Time Delay Estimation in Gravitationally Lensed Photon Stream Pairs

Michał Staniaszek
Supervisor: Peter Tiňo

The University of Birmingham

March 21, 2013

Outline

What is Gravitational Lensing?

- The bending of light due to gravitational effects
- Objects such as galaxy clusters affect the path of light
- Multiple images of the lensed object can be observed
- Source has a characteristic signal
- Images have the same signal, but with some time delay Δ

What is Gravitational Lensing?

- The bending of light due to gravitational effects
- Objects such as galaxy clusters affect the path of light
- Multiple images of the lensed object can be observed
- Source has a characteristic signal
- Images have the same signal, but with some time delay Δ

What is Gravitational Lensing?

- The bending of light due to gravitational effects
- Objects such as galaxy clusters affect the path of light
- Multiple images of the lensed object can be observed
- Source has a characteristic signal
- Images have the same signal, but with some time delay Δ

What is Gravitational Lensing?

- The bending of light due to gravitational effects
- Objects such as galaxy clusters affect the path of light
- Multiple images of the lensed object can be observed
- Source has a characteristic signal
- Images have the same signal, but with some time delay Δ

What is Gravitational Lensing?

- The bending of light due to gravitational effects
- Objects such as galaxy clusters affect the path of light
- Multiple images of the lensed object can be observed
- Source has a characteristic signal
- Images have the same signal, but with some time delay Δ

Strong vs. Weak Lensing

Strong Lensing

Time delays can be on the order of hundreds of days

- Daily measurements of photon flux used to observe variation

Weak Lensing

Time delays are on a much shorter timescale

- Variation in the signal observed on the order of hours rather than days
- Track individual photon arrival times (streams of photons)

Strong vs. Weak Lensing

Strong Lensing

Time delays can be on the order of hundreds of days

- Daily measurements of photon flux used to observe variation

Weak Lensing

Time delays are on a much shorter timescale

- Variation in the signal observed on the order of hours rather than days
- Track individual photon arrival times (streams of photons)

Strong vs. Weak Lensing

Strong Lensing

Time delays can be on the order of hundreds of days

- Daily measurements of photon flux used to observe variation

Weak Lensing

Time delays are on a much shorter timescale

- Variation in the signal observed on the order of hours rather than days
- Track individual photon arrival times (streams of photons)

Strong vs. Weak Lensing

Strong Lensing

Time delays can be on the order of hundreds of days

- Daily measurements of photon flux used to observe variation

Weak Lensing

Time delays are on a much shorter timescale

- Variation in the signal observed on the order of hours rather than days
- Track individual photon arrival times (streams of photons)

Strong vs. Weak Lensing

Strong Lensing

Time delays can be on the order of hundreds of days

- Daily measurements of photon flux used to observe variation

Weak Lensing

Time delays are on a much shorter timescale

- Variation in the signal observed on the order of hours rather than days
- Track individual photon arrival times (streams of photons)

Aim of the Project

Create a system to estimate the time delay Δ between pairs of photon streams from weakly lensed objects

Motivation

1. Form the base for a system to automatically flag potential lensed objects
   - Lots of data, but analysing it all is difficult
   - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
   - Improved estimates of H_0
   - Dark matter measurements
   - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
    - Lots of data, but analysing it all is difficult
    - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
    - Improved estimates of H_0
    - Dark matter measurements
    - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
    - Lots of data, but analysing it all is difficult
    - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
    - Improved estimates of H_0
    - Dark matter measurements
    - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
   - Lots of data, but analysing it all is difficult
   - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
   - Improved estimates of H_0
   - Dark matter measurements
   - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
   - Lots of data, but analysing it all is difficult
   - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
   - Improved estimates of H_0
   - Dark matter measurements
   - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
   - Lots of data, but analysing it all is difficult
   - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
   - Improved estimates of H_0
   - Dark matter measurements
   - Mass distribution for regions of space

Motivation

1. Form the base for a system to automatically flag potential lensed objects
   - Lots of data, but analysing it all is difficult
   - Flag interesting-looking objects for further investigation
2. Better estimates of time delay are useful
   - Improved estimates of H_0
   - Dark matter measurements
   - Mass distribution for regions of space

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

Photon Simulation

We use a nonhomogeneous poisson process to simulate arrival times.

- Rate parameter λ is the expected number of arrivals per unit time
- Waiting time until the next event has an exponential distribution
- Time to next event in homogeneous process $t = -\frac{1}{\lambda}\ln(U)$, where $U \sim U(0,1)$
- Use thinning on events generated using the above to generate times based on a nonhomogeneous process

Photon Simulation

We use a nonhomogeneous poisson process to simulate arrival times.

- Rate parameter λ is the expected number of arrivals per unit time
- Waiting time until the next event has an exponential distribution
- Time to next event in homogeneous process $t = -\frac{1}{\lambda}\ln(U)$, where $U \sim U(0,1)$
- Use thinning on events generated using the above to generate times based on a nonhomogeneous process

Photon Simulation

We use a nonhomogeneous poisson process to simulate arrival times.

- Rate parameter λ is the expected number of arrivals per unit time
- Waiting time until the next event has an exponential distribution
- Time to next event in homogeneous process $t = -\frac{1}{\lambda}\ln(U)$, where $U \sim U(0, 1)$
- Use thinning on events generated using the above to generate times based on a nonhomogeneous process

Photon Simulation

We use a nonhomogeneous poisson process to simulate arrival times.

- Rate parameter λ is the expected number of arrivals per unit time
- Waiting time until the next event has an exponential distribution
- Time to next event in homogeneous process $t = -\frac{1}{\lambda}\ln(U)$, where $U \sim U(0,1)$
- Use thinning on events generated using the above to generate times based on a nonhomogeneous process

Function Generation

To generate events, need some function $\lambda(t)$

- Randomly generate function by using Gaussians
- Centre Gaussians at uniform intervals Δt, with standard deviation $\alpha \cdot \Delta t$
- Sum the Gaussians to give a continuous function

Function Generation

To generate events, need some function $\lambda(t)$

- Randomly generate function by using Gaussians
- Centre Gaussians at uniform intervals Δt, with standard deviation $\alpha \cdot \Delta t$
- Sum the Gaussians to give a continuous function

Function Generation

To generate events, need some function $\lambda(t)$

- Randomly generate function by using Gaussians
- Centre Gaussians at uniform intervals Δt, with standard deviation $\alpha \cdot \Delta t$
- Sum the Gaussians to give a continuous function

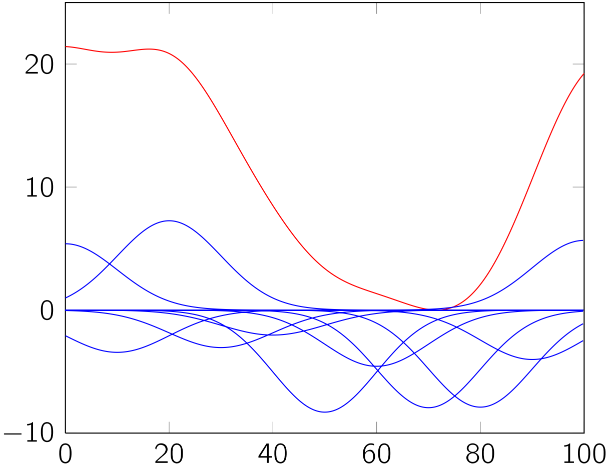


Figure: The red function is generated by summing the blue Gaussians. Gaussian values are multiplied by 3. Function is shifted so all $y \geq 0$

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

General Idea

1. Split the interval into bins
2. Count the number of events that occur in each bin
3. Estimate functions based on these counts

Iterative Weighted Least Squares

Estimate linear functions of the form $y = a + bx$ using Iterative Weighted Least Squares (IWLS)

- Find
$$\min_{\alpha,\beta} \sum_{k=1}^{n} w_k \cdot (Y_k - [\alpha + \beta x])^2$$
- α and β are estimators for a and b, w_k is the weight assigned to each value Y_k, which is the event count for the kth bin. x is the midpoint of the sub-interval.
- Update weights at each iteration by using estimated values of λ in each sub-interval.

Iterative Weighted Least Squares

Estimate linear functions of the form $y = a + bx$ using Iterative Weighted Least Squares (IWLS)

- Find
$$\min_{\alpha,\beta} \sum_{k=1}^{n} w_k \cdot (Y_k - [\alpha + \beta x])^2$$
- α and β are estimators for a and b, w_k is the weight assigned to each value Y_k, which is the event count for the kth bin. x is the midpoint of the sub-interval.
- Update weights at each iteration by using estimated values of λ in each sub-interval.

Iterative Weighted Least Squares

Estimate linear functions of the form $y = a + bx$ using Iterative Weighted Least Squares (IWLS)

- Find
$$\min_{\alpha,\beta} \sum_{k=1}^{n} w_k \cdot (Y_k - [\alpha + \beta x])^2$$
- α and β are estimators for a and b, w_k is the weight assigned to each value Y_k, which is the event count for the kth bin. x is the midpoint of the sub-interval.
- Update weights at each iteration by using estimated values of λ in each sub-interval.

Piecewise

Some parts of functions can be reasonably approximated by straight lines

- Split the interval into several subintervals and estimate each in turn
- Once an estimate is done, extend the line to probe the next interval
- If the extension matches the data, keep it

Piecewise

Some parts of functions can be reasonably approximated by straight lines

- Split the interval into several subintervals and estimate each in turn
- Once an estimate is done, extend the line to probe the next interval
- If the extension matches the data, keep it

Piecewise

Some parts of functions can be reasonably approximated by straight lines

- Split the interval into several subintervals and estimate each in turn
- Once an estimate is done, extend the line to probe the next interval
- If the extension matches the data, keep it

Baseline

Characteristic functions of photon streams are continuous - must make the piecewise estimate continuous as well.

- Modify each interval estimate to make a continuous function
- At each breakpoint, find the midpoint between the estimates
- Modify function values to make the end point of one interval estimate meet the start of the next

Baseline

Characteristic functions of photon streams are continuous - must make the piecewise estimate continuous as well.

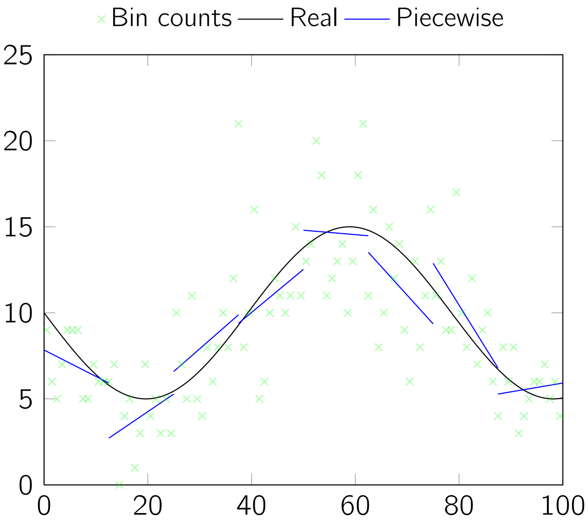
- Modify each interval estimate to make a continuous function
- At each breakpoint, find the midpoint between the estimates
- Modify function values to make the end point of one interval estimate meet the start of the next

Baseline

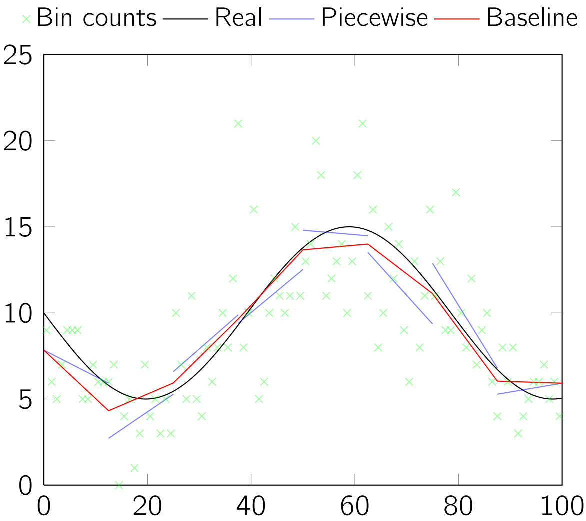
Characteristic functions of photon streams are continuous - must make the piecewise estimate continuous as well.

- Modify each interval estimate to make a continuous function
- At each breakpoint, find the midpoint between the estimates
- Modify function values to make the end point of one interval estimate meet the start of the next

Piecewise Estimate Example



Baseline Estimate vs Piecewise Estimate



Kernel Density

- Centre a Gaussian kernel at each event time
- Sum Gaussians to approximate the function
- Must be normalised depending on standard deviation used
- Use probability density function to automatically calculate normalisation constant

Kernel Density

- Centre a Gaussian kernel at each event time
- Sum Gaussians to approximate the function
- Must be normalised depending on standard deviation used
- Use probability density function to automatically calculate normalisation constant

Kernel Density

- Centre a Gaussian kernel at each event time
- Sum Gaussians to approximate the function
- Must be normalised depending on standard deviation used
- Use probability density function to automatically calculate normalisation constant

Kernel Density

- Centre a Gaussian kernel at each event time
- Sum Gaussians to approximate the function
- Must be normalised depending on standard deviation used
- Use probability density function to automatically calculate normalisation constant

Three main parts of the system

1. Photon stream simulation
2. Function estimation
3. Time delay estimation

General Idea

The actual Δ is not known, so we make guesses and check to see how good they are.

- Estimate each stream of photons
- Pick a value of Δ and shift the function estimate
- Compare it to the other estimate and see how good the match is
- Hierarchical - coarse first pass, improve estimate with finer second pass

General Idea

The actual Δ is not known, so we make guesses and check to see how good they are.

- Estimate each stream of photons
- Pick a value of Δ and shift the function estimate
- Compare it to the other estimate and see how good the match is
- Hierarchical - coarse first pass, improve estimate with finer second pass

General Idea

The actual Δ is not known, so we make guesses and check to see how good they are.

- Estimate each stream of photons
- Pick a value of Δ and shift the function estimate
- Compare it to the other estimate and see how good the match is
- Hierarchical - coarse first pass, improve estimate with finer second pass

General Idea

The actual Δ is not known, so we make guesses and check to see how good they are.

- Estimate each stream of photons
- Pick a value of Δ and shift the function estimate
- Compare it to the other estimate and see how good the match is
- Hierarchical - coarse first pass, improve estimate with finer second pass

Area Between Curves

1. Approximate the area between the two function estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$

$$
\begin{aligned}
d(\hat{\lambda}_1, \hat{\lambda}_2) &= \int (\hat{\lambda}_1(t) - \hat{\lambda}_2(t))^2 \, dt \\
&\approx \frac{1}{N} \sum_{i=1}^{N} (\hat{\lambda}_1(t) - \hat{\lambda}_2(t))^2
\end{aligned}
$$

2. Find the value of Δ for which $d(\hat{\lambda}_1, \hat{\lambda}_2)$ is minimised

Area Between Curves

1. Approximate the area between the two function estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$

$$
\begin{aligned}
d(\hat{\lambda}_1, \hat{\lambda}_2) &= \int (\hat{\lambda}_1(t) - \hat{\lambda}_2(t))^2 \, dt \\
&\approx \frac{1}{N} \sum_{i=1}^{N} (\hat{\lambda}_1(t) - \hat{\lambda}_2(t))^2
\end{aligned}
$$

2. Find the value of Δ for which $d(\hat{\lambda}_1, \hat{\lambda}_2)$ is minimised

Probability Density

1. Pick a value of Δ
2. Combine function estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$ into an "average" function $\overline{\lambda}$, where

$$\overline{\lambda}(t) = \frac{\hat{\lambda}_1(t) + \hat{\lambda}_2(t+\Delta)}{2}$$

3. See how well $\overline{\lambda}$ matches the data from the two streams by maximising

$$\log P(S_A, S_B \mid \overline{\lambda}(t)) = \sum_{t=\Delta}^{T-\Delta} \log P(S_A(t) \mid \overline{\lambda}(t)) + \log P(S_B(t+\Delta) \mid \overline{\lambda}(t))$$

Probability Density

1. Pick a value of Δ
2. Combine function estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$ into an "average" function $\overline{\lambda}$, where

$$\overline{\lambda}(t) = \frac{\hat{\lambda}_1(t) + \hat{\lambda}_2(t+\Delta)}{2}$$

3. See how well $\overline{\lambda}$ matches the data from the two streams by maximising

$$\log P(S_A, S_B \mid \overline{\lambda}(t)) = \sum_{t=\Delta}^{T-\Delta} \log P(S_A(t) \mid \overline{\lambda}(t)) + \log P(S_B(t+\Delta) \mid \overline{\lambda}(t))$$

Probability Density

1. Pick a value of Δ
2. Combine function estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$ into an "average" function $\overline{\lambda}$, where

$$\overline{\lambda}(t) = \frac{\hat{\lambda}_1(t) + \hat{\lambda}_2(t+\Delta)}{2}$$

3. See how well $\overline{\lambda}$ matches the data from the two streams by maximising

$$\log P(S_A, S_B \mid \overline{\lambda}(t)) = \sum_{t=\Delta}^{T-\Delta} \log P(S_A(t) \mid \overline{\lambda}(t)) + \log P(S_B(t+\Delta) \mid \overline{\lambda}(t))$$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha \cdot \Delta t$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha\Delta t$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha \cdot \Delta t$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha \cdot \Delta t$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha \cdot \Delta t$

Experimental Setup

Three sets of experiments

1. Preliminary sine function experiments
   - Vary α in $y = a - b\sin(\alpha t)$
   - Higher α leads to faster oscillation
2. Experiments on a smaller range to see degradation
3. Random functions
   - Vary α where standard deviation of Gaussian $\sigma = \alpha \cdot \Delta t$

Experimental Method

1. Perform model selection on each stream
   - withhold some event data
   - Determine optimal parameters for each set of streams
2. Estimate time delay for each pair of streams using optimal parameters (with all data)

Experimental Method

1. Perform model selection on each stream
   - withhold some event data
   - Determine optimal parameters for each set of streams
2. Estimate time delay for each pair of streams using optimal parameters (with all data)

Experimental Method

1. Perform model selection on each stream
   - withhold some event data
   - Determine optimal parameters for each set of streams
2. Estimate time delay for each pair of streams using optimal parameters (with all data)

Experimental Method

1. Perform model selection on each stream
   - withhold some event data
   - Determine optimal parameters for each set of streams
2. Estimate time delay for each pair of streams using optimal parameters (with all data)

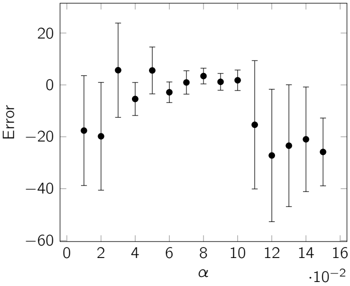
Results

- Area estimator better than PDF, but significance not high
- Both types of estimators not significantly different

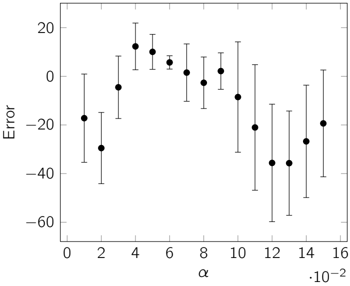
	Gaussian	Baseline
Area	15.95 ± 4.51	15.99 ± 3.09
PDF	16.53 ± 11.80	15.72 ± 14.06

Figure: Experimental results for $\alpha = 0.07$ in the second set of sine function experiments ($\mu \pm \sigma$, $n = 10$). Actual delay is 15.

Baseline Estimator Degradation

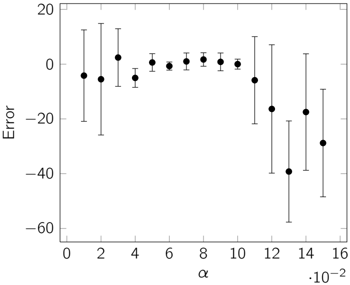


(a) Baseline area

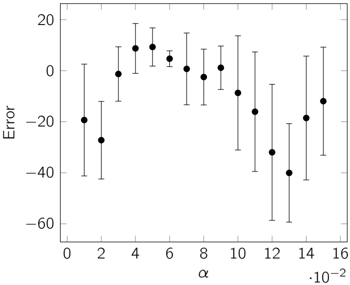


(b) Baseline PDF

Gaussian Estimator Degradation

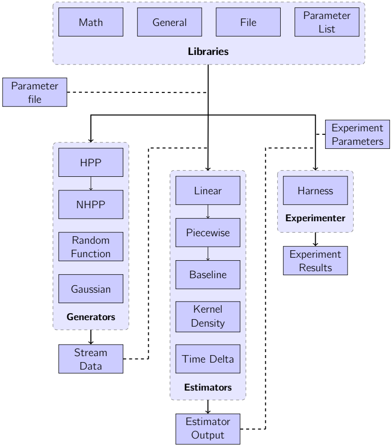


(c) Gaussian area



(d) Gaussian PDF

System Structure



- Modular structure
- Shared libraries for common functions
- Parameter files control behaviour
- Command line interface
- 7000 lines of C code
- Testing using Check utility
- Uses Automake
- In the style of a GNU package

Summary

- We want to find the value of Δ, the time delay between characteristic functions of photon streams
- Photon stream simulation using Poisson processes
- Estimation of characteristic function of stream using baseline or kernel density estimators
- Estimation of time delay with PDF or area estimators
- Experimental results indicate area estimator is better than PDF, but significance is not high

Summary

- We want to find the value of Δ, the time delay between characteristic functions of photon streams
- Photon stream simulation using Poisson processes
- Estimation of characteristic function of stream using baseline or kernel density estimators
- Estimation of time delay with PDF or area estimators
- Experimental results indicate area estimator is better than PDF, but significance is not high

Summary

- We want to find the value of Δ, the time delay between characteristic functions of photon streams
- Photon stream simulation using Poisson processes
- Estimation of characteristic function of stream using baseline or kernel density estimators
- Estimation of time delay with PDF or area estimators
- Experimental results indicate area estimator is better than PDF, but significance is not high

Summary

- We want to find the value of Δ, the time delay between characteristic functions of photon streams
- Photon stream simulation using Poisson processes
- Estimation of characteristic function of stream using baseline or kernel density estimators
- Estimation of time delay with PDF or area estimators
- Experimental results indicate area estimator is better than PDF, but significance is not high

Summary

- We want to find the value of Δ, the time delay between characteristic functions of photon streams
- Photon stream simulation using Poisson processes
- Estimation of characteristic function of stream using baseline or kernel density estimators
- Estimation of time delay with PDF or area estimators
- Experimental results indicate area estimator is better than PDF, but significance is not high