Resolution Sub-Pixel Shifted Images

http://www.stecf.org/newsletter/stecf-nl-22/adorf/adorf.html

Hans-Martin Adorf

Problem: Multiple Images of Galazy Clusters with subpixel shifts.



POCS Solution:

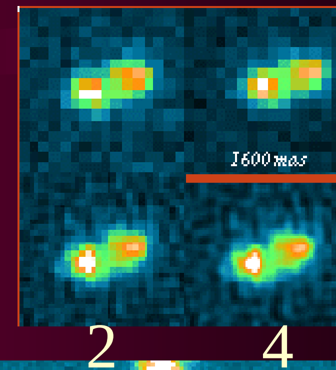
1. Sets of high resolution images giving rise to lower resolution images.
2. Positivity
3. Resolution Limit

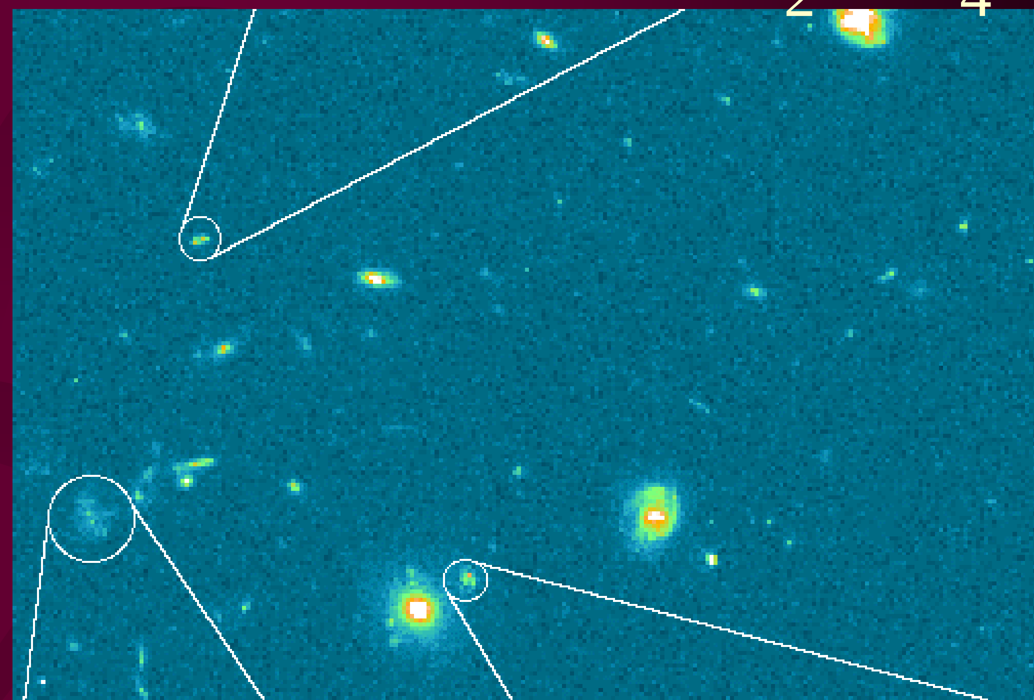
Application:

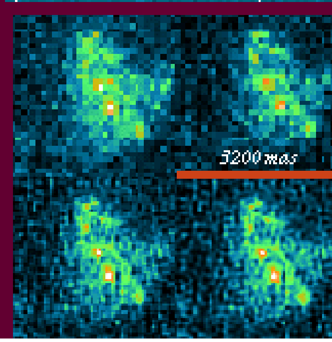
Combining Low Resolution Sub-Pixel Shifted Images

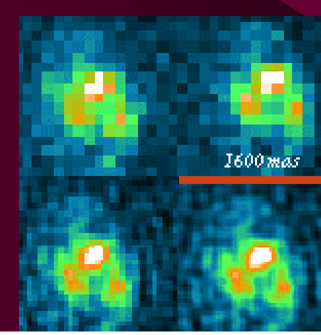
http://www.stecf.org/newsletter/stecf-nl-22/adorf/adorf.html

Hans-Martin Adorf









Problem: Multiple Images of Galazy Clusters with subpixel shifts.

POCS Solution:

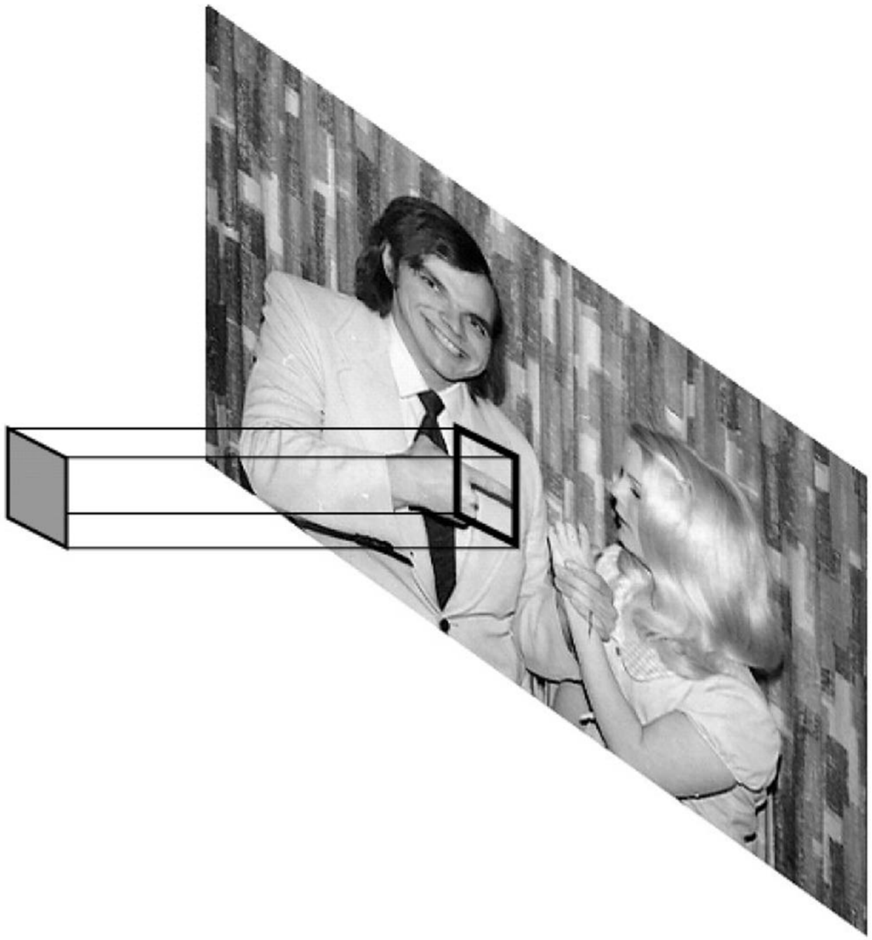
1. Sets of high resolution images giving rise to lower resolution images.
2. Positivity
3. Resolution Limit

Application:

Subpixel Resolution

using the world's worst camera

- Object is imaged with an aperture covering a large area. The average (or sum) of the object is measured as a single number over the aperture.
- The set of all images with this average value at this location is a convex set.
- The convex set is *constant area.*



Application:

Subpixel Resolution

Randomly chosen 8x8 pixel blocks. Over 6 Million projections.

Application:

Subpixel Resolution

Slow (7:41) Fast (3:00) Faster (1:30) Fastest (0:56)

Randomly chosen 8x8 pixel blocks. Over 6 Million projections.

Application:

Subpixel Resolution

Slow (0.58) Fast (0:29) Faster (0:14)

Blocks of random dimension <33 chosen at random locations. 2.7 Million blocks.

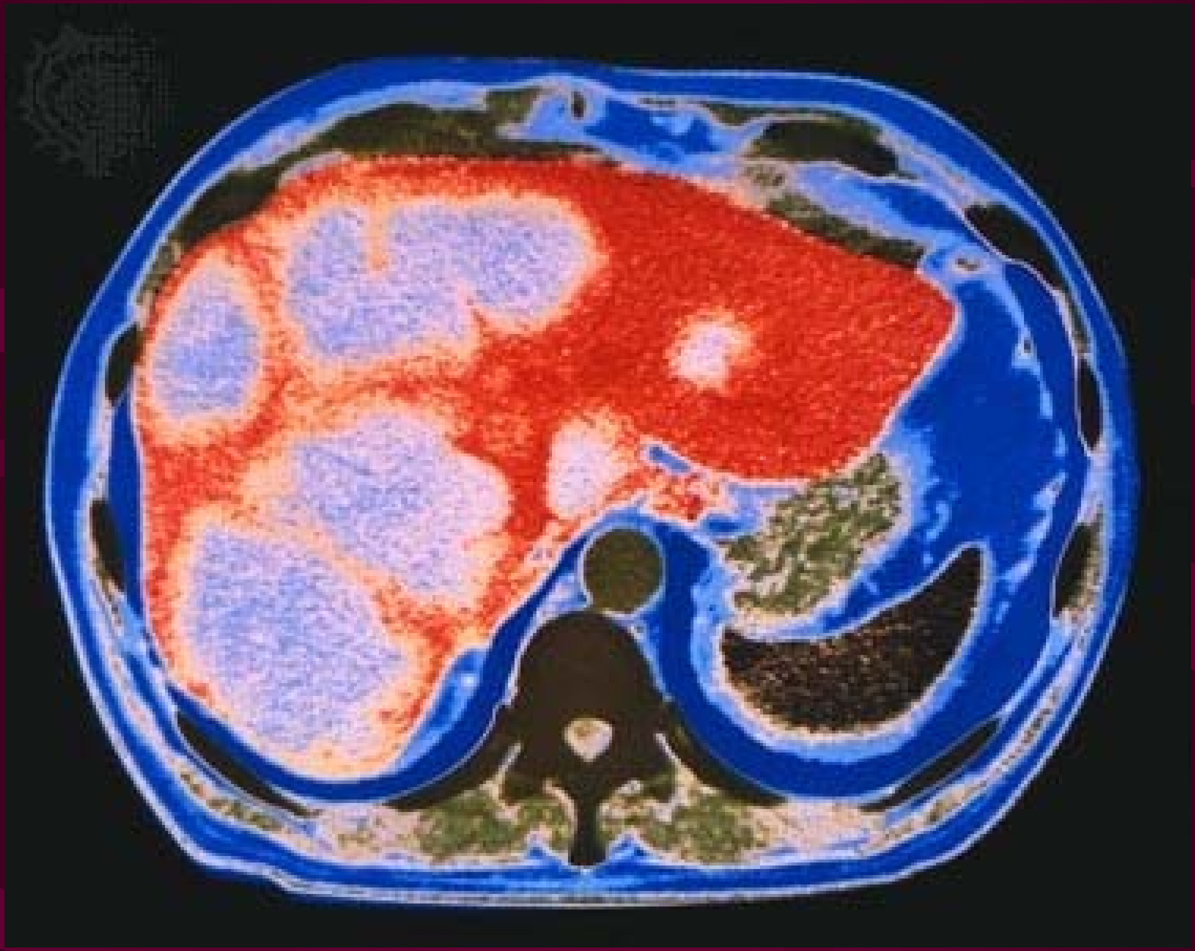
Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Syntesis
  - The Gerchberg-Saxton Algorithm

Application:

Tomography

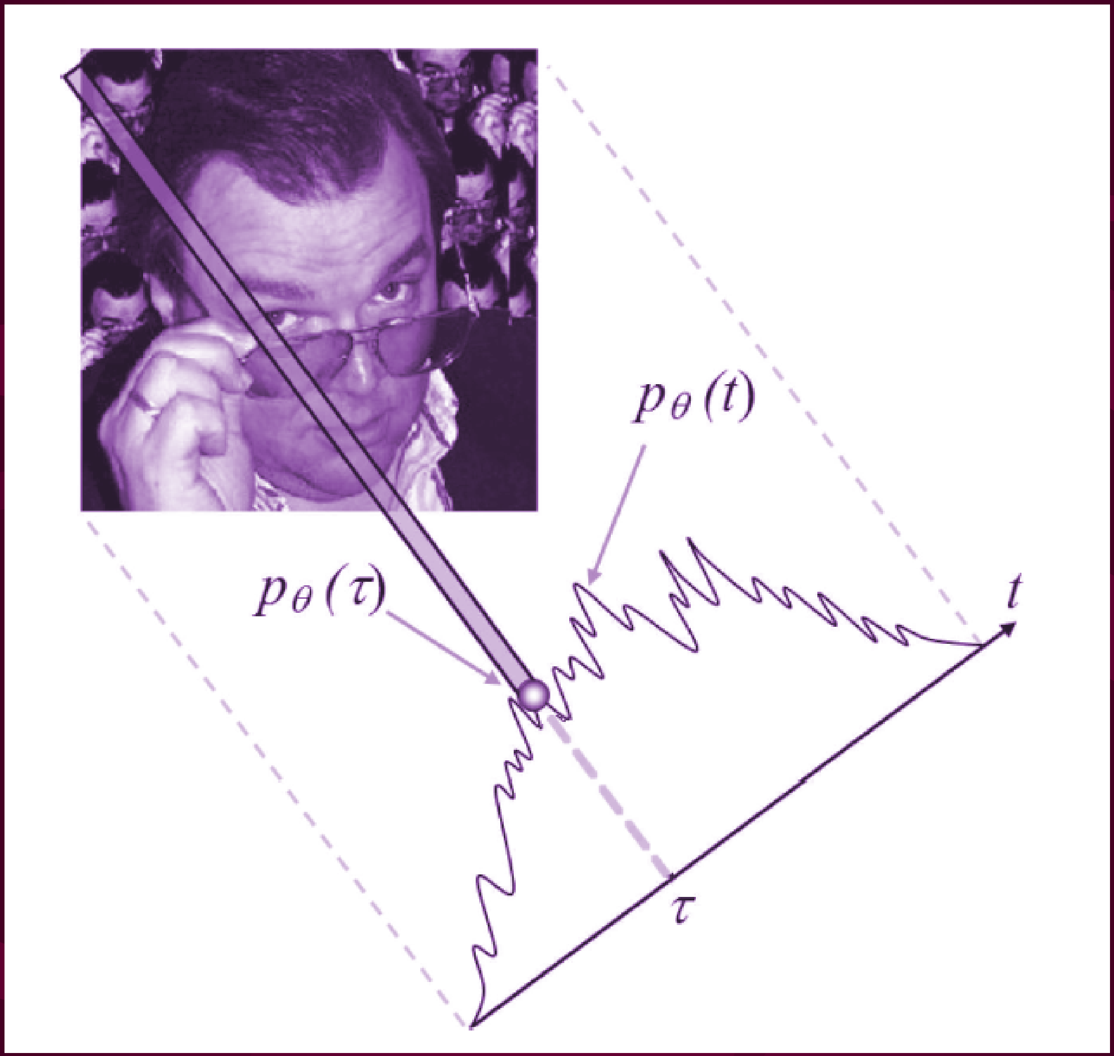
Same procedure as subpixel resolution.



Application:

Tomography

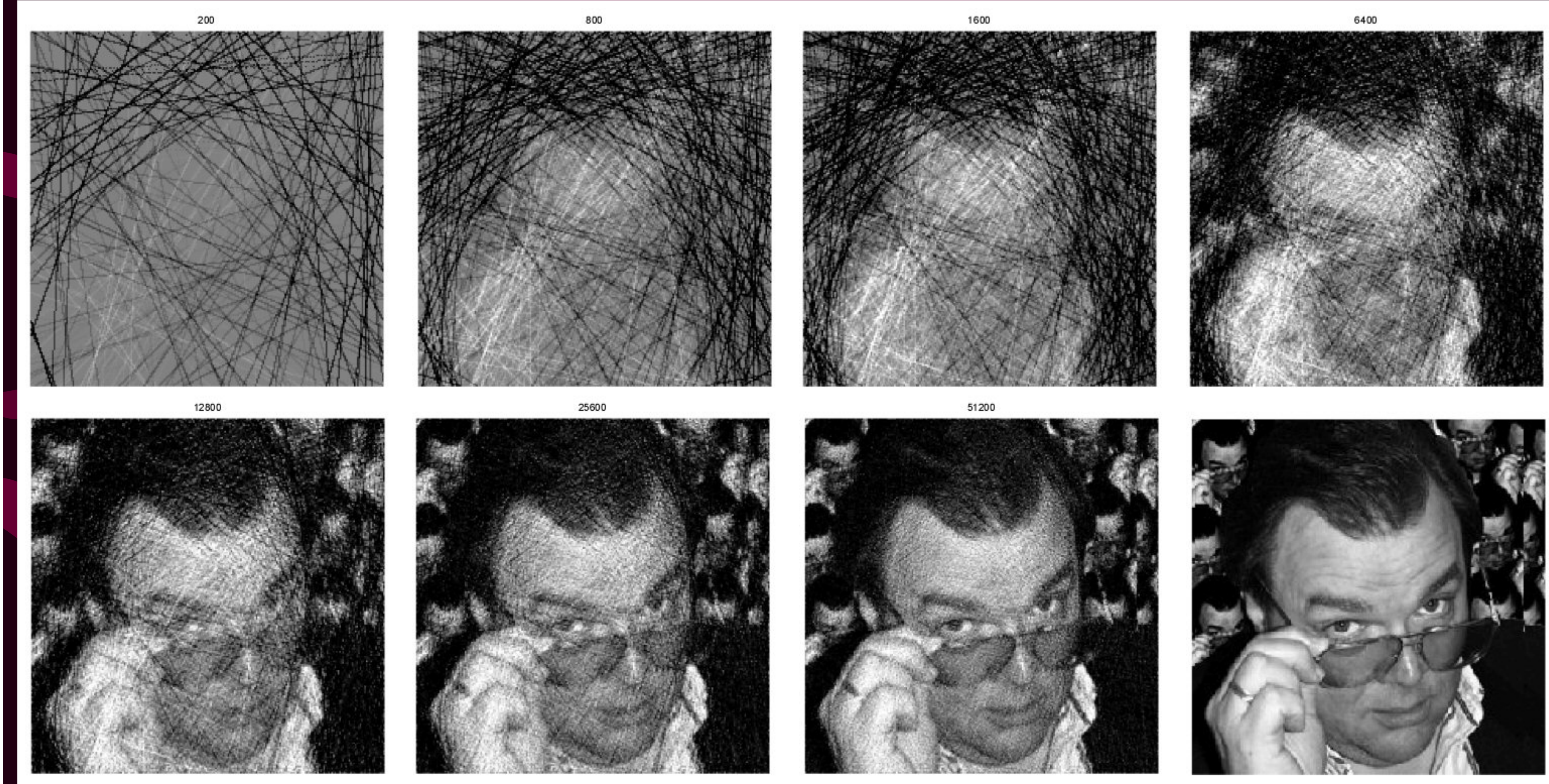
Same procedure as subpixel resolution.



Application:

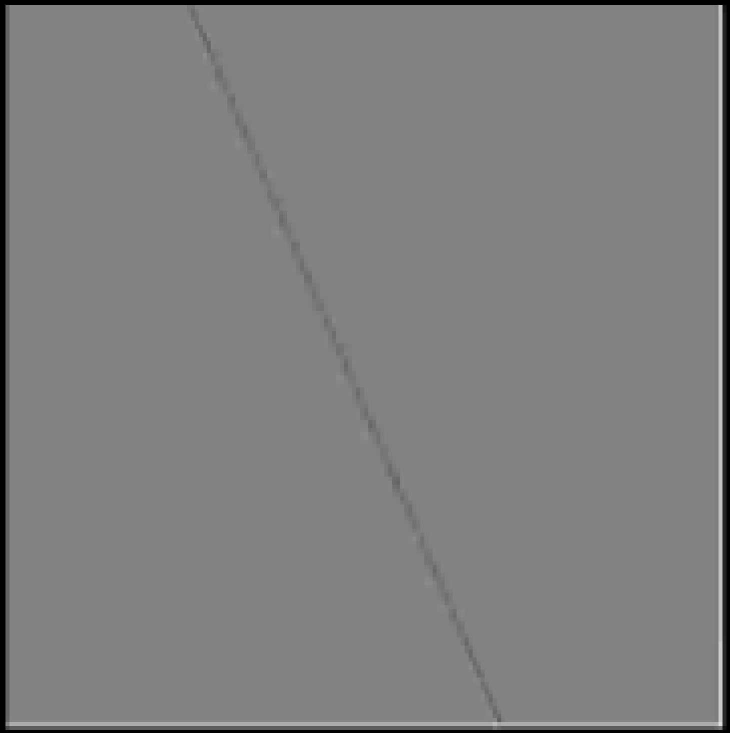
Tomography

Same procedure as subpixel resolution.



Tomography

Same procedure as subpixel resolution.



Slow (0.19) Fast (0:09)

102k iterations in ~log time

Tomography

Same procedure as subpixel resolution.

Slow (0.43) Fast (0:21)

409k iterations in ~log time

Application:

Radiation Oncology

Conventional Therapy Illustration

Application:

Radiation Oncology

Conventional Therapy Illustration

Application:

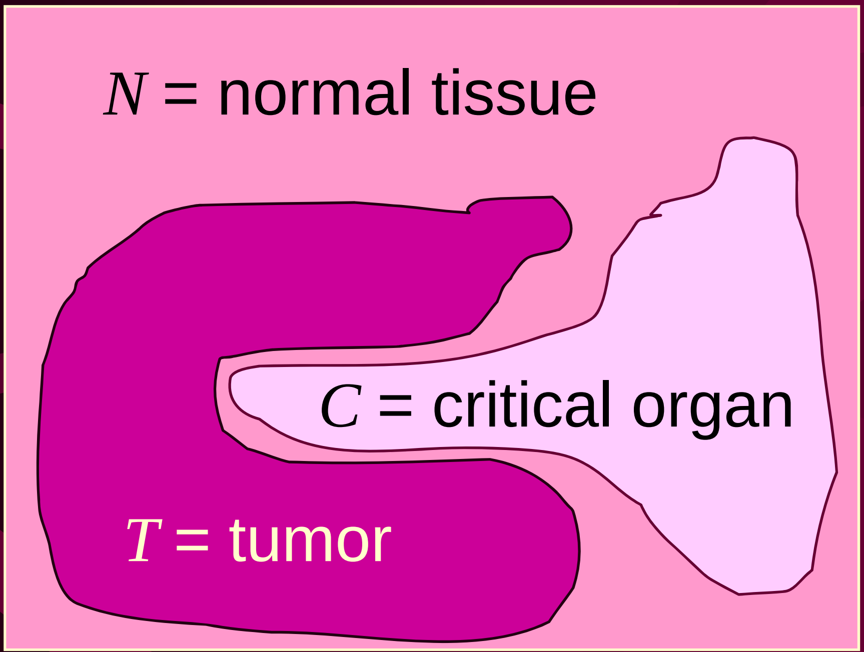
Radiation Oncology

Intensity Modulated Radiotherapy

Application:

Radiation Oncology

- Convex sets for dosage optimization



C_N = Set of beam patterns giving dosage in N less than $T_{N.}$

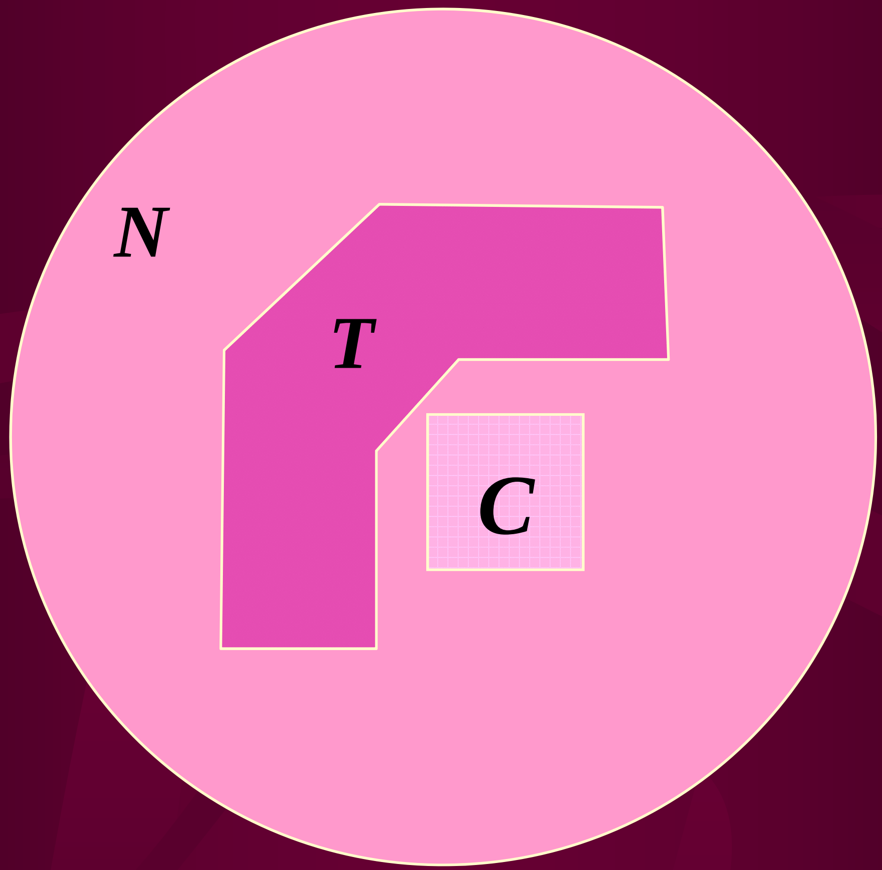
C_C = Set of beam patterns giving dosage in C less than $T_C < T_{N.}$

C_T = Set of beam patterns giving dosage in T between T_{Low} and T_{hi} (no cold spots or hot spots).

C_C = Set of nonnegative beam patterns.

Application:

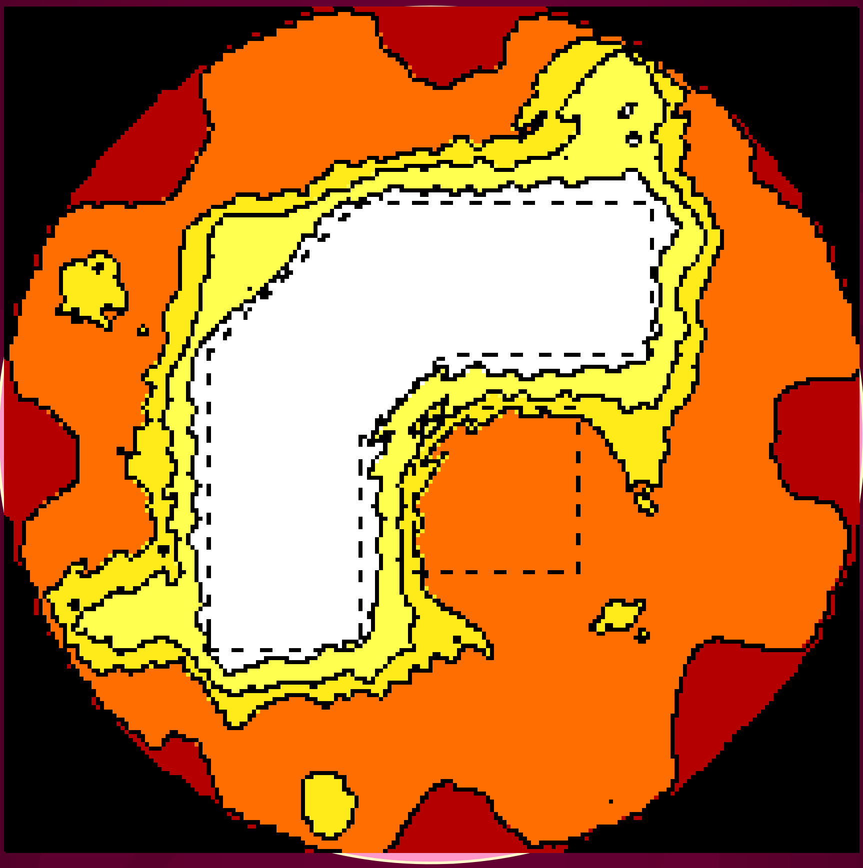
Radiation Oncology



Example POCS Solution

Application:

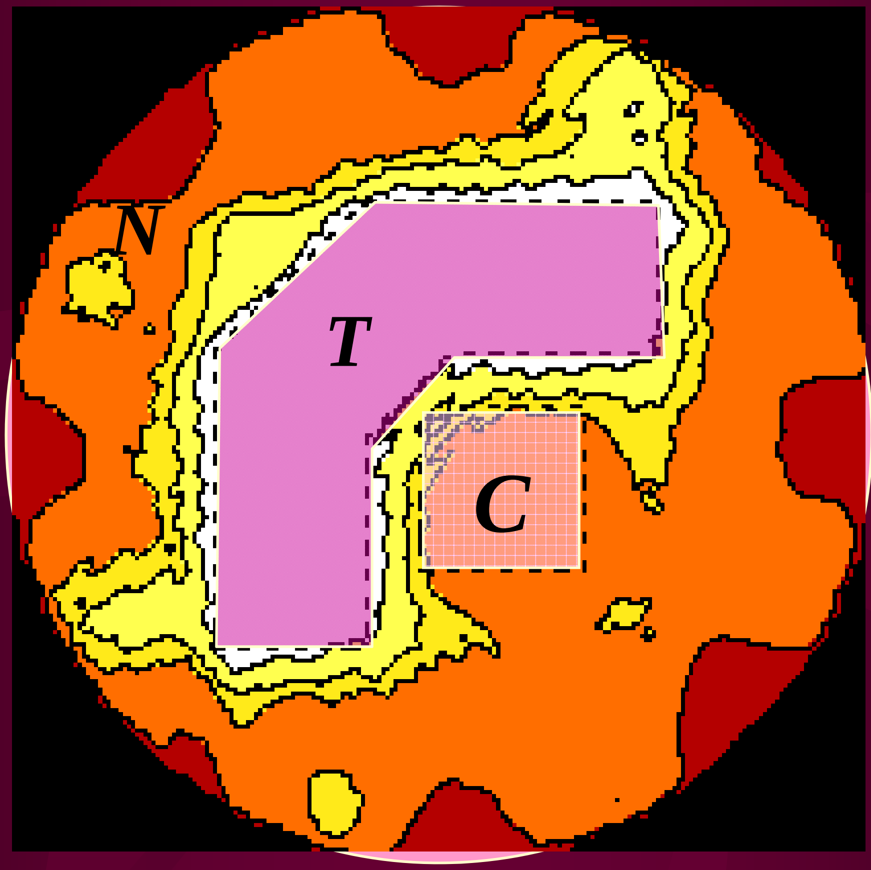
Radiation Oncology



Example POCS Solution

Application:

Radiation Oncology



Example POCS Solution

Outline

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Syntesis
  - The Gerchberg-Saxton Algorithm

Block Loss Recovery Techniques for Image and Video Communications

Transmission & Error

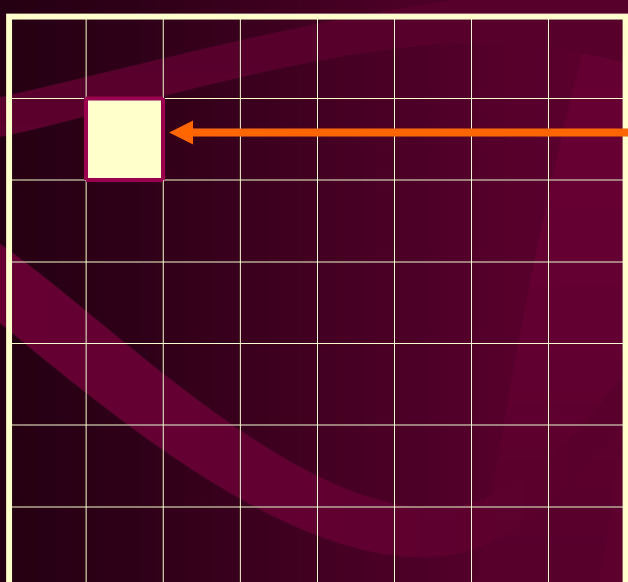
- **During jpg or mpg transmission, some packets may be lost due to bit error, congestion of network, noise burst, or other reasons.**
- Assumption: No Automatic Retransmission Request (ARQ)

Application

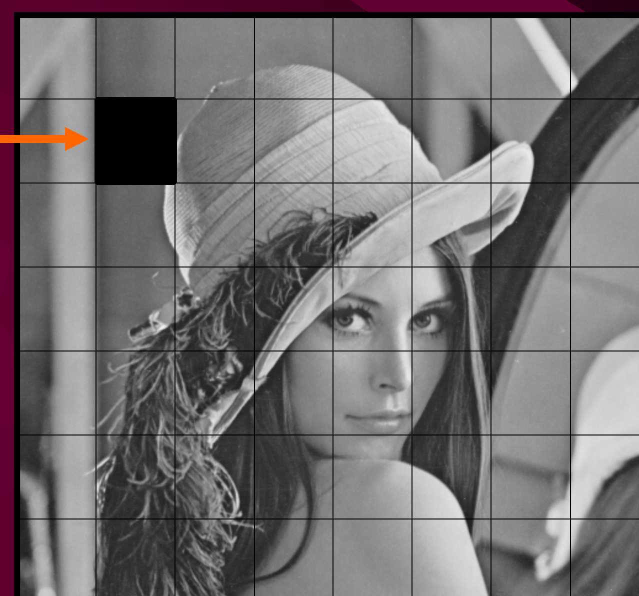
Block Loss Recovery Techniques for Image and Video Communications

Transmission & Error

- **During jpg or mpg transmission, some packets may be lost due to bit error, congestion of network, noise burst, or other reasons.**
- Assumption: No Automatic Retransmission Request (ARQ)



Missing 8x8 block of pixels



Application

Projections based Block Recovery – Algorithm

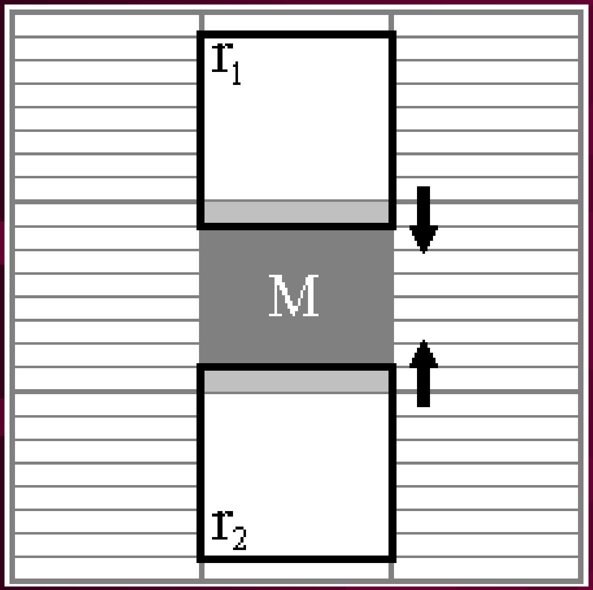
- **Two Steps**

  1. **Edge orientation detection**

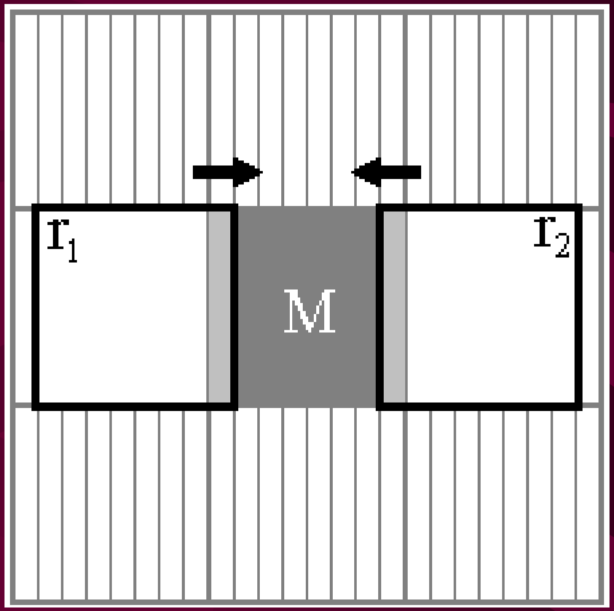
  2. **POCS using Three Convex Sets**

Application

- Recovery vectors are extracted to restore missing pixels.
- Two positions of recovery vectors are possible according to the edge orientation.



Vertical line dominating area



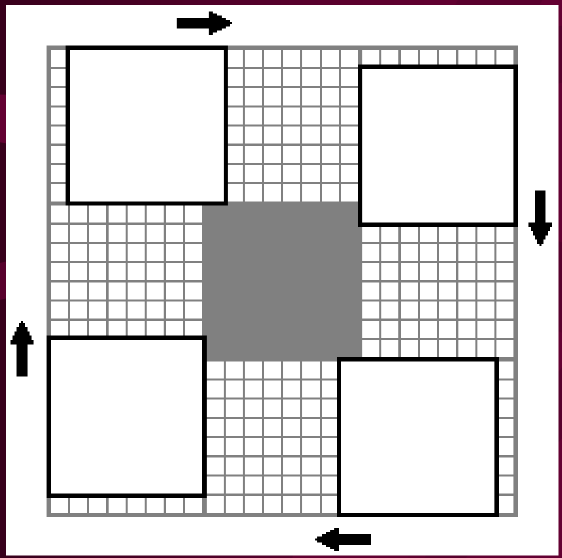
Horizontal line dominating area

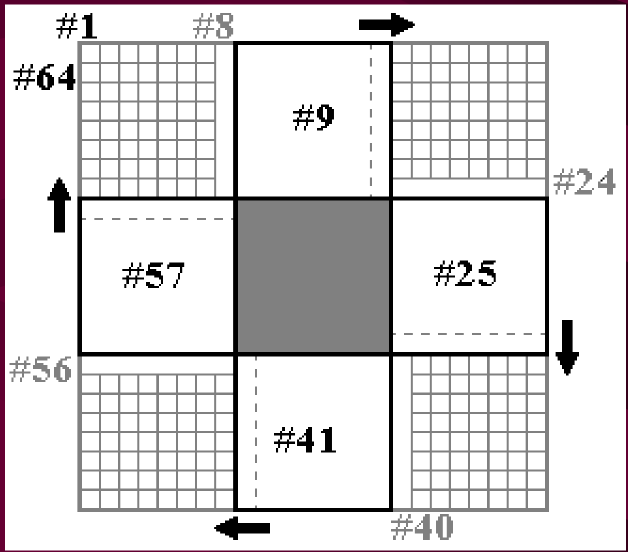
- Recovery vectors consist of known pixels(white color) and missing pixels(gray color).
- The number of recovery vectors, $\mathbf{r}_k$, is 2.

Application

Forming a Convex Constraint Using Surrounding Vectors

- **Surrounding Vectors, s_k, are extracted from surrounding area of a missing block by an *N* x *N* window.**

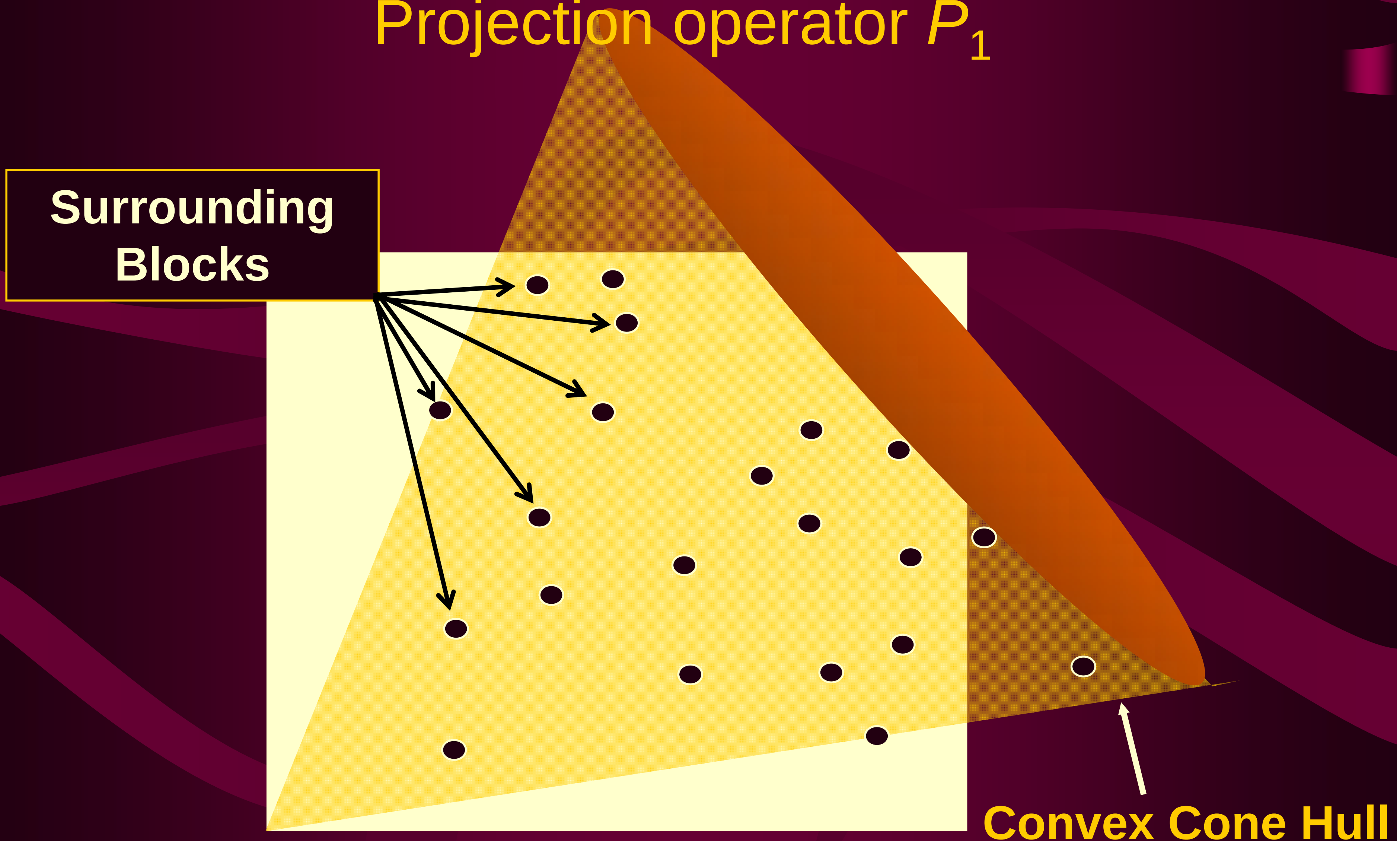




- **Each vector has its own spatial and spectral characteristic.**
- The number of surrounding vectors, s_k, is 8*N*.

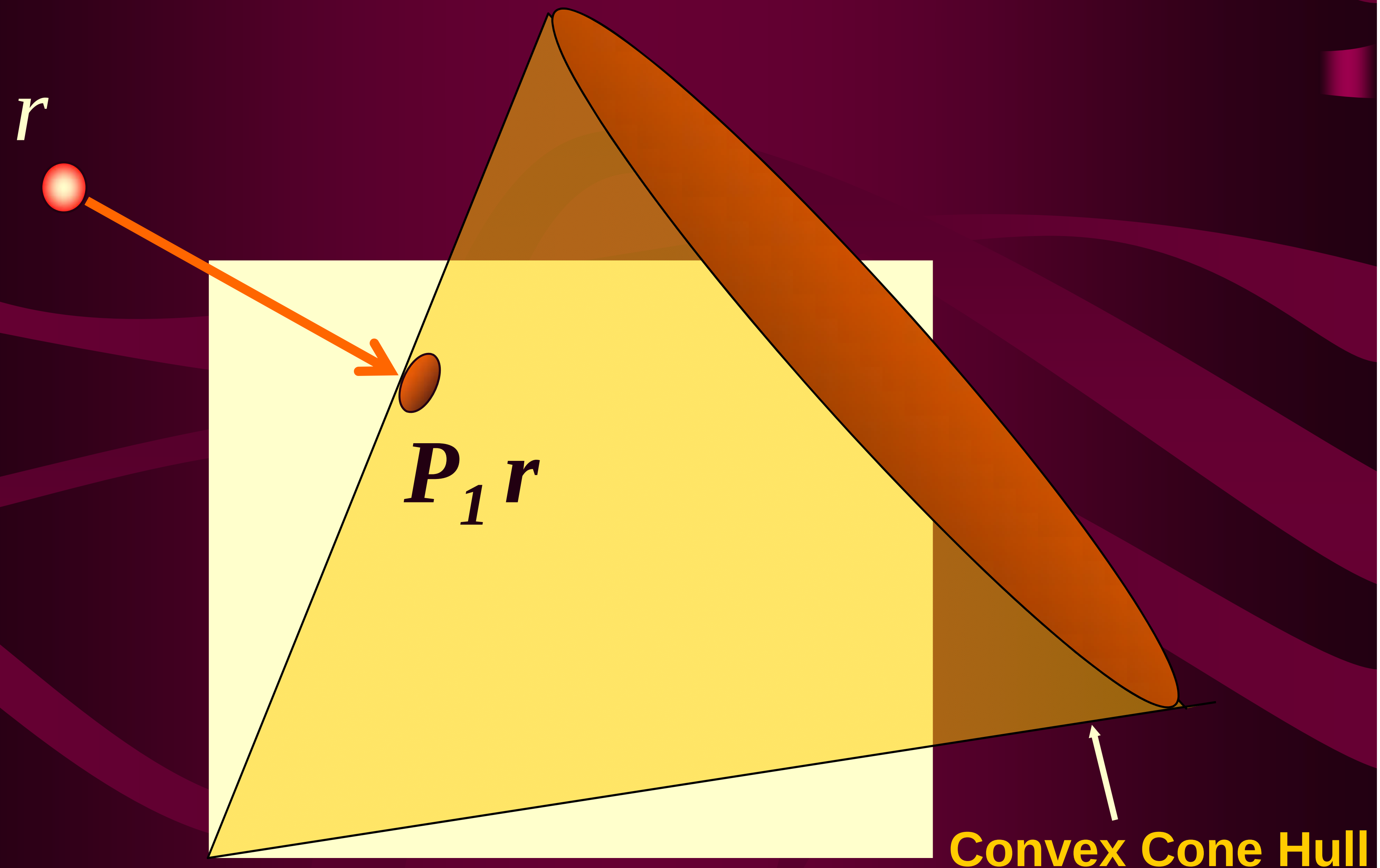
Application

Projections based Block Recovery – Projection operator P_1



Application

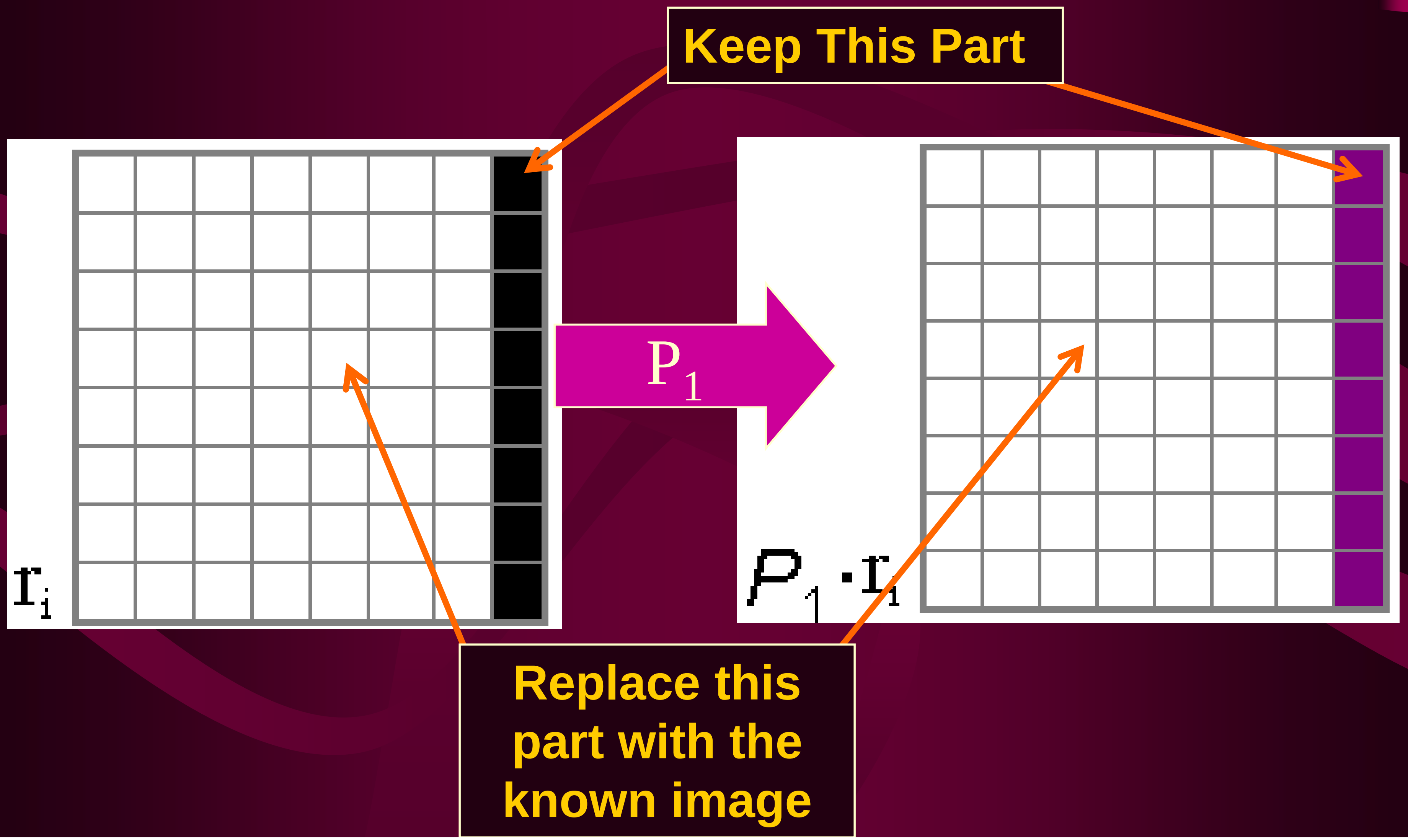
Projection operator P_1



Application

Projection operator P_2

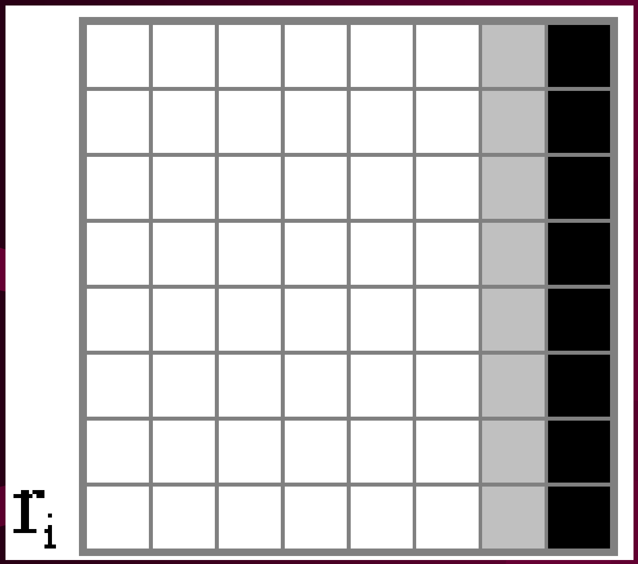
Identical Middles



Application

Projection operator P_3

Smoothness Constraint



f_{N-1} f_N

- **Convex set C_3 acts as a convex constraint between missing pixels and adjacent known pixels, (f_{N-1} f_N). The rms difference between the columns is constrained to lie below a threshold.**

$$\mathbf{g} = \{(\mathbf{f}_{N-1,0} - \mathbf{f}_{N,0}), \ldots, (\mathbf{f}_{N-1,N} - \mathbf{f}_{N,N})\}$$

Application

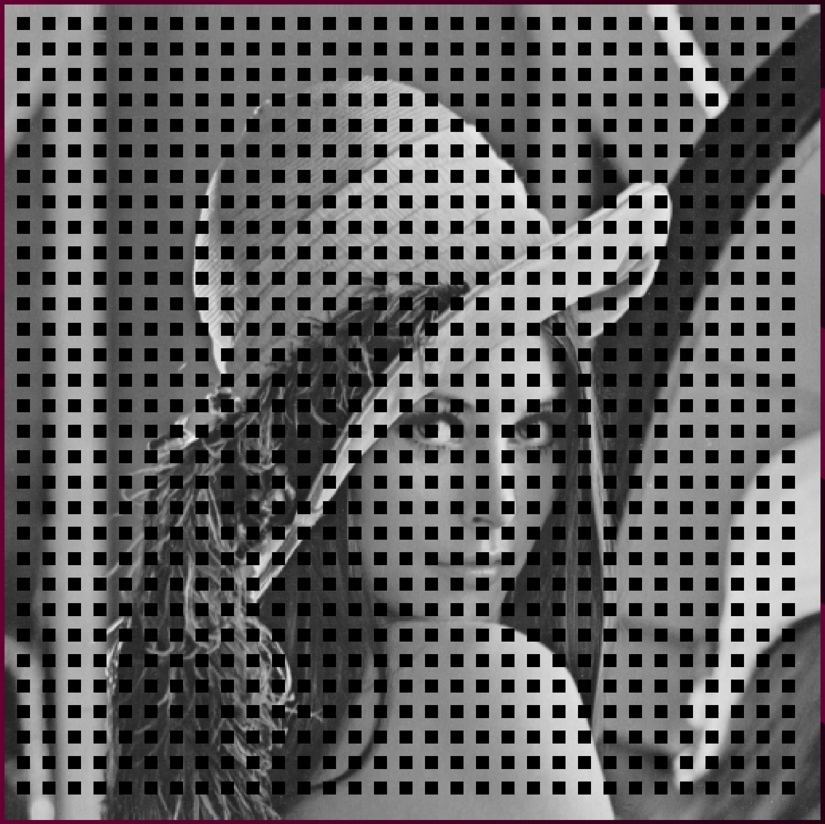
Simulation Results – Test Data and Error

- Peak Signal to Noise Ratio

$$PSNR = 10 \cdot \log\left(\frac{N \cdot M \cdot 255^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} | x(i,j) - \hat{x}(i,j) |^2} \right)$$

Simulation Results – Lena, 8 x 8 block loss

Original Image



Test Image

Simulation Results – Lena, 8 x 8 block loss

Ancis, PSNR = 28.68 dB

Hemami, PSNR = 31.86 dB

Simulation Results – Lena, 8 x 8 block loss

Ziad, PSNR = 31.57 dB

POCS, PSNR = 34.65 dB

Simulation Results – Lena, 8 x 8 block loss

Ancis

PSNR = 28.68 dB

Hemami

PSNR = 31.86 dB

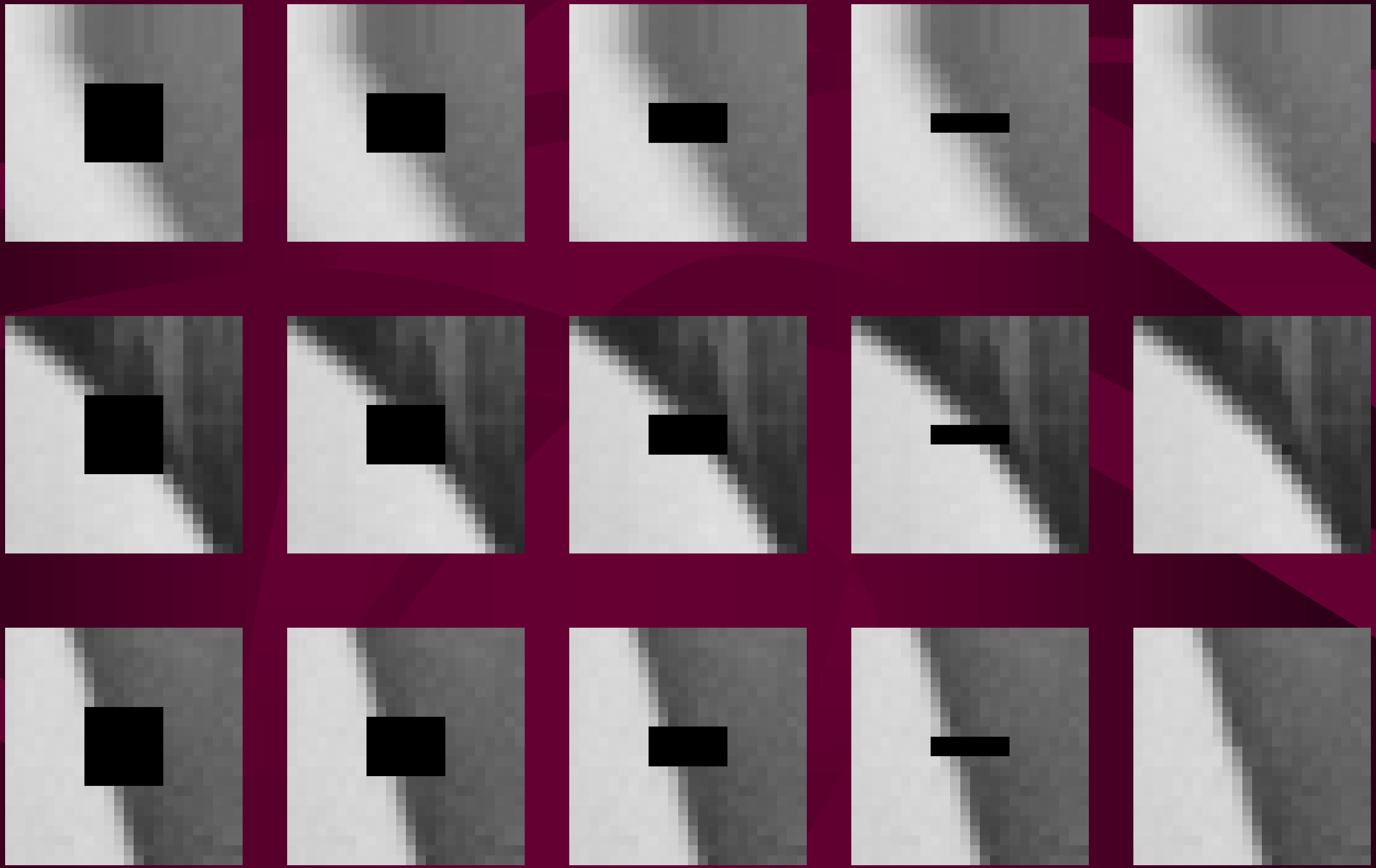
Ziad

PSNR = 31.57 dB

POCS

PSNR = 34.65 dB

Simulation Results – Each Step
Lena 8 x 8 block loss



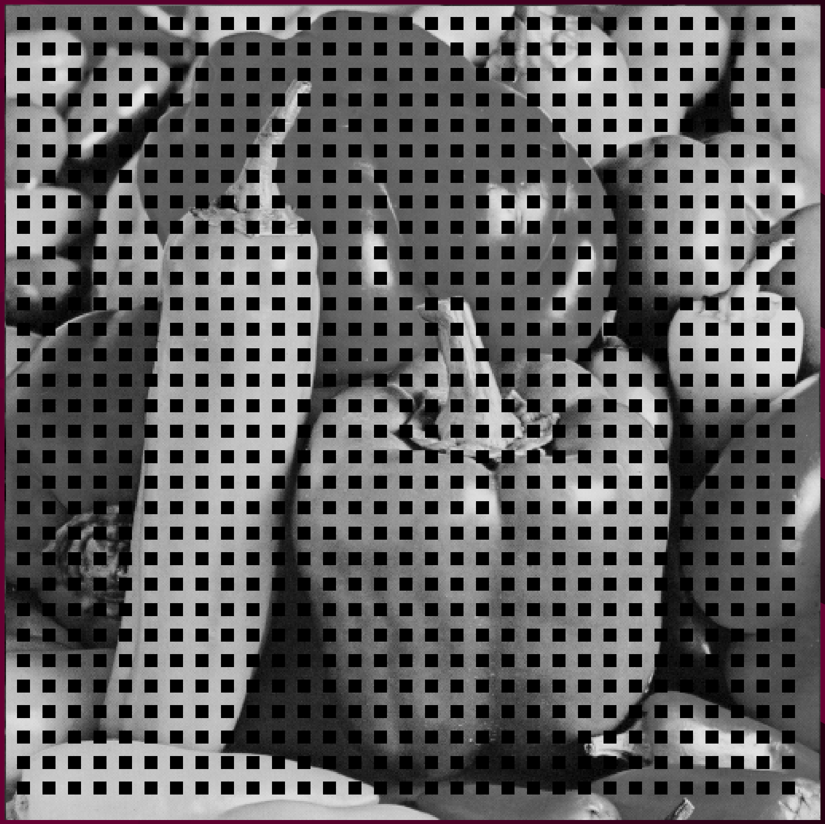
(a)

(b)

(c)

Simulation Results – Peppers, 8 x 8 block loss

Original Image



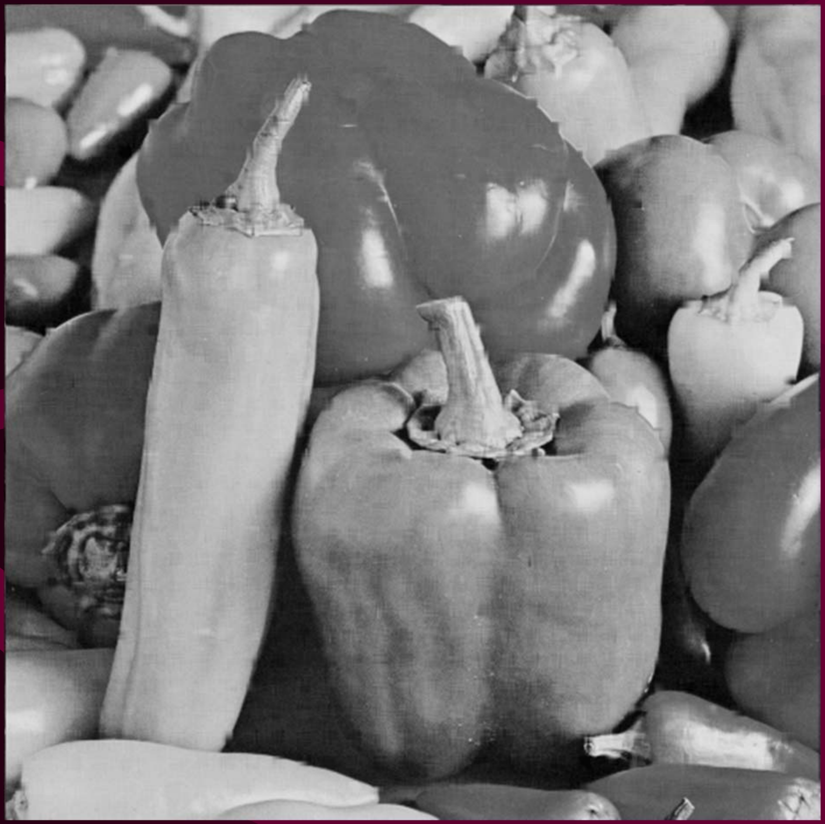
Test Image

Simulation Results – Peppers, 8 x 8 block loss

Ancis, PSNR = 27.92 dB

Hemami, PSNR = 31.83 dB

Simulation Results – Peppers, 8 x 8 block loss



Ziad, PSNR = 32.76 dB

POCS, PSNR = 34.20 dB

Simulation Results – PSNR (8 x 8)

	Lena	Masqrd	Peppers	Boat	Elaine	Couple
Ancis	28.68	25.47	27.92	26.33	29.84	28.24
Sun	29.99	27.25	29.97	27.36	30.95	28.45
Park	31.26	27.91	31.71	28.77	32.96	30.04
Hemami	31.86	27.65	31.83	29.36	32.07	30.31
Ziad	31.57	27.94	32.76	30.11	31.92	30.99
POCS	34.65	29.87	34.20	30.78	34.63	31.49

Simulation Results – Masquerade, 8 x one row block loss

Original Image

Test Image

Simulation Results – Masquerade, 8 x one row block loss

Hemami, PSNR = 23.10 dB

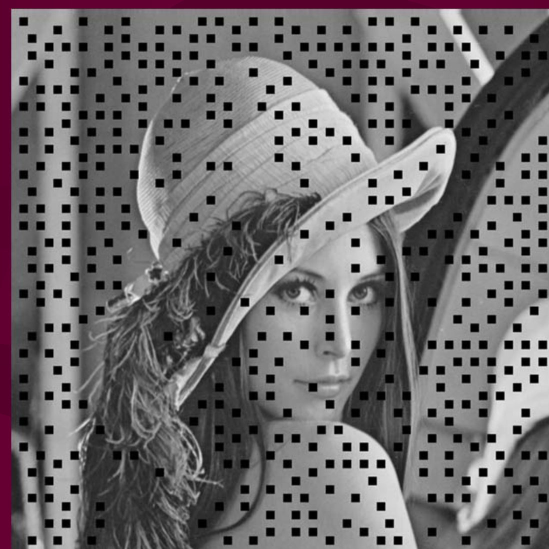
POCS, PSNR = 25.09 dB

Interpolation based Coding – Result 1

JPEG Coding

PSNR = 32.27 dB

Size = 0.30 BPP = 9,902 Byte



w/ Removed Blocks

Blocks : 447 / 4096 = 11%

Size = 0.29 BPP

I-based Coding

PSNR = 32.35 dB

Size = 0.29 BPP = 9,634 Byte

Interpolation based Coding – Result 2

JPEG Coding

PSNR = 32.27 dB

Size = 0.30 BPP = 9,902 Byte

w/ Removed Blocks

Blocks : 557 / 4096 = 14%

Size = 0.27 BPP

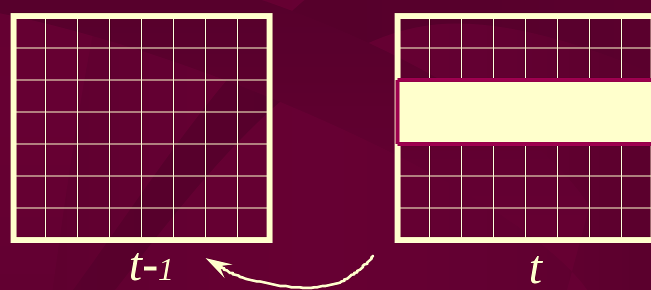
I-based Coding

PSNR = 32.37 dB

Size = 0.27 BPP = 9,570 Byte

Temporal Block Loss Recovery

- In video coding (e, g, MPEG), temporal recovery is more effective.



Simulation Results – Flower Garden

Original Sequence

Test Sequence

Simulation Results – Flower Garden

Zero Motion Vector,
PSNR = 16.15 dB

Average of
Surrounding Motion
Vectors, PSNR =
18.64 dB

Simulation Results – Flower Garden

Motion Flow Interpolation (1999), PSNR = 19.29 dB

Boundary Matching Algorithm (1993), PSNR = 19.83 dB

Simulation Results – Flower Garden

Decoder Motion Vector Estimation (2000), PSNR = 19.21 dB

POCS Based, PSNR = 20.71 dB

Simulation Results – Foreman

Original Sequence

Test Sequence

Zero Motion Vector PSNR = 24.71 dB

Average of Surrounding Motion Vectors PSNR = 26.22 dB

Simulation Results – Foreman

Motion Flow Interpolation (1999) PSNR = 27.09 dB

Boundary Matching Algorithm (1993), PSNR = 28.76 dB

Decoder Motion Vector Estimation (2000), PSNR = 27.46 dB

POCS Based PSNR = 29.82 dB

Simulation Results – Average PSNR

	Garden	Tennis	Football	Mobile	Foreman
MV	16.15	22.40	18.06	17.49	24.71
AV	18.64	21.98	18.72	19.03	26.22
BMA	19.83	23.55	19.41	19.75	28.76
DMVE	19.88	24.04	19.64	20.02	28.77
MFI	19.29	22.77	19.29	19.60	27.09
F-B BM	19.21	22.49	19.05	19.59	27.46
Proposed	20.71	24.52	20.32	20.66	29.82

Outline

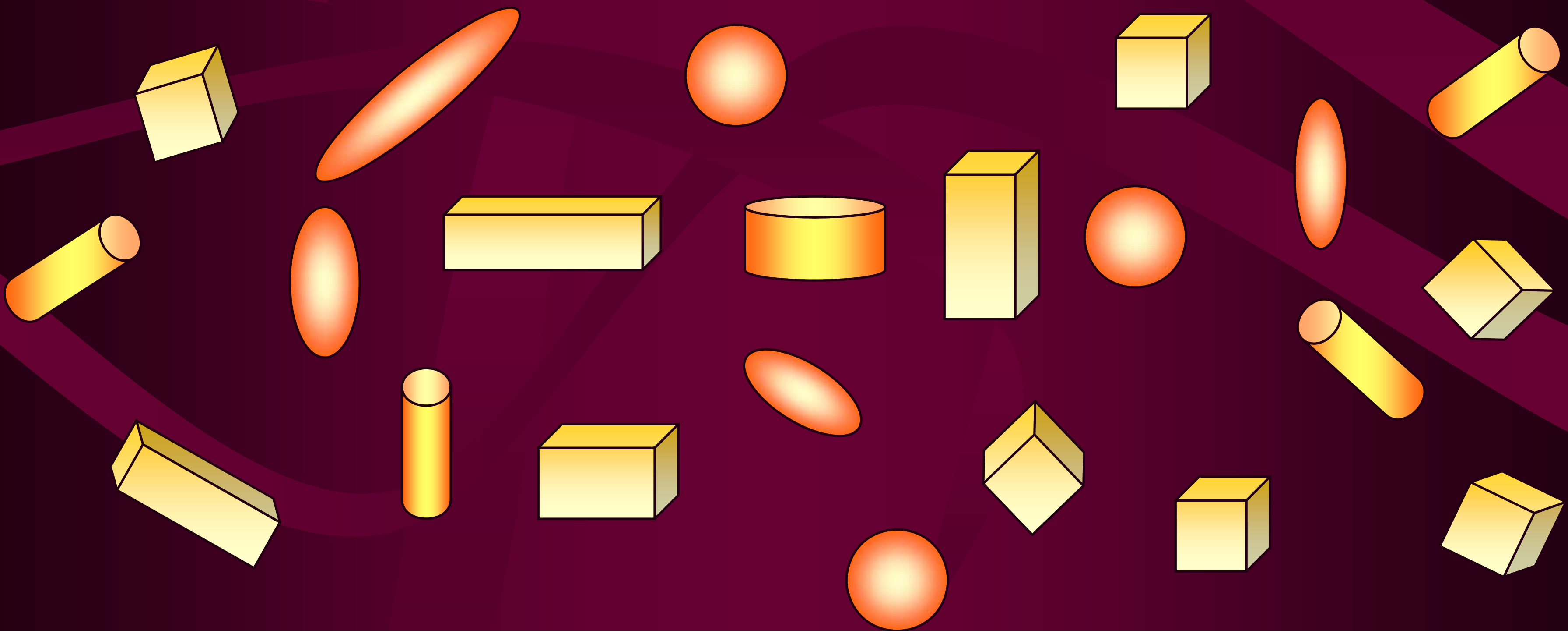
- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

Application

Missing Sensors

Idea:

- A collection of spatially distributed point sensors.
- Their readings are interrelated (e.g. temperature sensors in a room).



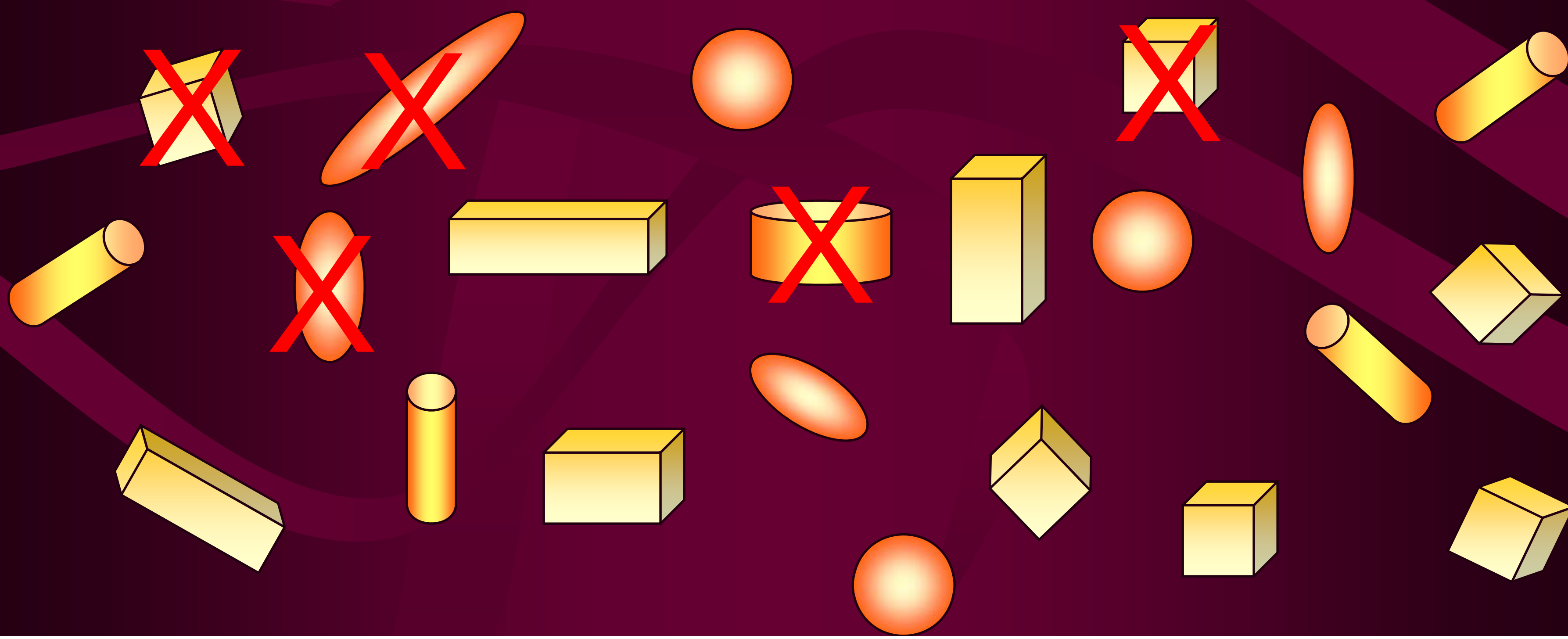
Application

Missing Sensors

- A plurality of sensors fail.

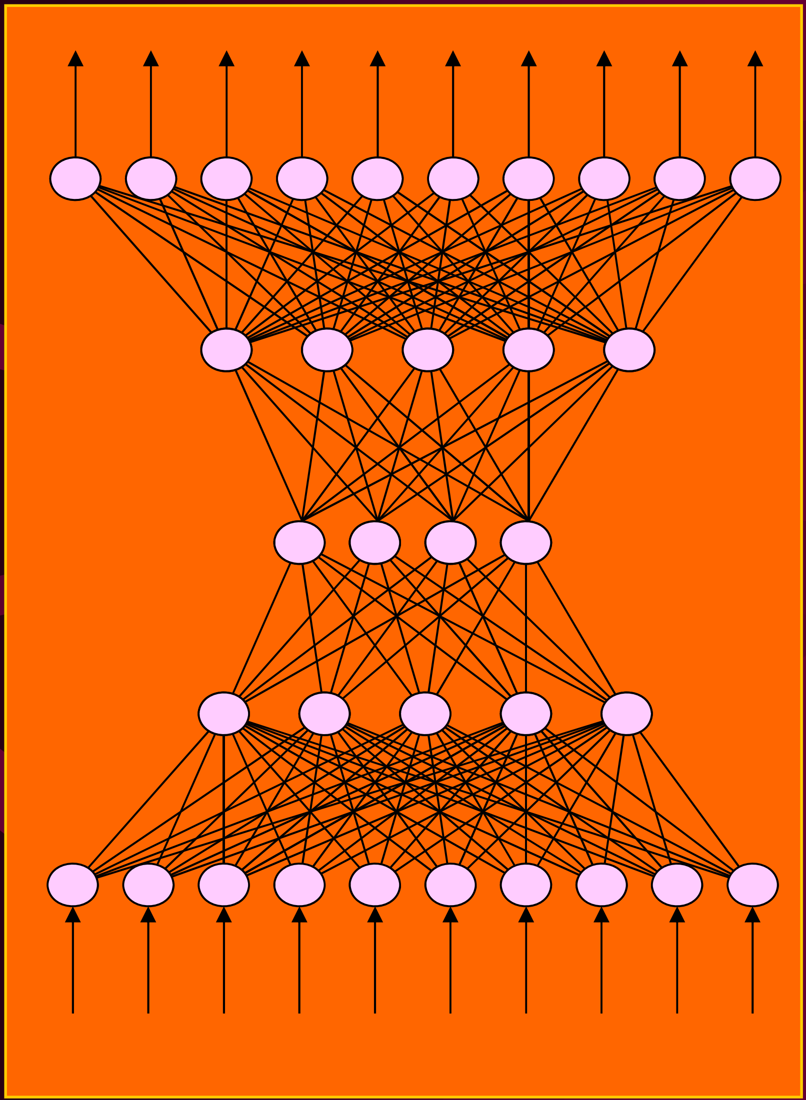
- Can the failed sensor readings be regained from those remaining without use of models?

- Applications: (1) Power Security Assessment (2) Engine Vibration Sensors



Application

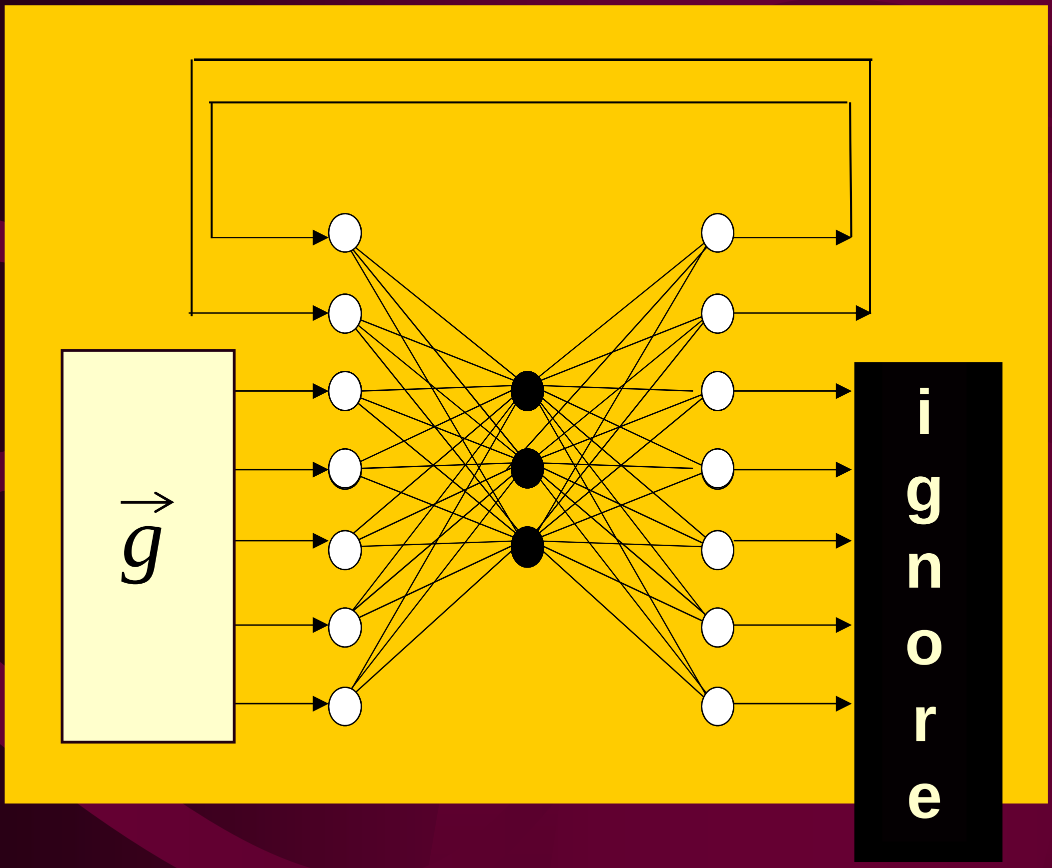
Missing Sensors



- Step 1: Learn the interrelationship among the sensors by training an auto-encoder neural network with historical data.
- The mapping of a properly trained neural network is a projection.

Application

Missing Sensors



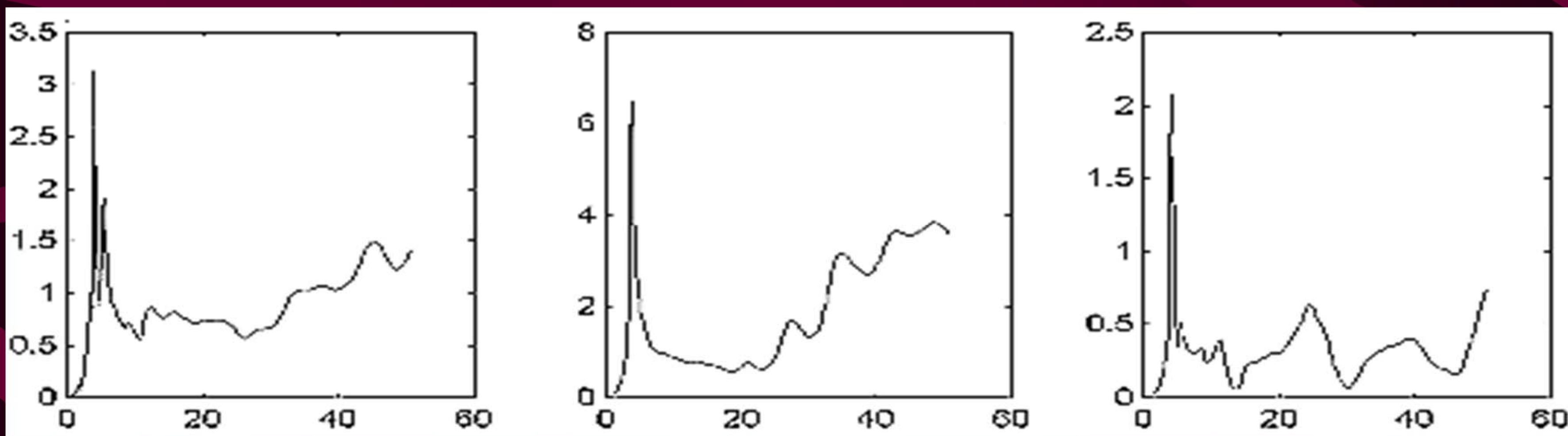
- Step 2: Impose the second convex constraint of known sensor values. If POCS, convergence is assured

Application

Missing Sensors

- Example: Vibration Sensors on a Jet Engine

x direction *y* direction *z* direction

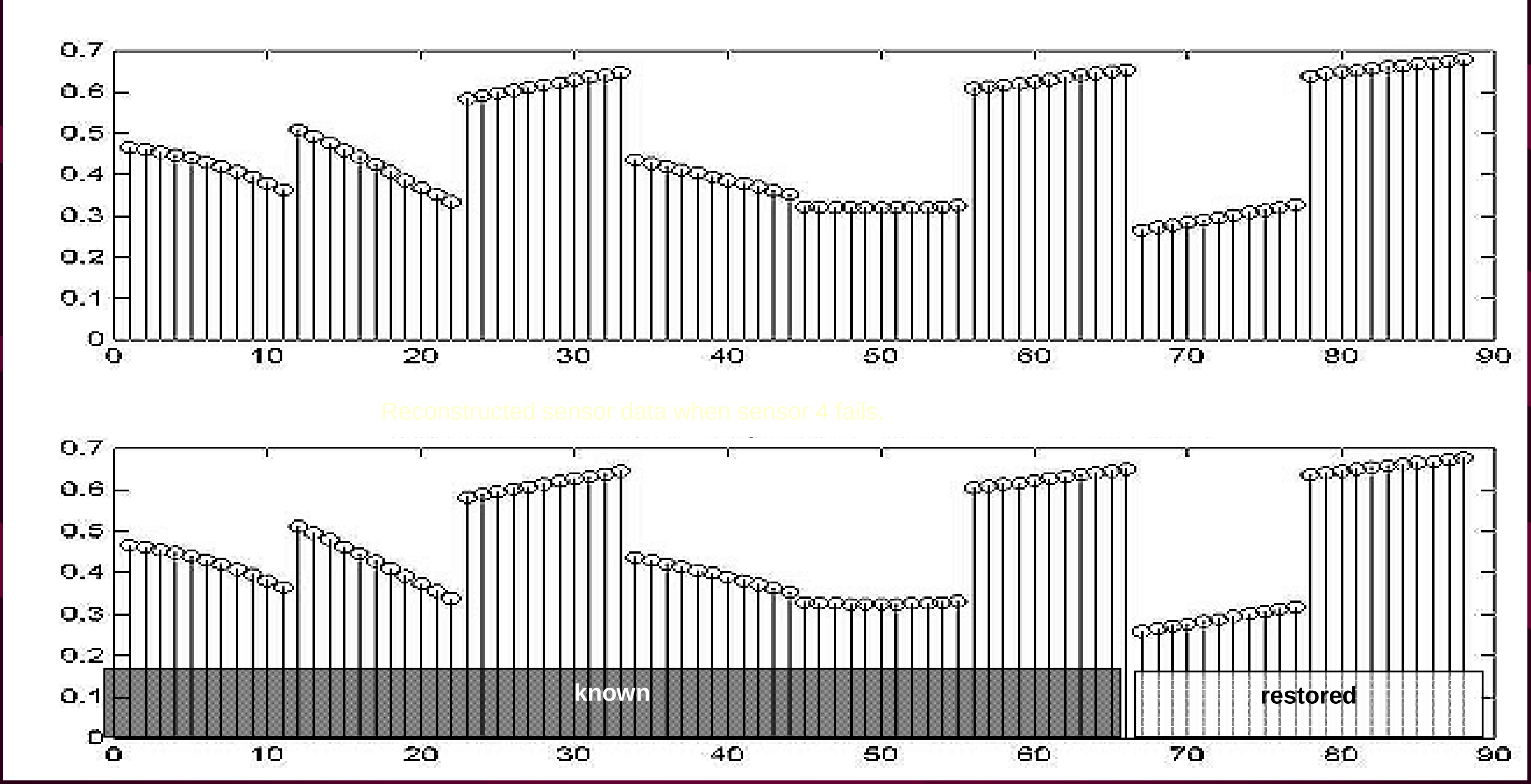


Application

Missing Sensors

- Four Sensors – One Missing

Original Sensor Data



Application

Missing Sensors

- Four Sensors – Two Missing

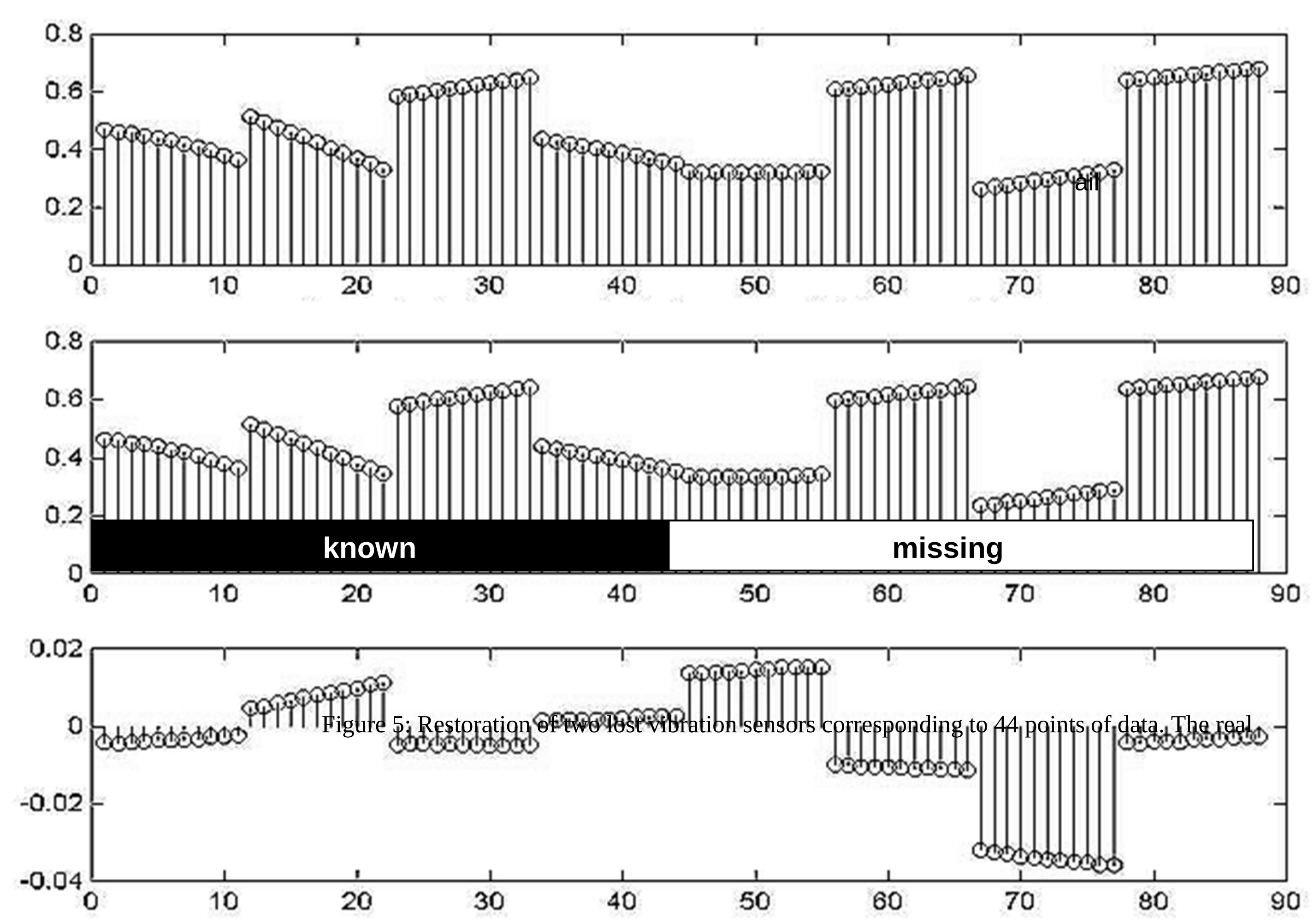


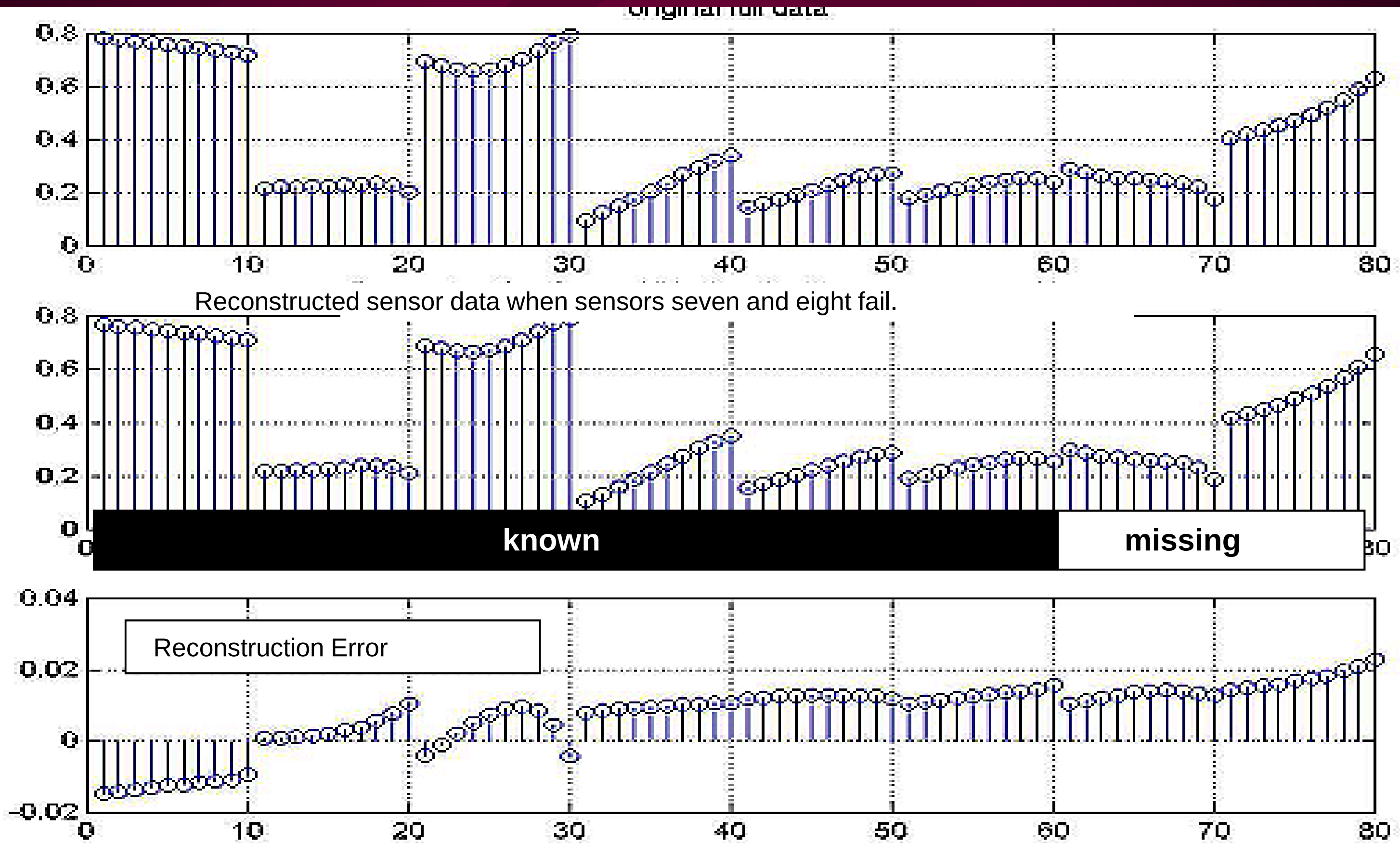
Figure 5: Restoration of two lost vibration sensors corresponding to 44 points of data. The real

Application

Missing Sensors

- Eight Sensors – Two Missing

(Magnitude Only)



Outline

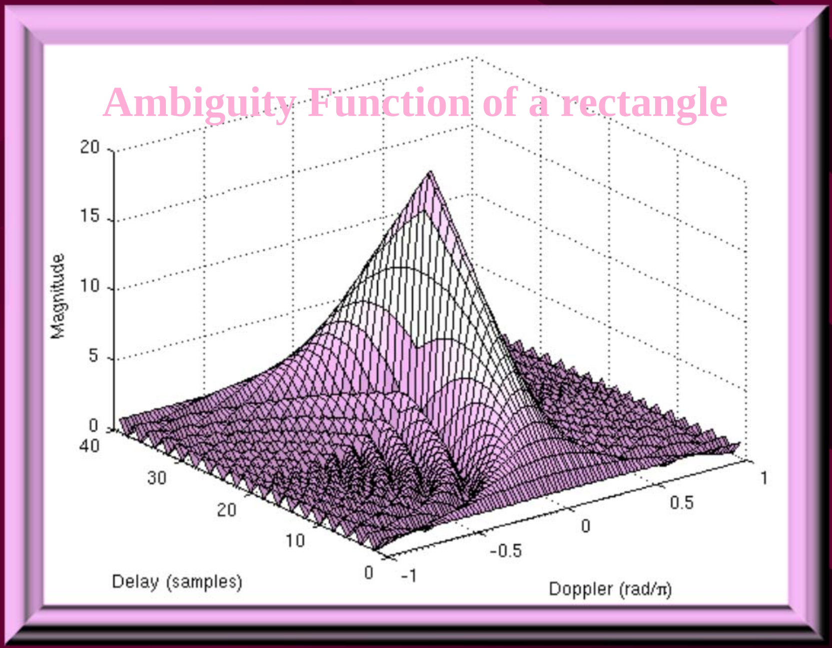
- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

Ambiguity Function Signal Synthesis

Woodward's Ambiguity Function

$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

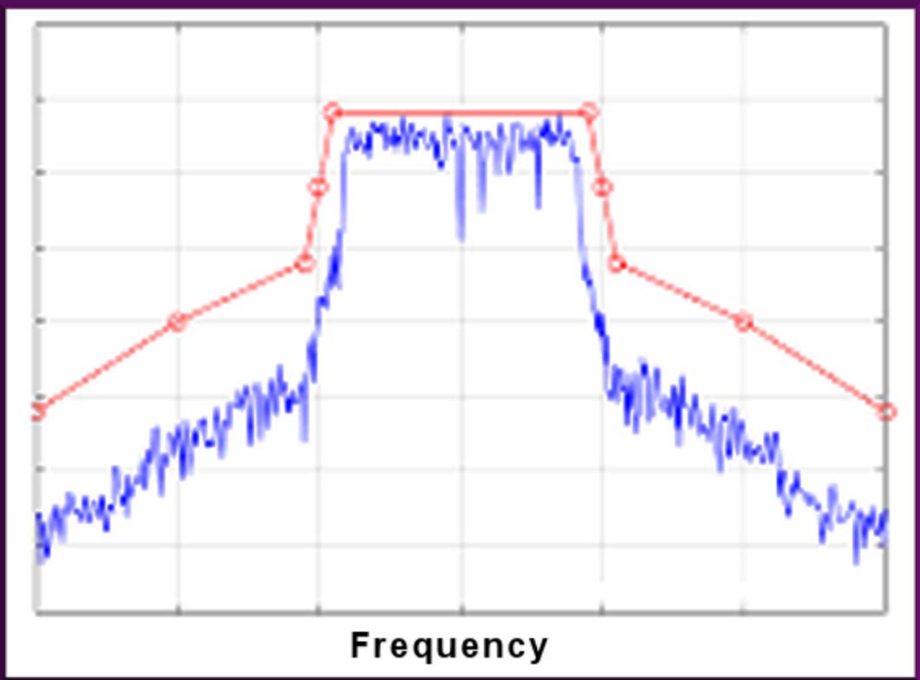
There exists an inherent trade-off in the ability of a signal to accurately measure both the range (determined by delay τ) and velocity (measured from Doppler shift u) of a target. Woodward's ambiguity function measures this uncertainty for narrowband RF signals for monostatic radar



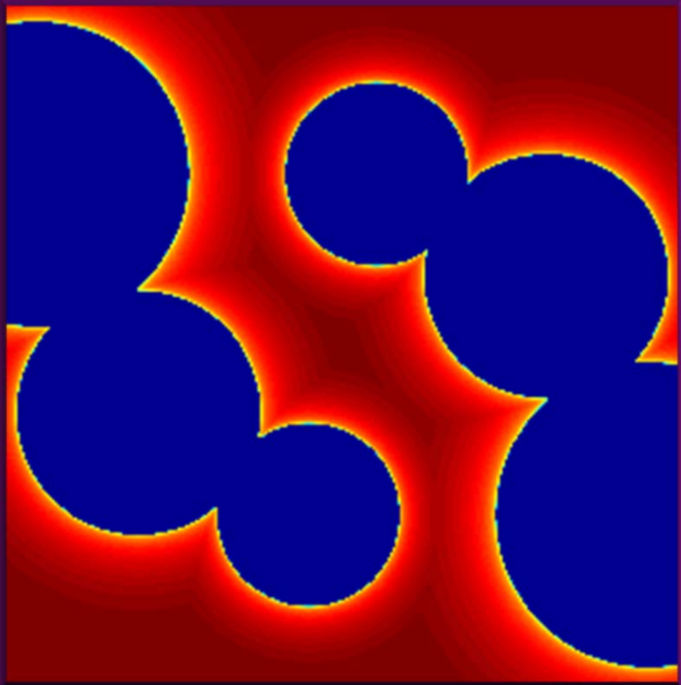
Ambiguity Function Signal Synthesis

Synthesized Signal Constraints:

1. **Limited Peak-to-average-power ratio (PAPR)**
2. **Spectral mask constraint**
3. **Targeted ambiguity function**

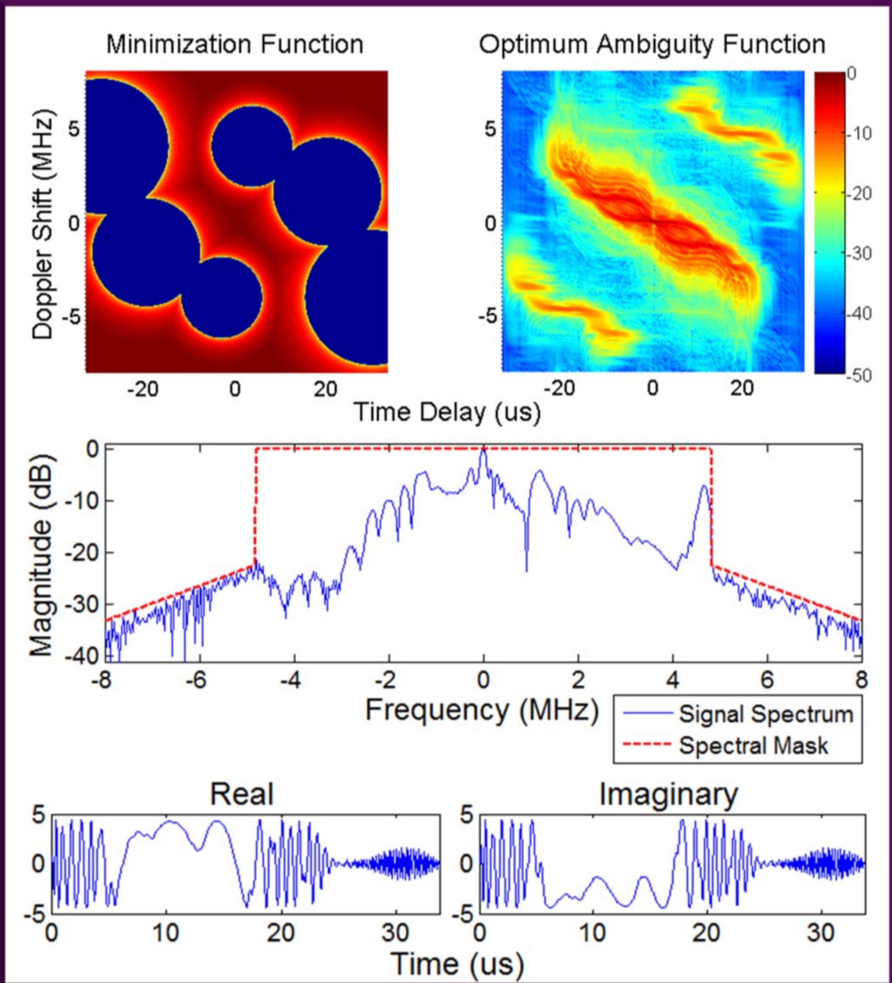


Spectral mask constraint



Ambiguity function constraint

Ambiguity Function Signal Synthesis



$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

$$\chi(\tau, u) = \int_t x(t)x^*(t-\tau)e^{-j2\pi ut}dt$$

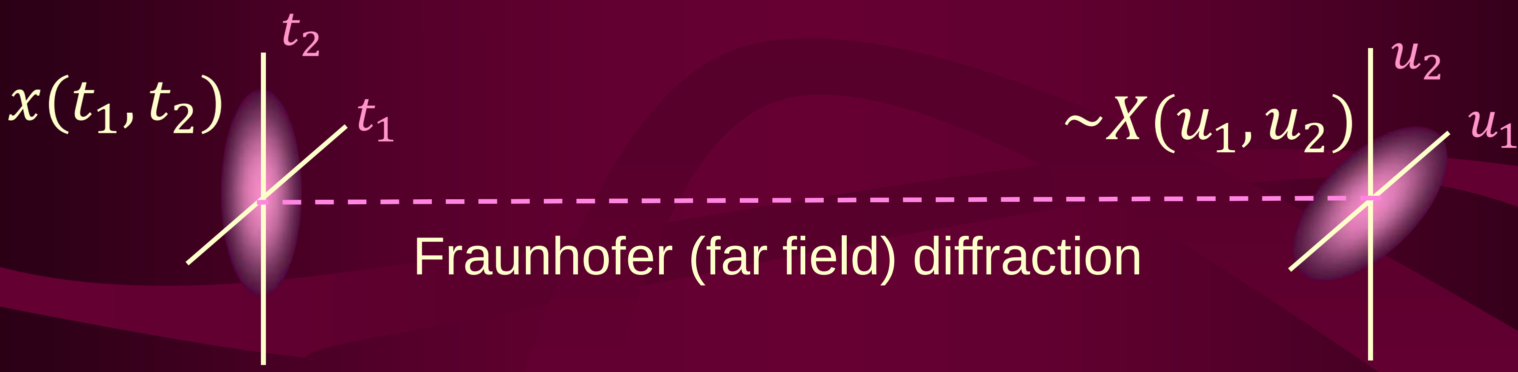
Dylan Eustice, Charles Baylis and Robert J. Marks II, "Woodward's Ambiguity Function: From Foundations to Applications," 2015 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS), April 23-24, 2015. Waco, Texas (pp. 1-17). DOI: 10.1109/WMCaS.2015.7233208

Dylan Eustice, Charles Baylis, Lawrence Cohen, and Robert J. Marks II. "Waveform synthesis via alternating projections with ambiguity function, peak-to-average power ratio, and spectrum requirements." 2016 IEEE Radio and Wireless Symposium (RWS), pp. 190-192. IEEE, 2016. DOI: 10.1109/RWS.2016.7444401

Outline

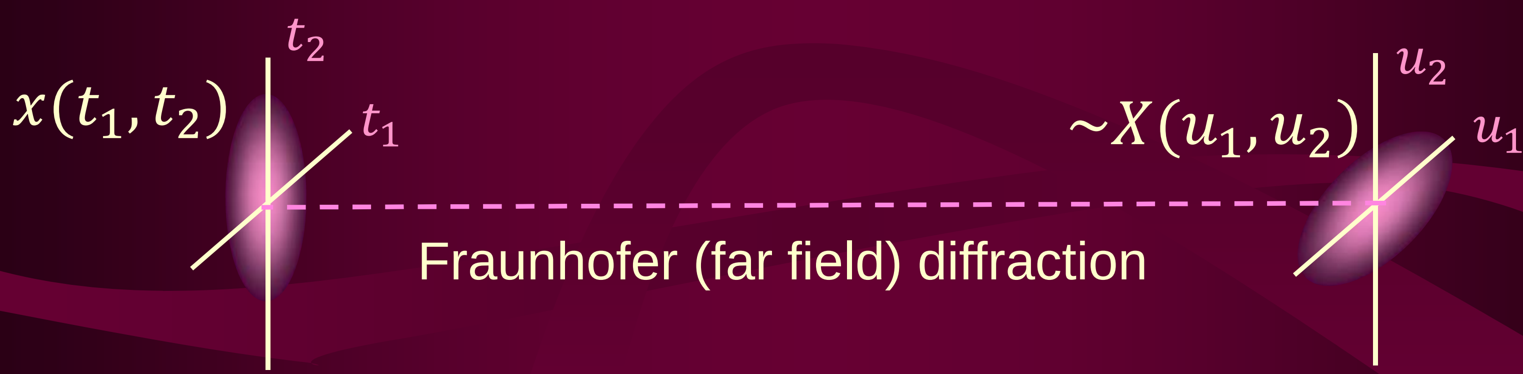
- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

Gerchberg-Saxton Algorithm



$$x(t_1, t_2) = |x|e^{j\angle x} \xrightarrow{\mathcal{F}} X(u_1, u_2) = |X|e^{j\angle X}$$

Gerchberg-Saxton Algorithm

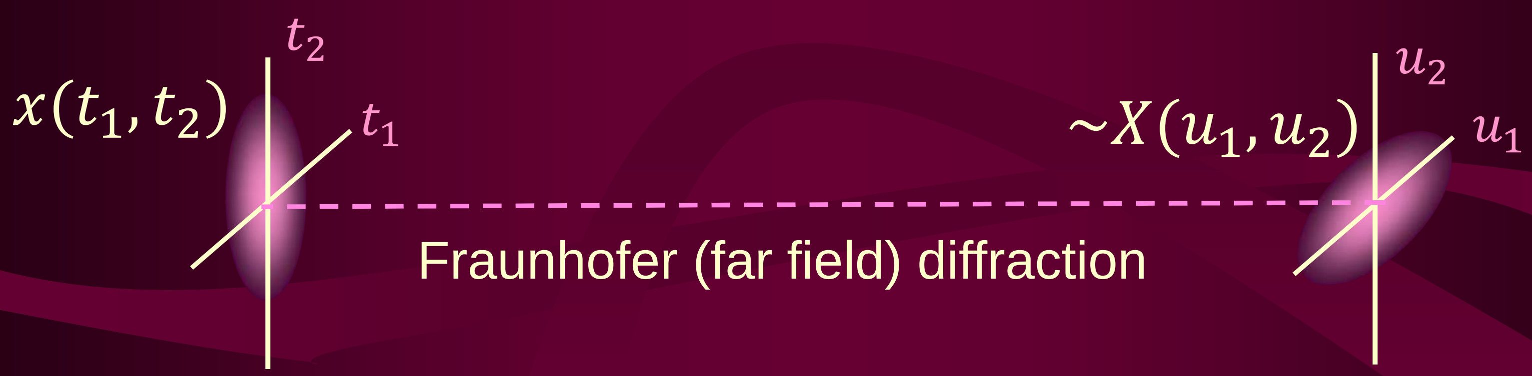


$$x(t_1, t_2) = |x|e^{j\angle x} \xrightarrow{\mathcal{F}} X(u_1, u_2) = |X|e^{j\angle X}$$

Square Law Detector measures $|x|$ and $|X|$.

Problem: Given $|x|$ and $|X|$, find phase $\angle x$ and $\angle X$

Gerchberg-Saxton Algorithm



$$x(t_1, t_2) = |x|e^{j\angle x} \xrightarrow{\mathcal{F}} X(u_1, u_2) = |X|e^{j\angle X}$$

Square Law Detector measures $|x|$ and $|X|$.

Problem: Given $|x|$ and $|X|$, find phase $\angle x$ and $\angle X$

Note: $C = \{y \ \mid \ |y| = |x|\}$ is *not* convex

But we can impose an idempotent projection: keep the phase and impose the magnitude: $\wp_C[y] = |x|e^{j\angle y}$

Gerchberg-Saxton Algorithm

Iterative Restoration:

$$g_n = |x|\, e^{j\angle f_n} \xrightarrow{\mathcal{F}} G_n = |G_{n-1}|\, e^{j\angle G_n}$$

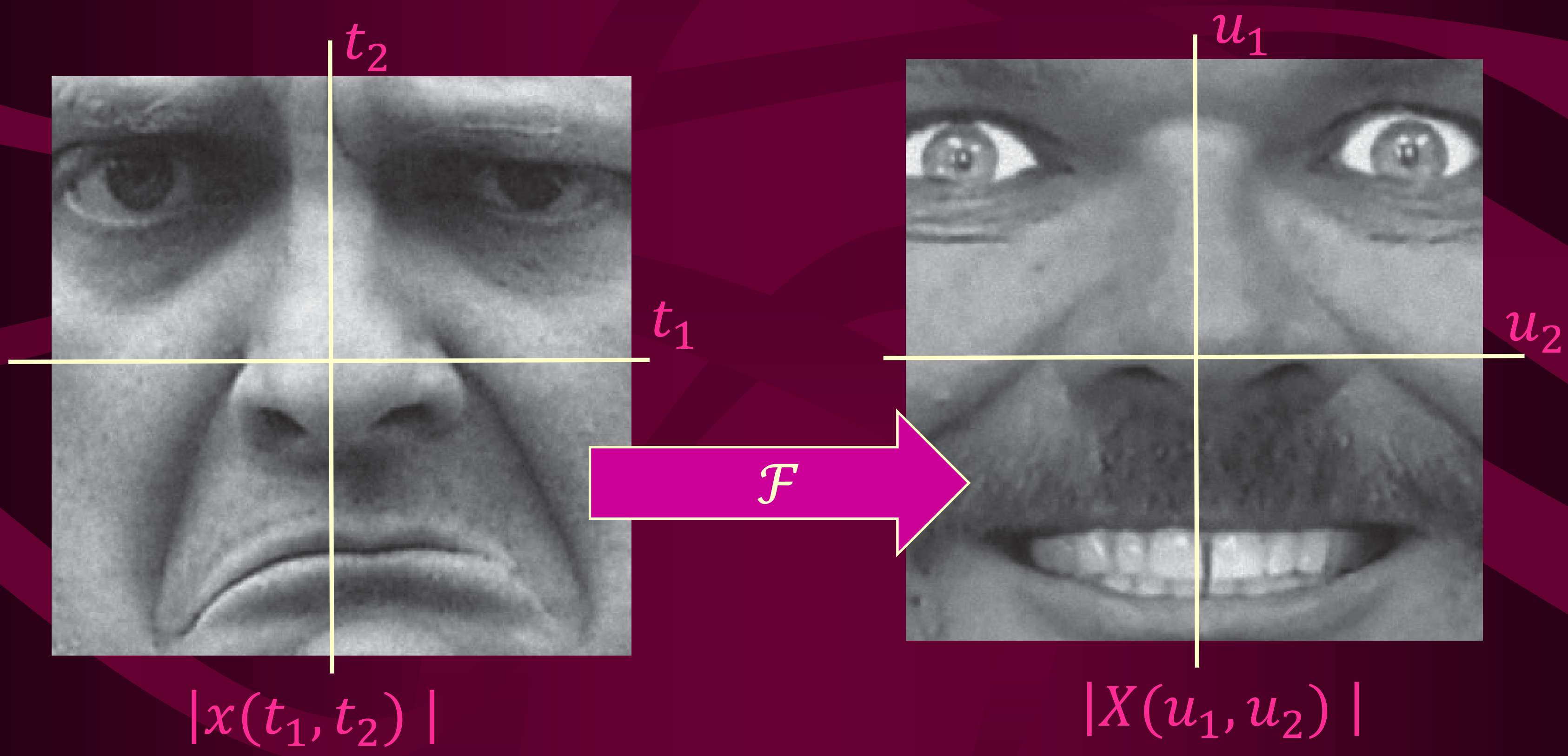
$$n = n + 1$$

Iteratively impose magnitude in both domains

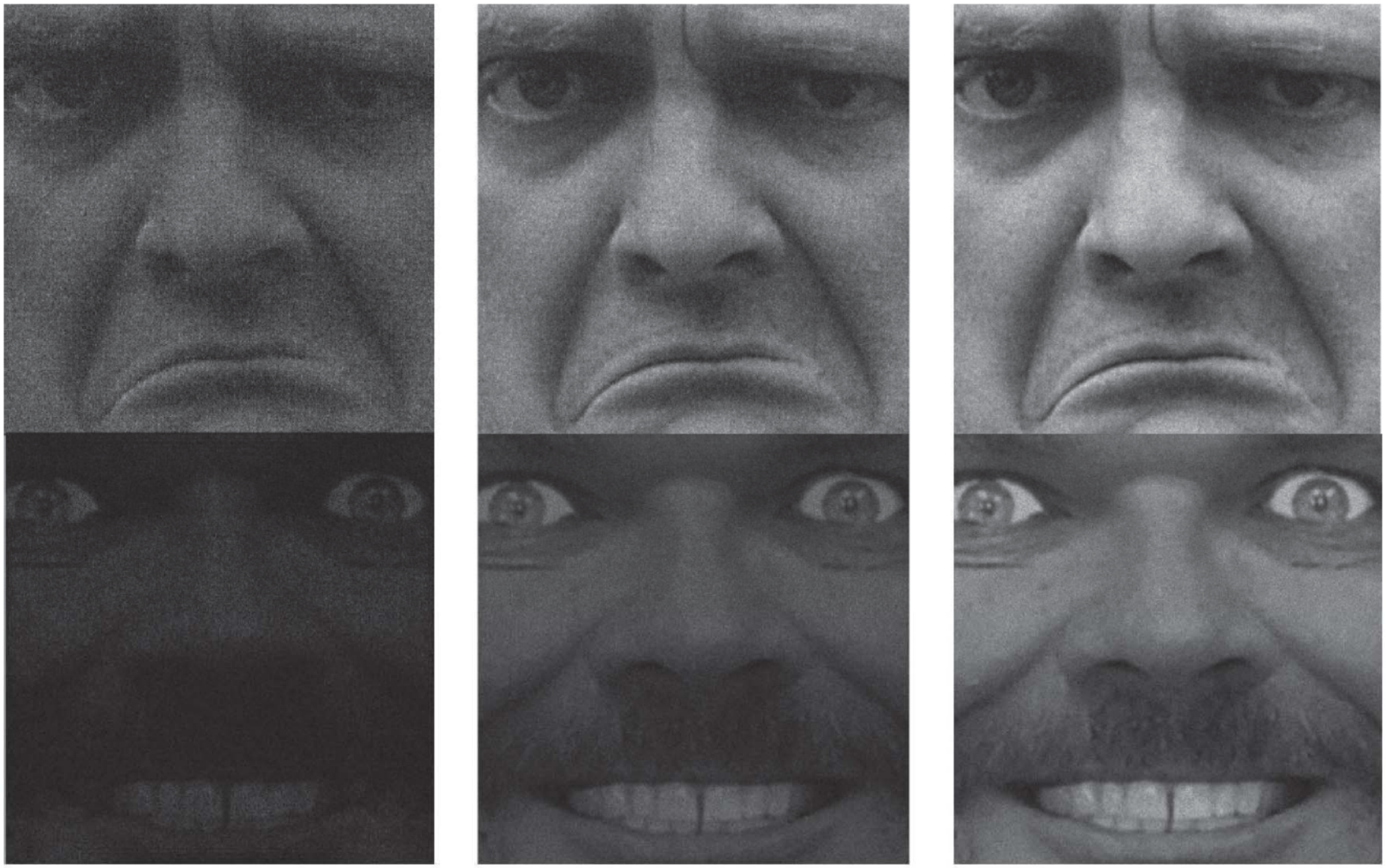
$$f_n = |f_n|\, e^{j\angle f_n} \xleftarrow{\mathcal{F}^{-1}} F_n = |X|\, e^{j\angle G_n}$$

Gerchberg-Saxton Algorithm

Example: Sad to Happy

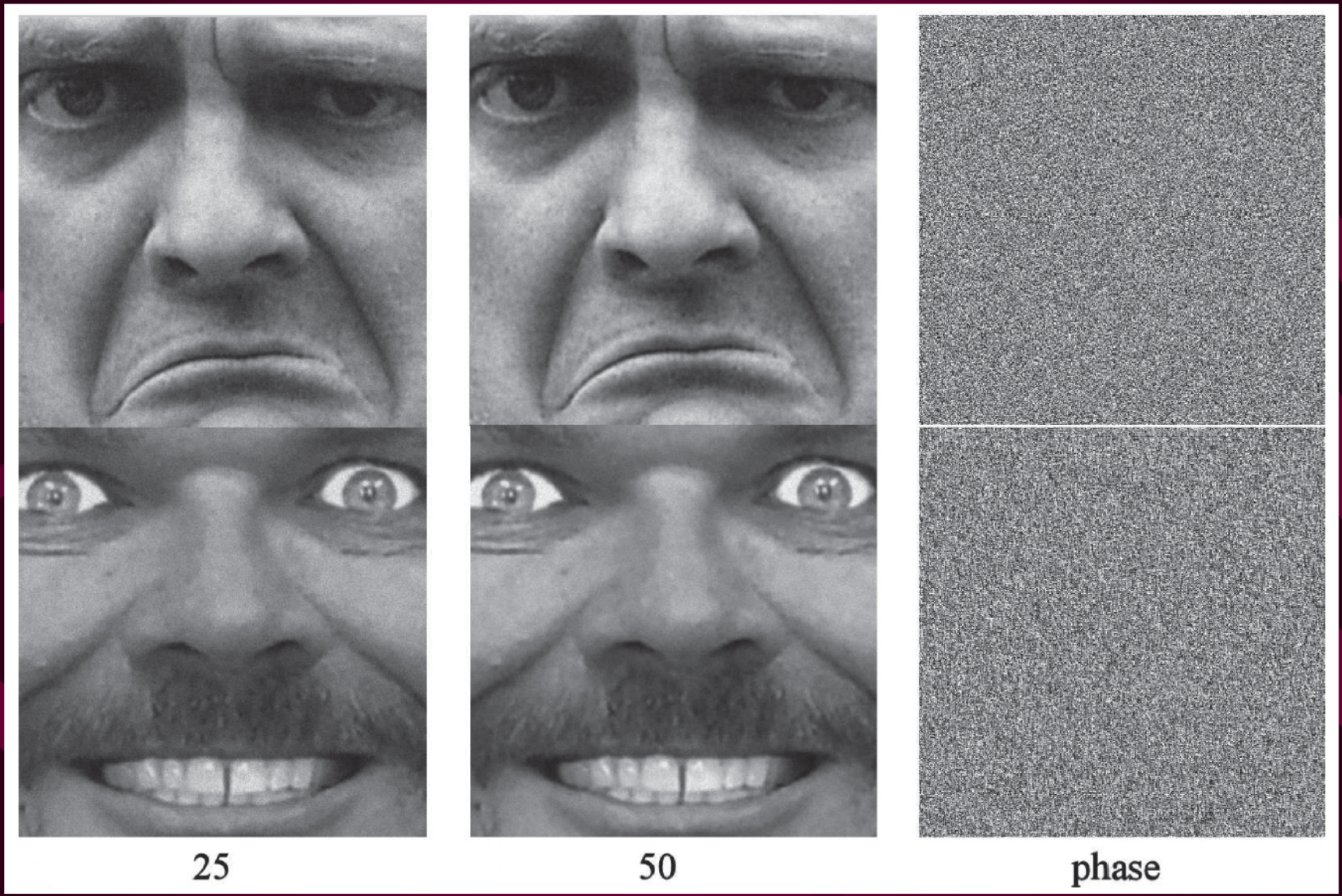


Gerchberg-Saxton Algorithm



2 5 10

Gerchberg-Saxton Algorithm



Final Comments

- POCS: What is it?
  - Convex Sets
  - Projections
  - POCS
- Applications
  - The Papoulis-Gerchberg Algorithm
  - Neural Network Associative Memory
  - Resolution at sub-pixel levels
  - Radiation Oncology / Tomography
  - JPG / MPEG repair
  - Missing Sensors
- Generalized Alternating Projections
  - Ambiguity Function Synthesis
  - The Gerchberg-Saxton Algorithm

- R.J. Marks II, Handbook of Fourier Analysis and Its Applications, Oxford University Press, (2009).
- http://robertmarks.org/REPRINTS/Marks-Pubs.htm