Геометрические преобразования растровых изображений

Содержание

- **Передискретизация**
- **Геометрическое преобразование**
- **Реконструкция**
- **Аппроксимации sinc**
- **Аппроксимация поверхности**
- **Супердискретизаци**
- **Разложение преобразований в композицию более простых**

Передискретизация

Построение дискретного изображения, которое было подвергнуто геометрическому преобразованию, при условии наличия дискретизации исходного изображения является фактически проблемой **передискретизации**.

Пусть исходное (непрерывное) изображение задано функцией I(x, y), тогда его *дискретизация* $I_s(x, y)$ (считаем частоту равной f_s) будет получена умножением на функцию двумерной "решетки«:

(*)

$$\begin{aligned}
I_s(x,y) &= I(x,y)\cdot Comb_{f_s}(x,y)\\
&= I(x,y)\cdot\sum_{i,j}\sigma\left(x-\tfrac{i}{f_s},y-\tfrac{j}{f_s}\right),(i,j)\in\mathbb{Z}^2\\
&= \sum_{i,j} I\left(\tfrac{i}{f_s},\tfrac{j}{f_s}\right)\cdot\sigma\left(x-\tfrac{i}{f_s},y-\tfrac{j}{f_s}\right).
\end{aligned}$$

Следовательно, мы получили растр, с атрибутами

$$I_s(i,j)\overset{def}{=} I\left(\frac{i}{f_s},\frac{j}{f_s}\right).$$

Геометрическое преобразование

Пусть преобразование задано функцией :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix}.$$

Тогда преобразованное изображение равно $I'(x', y') = I(T^{-1}(x', y'))$,

дискретизация (с частотой f'_s) преобразованного изображения должна быть получена как

$$\begin{aligned} I'(x', y') &= I'(x', y') \cdot Comb_{f'_s}(x', y') \\ &= I'(x', y') \cdot \sum_{i', j'} \sigma\left(x' - \frac{i'}{f'_s}, y' - \frac{j'}{f'_s}\right), (i', j') \in \mathbb{Z}^2 \\ &= \sum_{i', j'} I'\left(\frac{i'}{f'_s}, \frac{j'}{f'_s}\right) \cdot \sigma\left(x' - \frac{i'}{f'_s}, y' - \frac{j'}{f'_s}\right) \\ &= \sum_{i', j'} I\left(T^{-1}\left(\frac{i'}{f'_s}, \frac{j'}{f'_s}\right)\right) \cdot \sigma\left(x' - \frac{i'}{f'_s}, y' - \frac{j'}{f'_s}\right). \end{aligned}$$

Реконструкция

Мы рассматриваем задачу, когда нам неизвестно исходное изображение, а известна только его *дискретизация* I_s.

В таком случае в качестве I следует подставить реконструированное изображение I_r.

Реконструкция производится с помощью реконструирующего фильтра RFilter:

(**)

$$I_r(x,y) = \int I_s(s,t) \cdot RFilter(x-s, y-t)dsdt$$

$$\stackrel{7.11}{=} \sum_{i,j} I_s(i,j) \cdot RFilter\left(x - \frac{i}{f_s}, y - \frac{j}{f_s}\right).$$

Аппроксимации sinc

В идеальном случае, когда начальная частота дискретизации была больше частоты Найквиста и в качестве RFilter выступает *функция* sinc изображение будет реконструировано точно, т.е. $I_r = I$.

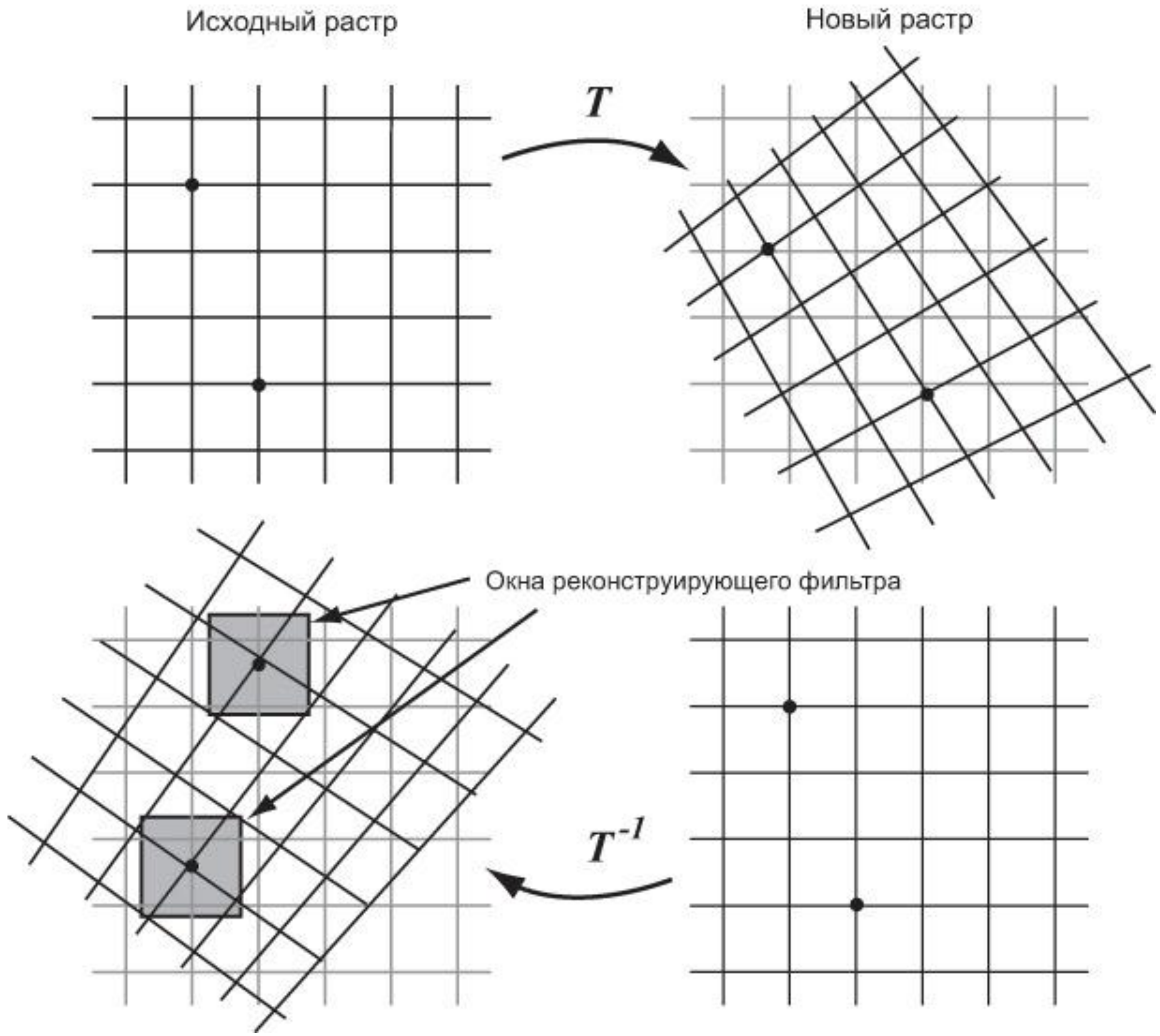
На практике же в качестве RFilter выступают различные аппроксимации sinc с локальным носителем (см. таблицу), что дает некоторые искажения (в частности, *размытие*).

Если в *выражение* (*) подставить I_r, вычисленное по формуле (**) вместо I и воспользоваться определением (7.11) для атрибутов нового растра $I'_s(i', j')$ то получаем, что

$$I'(x', y') = \sum_{i,j} I_s(i,j) \cdot RFilter\left(T^{-1}\left(\begin{pmatrix} \frac{i'}{f'_s}, \\ \frac{j'}{f'_s} \end{pmatrix}\right) - \begin{pmatrix} \frac{i}{f_s}, \\ \frac{j}{f_s} \end{pmatrix}\right).$$

Таким образом, задача передискретизации сводится к применению дискретной свертки (фактически суммированию) исходного дискретизованного изображения I_s с функцией фильтра RFilter, центрированной в прообразе нового пикселя при преобразовании T

Передискретизация



Аппроксимация поверхности

Линейные фильтры, используемые в качестве RFilter, фактически осуществляют локальную интерполяцию или аппроксимацию поверхности I_r по точкам дискретизации.

Если мы просто будем считать I_r некоторой поверхностью, то в этом случае

$$I'_s(i', j') = I_r\left(T^{-1}\left(\frac{i'}{f'_s}, \frac{j'}{f'_s}\right)\right).$$

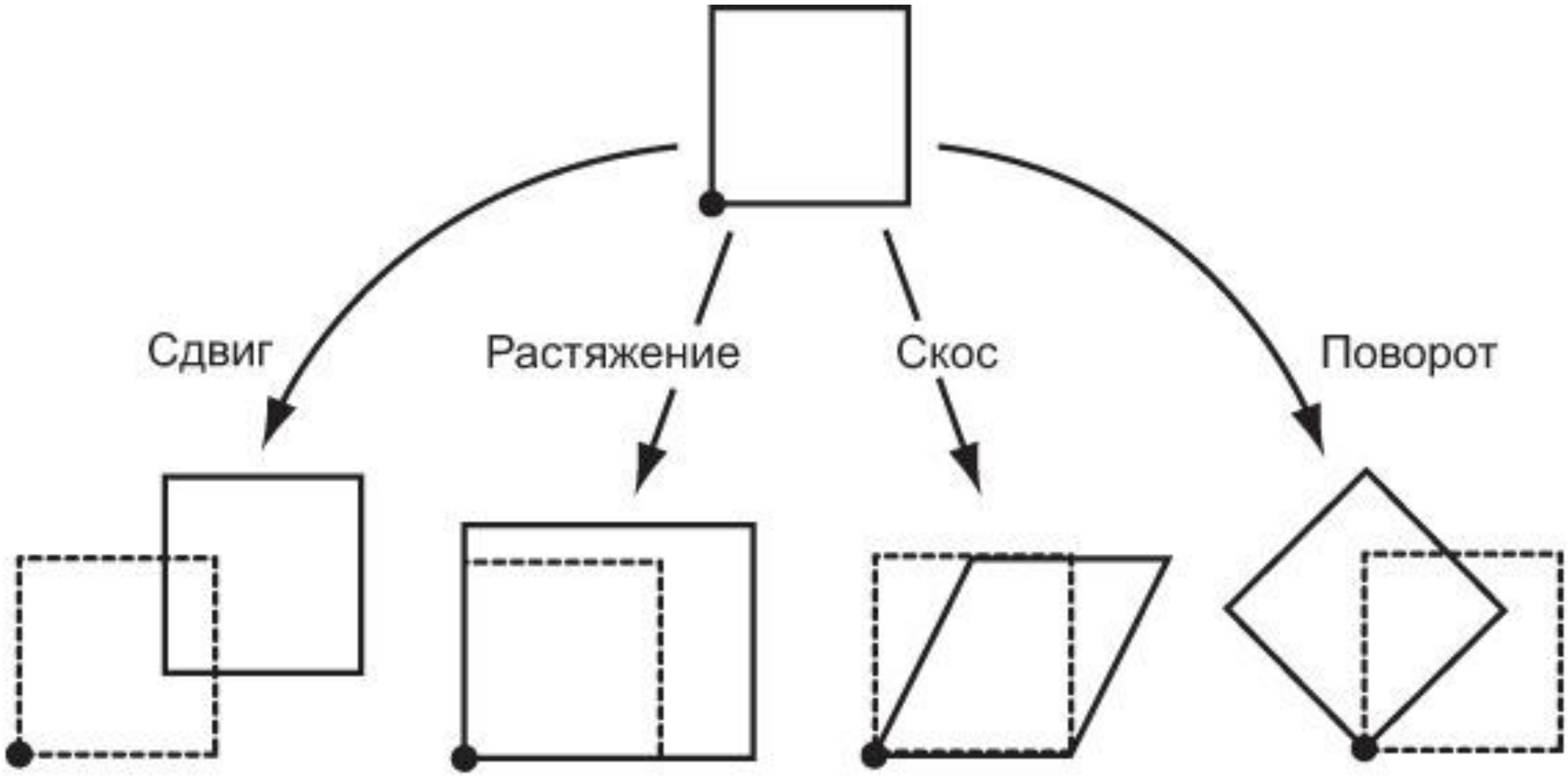
Если в качестве фильтра выступает *функция*-параллелепипед, то при передискретизации новое *значение* будет просто средним по пикселям, попавшим в область носителя (для малого радиуса (1/2) - просто *значение* ближайшего пикселя).

Пирамидальному фильтру соответствует билинейная *интерполяция*.

Более качественные результаты можно получить при использовании бикубической интерполяции, которая соответствует сепарабельному кубическому фильтру (см. таблицу).

Она является стандартной в популярном растровом редакторе Adobe Photoshop.

Аффинные преобразования



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Аффинные преобразования

Предметом нашего рассмотрения будут в основном аффинные преобразования:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

частным случаем которых являются сдвиги, растяжения, скосы и повороты (см. рис.).

В трехмерной графике подобные проблемы возникают при текстурировании, т.е. наложении искаженных растровых изображений на поверхности объектов, которые затем проецируются на экран.

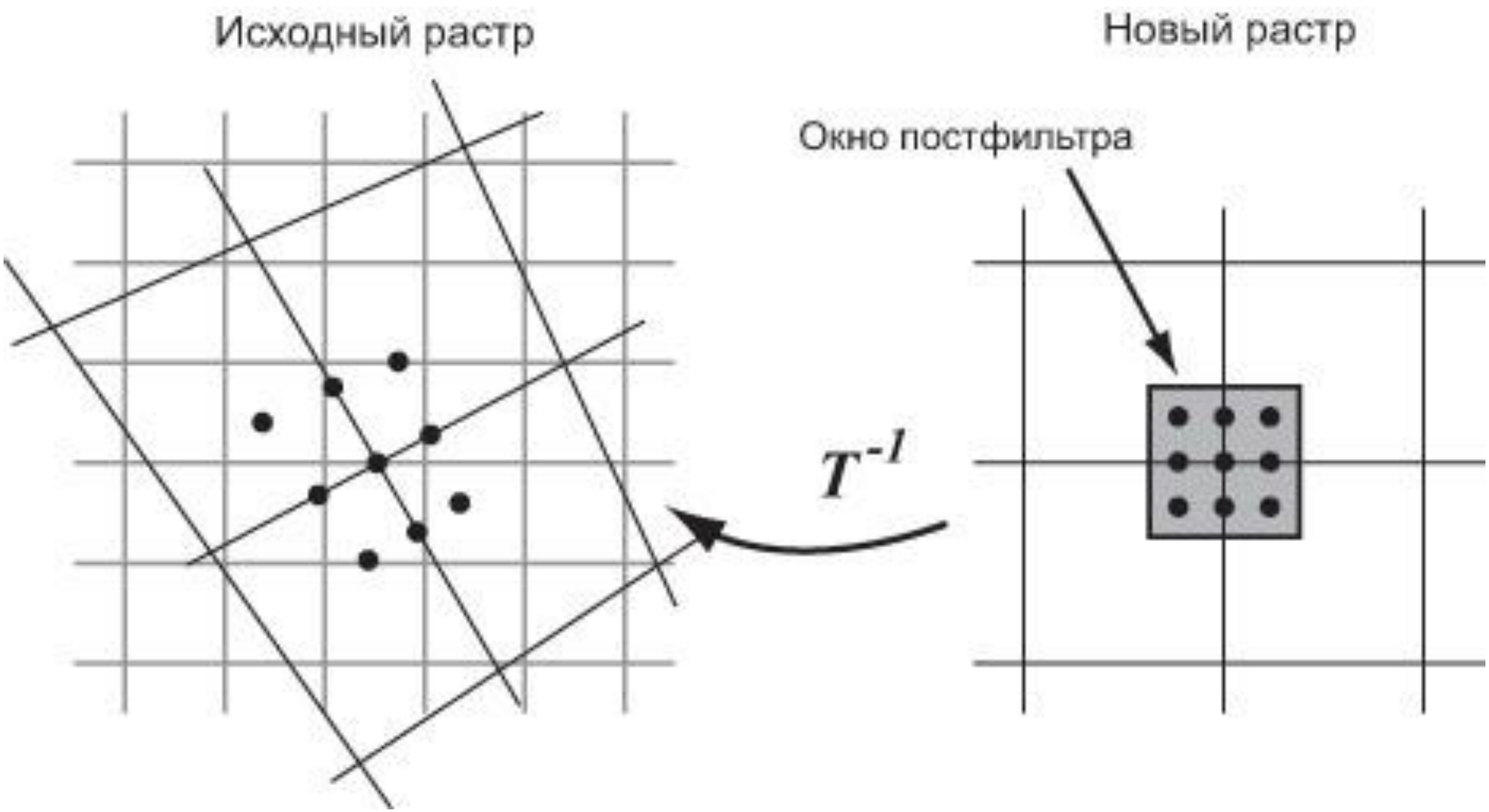
Там, как правило, речь идет о перспективных преобразованиях (задаются невырожденной трехмерной матрицей)

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}},$$

$$y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}}.$$

☹

Супердискретизация



Супердискретизация

При перспективных преобразованиях возможны значительные искажения, при которых *эффективная частота* дискретизации, обратно пропорциональная расстоянию между переведенными точками , может быть значительно больше исходной, которая полагается равной 1 (в пикселях исходного изображения).

В этом случае прибегают к супердискретизации.

Локально для одного пикселя в I' строится промежуточная *дискретизация* с частотой, большей чем f' (обычно в *целое число* раз, которое зависит от степени искажения[3]), а затем по ней с помощью второго этапа дискретной фильтрации (зачастую здесь применяется простейший усредняющий фильтр-параллелепипед) получаем *значение* интенсивности результирующего пикселя (см. рис.).

Как всегда, на практике, пожертвовав точностью, можно получить более быстродействующие алгоритмы.

Разложение преобразований в композицию более простых

В некоторых случаях имеет смысл раскладывать сложное преобразование в последовательность более простых, для которых существуют эффективные алгоритмы.

Самый простой и полезный пример - представление общего преобразования T в виде композиции преобразований по столбцам и по строкам. Преобразования, сохраняющие столбцы или строки, можно эффективно распараллелить (по столбцам и строкам соответственно).

Пусть , $T = R \circ C$ где C сохраняет столбцы, а R - строки.

Пусть , $T\colon (x, y) \mapsto (x'', y'')$

а промежуточный результат - $(x', y')(C\colon (x, y) \mapsto (x', y'), R\colon (x', y') \mapsto (x'', y''))$

Вывод

Тогда

$$T = \begin{pmatrix} t_1(x,y) \\ t_2(x,y) \end{pmatrix}; C = \begin{pmatrix} x \\ c_2(x,y) \end{pmatrix}; R = \begin{pmatrix} r_1(x',y') \\ y' \end{pmatrix}.$$

Следовательно

$$T = \begin{pmatrix} t_1(x,y) \\ t_2(x,y) \end{pmatrix} = \begin{pmatrix} r_1(x, c_2(x,y)) \\ c_2(x,y) \end{pmatrix}.$$

Таким образом, $c_2(x, y) = t_2(x, y)$ и $r_1(x, t_2(x, y)) = t_1(x, y)$.

Для того, чтобы найти r_1, надо в выражении t_1 вычленить все подвыражения, содержащие y, и привести их к виду $t_2(x, y)$.

После этого, заменив эти подвыражения на y, получим выражение для r_1.

Пример поворота на угол

$$T(x,y) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Тогда

$$t_1(x,y) = \cos\theta \cdot x - \sin\theta \cdot y,$$
$$c_2(x,y) = t_2(x,y) = \sin\theta \cdot x + \cos\theta \cdot y,$$

тогда $y = (t_2(x,y) - \sin\theta \cdot x)/\cos\theta$ (это, конечно, возможно, когда $\theta \neq \pm 90^\circ$). Отсюда

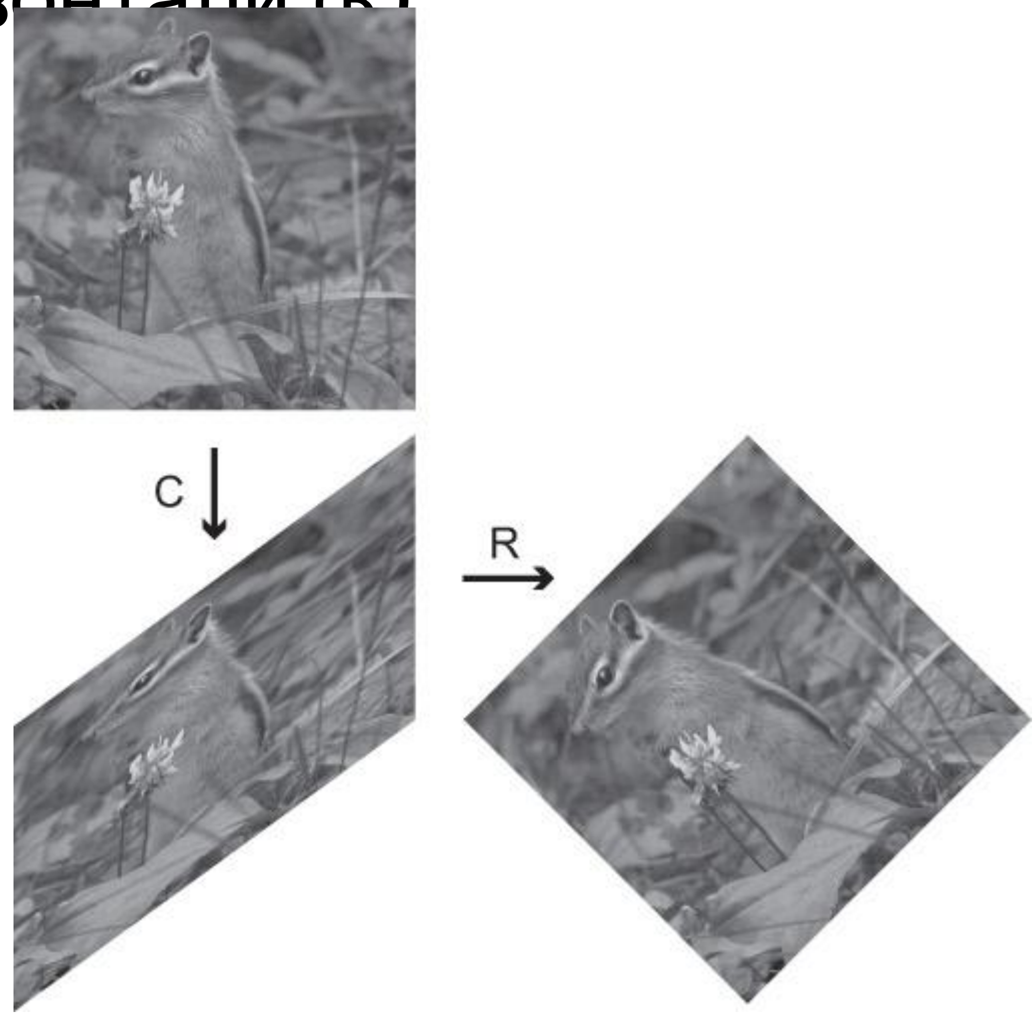
$$t_1(x,y) = \cos\theta \cdot x - \sin\theta \cdot \left(\frac{t_2(x,y) - \sin\theta \cdot x)}{\cos\theta}\right)$$
$$= \sec\theta \cdot x - \tan\theta \cdot t_2(x,y).$$

Получаем, что $r_1(x,y) = \sec\theta \cdot x - \tan\theta \cdot y$. Итак, получено следующее разложение (наглядно на рис.

$$\overset{T}{\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}} = \overset{R}{\begin{pmatrix} \sec\theta & -\tan\theta \\ 0 & 1 \end{pmatrix}} \cdot \overset{C}{\begin{pmatrix} 1 & 0 \\ \sin\theta & \cos\theta \end{pmatrix}}$$

Разложение вращения на скос и смещение по вертикали (C), а затем по горизонтали (R)

- Преобразования C и R представляют собой композицию сжатия и скоса по вертикали и горизонтали соответственно. Поэтому данная декомпозиция позволяет реализовать поворот с помощью нескольких последовательных применений алгоритма для скосов и масштабирования.

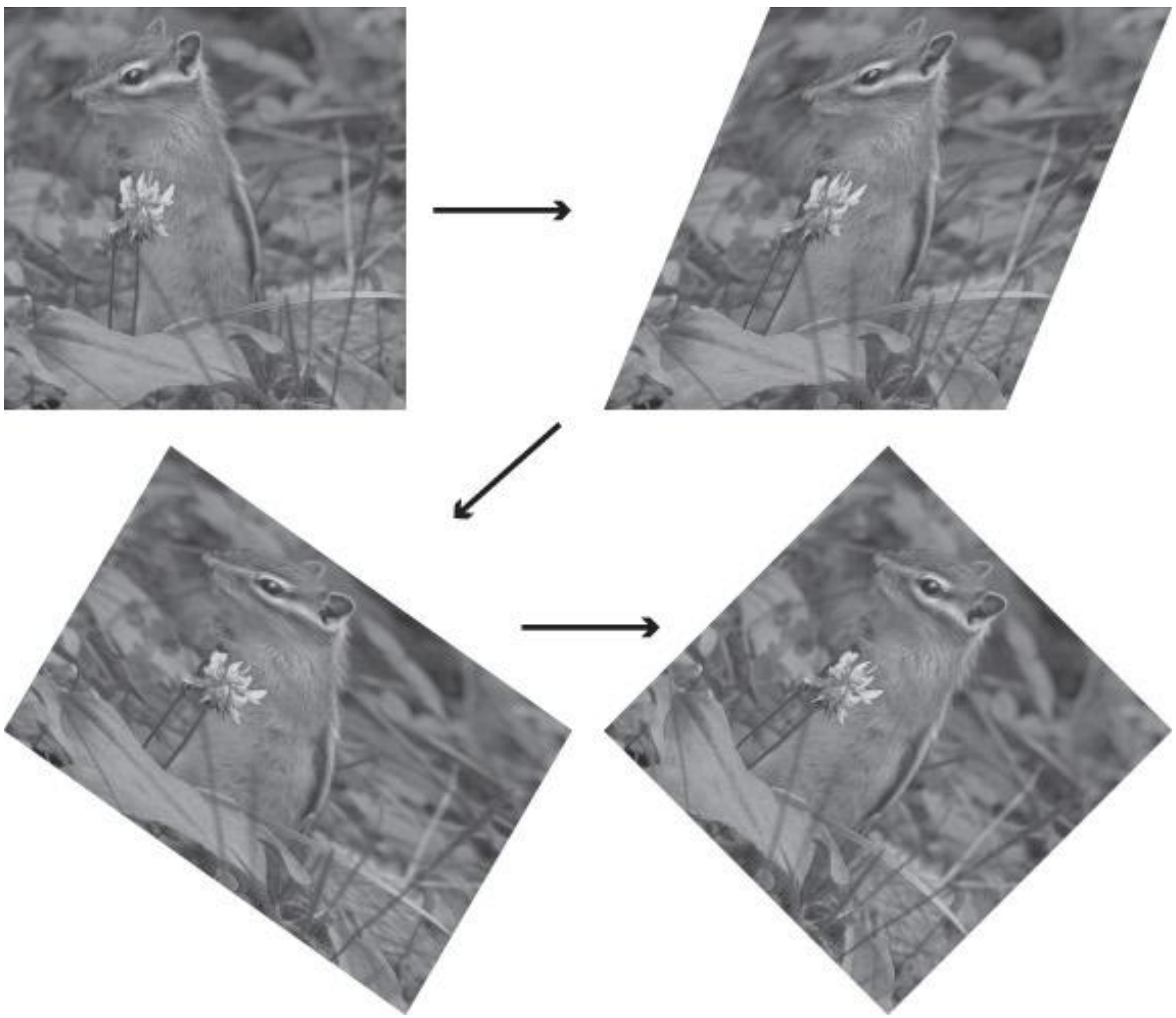


Разложение вращения на 3 скоса

Существует альтернативное разложение на 3 скоса для матрицы поворота (наглядно на рис.), которое позволяет избавиться от существенных искажений при $\theta \approx \pm 90^\circ$, присущих вышеизложенному алгоритму:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$

Разложение вращения на 3 скоса.



Литература

- https://www.intuit.ru/studies/courses/993/163/lecture/4503?page=9