

**Дискретизация.  
Антиалиасинг.  
Геометрические  
преобразования растровых  
изображений**

# Содержание

- Дискретизация
- Частотная область
- Гребенчатый фильтр
- Теорема Найквиста-Котельникова
- Дискретизация аналогового сигнала
- Ряд Котельникова и Ряд Уиттакера
- Теорема (Найквиста-Котельникова)
- Алиасинг
- Префильтрация
- Алгоритм растеризации Гупты-Спрулла

# Дискретизация

Этот раздел посвящен проблемам представления непрерывного двумерного цветового сигнала, которым является изображение, на дискретной растровой сетке.

Процесс получения дискретной аппроксимации непрерывного сигнала называется **дискретизацией** (англ. sampling).

Изображение в компьютере хранится в дискретном виде.

Для получения такого дискретного представления из непрерывных аналоговых изображений реального мира (как мы их видим глазами или через оптические устройства) и применяется *дискретизация*.

*Дисплей* фактически осуществляет реконструкцию аналогового изображения по его дискретизированному представлению.

# Обработка сигналов

Наиболее корректно рассматривать возникающие проблемы в рамках теории **обработки сигналов** (англ. *signal processing*).

Двумерное изображение можно представлять как двумерный сигнал, рассматривая его как *отображение*

$$I: D \rightarrow C$$

, где  $C$  - *атрибут* изображения, - интенсивность ( $C$  - *отрезок*) или цвет; в последнем случае чаще всего  $C$  - RGB-куб

В аналоговой форме этот сигнал непрерывный (определенный на прямоугольнике  $D$ ), а в дискретной - определенный лишь на точках раstra  $D$

Теория обработки сигналов рассматривает периодические сигналы, определенные на всем континууме .

Поэтому для рассмотрения изображений, определенных на прямоугольнике, поступают двояко: либо трактуют как сигнал, равный 0 везде кроме данного прямоугольника, либо считают длину и ширину периодами и производят "замощение" всей плоскости сдвинутыми на периоды копиями изображения.

# Частотная область

Сигналы рассматривают как **в пространственной области** (англ. *spatial domain*) - это обычная область определения, так и **в частотной области** (англ. *frequency domain*), которая получается как область определения коэффициентов разложения сигнала по тригонометрической системе Фурье.

Частотная область представляет из себя .

Частотное представление  $F(f)$  для одномерной функции  $I(x)$  рассчитывается при помощи преобразования Фурье

$$F(f) = \int_{-\infty}^{\infty} I(x)[\cos 2\pi f x - i \sin 2\pi f x] dx. \quad (7.1)$$

Если представить  $F(f)$  в виде  $F(f) = Re(f) + iIm(f)$ , то *модуль*, или амплитуда, - это  $\sqrt{Re(f)^2 + Im(f)^2}$ , а фазовый сдвиг, или фазовый угол, -  $\arctan\left[\frac{Im(f)}{Re(f)}\right]$ . На графиках далее будут представлены только более наглядные значения амплитуды. В многомерном случае *произведение*  $f \cdot x$  заменяется на *скалярное произведение* векторов  $f$  и  $x$  ( $(f, x) \stackrel{def}{=} \sum_{i=1}^n x_i f_i$ ). Для интересующего нас двумерного случая преобразование Фурье запишется как

$$F(f_x, f_y) = \int_{\mathbb{R}^2} I(x, y)[\cos(2\pi(f_x x + f_y y)) - i \sin(2\pi(f_x x + f_y y))] dx dy. \quad (7.2)$$

Далее для упрощения рассуждения будут проводиться в одномерном случае, впрочем, их несложно обобщить на общий  $n$ -мерный случай. Обрато к пространственной области можно перейти с помощью обратного преобразования Фурье:

$$I(x) = \int_{-\infty}^{\infty} F(f)[\cos 2\pi f x + i \sin 2\pi f x] df. \quad (7.3)$$

В дискретном случае, если сигнал представлен в виде функции, определенной на  $N$  точках  $x \in \overline{0, N-1}$ , используется дискретное преобразование Фурье (ДПФ), которое по сути является дискретизацией непрерывного преобразования Фурье, примененного к дискретному сигналу:

$$F(f) = \sum_{x=0}^{N-1} I(x) \left[ \cos \frac{2\pi f x}{N} - i \sin \frac{2\pi f x}{N} \right], \quad (7.4)$$

# Гребенчатый фильтр

Обычно *дискретизация* происходит путем измерения сигнала (взятия значения функции) через равные промежутки в области определения. Эта операция математически описывается как *умножение* функции  $I(x)$  на гребенчатый фильтр, состоящий из последовательности равномерно сдвинутых функций Дирака  $\delta(x)$

$$\text{Comb}(x) = \sum_{n=-\infty}^{+\infty} \sigma(x - nT),$$

где  $T = 1/f_s$  - период, а  $f_s$  - частота дискретизации  
Интересным свойством функции  $\text{Comb}$  является то, что ее преобразование Фурье также есть *функция*  $\text{Comb}$ , но с другими амплитудой (не 1, а  $f_s$ ) и частотой (также  $f_s$ ).

# Теорема Найквиста-Котельникова

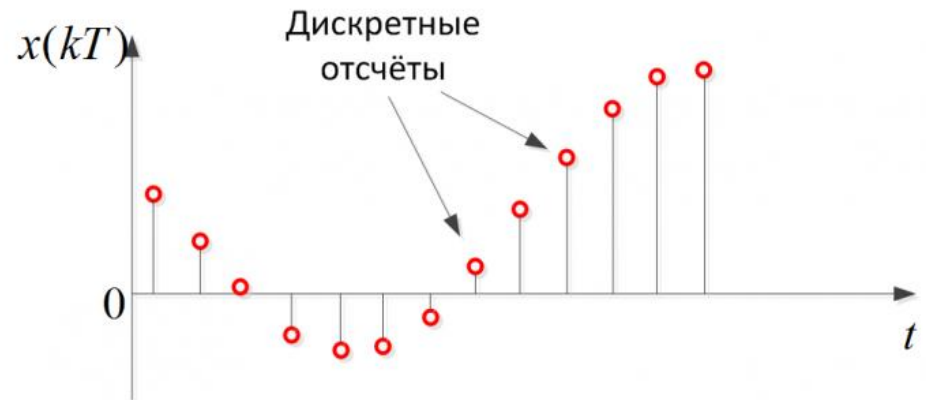
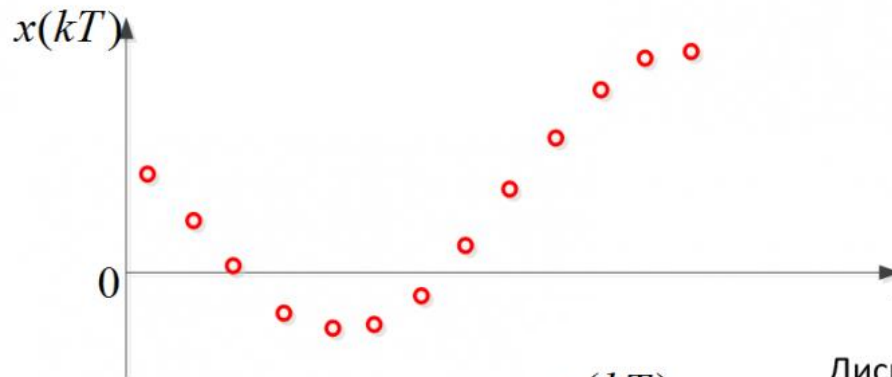
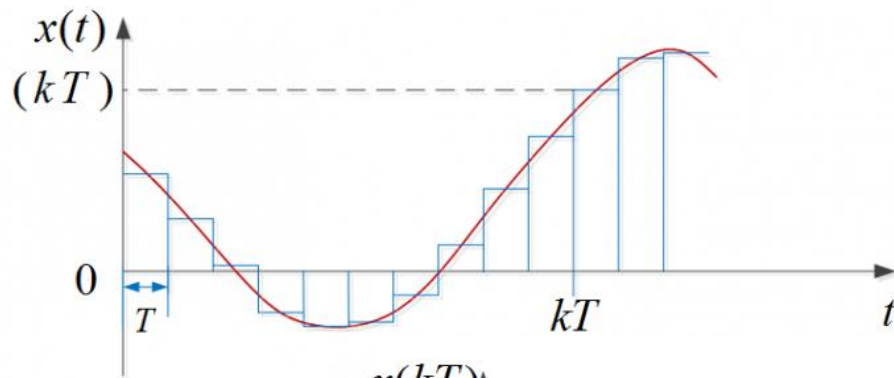
*Теорема Найквиста-Котельникова* дает ответ на вопрос, какой частоты дискретизации  $f_s$  достаточно для того, чтобы не произошло потери информации, т.е. чтобы по дискретизованному сигналу можно было восстановить исходный.

Применительно к изображениям "Какая разрешающая способность должна быть у раstra, чтобы он сохранил все детали исходного аналогового изображения".

Хотя потеря информации даже в случае соблюдения условий теоремы Котельникова произойдет из-за того, что значения дискретизованной функции (растрового изображения) в компьютере сами хранятся с ограниченной точностью.



# Дискретизация аналогового сигнала



Дискретизация аналогового сигнала есть преобразование функции непрерывной переменной  $x(t)$  в последовательность дискретных значений  $x(kT)$ ,  $k=1,2,3\dots$

Дискретный сигнал можно отображать разными способами, например, в виде дискретной последовательности импульсов, имеющих на интервалах  $T$  значения  $x(kT)$ , или в виде точечных значений сигнала

# Ряд Котельникова

Члены ряда Котельникова представляют собой функции отсчётов, сдвинутых друг относительно друга по времени на величину  $T$ , умноженных на величину дискретного значения сигнала  $x(kt)$ .

Суммирование конечного числа членов ряда позволяет получить сигнал, который будет приближаться к аналоговому сигналу

# Ряд Котельникова и Ряд Уиттакера

**Теорема 1.** Любую функцию  $s(t)$ , состоящую из частот от 0 до  $f_B$  периодов в секунду, можно представить рядом

$$s(t) = \sum_{k=-\infty}^{\infty} s(kT) \frac{\sin 2\pi f_B(t - kT)}{2\pi f_B(t - kT)}$$

где  $k$  – целые числа;  $s(kT)$  – дискретные значения функции,  $f_B$  – верхняя частота спектра.

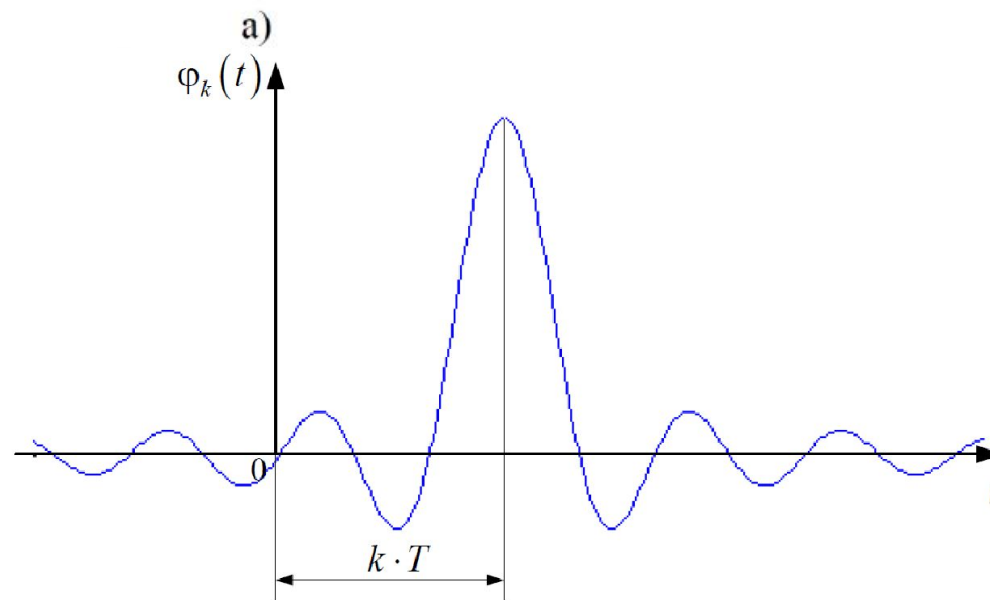
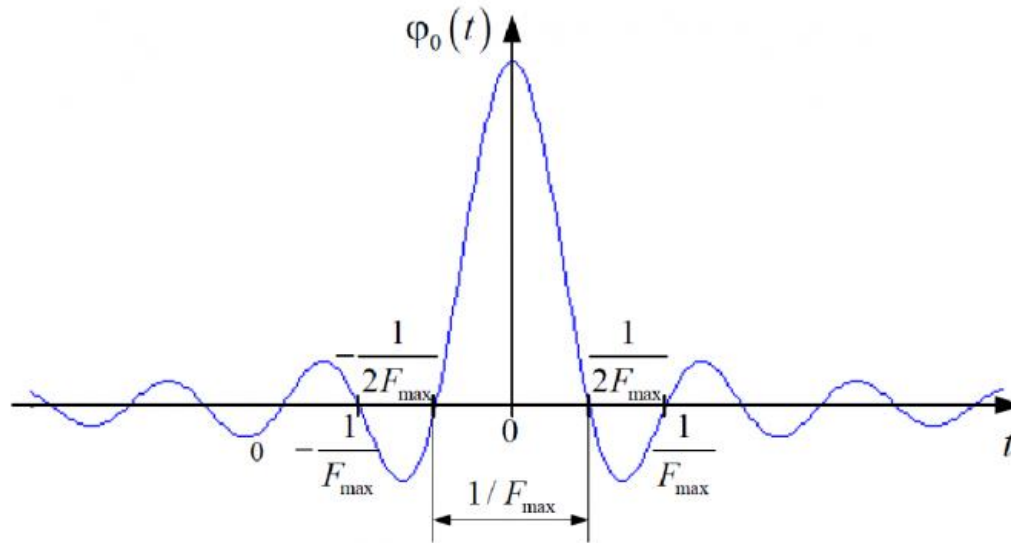
$$x(t) = \sum_{k=-\infty}^{\infty} x(a + kT) \frac{\sin(\omega_{\max}(t - a - kT))}{\omega_{\max}(t - a - kT)},$$

$$T = \frac{\pi}{\omega_{\max}}$$

# Теорема отсчетов Котельникова

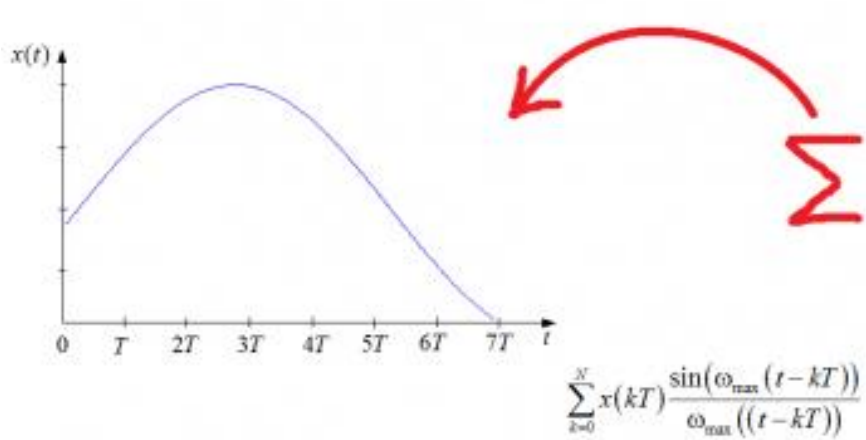
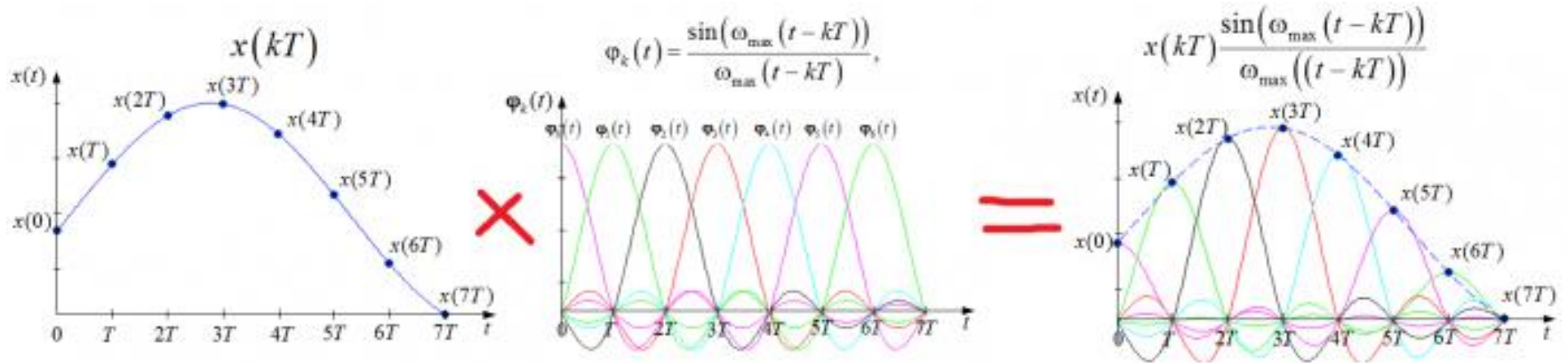
Теорема 2. (*Теорема отсчетов*). Любую функцию (сигнал)  $s(t)$ , состоящую из частот от 0 до  $f_B$  периодов в секунду, можно непрерывно передавать с любой точностью с помощью чисел, следующих друг за другом через интервалы времени  $T = \frac{1}{2f_B}$  секунд.

# Функции отсчётов

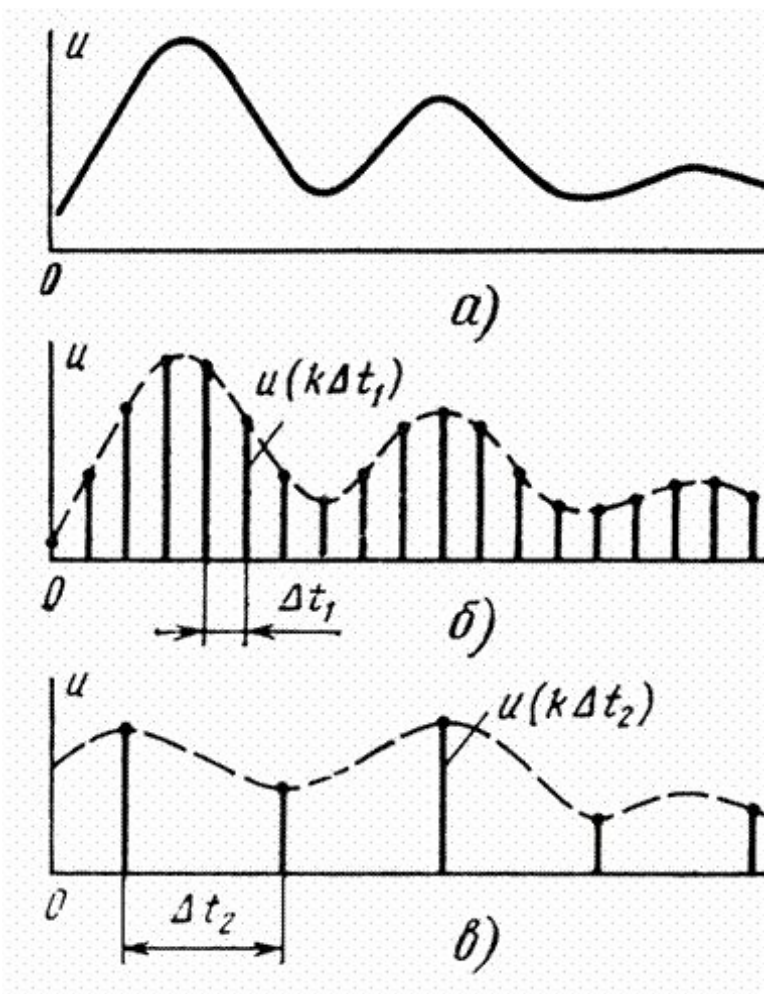


б)

# Восстановление сигнала с помощью ряда Уиттакера-Котельникова



# Величина шага отсчета



При малом шаге последовательность отсчетов достаточно точно описывает сигнал, а при большом шаге по отсчетам нельзя восстановить форму сигнала, так как пропущены его характерные экстремальные точки.

Как же часто следует брать отсчеты, чтобы по ним можно было полностью восстановить сигнал?

Ответ на этот вопрос дает теорема, доказанная в 1933 г. Советским ученым академиком [В.А. Котельниковым](#) и названная его именем.

Согласно этой теореме любой непрерывный сигнал с конечным спектром (имеющим максимальное значение) можно представить в виде дискретных отсчетов, частота дискретизации которых должна быть выбрана не менее чем в два раза выше максимального значения спектра сигнала; передать его по линии связи, а затем восстановить исходный аналоговый сигнал.

# Физический смысл теоремы Котельникова.

Теорема Котельникова утверждает, что если требуется передать непрерывный сигнал с ограниченным спектром по каналу связи, то можно не передавать все его значения: **достаточно лишь передать его мгновенные значения (отсчеты) через интервал** .

Поскольку сигнал полностью определяется этими значениями, то по ним он может быть восстановлен на приемном конце системы связи.

Для этого достаточно **соединить отсчеты плавной кривой**. Это можно объяснить тем, что сигнал между отсчетами может изменяться только плавно, так как частоты выше дающие быстрые изменения, в сигнале отсутствуют.

Отсчеты берутся достаточно часто, и тем чаще, чем выше максимальная частота .

## Пример

Пусть есть речевой сигнал. Он содержит частоты от 300 Герц до 3 килогерц. Для его передачи по цифровому каналу связи вам потребуется "выхватывать" мгновенные значения этого сигнала с частотой, в два раза превышающей самую высокочастотную компоненту сигнала, т. е. 6 кГц. Иными словами, шесть тысяч раз в секунду вы будете измерять значение сигнала, который изменяется не чаще трех тысяч раз в секунду.



# Теорема (Найквиста-Котельникова)

В доказательстве теоремы и далее будет использоваться операция **свертки** функций  $I(x)$ ,  $J(x)$ , определяемая так:

$$(I * J)(x) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} I(u) \cdot J(x - u) du.$$

**Теорема (Найквиста-Котельникова).** Для того чтобы сигнал  $I(x)$  можно было восстановить по его дискретному образу, его спектр должен быть ограничен максимальной частотой  $f_H$  и частота дискретизации  $f_s$  должна быть более  $2f_H$ .

Пусть  $I_s(x)$  – дискретный образ исходного сигнала  $I(x)$ , как обычно,  $T = 1/f_s$  – период дискретизации, тогда

$$I_s(x) = I(x) \cdot \text{Comb}(x) = I(x) \sum_{n=-\infty}^{+\infty} \sigma(x - nT) = \sum_{n=-\infty}^{+\infty} I(nT) \cdot \sigma(x - nT).$$

Образом функции  $\text{Comb}$  в частотной области является функция  $F\text{Comb}(f) = f_s \sum_{k=-\infty}^{+\infty} \sigma(f - kf_s)$ , а Фурье-образ  $I(x)$  по-прежнему будем обозначать  $F(f)$ .

Умножение функций в пространственной области соответствует их свертке (будем обозначать ее  $*$ ) в частотной и наоборот.

# Теорема

Соответственно, рассмотрим свертку  $F$  и  $FComb$ , являющуюся Фурье-образом  $I_s(x)$  (обозначим его  $F_s(f)$ ):

$$F_s(f) = F * FComb(f) = F(f) * f_s \sum_{k=-\infty}^{+\infty} \sigma(f - kf_s) \stackrel{(1)}{=}$$

$$(*) \quad f_s \sum_{k=-\infty}^{+\infty} F(f - kf_s),$$

где переход (1) произошел благодаря сдвигающему свойству дельта-функции при свертке. Как видно из последнего выражения,  $F_s(f)$  представляет собой бесконечную сумму функций  $F(f)$ , умноженных на  $f_s$  и сдвинутых на  $f_s$  относительно друг друга, поэтому при условии  $f_s > 2f_H$  носители соседних сдвинутых версий не пересекаются, и отдельно, взяв центральную копию  $F(f)$  ( $k = 0$ ) и применив к ней обратное преобразование Фурье, можно получить исходный сигнал  $I(x)$ . Центральная копия берется путем умножения  $F_s(f)$  на прямоугольную функцию

$$\text{где, } T \cdot \text{Rect}(T \cdot f) \\ \text{Rect}(f) = \begin{cases} 1, & |f| \leq 1 \\ 0, & |f| > 1 \end{cases}$$

Т.е.  $F(f) = F_s(f) \cdot T \cdot \text{Rect}(T \cdot f)$  - образ исходной функции получен. Заметим, этому умножению в частотной области соответствует свертка в пространственной области.

# Интерполяционная формула Найквиста-Шеннона

Применив обратное преобразование Фурье к  $T \cdot \text{Rect}(T \cdot f)$  получим функцию  $\text{sinc}(\pi x/T) \stackrel{\text{def}}{=} \frac{\sin \pi x/T}{\pi x/T}$ .

Применив свертку с  $I_s(x)$ , получаем

$$\begin{aligned} I(x) &= I_s(x) * \text{sinc}(\pi x/T) \\ &= \sum_{n=-\infty}^{+\infty} (I(nT) \cdot \sigma(x - nT)) * \text{sinc}((x - nT)/T) \\ &\stackrel{(2)}{=} \sum_{n=-\infty}^{+\infty} I(nT) \cdot \text{sinc}((x - nT)/T), \end{aligned}$$

где переход (2) также произошел благодаря сдвигавшему свойству дельта-функции при свертке. Последняя формула называется **Интерполяционной формулой Найквиста-Шеннона**

# Пример

Для завершения доказательства осталось показать, что невозможно однозначно восстановить сигнал при  $f_s \leq 2f_H$ .

Приведем соответствующий пример.

Зафиксируем две частоты -  $f_s$  и  $f_H \leq 2f_H$ ; для упрощения рассуждений предположим, что  $f_s > f_H$  (в общем случае может быть более двух наложений сдвинутых образов, что усложнит построение контрпримера).

Из формулы (\*) следует, что Фурье-образ  $I_s(x)$  является периодической функцией с периодом  $f_s$ , поэтому вся информация для восстановления содержится в одном периоде (например  $f \in [-\frac{f_s}{2}, \frac{f_s}{2}]$ ).

# Пример не однозначности образа

Рассмотрим две функции, Фурье-образы которых равны

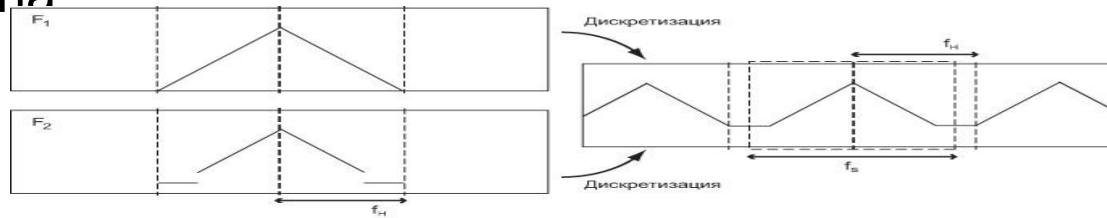
$$F_1(f) = \begin{cases} f_H - |f|, & |f| \in [0, f_s - f_H) \\ \frac{2f_H - f_s}{2}, & |f| \in [f_s - f_H, f_H] \\ 0, & |f| > f_H \end{cases}$$

$$F_2(f) = \begin{cases} f_H - |f|, & |f| \in [0, f_H] \\ 0, & |f| > f_H \end{cases}$$

(функция однозначно задается своим Фурье-образом). При дискретизации с частотой  $f_s$  в соответствии с формулой (\*) Фурье-образ в интервале  $[-\frac{f_s}{2}, \frac{f_s}{2}]$  для обеих функций будет равен

$$F_s(f) = \begin{cases} f_s(f_H - |f|), & |f| \in [0, f_s - f_H) \\ f_s(2f_H - f_s), & |f| \in [f_s - f_H, f_s/2] \end{cases}$$

Таким образом, в этом случае однозначная реконструкция невозможна



# Искажение сигнала и борьба с этим эффектом

Как было показано при доказательстве *теоремы Найквиста-Котельникова*, при недостаточной частоте дискретизации восстановленный сигнал будет искажен.

Это наглядно можно наблюдать в частотной области ([см. рис.](#)), т.к. при этом копии частотного спектра исходного сигнала будут суммироваться в пересекающихся областях, что даст мнимое увеличение веса компонент с этими частотами в спектре, некую подмену высокочастотных компонент низкочастотными, эффект, известный как **алиасинг** (англ. aliasing).

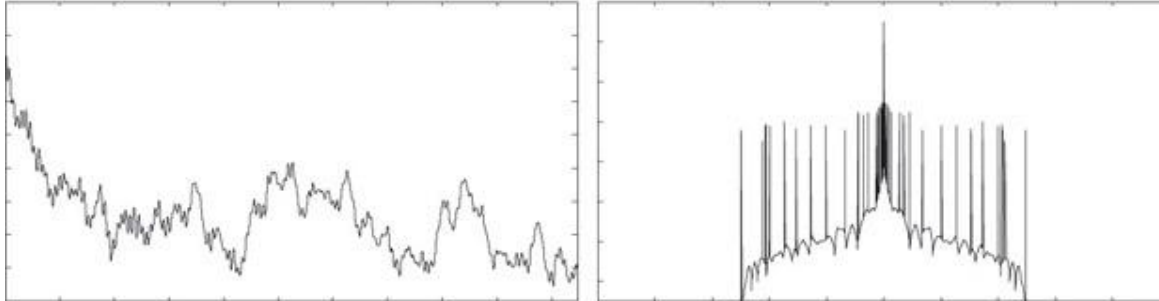
Особенно наглядно этот эффект можно наблюдать на резких контрастных изменениях яркости

# Алиасинг

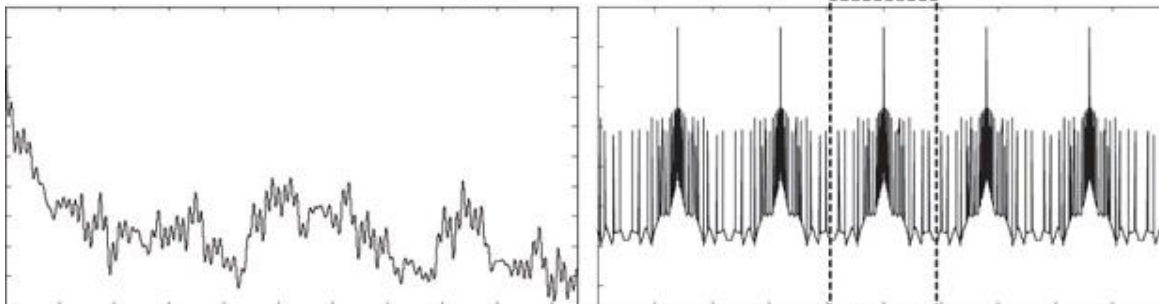
Пространственная область

Частотная область  
(логарифмический масштаб)

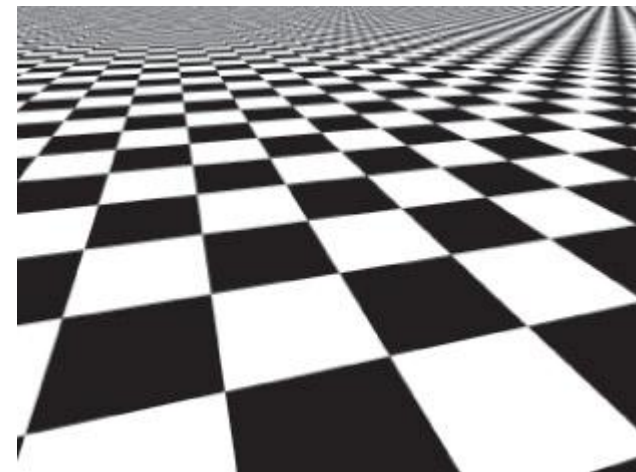
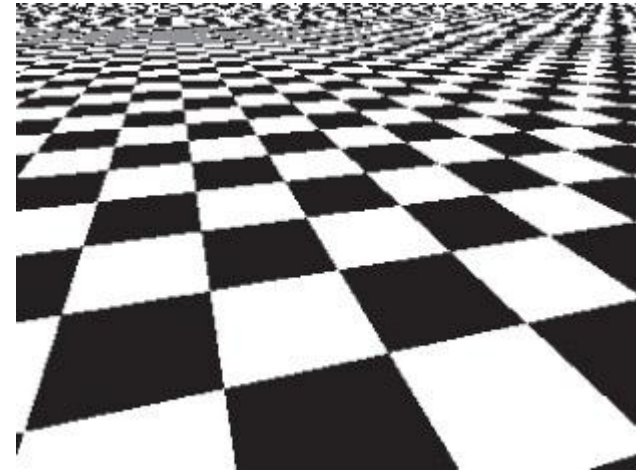
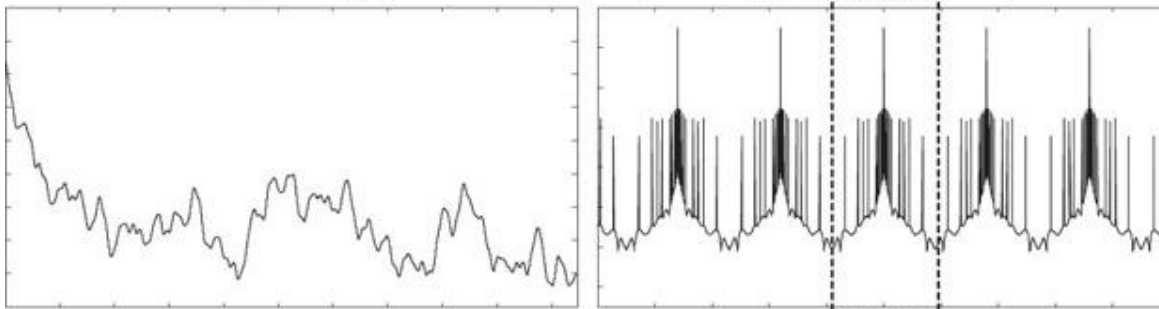
Исходный сигнал



Сигнал, реконструированный по дискретизированному



Сигнал, реконструированный по префильтрованному дискретизированному



# Префильтрация

Для борьбы с подобными явлениями применяют префильтрацию - свертку с некой функцией фильтра что эквивалентно умножению на Фурье-образ функции-фильтра в частотной области, перед тем как производится дискретизация.

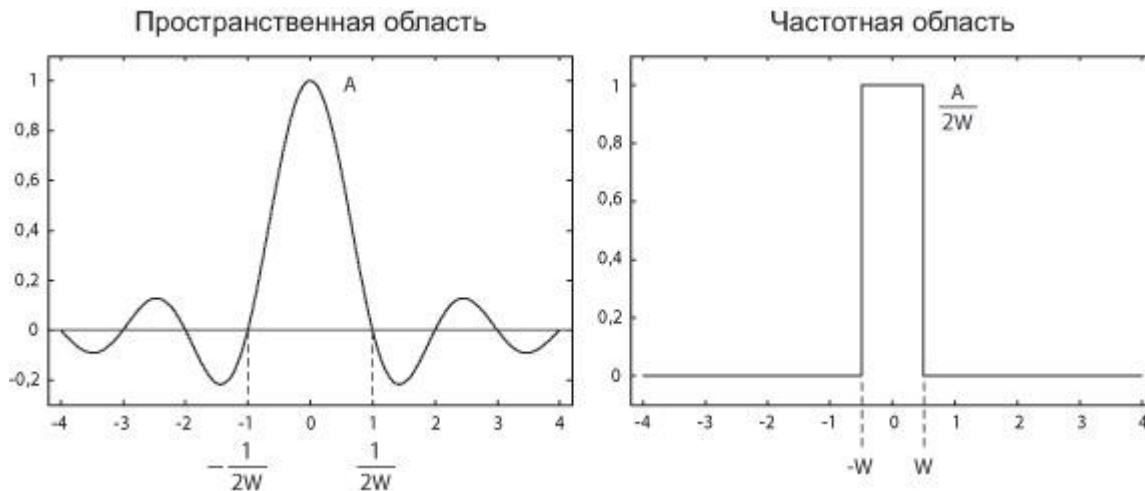
Цель префильтрации - заранее отсечь высокочастотные компоненты, которые могут привести к алиасингу.

В идеале, для этого следовало бы применять функцию sinc,

$$\text{sinc}(\pi x/T) \stackrel{\text{def}}{=} \frac{\sin \pi x/T}{\pi x/T}.$$

действие которой как раз и состоит в отсечении высоких частот, но она имеет бесконечный носитель<sup>1</sup>, что затрудняет ее применение в пространственной области.

На практике используются различные аппроксимации с ограниченным носителем





# Префильтрация

В двумерном случае фильтрация описывается следующим образом: интенсивность пикселя  $I(x, y)$  с центром в точке  $(x, y)$  будет определяться формулой двумерной свертки

$$(1) \quad I(x, y) = \int_{\text{supp}(F)} F(s, t) \text{Img}(x - s, y - t) ds dt,$$

где  $F(s, t)$  - функция фильтра с центром в 0,  $\text{supp}(F)$  - ее носитель (область, где она не равна 0),  $\text{Img}(x, y)$  - непрерывное аналоговое изображение.

# Одномерные функции-фильтры для антиалиасинга

Название	Функция фильтра F(x)
Импульсный ( <i>pulse</i> )	$F_p(x) = \begin{cases} 1, &  x  \leq 1/2 \\ 0, &  x  > 1/2 \end{cases}$
Треугольный ( <i>triangle</i> )	$F_t(x) = \max\{1 -  x , 0\}$
Гауссовский (Gaussian)	$F_{G(\sigma,R)}(x) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}, &  x  \leq R \\ 0, &  x  > R \end{cases}$
Кубический ( <i>cubic</i> )	$F_{c(a)}(x) = \begin{cases} (a + 2) x ^3 - (a + 3) x ^2 + 1, & 0 \leq  x  < 1 \\ a x ^3 - 5a x ^2 + 8a x  - 4a, & 1 \leq  x  < 2 \\ 0, &  x  > 2 \end{cases}$
Ланцоша (Lanzcos)	$F_{L(R)}(x) = \begin{cases} \text{sinc}(x/R) \cdot \text{sinc}(x), & 0 \leq  x  \leq R \\ 0, &  x  > R \end{cases}$

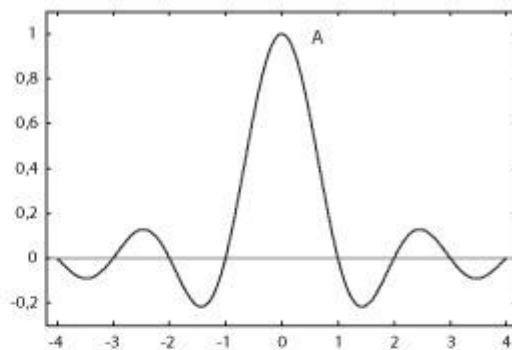
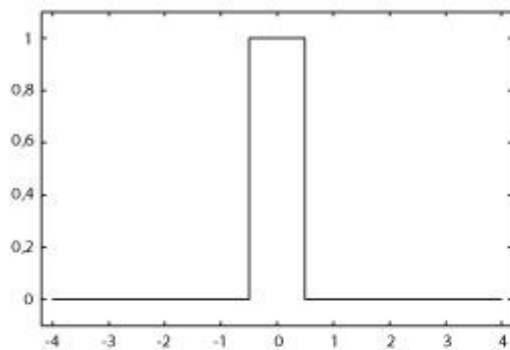
# Комментарий

- Гауссовский фильтр обычно применяется с  $\sigma = 0,5$  (так он лучше всего приближает sinc в частотной области),  $R$  берется порядка 2-3.
- У кубического фильтра присутствует параметр  $a$ , с помощью которого можно регулировать степень размытия (увеличение  $a$ ) или, наоборот, более четкой передачи краев несмотря на некоторый алиасинг (уменьшение  $a$ ), стандартным компромиссным значением является  $a = -1$ .
- Для Lanczos, который представляет собой обрезанный и сглаженный sinc, радиус  $R$  обычно равен 2 (так называемый Lanczos2), реже - до 4. Большие значения не используются ввиду того, что носитель, а значит, и область интегрирования при свертке, будет слишком большой.
- Понятно, что для дискретизации достаточно знать значения префильтрованной функции только в точках дискретизации..

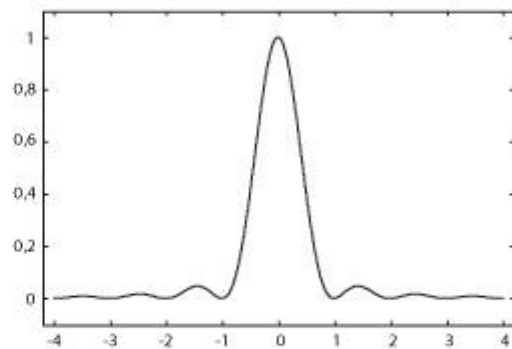
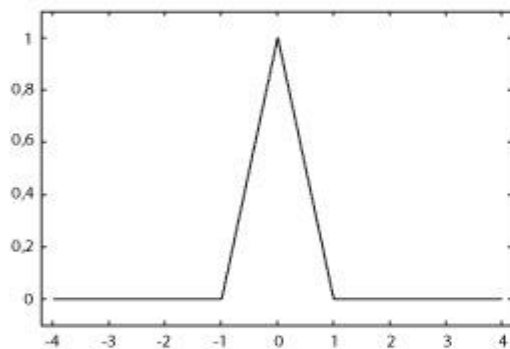
Пространственная область

Частотная область

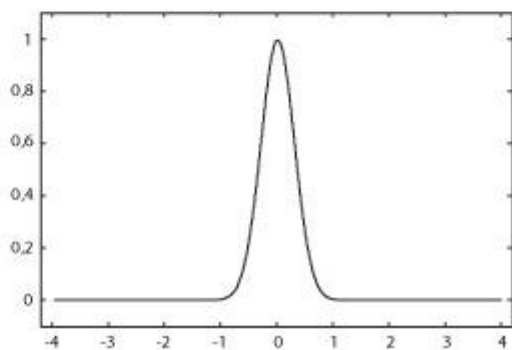
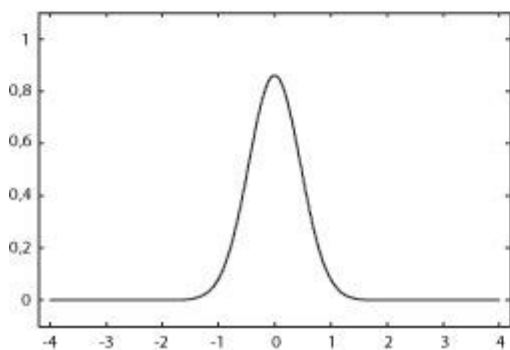
Импульсный фильтр (pulse)



Треугольный фильтр (triangle)



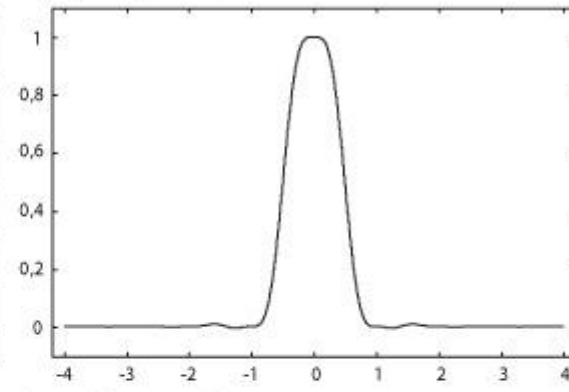
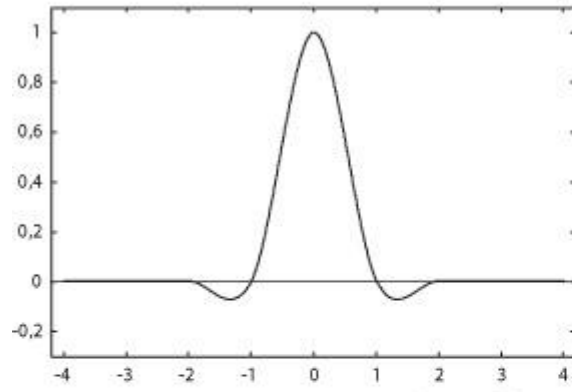
Гауссовский фильтр (Gaussian)



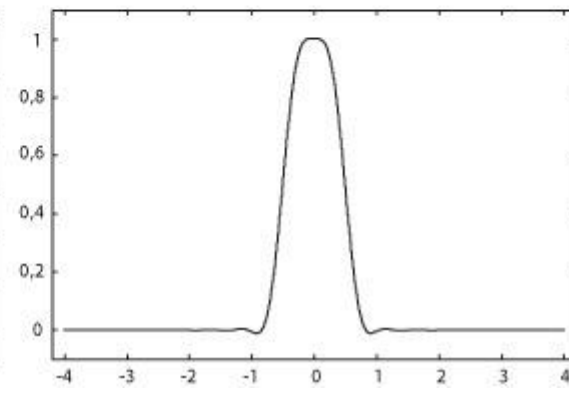
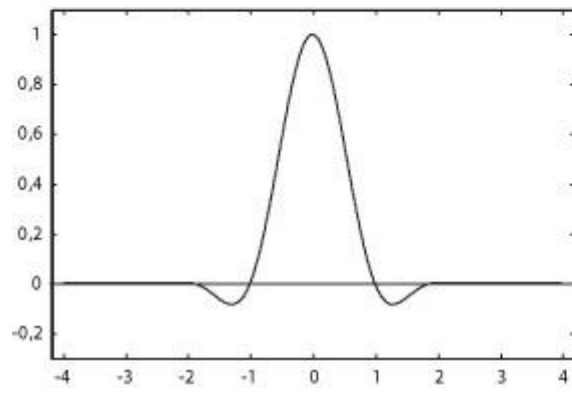
Пространственная область

Частотная область

Кубический фильтр ( $a = -1$ ) (cubic)



Фильтр Ланцоша ( $R = 2$ ) (Lanczos2)



# Двумерные аналоги

Двумерные аналоги одномерного фильтра  $f_1(x)$  строятся двумя путями:

как функция от радиуса:  $f_2(x, y) = f_1(r) = f_1(\sqrt{x^2 + y^2})$

как произведение:

$$f_2(x, y) = f_1(x) \cdot f_1(y)$$

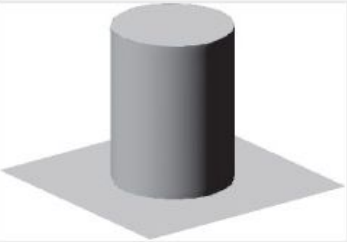
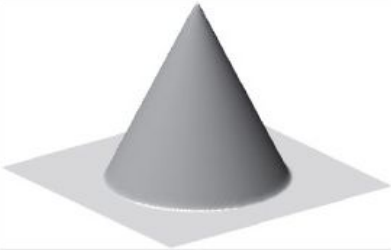
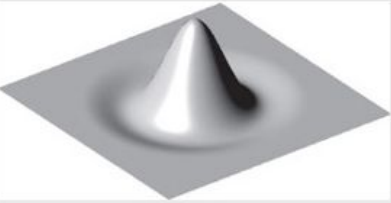
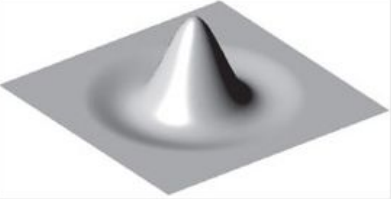
Первый вариант - более корректный, но второй обладает свойством сепарабельности, т. е. в выражении можно двумерное интегрирование разбить на два одномерных:

$$\begin{aligned} I(x, y) &= \int_{\text{supp}(F_1) \times \text{supp}(F_1)} F_1(s) F_1(t) \text{Img}(x - s, y - t) ds dt \\ &= \int_{\text{supp}(F_1)} ds F_1(s) \left( \int_{\text{supp}(F_1)} F_1(t) \text{Img}(x - s, y - t) dt \right) \end{aligned}$$

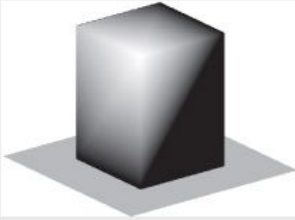

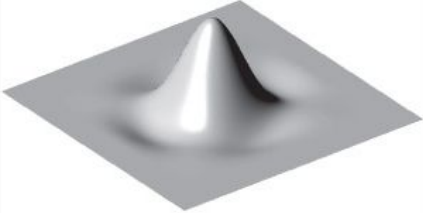
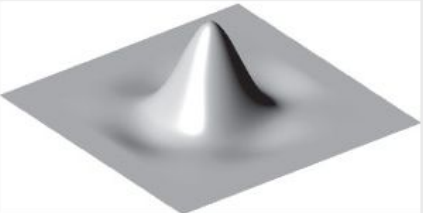
В случае передискретизации интегрирование заменяется суммированием и в этом случае быстрее производить вычисления в два этапа, каждый из которых состоит в дискретной свертке по одному измерению и может быть вычислен достаточно эффективно (в том числе и с применением параллельных алгоритмов).

В алгоритмах *растеризации* с префильтрацией более популярны фильтры первого типа, а для геометрических преобразований, где происходит передискретизация - второго.

# Радиально-симметричные фильтры для антиалиасинга

Название	Функция фильтра $F(s, t) = F_1(r)$ ( $r = \sqrt{s^2 + t^2}$ )	Изображение
Цилиндрический	$F_1(r) = F_p(r)$	
Конусообразный	$F_1(r) = F_t(r)$	
Кубический	$F_1(r) = F_{c(a)}(r)$	
Ланцоша	$F_1(r) = F_{L(R)}(r)$	

# Одномерные функции-фильтры для антиалиасинга

Название	Функция фильтра $F(s, t) = F_1(s) \cdot F_1(t)$	Изображение
Параллелепипед	$F_1(x) = F_p(x)$	
Пирамидальный	$F_1(x) = F_t(x)$	
Кубический	$F_1(x) = F_{c(a)}(x)$	
Ланцоша	$F_1(x) = F_{L(R)}(x)$	



# Комментарий

Двумерные аналоги фильтров, приведены в (для простоты частота дискретизации равна 1, произвольный случай получается масштабированием по осям).

Прямоугольный фильтр соответствует либо цилиндру (первый тип), либо параллелепипеду с квадратным основанием (англ. **Box filter**) (второй тип);

треугольный соответствует либо конусу, либо пирамиде соответственно.

По смыслу прямоугольные и треугольные аналоги соответствуют так называемым **невзвешенной** и **взвешенной площадным выборкам**.

Если рассмотреть Фурье-образы их одномерных аналогов на , можно заметить, что **взвешенная площадная выборка лучше** отфильтровывает высокие частоты, поэтому рекомендуется применять именно ее.

# Антиалиасинг или Фильтрация-сглаживание

(англ. *antialiasing*) – устранение видимых артефактов дискретизации, особенно для изображений линий или краев объектов. Возможны два основных подхода.

- Учет эффектов дискретизации в алгоритмах построения изображений. То есть, фактически, встраивание в них префильтрации.
- Адаптивная дискретизация с постфильтрацией.

Получение нового дискретизированного изображения по другому дискретизированному с помощью дискретной фильтрации, т. н. **постфильтрации**.

Данный подход может быть эффективен при переходе от промежуточной большей частоты дискретизации к меньшей.

Процесс *растеризации* с увеличенной по отношению к требуемой частотой первичной дискретизации с последующей постфильтрацией называется **супердискретизацией** (англ. *supersampling*).

Первичная частота дискретизации может варьироваться в разных частях изображения в зависимости от его свойств.

# Растреризация с антиалиасингом

При *растреризации* абстрактно определенных примитивов (точек, линий, многоугольников) можно рассматривать их как двумерный сигнал и применить к нему префильтрацию как описано в предыдущем разделе.

В качестве одной из моделей понятия растра рассматривалось замощение плоскости, где каждый пиксель представляет собой квадрат. При *растреризации* этому квадрату нужно приписать цвет так, чтобы при взгляде на изображение растра на каком-либо устройстве вывода воспринимаемая картинка была наиболее похожа на исходный непрерывный двумерный сигнал.

Заметим, что выражение (1) линейно по  $Img$  и, следовательно, обладает свойством суперпозиции по отношению к  $Img$ .

Т.е. если  $Img$  состоит из нескольких непересекающихся объектов, то в (1) достаточно будет просуммировать вклад от этих объектов.

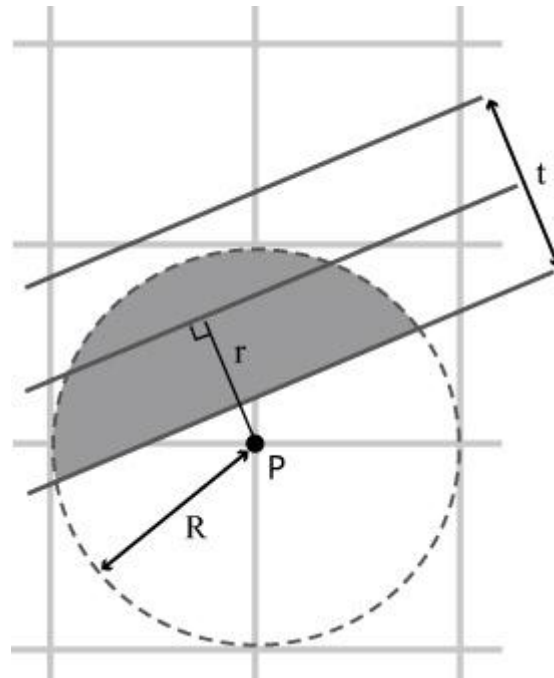
Понятно, что с увеличением радиуса фильтра изображение будет все более сглаженным и даже размытым, а с уменьшением радиуса уменьшится его эффект для антиалиасинга.

# Алгоритм Гупты-Спрулла

Данный алгоритм для *растеризации* отрезков с целочисленными координатами концов был предложен Гуптой и Спруллом .

В нем предполагается использование радиально-симметричного фильтра (такого как конический).

Пусть его радиус равен  $R$ . Для того чтобы выяснить, каким цветом следует закрасить пиксель, необходимо посчитать значение свертки фильтра с функцией прямой (  $1$  в вытянутом прямоугольнике толщиной  $t$  и  $0$  вне него), см. рис.



# Алгоритм Гупты-Спрулла

Обозначим результат этой операции  $F_{R,t}(r)$ , где  $r$  - расстояние от центра закрашиваемого пикселя до центральной оси прямой.

Заметим, что  $F_{R,t}(r) = 0$  для  $r > R + t/2$ .

Идея алгоритма состоит в модификации алгоритма Брезенхема для одновременной закрашки нескольких пикселей разной интенсивности в столбце.

Так же, как и в алгоритме Брезенхема, речь идет о случае наклона прямой меньше чем на  $45^\circ$ .

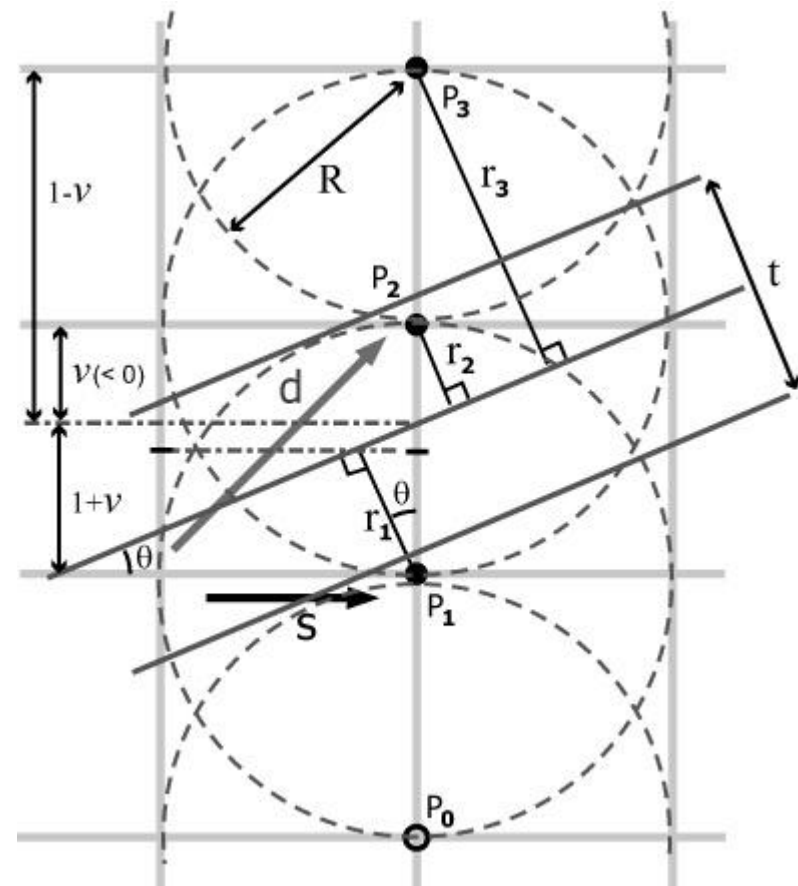
Случай вертикального наклона аналогично просто сводится к первому случаю.

# Алгоритм Гупты-Спрулла

Количество пикселей, которые могут быть закрашены в одном столбце зависит от  $t$  и  $R$ , а также от наклона прямой.

Канонически будем полагать радиус фильтра  $R = 1$  и толщину отрезка  $t = 1$ . Тогда максимально может быть закрашено 4 пикселя при наклоне, близком к 45.

Если рассматривать конический фильтр, то в случае 4 пикселей возможный вклад самого верхнего или нижнего будет достаточно мал, поэтому для упрощения алгоритма будем закрашивать только ближайшие 3 к пересечению прямой с вертикалью пикселя.



# Алгоритм Гупты-Спрулла

Пусть, как и в "Алгоритмы растеризации отрезков, окружностей и эллипсов", строится отрезок  $(0, 0) \rightarrow (a, b)$ .

Обозначим нижний текущий (как в алгоритме Брезенхема) пиксель  $P_2$ , нижний -  $P_1$ , а верхний -  $P_3$  (см. рис. 7.12). Пусть  $v$  - расстояние от центральной точки между рассматриваемыми пикселями до выбранного пикселя  $P_2$ . Оно может быть как положительным (если выбран шаг  $s$ ), так и отрицательным (если выбран шаг  $d$ ). Тогда соответствующие расстояния до оси прямой равны

$$r_1 = (1 + v) \cos \theta, r_2 = |v| \cos \theta, r_3 = (1 - v) \cos \theta,$$

где  $\theta$  равен углу наклона прямой (см. рис. 7.12), т.е.  $\cos \theta = a / \sqrt{a^2 + b^2}$ .

Если обратиться к исходному алгоритму Брезенхема ("Алгоритмы растеризации отрезков, окружностей и эллипсов"), несложно заметить, что

$$e = (v - \frac{1}{2}) \cdot 2a$$

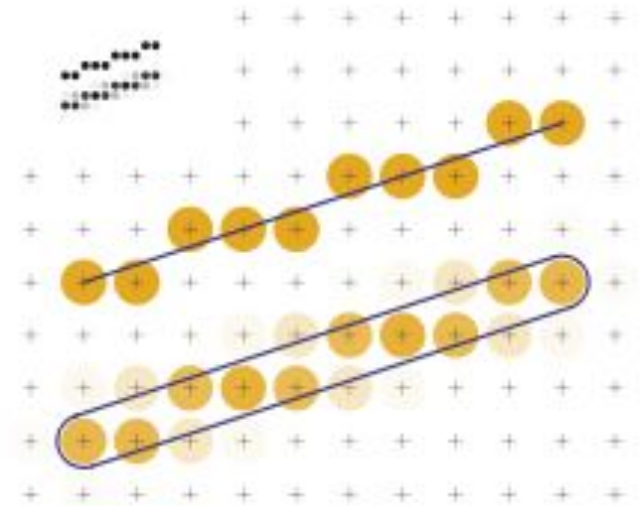
или  $2av = e + a$ . Тогда  $v \cos \theta = \frac{e+a}{2\sqrt{a^2+b^2}}$ .

Для ускорения вычислений построим дискретную аппроксимацию  $F_{R,t}(r)$ , разбив интервал  $[0, R + t/2]$  на равные части и аппроксимируя значение  $F$  в каждом интервале значением в центре этого интервала. Получим таблицу значений интенсивности  $I_{Table}$ .

# Алгоритм Гупты-Спрулла

Пусть функция `round` находит номер интервала по действительному значению в  $[0, R + t/2]$ . Тогда, модифицируя алгоритм Брезенхема, получаем следующий алгоритм:

```
// plot(x,y,I) закрашивает пиксель (x,y) с интенсивностью I
e = 2b - a;
Delta eS = 2b;
Delta eD = 2b - 2a;
denom = 1/(2*sqrt(a*a+b*b));
fixPart = 2a*denom;
// фикс. часть r_1 и r_3
// (x,y) - Координаты текущей точки
x= 0; y = 0; while( x < a ) { av2 = (e+a)*denom;
plot(x, y-1, ITable[round(fixPart+av2)] );
plot(x, y, ITable[round(abs(av2))] );
plot(x, y+1, ITable[round(fixPart-av2)] );
if( e > 0 ) {
// d : диагональное смещение
x++;
y++; e += Delta eD; } else {
// s : горизонтальное смещение
x++; e += Delta eS; } }
```





# Литература

- <https://www.intuit.ru/studies/courses/993/163/lecture/4503?page=3>
- <https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0%D0%9A%D0%BE%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0>