



Спецкурс ОСФИ

Лекция 9

20 апреля 2011

Сэмплирование и реконструкция

Алексей Игнатенко, к.ф.-м.н.

Лаборатория компьютерной графики и
мультимедиа ВМК МГУ

Информация для 2-го курса

Собеседование

Лаборатория компьютерной графики и мультимедиа

25 апреля, в 16-20, в ауд. 702

План

- Сэмплирование (дискретизация) и реконструкция
- Напоминание основ ЦОС (DSP)
- Алиасинг и анти-алиасинг

Сенсор = сэмплер (дискретизатор)

Все сенсоры преобразуют непрерывное изображение в дискретное (сэмплированное) с помощью интегрирования по площади сенсора

Пример:

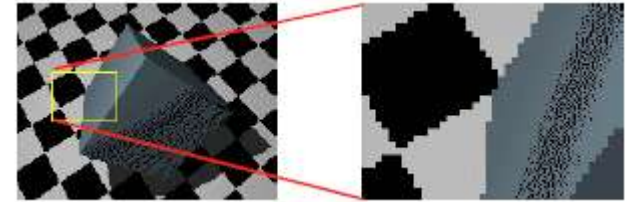
- Сетчатка: колбочки, палочки
- CCD-матрица

Дисплей – Реконструкция сигнала

Дисплеи реконструируют непрерывный сигнал с помощью источников света конечного размера для каждого пикселя

Проблемы сэмплирования в компьютерной графике

- «Лестничный эффект»
- Муар
- Мерцание мелких объектов
- Мерцание бликов
- Стробирование



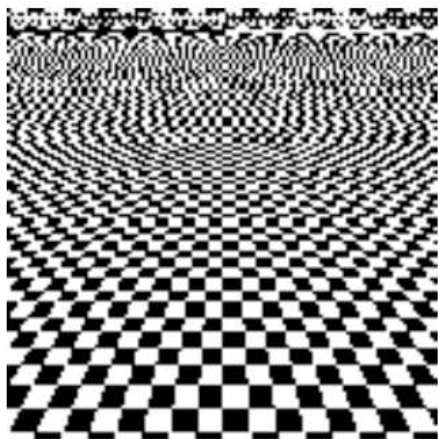
Устранение этих эффектов - антиалиасинг

Анти-алиасинг

Aliasing

sample

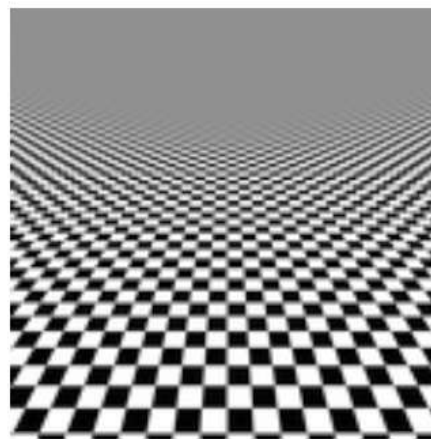
sample



Antialiasing

sample

sample

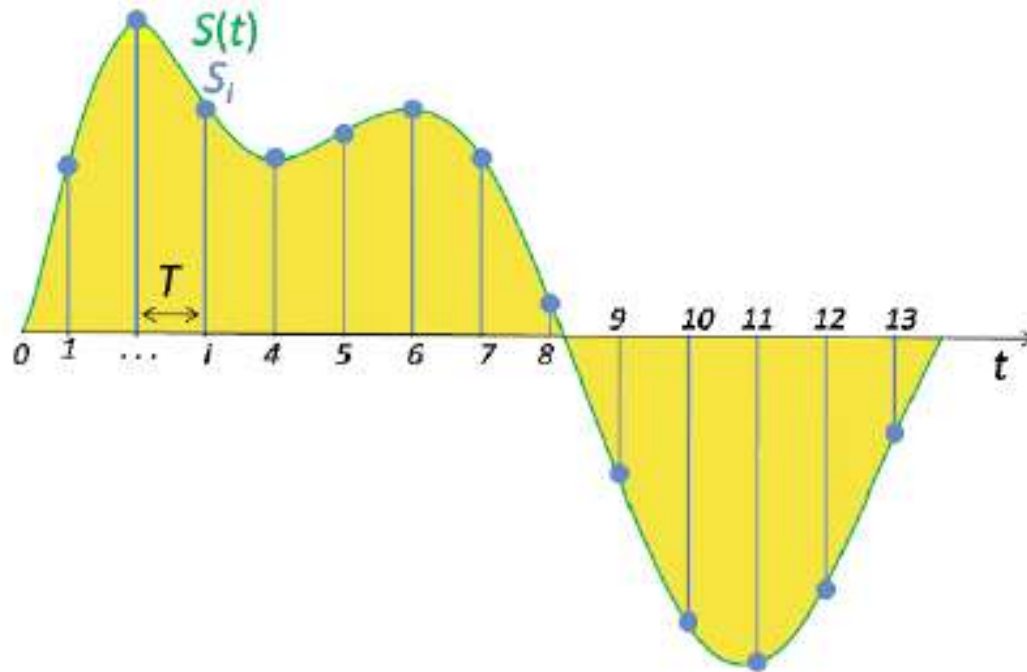


А какое отношение это имеет к синтезу изображений?

- При генерации изображения мы имеем дело с непрерывным L
- Надо преобразовать в дискретное изображение
- Может это сделать с помощью сэмплирования L в некоторых точках
- Исходная функция теряется
- Какие точки выбрать, чтобы
 - 1) была приемлемая производительность
 - 2) максимально приблизиться к непрерывной исходной функции

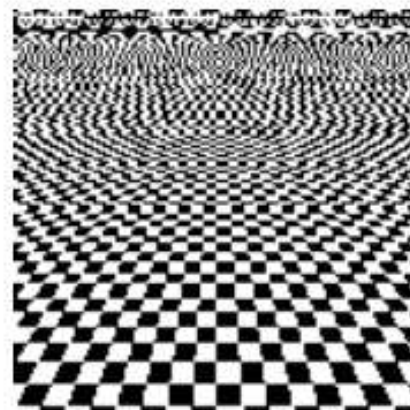
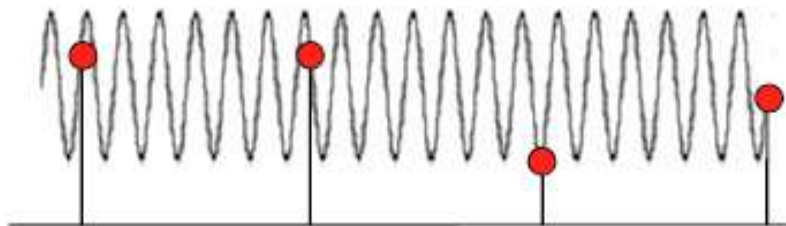
Основная идея дискретизации

Дискретизация сигнала – чтение непрерывного сигнала в некоторых точках



Что является причиной алиасинга?

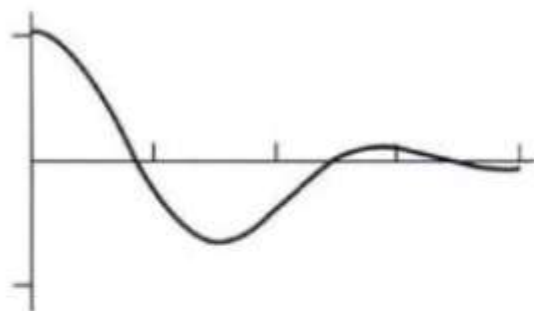
Слишком низкая частота дискретизации не позволяет оцифровать высокочастотные вариации сигнала



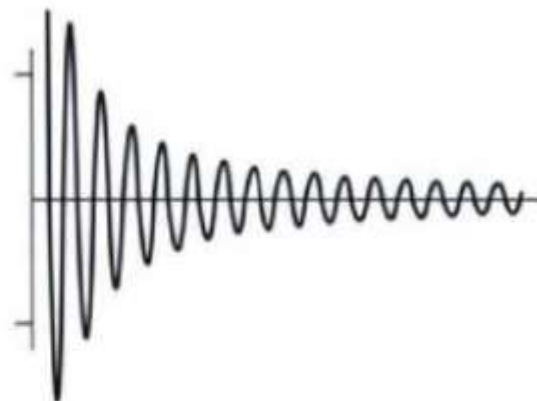
«Заворачивание» частот

Формализация

- Необходимо формализовать процесс, чтобы численно оценить ошибку, вносимую во время дискретизации и реконструкции
- Преобразование Фурье



(a)



(b)

Figure 7.2: (a) Low-frequency function, and (b) high-frequency function. Roughly speaking, the higher frequency a function is, the more quickly it varies over a given region.

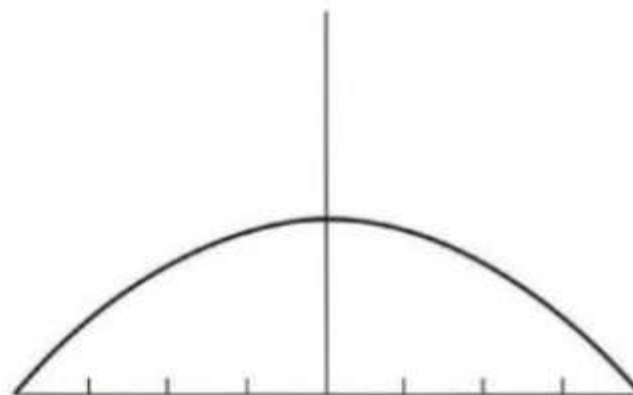
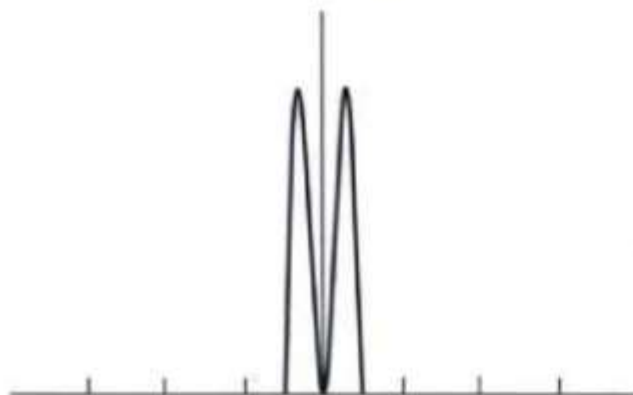
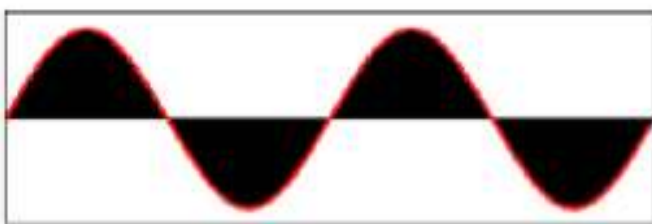


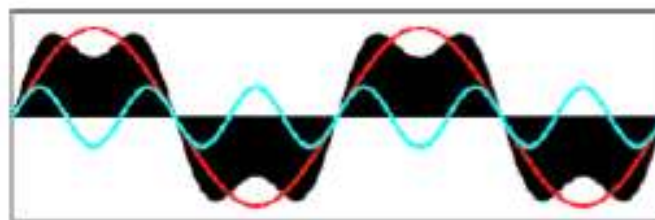
Figure 7.3: Frequency Space Representations of the Functions in Figure 7.2. The graphs show the contribution of each frequency ω to each of the functions in the spatial domain.

Преобразование Фурье

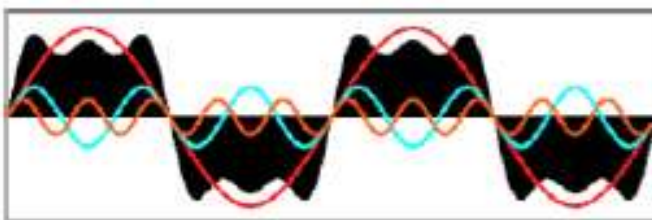
Сигнал может быть преобразован в компоненты различных частот с помощью преобразования Фурье



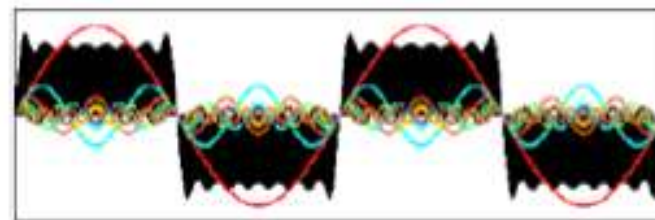
Frequencies: f



Frequencies: $f + 3f$



Frequencies: $f + 3f + 5f$



Frequencies: $f + 3f + \dots + 15f$

Прямое и обратное преобразование Фурье

$$F(\omega) = \int f(x) e^{-i2\pi\omega x} dx$$

$$e^{ix} = \cos x + i \sin x$$

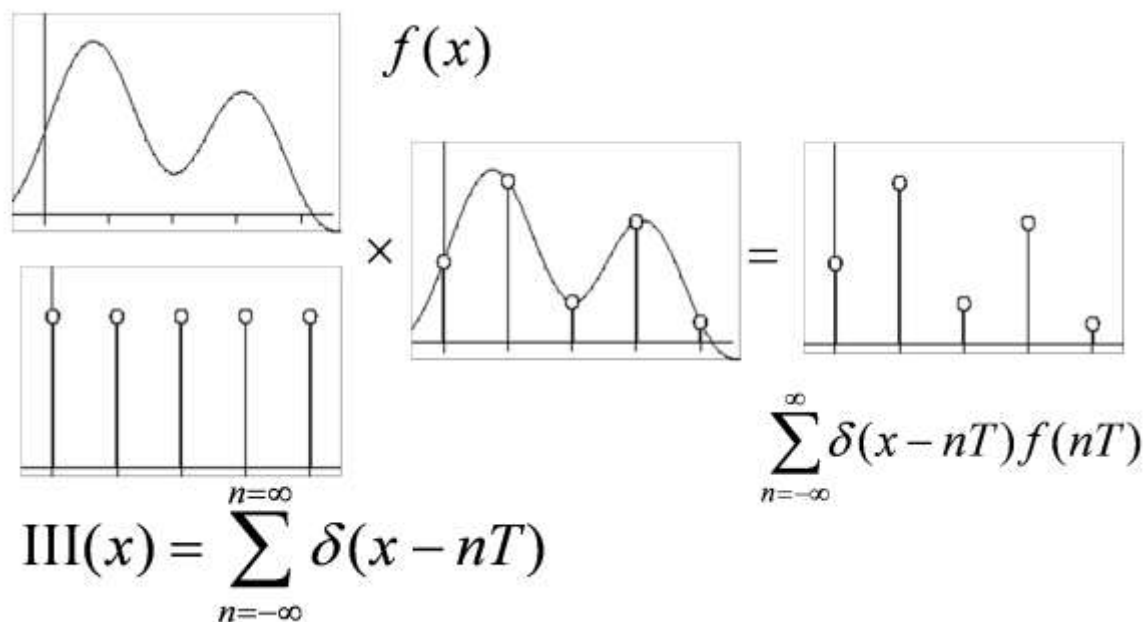
$$f(x) = \int F(\omega) e^{i2\pi\omega x} d\omega$$

-
- Каждая функция имеет два представления
 - Пространственное
 - Спектральное
 - Преобразование Фурье преобразует между пространственным и спектральным представлением

Преобразования Фурье для некоторых функций

Spatial Domain	Frequency Space Representation
Box: $f(x) = 1$ if $ x < 1/2$, 0 otherwise	Sinc: $f(\omega) = \text{sinc}(\omega) = \sin(\pi\omega)/(\pi\omega)$
Gaussian: $f(x) = e^{-\pi x^2}$	Gaussian: $f(\omega) = e^{-\pi\omega^2}$
Constant: $f(x) = 1$	Delta: $f(\omega) = \delta(\omega)$
Sinusoid: $f(x) = \cos x$	Translated delta: $f(\omega) = \pi(\delta(1/2 - \omega) + \delta(1/2 + \omega))$
Shah: $f(x) = \text{III}_T(x) = T \sum_i \delta(x - Ti)$	Shah: $f(\omega) = \text{III}_{1/T}(\omega) = (1/T) \sum_i \delta(\omega - i/T)$

Формализация сэмплирования



- Равномерная дискретизация моделируется «гребнем дирака» – набором дельта-функций с периодом T

Реконструкция с помощью свертки

$$\text{III}_T(x)f(x) = T \sum_i \delta(x - iT)f(iT).$$

These sample values can be used to define a reconstructed function \tilde{f} by choosing a reconstruction filter function $r(x)$ and computing the *convolution*

$$(\text{III}_T(x)f(x)) \otimes r(x),$$

where the convolution operation \otimes is defined as

$$f(x) \otimes g(x) = \int_{-\infty}^{\infty} f(x')g(x - x') \, dx'.$$

Реконструированная функция

For reconstruction, convolution gives a weighted sum of scaled instances of the reconstruction filter centered at the sample points:

$$\tilde{f}(x) = T \sum_{i=-\infty}^{\infty} f(iT)r(x - iT).$$

Свертка и преобразование Фурье

Definition

$$h(x) = f \otimes g = \int f(x')g(x - x') dx'$$

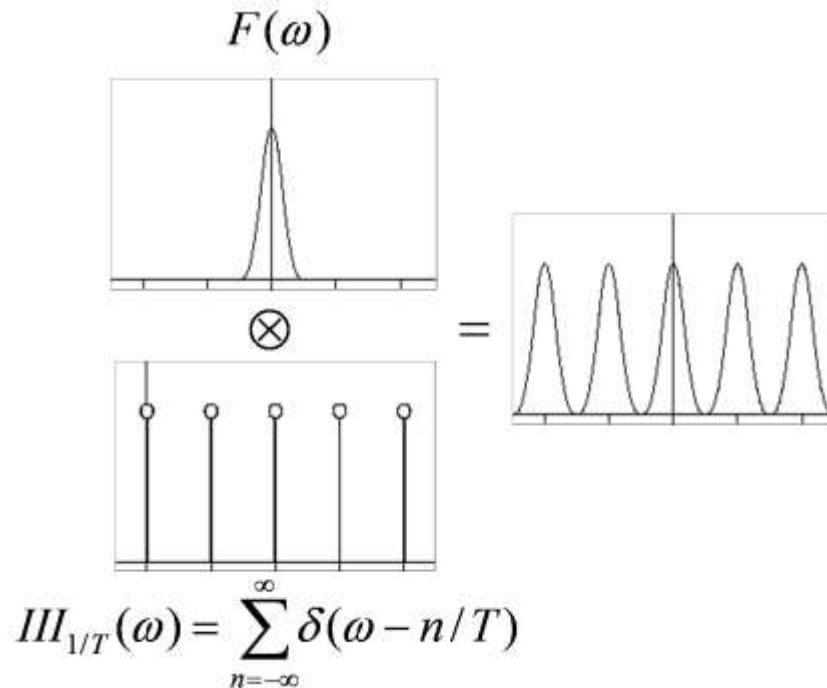
Convolution Theorem: Multiplication in the frequency domain is equivalent to convolution in the space domain.

$$f \otimes g \leftrightarrow F \times G$$

Symmetric Theorem: Multiplication in the space domain is equivalent to convolution in the frequency domain.

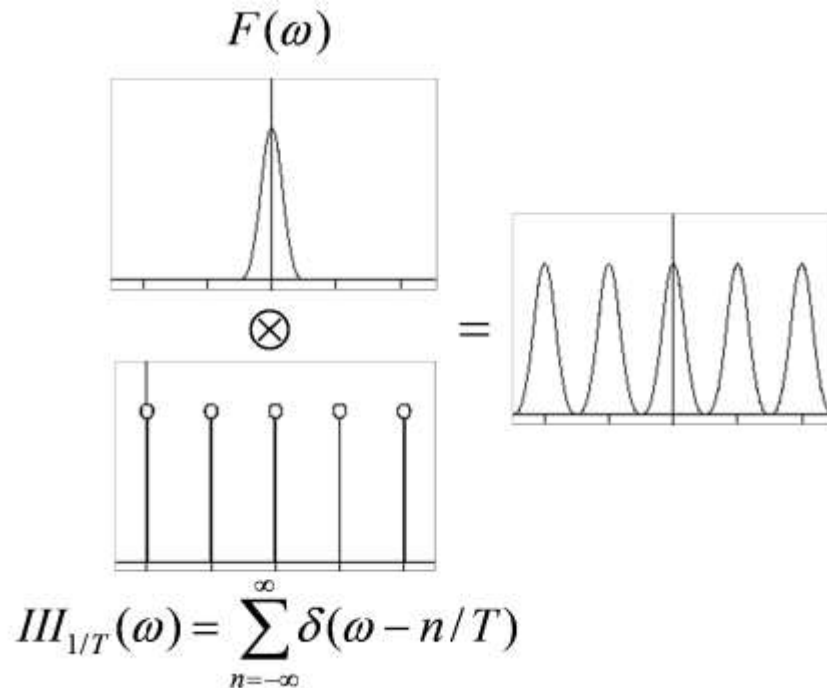
$$f \times g \leftrightarrow F \otimes G$$

Сэмплирование в спектральном пространстве



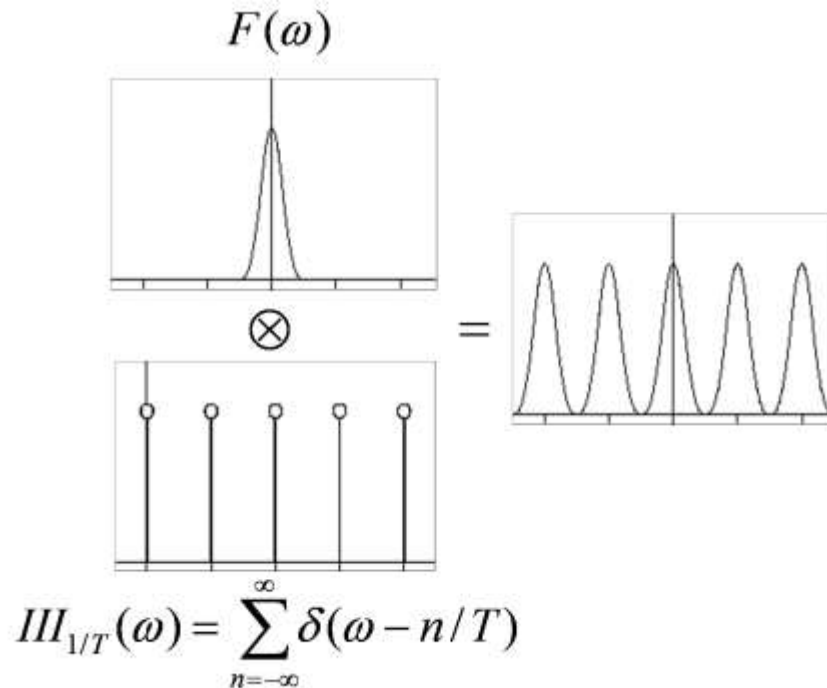
- Получаем спектр функции f
- Получаем спектр гребня Дирака (шаг $1/T$)

Сэмплирование в спектральном пространстве



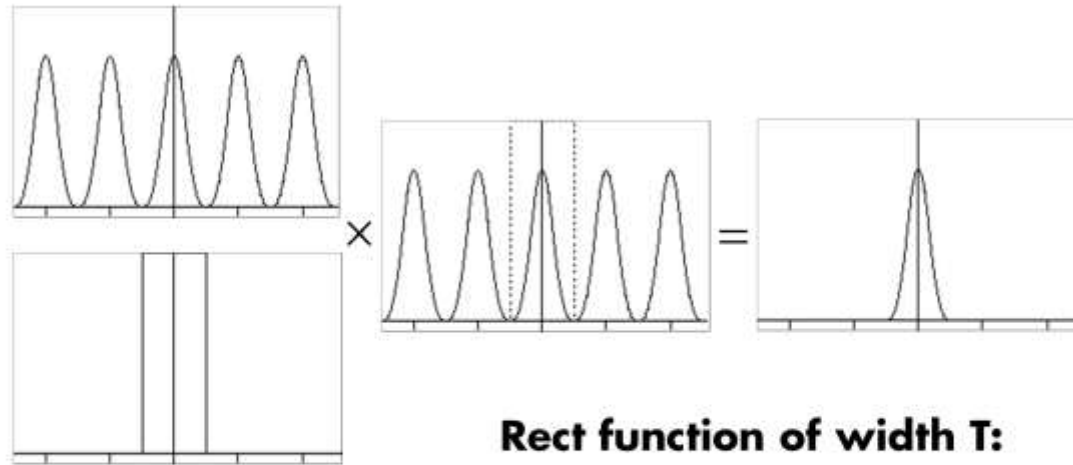
- Проводим свертку
 - Свертка сигнала дельта-функцией есть сигнал
 - Т.е. свертка гребенкой есть размножение сигнала с шагом $1/T$

Сэмплирование в спектральном пространстве



- Обратите внимание: чем больше T , тем ближе становятся копии сигналов

Реконструкция функции в частотной области

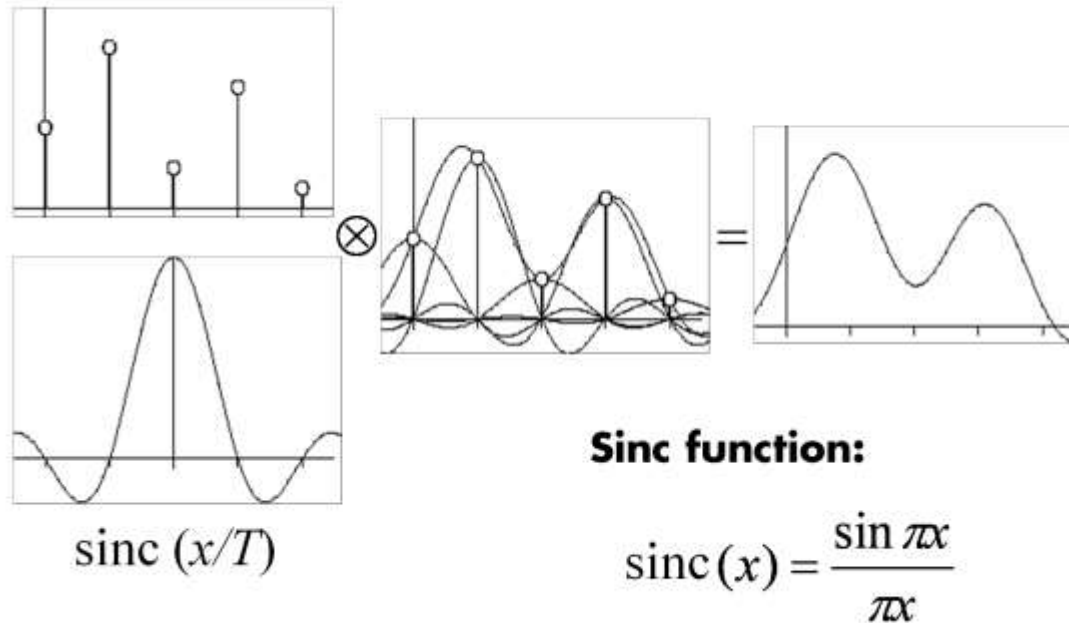


$$\Pi_{1/T}(x)$$

$$\Pi_T(x) = \begin{cases} 1 & |x| \leq \frac{T}{2} \\ 0 & |x| > \frac{T}{2} \end{cases}$$

- Как реконструировать функцию из результирующего спектра?
- Отрезаем лишнее с помощью box-функции

Реконструкция функции в пространственной области

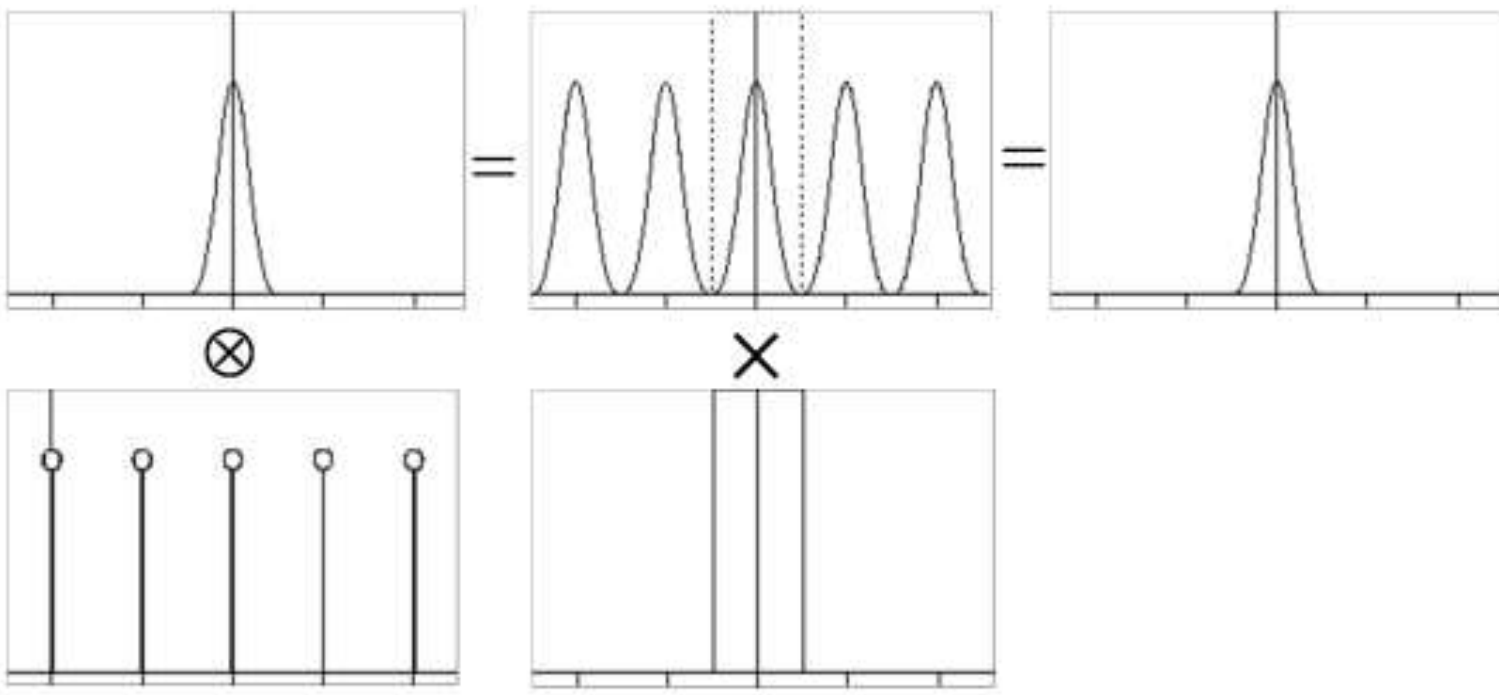


- Эквивалентна свертке
- Преобразование Фурье от box-функции -- sinc

Идеальная реконструкция – проблемы

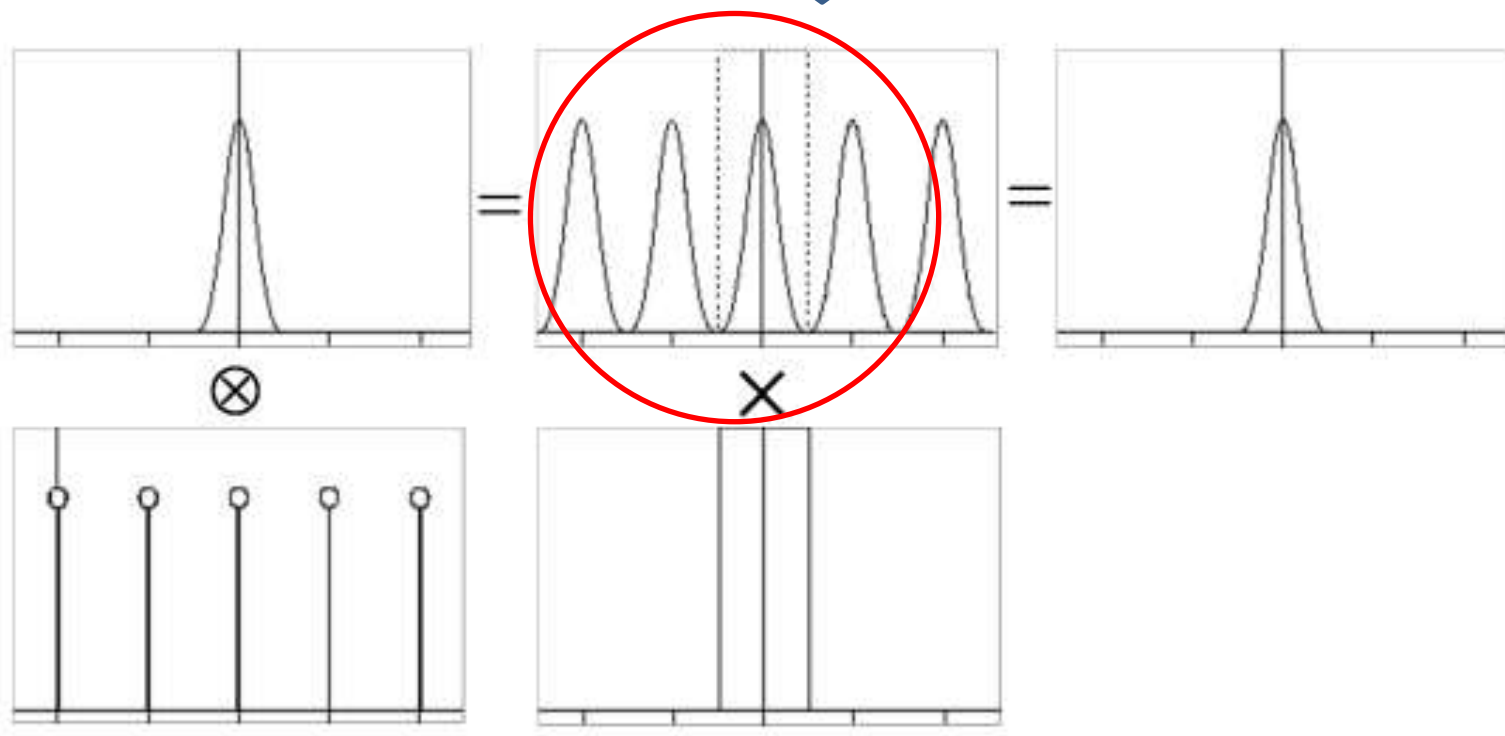
- Sinc имеем бесконечную поддержку («ширину»), что неудобно на практике
- Применение sinc-функции может привести к появлению заметных артефактов в виде колец

Сэмплинг и реконструкция

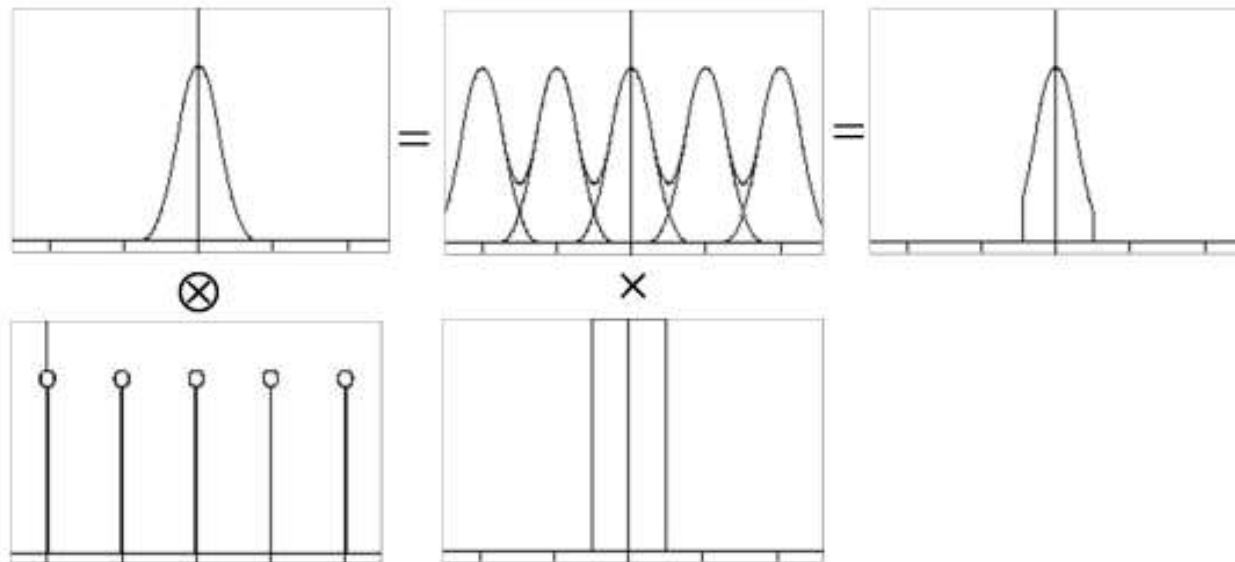


Алиасинг

Что если копии функций будут пересекаться?

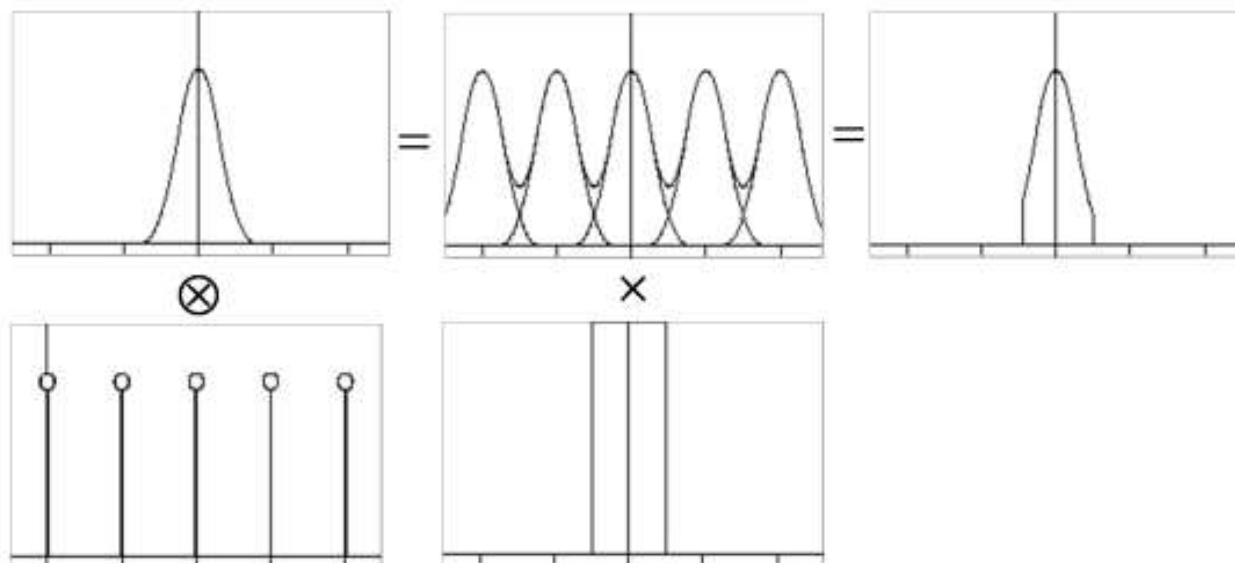


Алиасинг



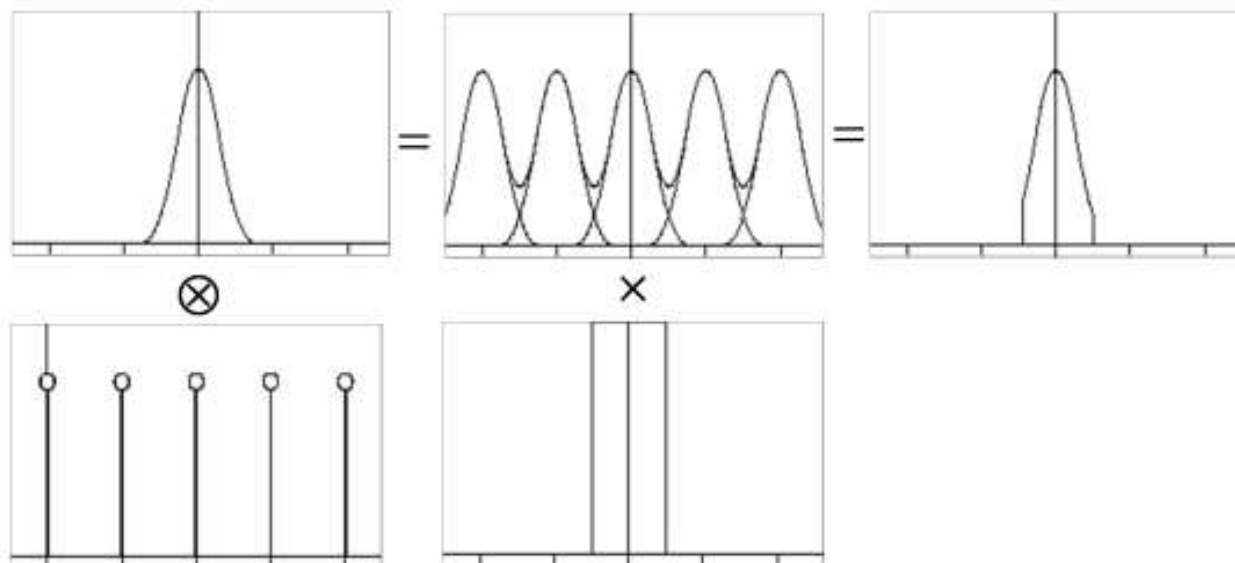
- Если T недостаточно большой, функции начинают накладываться друг на друга
- Это приводит к **наложению спектров**
- Еще называется алиасинг

Алиасинг



- Высокие частоты «протекают» в низкие частоты
- Alias = маска (высокие частоты маскируются как низкие частоты)

Алиасинг



- Какой нужно взять шаг, чтобы не получить наложения спектров?
- Ответ: теорема Найквиста-Котельникова-Шеннона

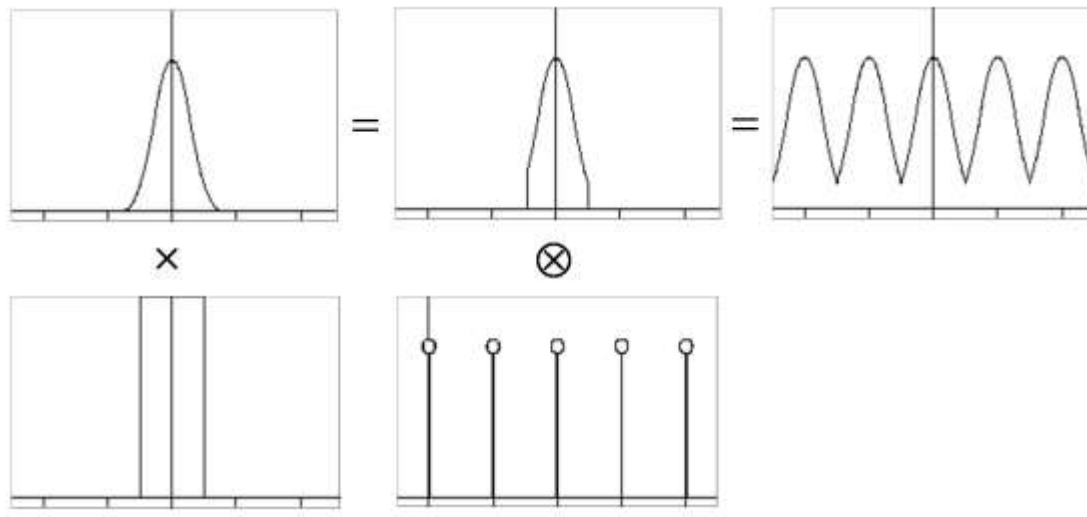
Теорема Найквиста-Шеннона (теорема Котельникова)

- Если сигнал
 - Не имеет компонент частотой выше B
 - Дискретизируется с частотой как минимум $2B$
- То он может быть полностью восстановлен после дискретизации
- Если система проводит дискретизацию с частотой B (например, частота пикселей на экране), то $B/2$ – максимальная частота сигнала, которую она может воспроизвести

Способы антиалиасинга

- **Пре-фильтрация.** Рассчитать низко-частотную версию из непрерывного представления, затем оцифровывать
 - Пример: фильтры перед сенсорами камеры
 - Пример: аналитический расчет заполнения пикселей (для линий, полигонов)
- **Равномерный суперсэмплинг и пост-фильтрация.** Дискретизация с заведомо высокой частотой, затем фильтрация с целью удалить высочастотные компоненты
- **Неравномерный или стохастический суперсэмплинг**

Пре-фильтрация



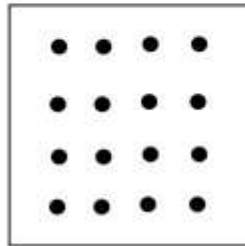
- Заранее удаляем частоты из функции, которые мы не сможем оцифровать
- В теории нужна фильтрация (размытие) sinc-функцией. На практике работает размытие конечной ширины

Однородный супер-сэмплинг

Uniform Supersampling

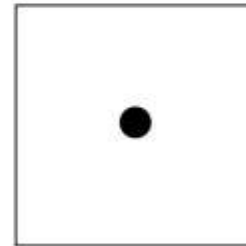
Increasing the sampling rate moves each copy of the spectra further apart, potentially reducing the overlap and thus aliasing

Resulting samples must be resampled (filtered) to image sampling rate



Samples

$$Pixel = \sum_s w_s \cdot Sample_s$$

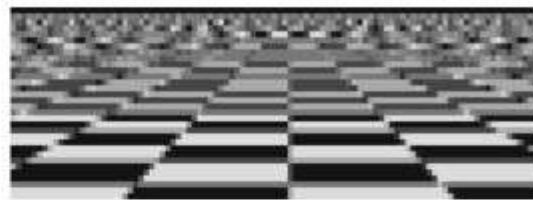


Pixel

Point vs. Supersampled



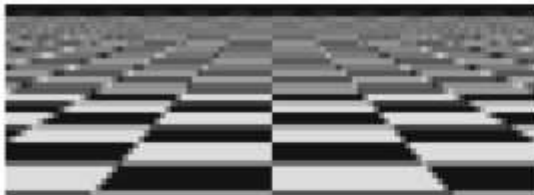
Point



4x4 Supersampled

Checkerboard sequence by Tom Duff

Analytic vs. Supersampled



Exact Area

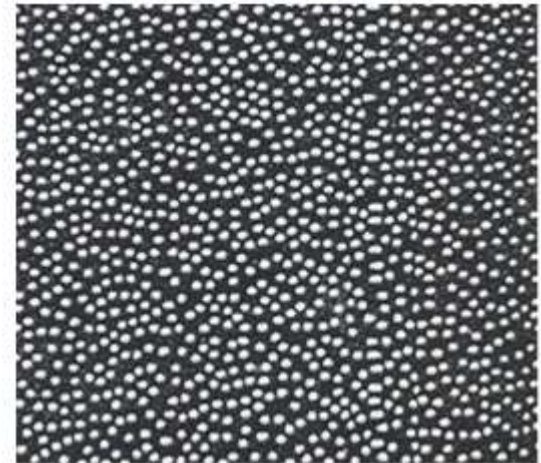


4x4 Supersampled

Стохастический суперсэмплинг

Равномерное сэмплирование

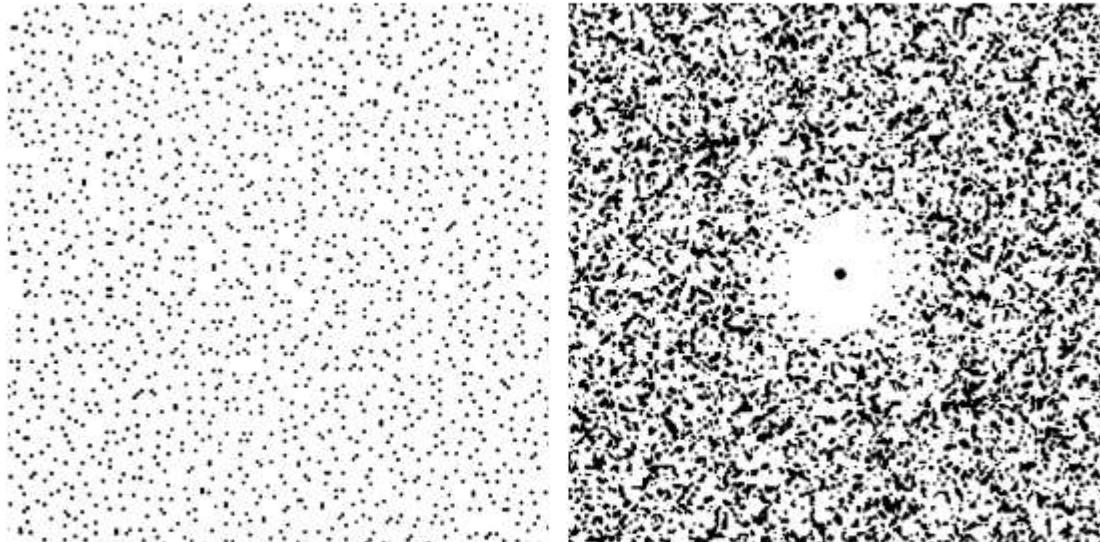
- Спектр равномерных пиков тоже равномерный
- Произведение сигнала на гребенку спектра равно помещению копии сигнала на каждый пик (в частотной области)
- => Артефакты равномерны и поэтому очень заметны



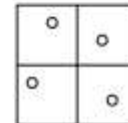
Неравномерное сэмплирование

- Расположить то же количество пиков (выборок) на случайных местах
- Это не поможет восстановить исходный сигнал в точности
- Однако паттерн артефактов примет форму шума и станет менее заметным, т.е. зрение менее чувствительно к высокочастотному шуму

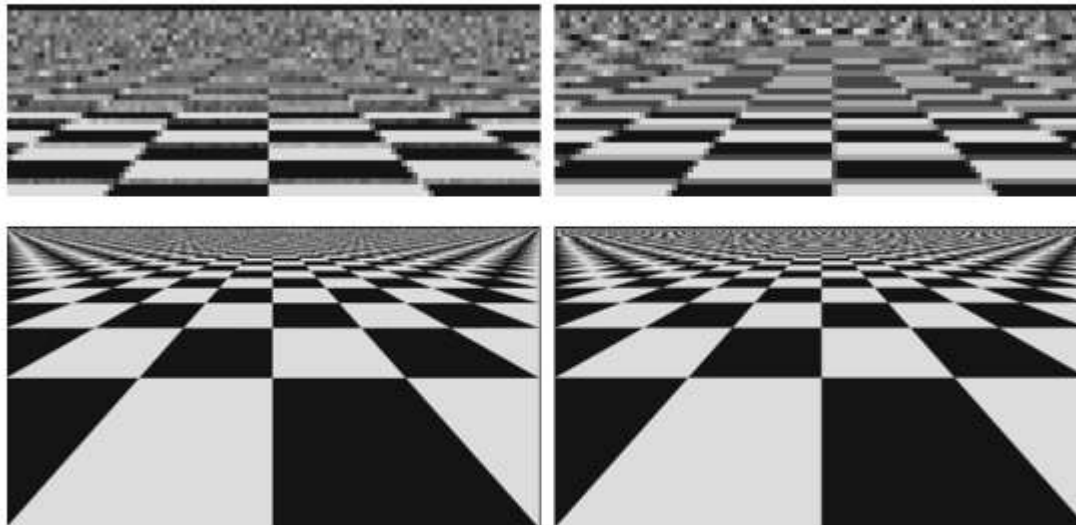
Jittered Sampling



Add uniform random jitter to each sample



Jittered vs. Uniform Supersampling



4x4 Jittered Sampling

4x4 Uniform

Адаптивное сэмплирование

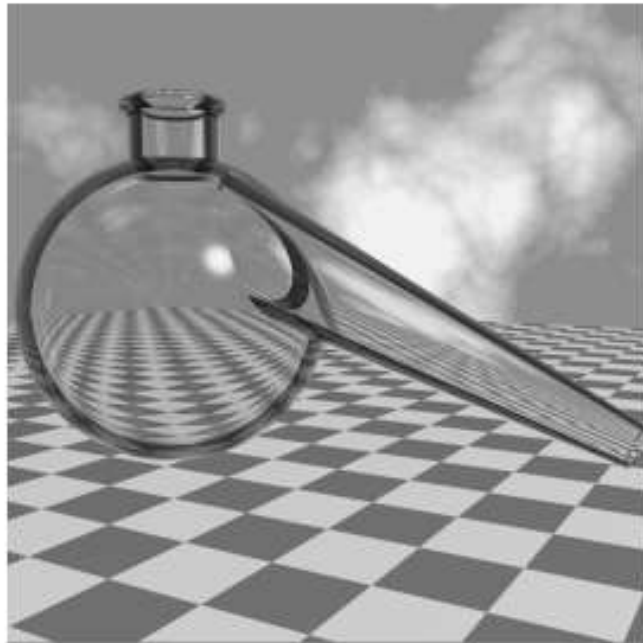
- Найти места сигнала с потенциально высокими частотами и увеличить частоту дискретизации локально
- Часто нужно найти после равномерной дискретизации. Сложная задача!

Применение к синтезу изображений

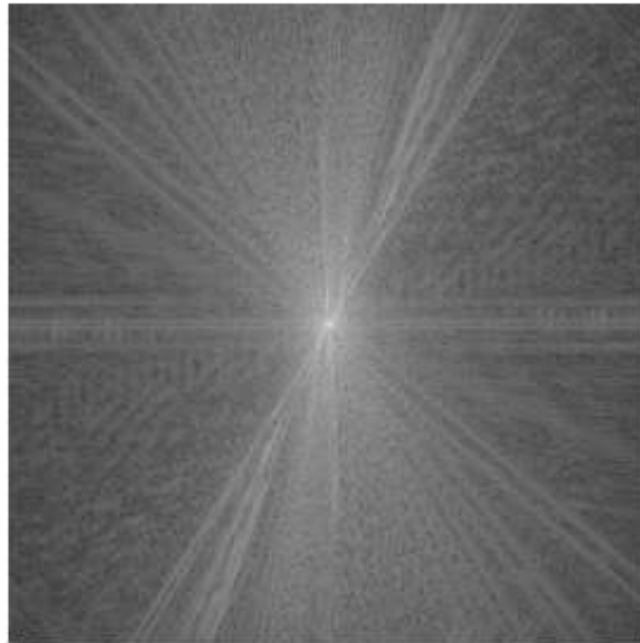
- Изображение есть функция $L(x,y)$
- Необходимо дискретизировать эту функцию на растр.
- Префильтрация невозможна в нашем случае!
 - Остается: суперсэмплинг
 - Остается: неравномерное сэмплирование

Spatial and Frequency Domain

Spatial Domain

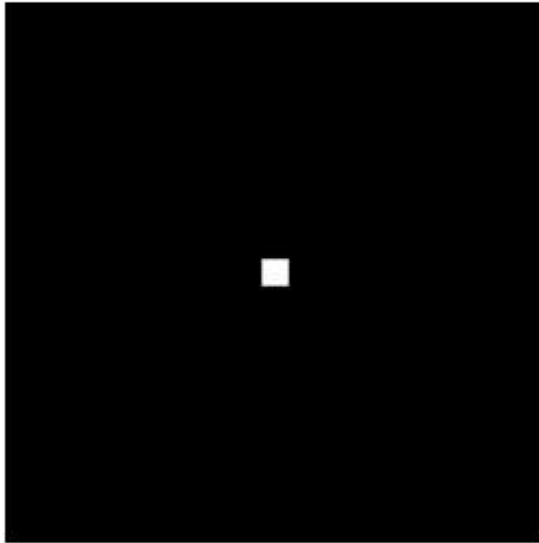


Frequency Domain

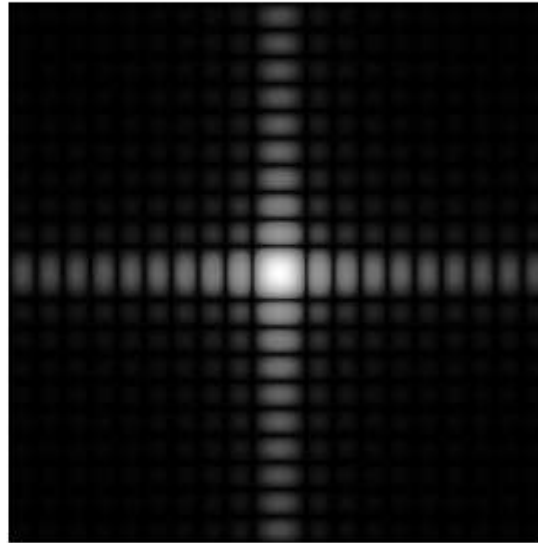


More Examples

Spatial Domain

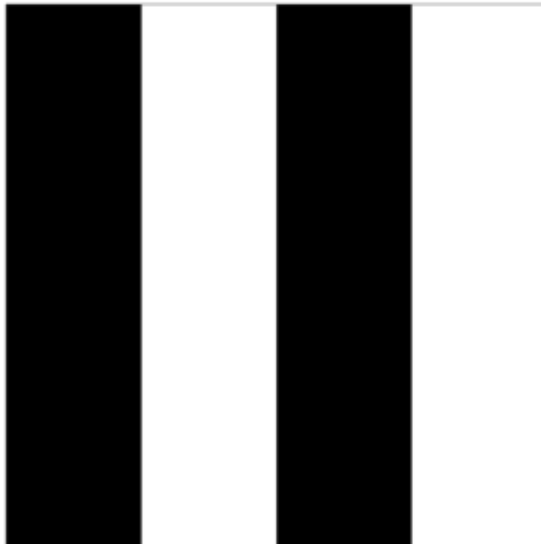


Frequency Domain



More Examples

Spatial Domain

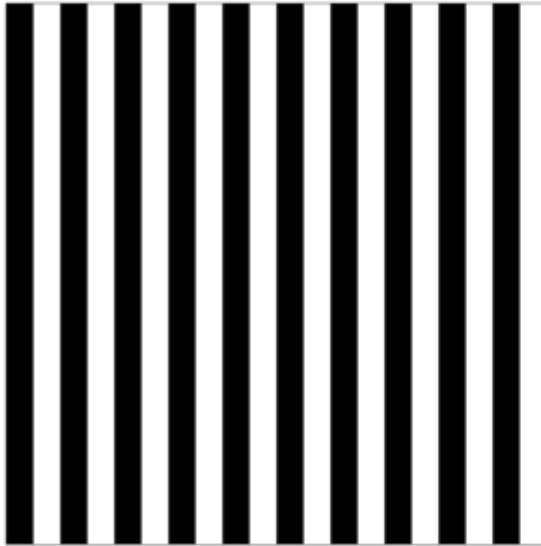


Frequency Domain



More Examples

Spatial Domain

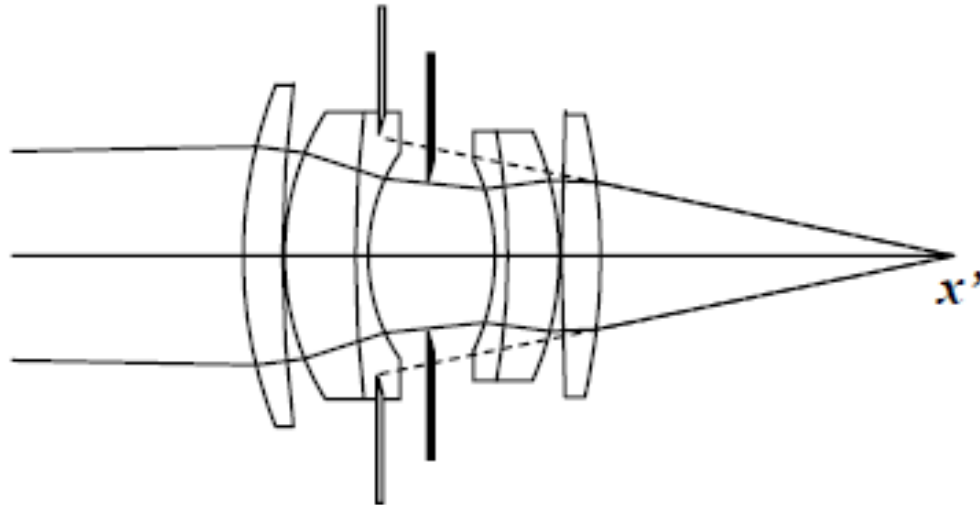


Frequency Domain



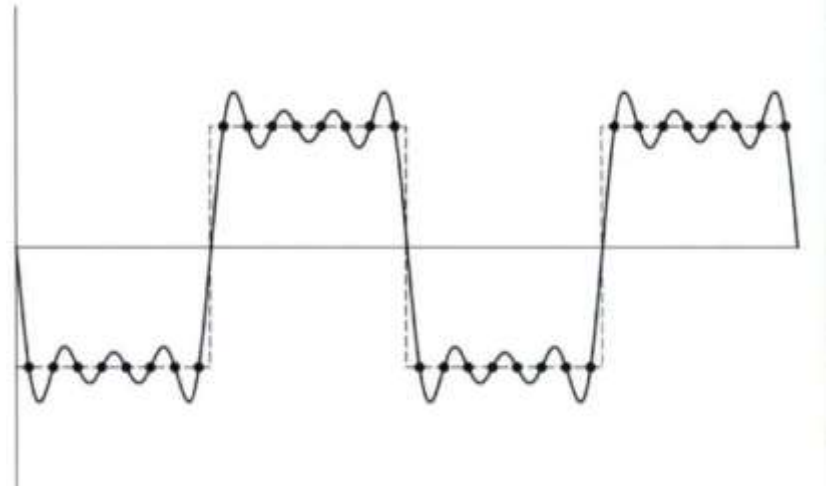
Обобщение функции сцены

- (x, y) – координаты на плоскости проекции (картинной плоскости)
- (u, v) – координаты на выходном зрачке (или на диафрагме)
- t - время

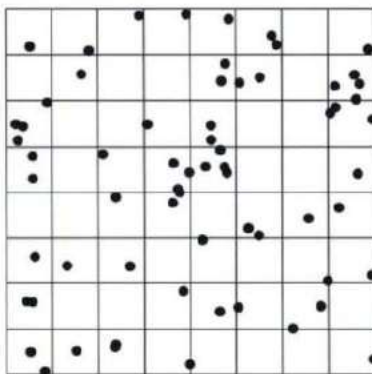


Причины алиасинга в синтезе изображений

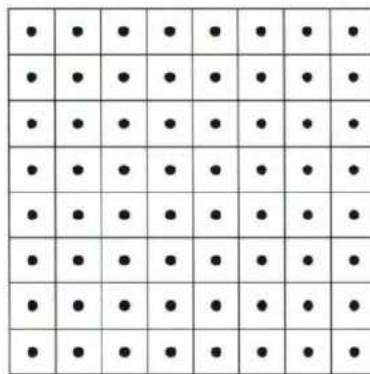
- Геометрия.
 - Границы объектов создают границы на изображении (бесконечные частоты)
 - При реконструкции sinc появляется эффект Гиббса.
- Маленькие объекты
- Текстуры и материалы
 - Тени, блики и т.п.



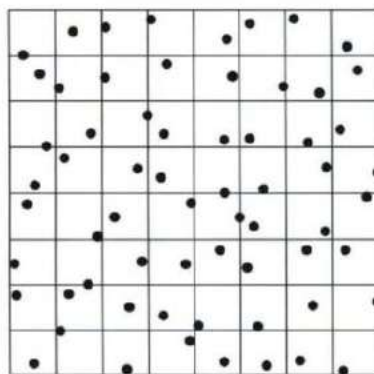
Стратифицированный сэмплинг



(a)



(b)



(c)

Изображение
для сравнения
(256 лучей на
пиксель)



(a)

1 луч на
пиксель, без
джиттеринга



(b)

1 луч на
пиксель,
джиттеринг



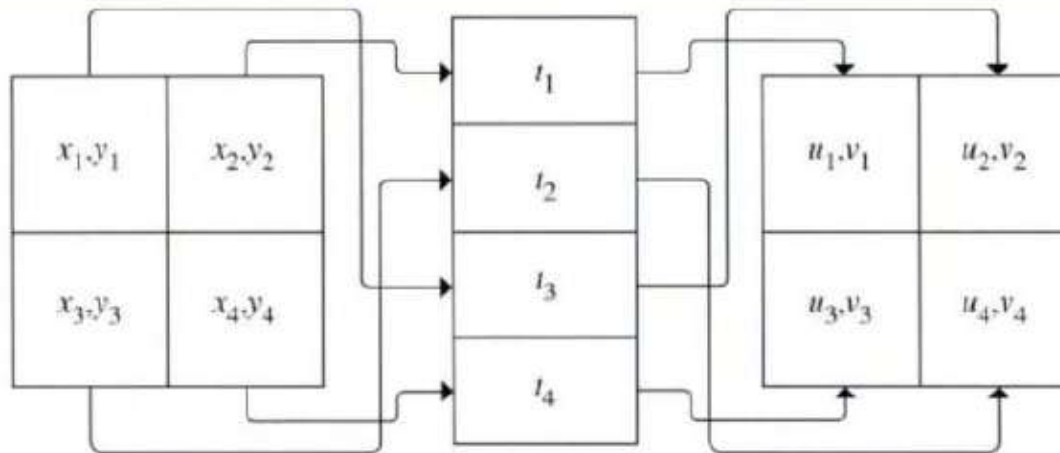
(c)

4 луча на
пиксель,
джиттеринг



Стратификация в многомерном случае

- Пятимерная функция (картинка, линза, время) на четыре страты.
- Уже 1024 сэмпла!
- Выход: генерировать страты по каждой размерности отдельно, затем случайным образом соединять



Пример: разфокусированная сфера



Образец



Случайное
сэмплирование
без
стратификации



Случайное
сэмплирование
+ стратификация
(в том числе по
линзе)

Другие методы сэмплирования

Существует множество более сложных стратегий сэмплирования

- Low-discrepancy sampling
- Best-candidate sampling
- ...

Реконструкция изображения

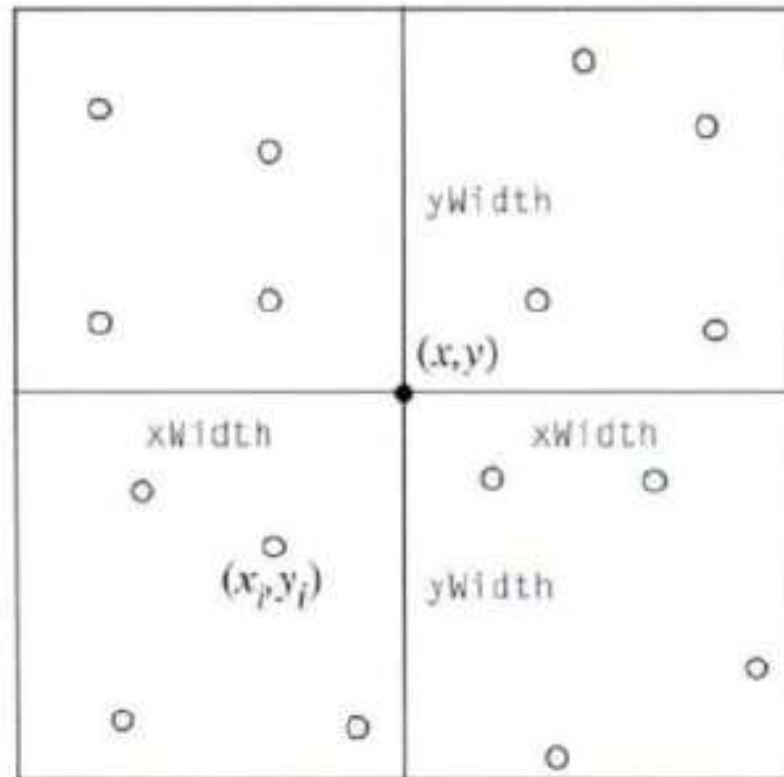
Этапы

- Построить непрерывную функцию L из набора сэмплов
- Префильтровать функцию с учетом лимита для пикселей
- Сэмплировать в положениях пикселя

Мы будем сэмплировать только в центрах пикселей, поэтому этап 1 можно пропустить и получить один фильтр на этапы 1+2.

Получение значения пикселя по набору сэмплов

$$I(x, y) = \frac{\sum_i f(x - x_i, y - y_i) L(x_i, y_i)}{\sum_i f(x - x_i, y - y_i)}$$



Как выбрать фильтр?

- Sinc не подходит (бесконечно широкий + создает эффект Гиббса)
- Фильтр может неожиданным образом взаимодействовать с алгоритмов получения сэмплов
- Сложный выбор, на стыке науки и искусства 😊

Примеры работы фильтров



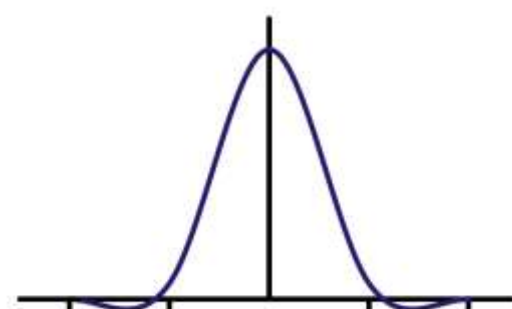
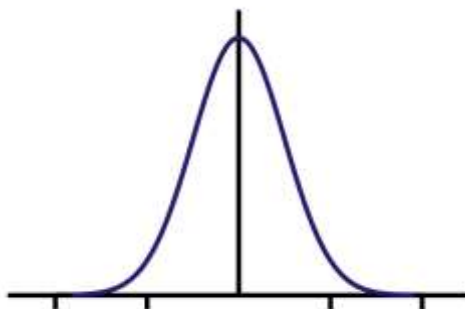
Вох-фильтр



Гауссиан



Митчелл



Ссылки

- http://graphics.stanford.edu/courses/cs148-10-summer/docs/14a_sampling.pdf
- <http://www.siggraph.org/education/materials/HyperGraph/aliasing/alias0.htm>
- <https://graphics.stanford.edu/wikis/cs348b-09/Lectures#Sampling>