Samsung Test Task by Oleg Kuzik

The algorithm for the test task is implemented in **Wave** class and comprises such performance steps:

1. The angle of wave propagation ($0° < \alpha < 360°$) is randomized. For the first pass, $\alpha = 45$.

2. The corner (from which the wave starts propagating) is selected, depending on the angle. The basic coordinates of the wave (x,y) are initialized with either zeros or width/height of the background image respectively.

3. The wave function (Bell-Shaped Normal Distribution Function) is estimated into an array of elevations of wave in different displacements along the wavelength)

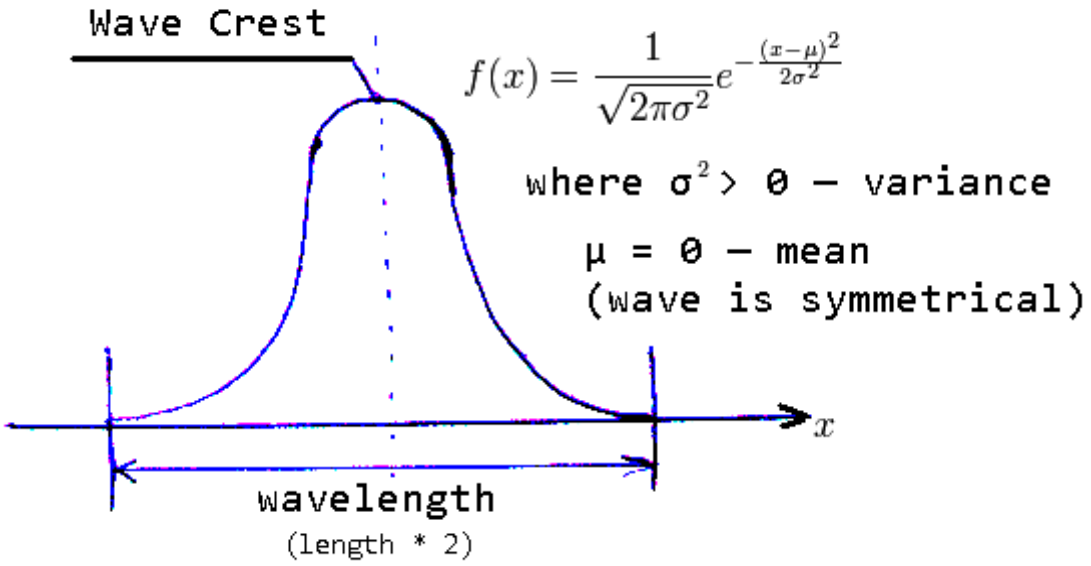


Figure 1. Cross-section of a bell-shaped wave.

by Oleg Kuzik, thesky.himself@gmail.com 2012

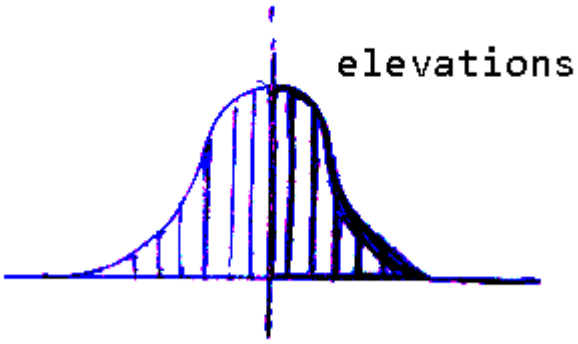


Figure 2. Elevations (set of points representing the height of wave at certain points throughout the wavelength)

4. The wave is propagated for one step. In case of propagation at a certain angle, displacement coefficients Δx and Δy should be calculated. Also at this point the **speed** parameter is involved:

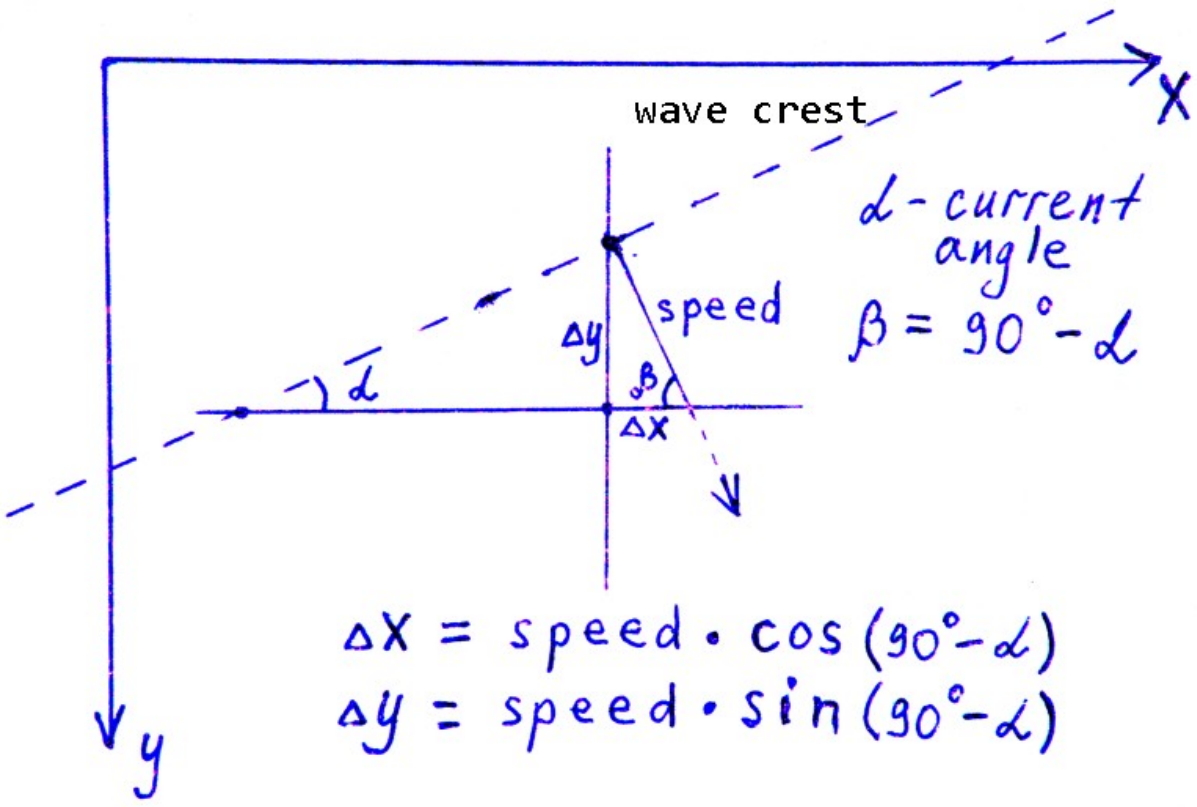


Figure 3. implementation of wave propagation along the slope with predefined speed.

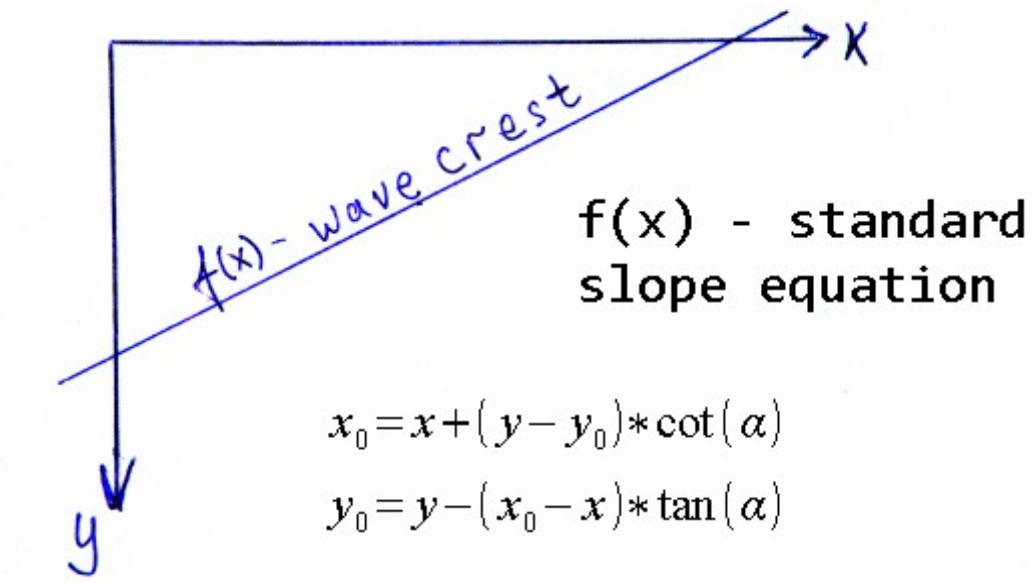


Figure 4. Slope equations for wave crest-obtaining methods

5. The plane wave effect (stretch distortion) is applied to the overlay bitmap. The algorithm is implemented as follows:

At first, two wavelength-sized arrays are initialized with precise displacement values for x and y axes, taking into consideration the slope correction coefficients:

$$dx = \cos(90° - \alpha)$$
$$dy = \sin(90° - \alpha)$$

Thus,

$$displacement\_x = dx \cdot (elev[current] - elev[next])$$
$$displacement\_y = dy \cdot (elev[current] - elev[next])$$

As we have the arrays initialized (this is done only once), the main distortion algorithm is applied to the overlay image.

However taking into account the application performance considerations (and the constraint of non-usage of graphics processing frameworks as well as the inefficiency of per-pixel editing methods), the algorithm itself is divided into two separate algorithms for horizontal and vertical processing of the bitmap.

This means, that there are two for-loops, the nested loop iterates only the area affected by the wave deformation.

The processing methods are selected depending on current angle and are based on moving pixels of the bitmap according to the displacements arrays.

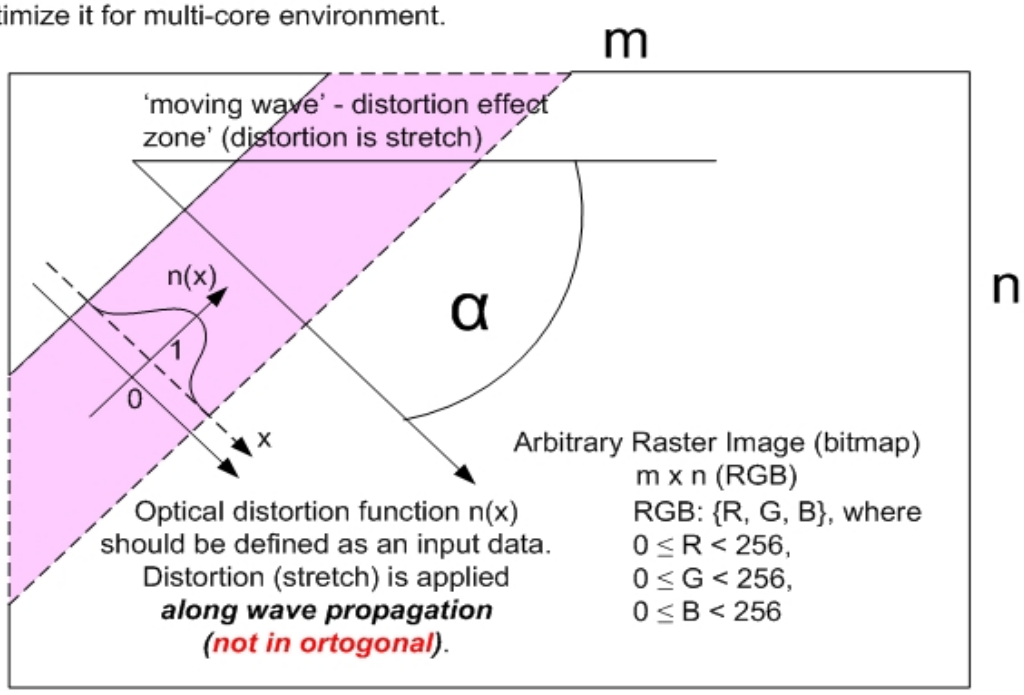
The only drawback of such method is the small deviation of the visible wavelength depending on the current angle, however this algorithm is far more efficient than updating all pixels on the screen.

Original Task:

Time-frame for this mini-project is 10 days

Flat surface 'wave' animation mini-project
By Andrey Fisunenko

Imagine,... you are observing a picture on the bottom of a borderless aquarium through a rectangular hole.
Periodically you can see a flat wave spreading over the surface of water…
Task is
1) to simulate 'flat wave' visual effect with a proper algorithm
2) to optimize it for multi-core environment.



Basic requirements:
0) program should be Win32 application (C or C++)
1) it should obtain BMP image as an input;
2) application has to provide a way for tuning parameters of 'wave' animation (e.g. 'speed of propagation, etc) and mean for distortion function n(x).
3) angle α has random value for every pass of wave ($0 \le \alpha < 360$), for first pass it must be 45° (direction must be as shown on the figure above);
4) every pass of 'wave' animation has to start from random corner pixel of the image automatically.
5) application has to provide for user switching between 'parallel' and 'sequential' animation mode.
6) any external 3D libraries (OpenGL, etc) must not be used.

Two variants have to be implemented:
1) sequential algorithm;
2) 'parallelized' algorithm (performance optimization with thread API);

The final result is **source code** for Visual Studio 2005 or 2008 and an accompanying **algorithm description document**.

Acceptance criteria:
A) An application runs animation of 'surface wave' after pressing F5 in Visual Studio.
B) An application complies fully with 0) – 6).
C) There is an algorithm description document (diagrams are more preferable).

Notes:
- You are free to choose a way showing efficiency of parallel version comparing to sequential one

Samsung Ukraine R&D Center

by Oleg Kuzik, thesky.himself@gmail.com 2012