**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 m** **56341**

**Online – April 2021**

**Title: Clarifying linear and radial gradients in ISO/IEC 14496-22 AMD2**

**Authors: Peter Constable, Cosimo Lupo, Dominik Röttsches, Roderick Sheeter, Vladimir Levantovsky (on behalf of AHG)**

During the break-out group discussion at the last SC29/WG3 meeting in January 2021 (see report in **m56219**), the group raised three issues related to definitions of gradients:

* The description of rotation vector, and the effect that transforms would have on final results for linear gradients;
* The expected behavior for radial gradients when two circles defining the gradient are the same, and
* Underspecified behavior of radial gradients when applying a transform flattens both circles and their centers to a line.

This contribution summarizes the results of the discussions and proposes new updates to the relevant sections of the sub-clause 7.7.11.1.2 “Gradients” defined as part of the ISO/IEC 14496-22:2019/AMD2, and resolves the issues outlined by the BoG report.

*5.7.11.1.2.2 “Linear gradients”*

*Replace the entire content of the sub-clause with the following:*

A linear gradient provides gradation of colors along a straight line. The gradient is defined by three points, p₀, p₁ and p₂, plus a color line. The color line is positioned in the design grid with stop offset 0 aligned to p₀ and stop offset 1.0 aligned to p₁. (The line passing through p₀ and p₁ will be referred to as line p₀p₁.) Colors at each position on line p₀p₁ are interpolated using the color line. For each position along line p₀p₁, the color at that position is projected on other side of the line.

The additional point, p₂, is used to rotate the gradient orientation in the space on either side of the line p₀p₁. The line passing through points p₀ and p₂ (line p₀p₂) determines the direction in which colors are projected on either side of the color line. That is, for each position on line p₀p₁, the line that passes through that position on line p₀p₁ and that is parallel to line p₀p₂ will have the color for that position on line p₀p₁.

NOTE For convenience, point p₂ can be referred to as the *rotation point*, and the vector from p₀ to p₂ can be referred to as the *rotation vector*. However, neither the magnitude of the vector nor the direction (from p₀ to p₂, versus from p₂ to p₀) has significance.

If either point p₁ or p₂ is the same as point p₀, the gradient is ill-formed and shall not be rendered.

If line p₀p₂ is parallel to line p₀p₁ (or near-parallel for an implementation-determined definition), then the gradient is ill-formed and shall not be rendered.

NOTE An implementation can derive a single vector, from p₀ to a point p₃, by computing the orthogonal projection of the vector from p₀ to p₁ onto a line perpendicular to line p₀p₂ and passing through p₀ to obtain point p₃. The linear gradient defined using p₀, p₁ and p₂ as described above is functionally equivalent to a linear gradient defined by aligning stop offset 0 to p₀ and aligning stop offset 1.0 to p₃, with each color projecting on either side of that line in a perpendicular direction. This specification uses three points, p₀, p₁ and p₂, as that provides greater flexibility in controlling the placement and rotation of the gradient, as well as variations thereof.

Figures 5.12 – 5.14 illustrate linear gradients using the three different color line extend modes. Each figure illustrates linear gradients with two different rotation vectors. In each case, three color stops are specified: red at 0.0, yellow at 0.5, and blue at 1.0.



**Figure 5.12 Linear gradients with different rotations using the pad extend mode**



**Figure 5.13 Linear gradients with different rotations using the repeat extend mode**



**Figure 5.14 Linear gradients with different rotations using the reflect extend mode**

NOTE When a linear gradient is combined with a transformation (see 5.7.11.1.5), the appearance will be the same as if the gradient were defined using the transformed positions of points p₀, p₁ and p₂.

Linear gradients are specified using a PaintVarLinearGradient or PaintLinearGradient table, with or without variation support, respectively. See 5.7.11.2.5.3 for format details.

See 5.7.11.1.3 for details on how fills are applied to a shape.

*5.7.11.1.2.3 “Radial gradients”*

*In the third paragraph describing the radial gradients rendering steps, replace the description of step 3 with the following:*

1. For all values of ω where r(ω) > 0, starting with the value of ω nearest to positive infinity and ending with the value of ω nearest to negative infinity, draw the circular line with radius r(ω) centered at position (x(ω), y(ω)), with the color at ω, but only painting on the parts of the bitmap that have not yet been painted on in this step of the algorithm for earlier values of ω.

*In the fourth paragraph describing the algorithm for rendering radial gradients, insert the first bullet point with the following text:*

* When the circles are identical, then nothing is painted.

*In the caption text for Figures 18 – 20, replace the word “string” with the “strip”.*

*In the end of the sub-clause, replace the very last NOTE [after Figure 5.26] with the following text, including additional figures. Re-number the remaining figures within the clause 5:*

NOTE When a radial gradient is combined with a transformation (see 5.7.11.1.5), the appearance will be the same as if the geometry of the two circles were transformed and step 3 of the algorithm were performed by interpolating the shapes derived from the two transformed circles. For the condition r(ω) > 0, the pre-transformation values of r(ω) can be used.

NOTE A scale transformation can flatten shapes to resemble lines. If a radial gradient is nested in the child sub-graph of a transformation that flattens the circles so that they are nearly lines, the centers could still be separated by some distance. In that case, the radial gradient would appear as a strip or a cone filled with a linear gradient.

If a radial gradient is nested in the sub-graph of a transformation that flattens the circles so that they form a single line (or nearly a line, for an implementation-determined definition), with both centers on that line, then the resulting gradient is degenerate and shall not be rendered.

NOTE As seen in the figures above, the gradient fills the space when one circle is contained within the other, but not when neither circle is contained within the other. In a variable font, if the placement or radii of the circles vary, then a sharp transition can occur if the variation results in one circle being contained with the other for some instances but not for other instances. This transition will occur when the inner circle just touches the outer circle (i.e., they have exactly one point in common). In this case, the gradient will fill exactly one half of the space. This is illustrated in figure 5.27 using the pad extend mode.



**Figure 5.27 Radial gradient with inner circle just touching the outer circle, pad extend mode**

When the repeat or reflect extend modes are used, having the two circles in very close proximity results in very high spatial-frequency transitions that can lead to Moiré patterns or other display artifacts. This is illustrated in figure 5.28, which shows the display result, for one particular rendering context, of a radial gradient defined using nearly-identical circles and the reflect extend mode.



**Figure 5.28 Radial gradient defined using nearly-identical circles, showing interference patterns**

The artifacts seen can be affected by a combination of several factors, such as image scaling, sub-pixel rendering, display technology, and limitations in software implementation or display capabilities. For this reason, the appearance can be very different in different situations. Font designers should exercise caution if the circles are in close proximity (either in a static design or for some variable font instances), and should not rely on these display artifacts to obtain a particular pattern.

Radial gradients are specified using a PaintVarRadialGradient or PaintRadialGradient table, with or without variation support, respectively. See 5.7.11.2.5.4 for format details.

See 5.7.11.1.3 for details on how fills are applied to a shape.