SIXTH FRAMEWORK PROGRAMME
PRIORITY 1.6. Sustainable Development, Global Change and Ecosystem
1.6.2: Sustainable Surface Transport

506716

Title	**Design guidelines for bitmap displays**
Authors	**Bureau Mijksenaar b.v.**
Summary	**– Bitmap vs. vector** **– Bitmap types** **– Lines/Angles** **– Roundings** **– Image conversion** **– Software to use** **– File types**
Status	(**F: final,** D: draft, RD: revised draft)
Date	**2005-11-07**
Revisions	
Distribution	
Document ID	**Designing for bitmap #E877B.pdf**
Attachments	

Design guidelines for bitmap displays

Version 1.0

Date 2005-11-10

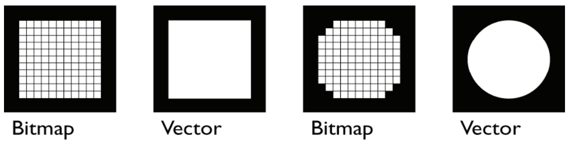
1. Introduction

This document was written for the project partners of In-Safety work packets 2.4 and 2.6. It will recommend some practices and guidelines for the production of bitmap pictograms for Variable Message Signs (VMS) located on the European highways. Its first and foremost goal is to standardize the output (file types, characteristics) of the different project partners.

For the sake of simplicity all designs and testing during the In-Safety project will adheres to a 64x64 pixel non-anti-aliased bitmap display. In practice the selected designs might be used on displays with different characteristics (higher and lower resolution, different colours, possibility of anti-aliasing or display of vectorized information). Another goal of these guidelines is to make adaptation of an existing design to a different display as easy as possible.

2. Bitmap characteristics

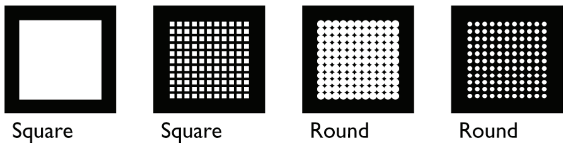
2.1 Bitmap versus vector images



The fundamental difference between bitmap images (such as digital photo's) and vector images (such as Adobe Illustrator files) is the fact that a bitmap image has a discreet resolution where a vector image is infinitely scalable. A bitmap image has a discreet number of fixed elements, a vector image has elements that are described within a coordinate system, and that are not limited to a pre-defined grid.

All displays are bitmap displays. Thus a vector image cannot be show on a bitmap display as it is, it has to be converted to a bitmap first. At higher resolutions (computer monitors) this conversion is not noticeable and can be performed by an automated system. At lower resolutions (our 64x64 VMS) this conversion is very much noticeable, and cannot be performed by an automated system. The human eye is needed to adjust the design to the coarse resolution and the limitations of the display used.

2.2 Bitmap types

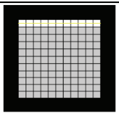


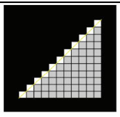
The bitmap images we display on our computer monitors (and our computer monitors proper) have square, gapless pixels. This is also the kind of pixel the pictogram designs will be tested with.

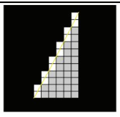
In reality however, pixels on large displays are hardly ever square and hardly ever gapless. They can be square, rectangular, round and other shapes, and can have a lot of black in between the pixels. Especially the black gaps are a design constraint: they will make visual recognition of lines and surfaces harder.

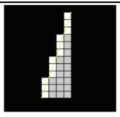
2.3 Bitmap characteristics

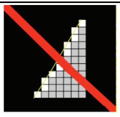
Bitmaps have characteristics that have to be taken into account when designing for low-resolution displays. Because of the discreet character of the pixels, only rectangular lines can be shown as they are: straight. All other lines will be divided into smaller straight-line segments. When the size of these smaller segments varies, the line will look very wobbly and irregular.


0°, 90°


45°

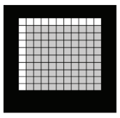

26.6°, 63.4°

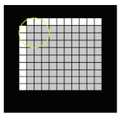

18.4°, 71.6°

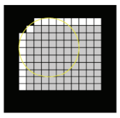

30°, 60°

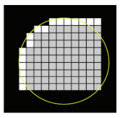
As can be read from the illustration, it is best to limit the angle of straight lines to those angles that can be properly displayed on a bitmapped display. Other angles (most notably, but not limited to, 30° and 60° angles) should be avoided for they will appear very jagged[1].

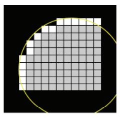
Bitmaps are also unable to draw round corners. The human visual system however will see a rounded corner when any of the radiuses below are drawn:


right angle

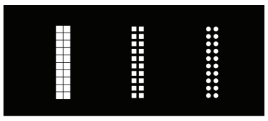

r=2

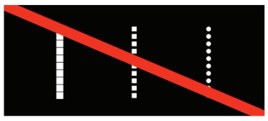

r=4


r=6

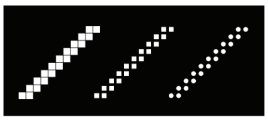

r=8

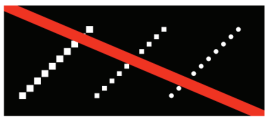
When it comes to drawing smaller elements, we will have to compensate for displays that show large gaps between pixels. Thin lines might be seen as separate pixels, so – whenever possible – lines of at least two pixels wide should be preferred.





This applies as well to non-orthogonal lines:





[1] This 'jaggedness' will be hardly noticeable of anti-aliased displays, displays that are not limited to black and white pixels but that can also display the grays in between. Anti-aliasing however does not work well on lower resolution displays, and is not something that will be used in the In-Safety tests. Also, designs that are optimized for our 64x64 pixel displays can easily be converted to designs for anti-aliased displays, but this does not work the other way round. For this reason anti-aliasing will not be discussed in the rest of this document.

3. Design process and software

3.1 Bitmap design strategies

There are two fundamental different strategies for designing a series of bitmap images (such as our 'car' image, that will be used in various sizes and positions):

1. Design the bitmaps directly as bitmaps. If vector images are needed as well (for instance for larger sizes) then design these with the bitmaps as basis.

2. Design a vector image first. Scale this vector image to the appropriate sizes and draw bitmaps on basis of the scaled vector images.

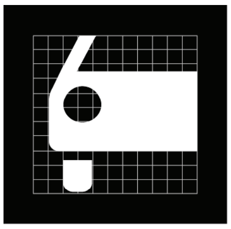
Of these two strategies, the first will give you the best bitmaps. This is the way the best screen fonts (such as Verdana) have been designed. This strategy has one major drawback however: it is very labour intensive, for there is no easy way to generate a new size from an existing bitmap image. Each and every image has to be drawn form scratch.
The second strategy will lead to less clearer bitmaps, but has the advantage of being easy to scale and modify.

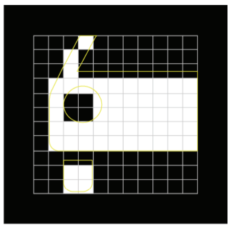
This is why we propose to use the second design strategy in this phase of the In-Safety project. As long as we are still designing for variants (both in pictogram design and in pictogram application) to be used in user testing, it is paramount to be able to design these variants quick and easy.

In a later phase of the In-Safety project (when pictograms have been selected, image details such as the style of the car have been agreed upon, and when a 'red triangle variant' has been chosen) it will be wise to switch to first design strategy.

3.2 Converting a vector image to a bitmap image

There are a lot of programs that can convert vector images to bitmap images. What happens during that conversion is that the average colour value of every pixel is calculated (for instance 75% black), and then that pixel gets mapped to the nearest available colour[2] (in our case that would be 100% black).

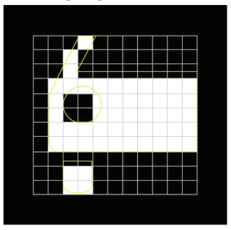

vector image

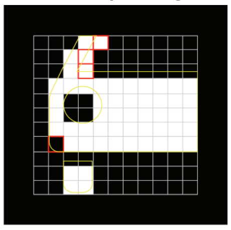

bitmap

[2] In case of a vector to bitmap conversion with anti-aliasing, the assigned colour value will be exactly the average colour value, no mapping to a limited colour map will take place. This has become the standard for most image editors. This is inappropriate however for our four or less colour VMS displays.

This automated conversion will always be very coarse, and the resulting image will need to be adjusted. There are three different ways to adjust the resulting bitmap image:

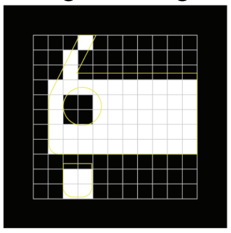
1. Change pixels directly in the bitmap image

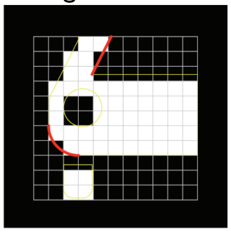

standard conversion


pixels edited

This method has the same drawback as the strategy to directly design bitmaps: all adaptations are size specific and can only be re-used in images of exactly the same size.

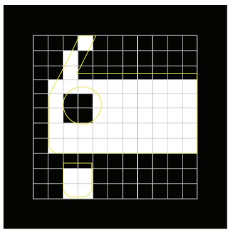
2. Change the original vector image

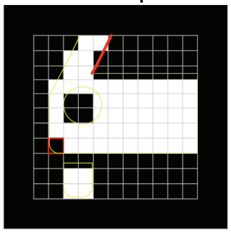

standard conversion


vectors edited

This method has the advantage that adaptations can be re-used, even in images of different sizes. Since a lot of the adaptations however will be size-specific, this advantage has only limited application possibilities.

3. A combination of the previous two options


standard conversion


both edited

Of these three adjustment methods, the third one is most fitting to our current phase in the design of VMS pictograms. Small adaptations (such as correction of a rounded angle) can be performed very quick; recurrent adaptations (angles, alignment, repeating elements such as headlights) can be easily re-used in other pictograms and other sizes.

3.3 Software

There is only one program that easily combines the two adjustment methods from the previous paragraph, and that is Macromedia Fireworks. This is also one of the very few programs that were conceived for the design of small and coarse bitmap images. Other, more popular software such as Adobe Photoshop and Adobe Illustrator are generally conceived with the design of large print images in mind. As such the use of these programs, even if they can perform the tasks needed, will be akin to hunting a mosquito with a cannon.

We therefore strongly advise all project partners to use Macromedia Fireworks for the conversion of vector images to bitmap images, and for the corrections of the vector drawings and bitmap images.

3.4 Resulting files

Use of Fireworks will result in a source file (PNG). This file contains the vector images, grid settings, export settings, etc. When opened in Fireworks all these edit options will be visible and editable, when opened in another image application only the bitmap image will be shown.

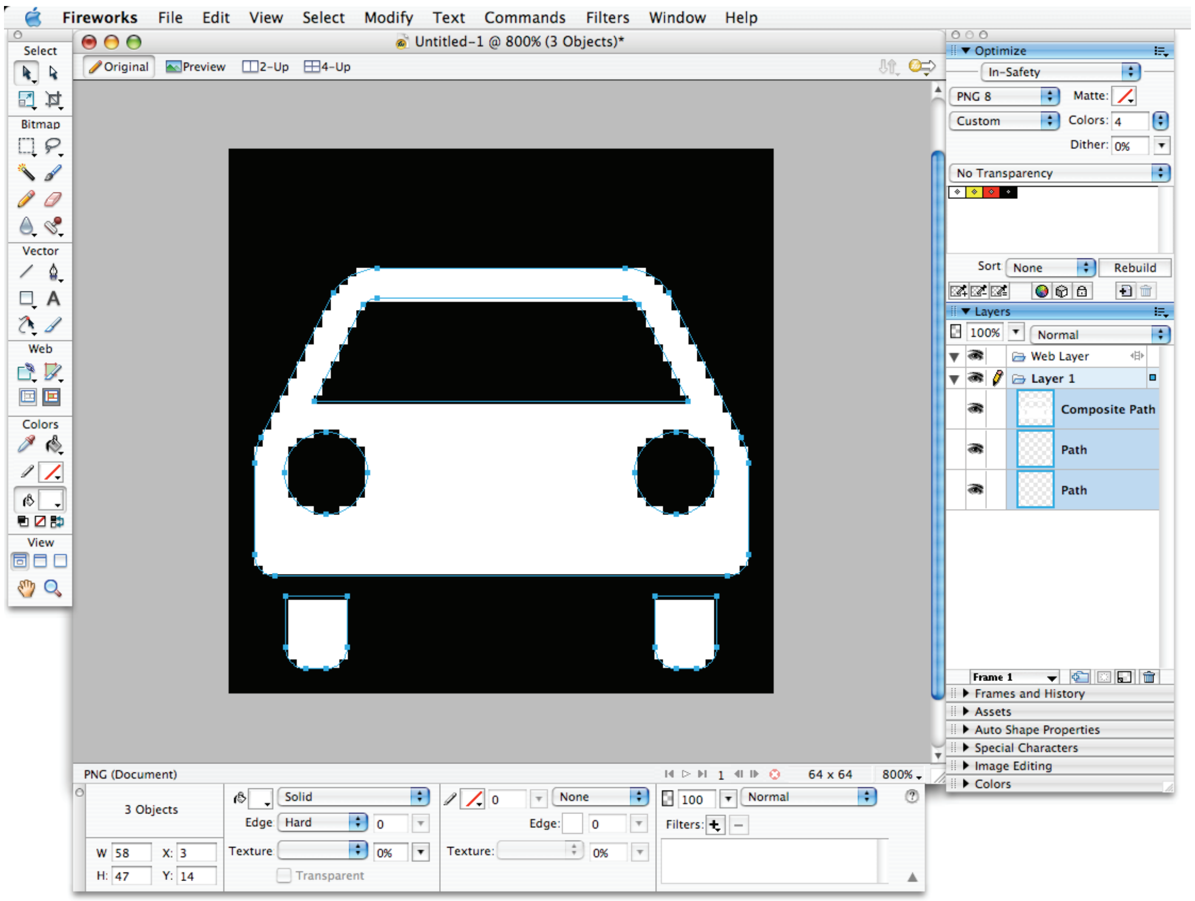
For reasons of file size and compatibility it is advisable to export the source files into a flat file format, which only contains the bitmap in four colours, and does not contain any of the Fireworks baggage. Our advice is to export all images in a four colour PNG8. Take note however that the resulting file will be named identical to the source file; take care not to overwrite your original file!

The four colours used should be:
- Black (#000000)
- White (#FFFFFF)
- Red (#FF0000)
- Yellow (#FFFF00)

As a safe alternative export of the images as GIF may be used.
Do not use other file types for export, for they may be lossy (JPG) ad thus unsuitable for our type of bitmap image, or prone to conversion problems (BMP, PICT).

4. Addendum

4.1 Fireworks edit window



In-Safety
PNG 8 Matte:
Custom Colors: 4
Dither: 0%
No Transparency
Sort None
100% Normal
64 x 64 800%
Solid
Edge Hard 0
Texture 0%
0 None
Edge: 0
Texture: 0%
100 Normal
W 58 X: 3
H: 47 Y: 14