

HINTING EN PANTALLAS DE ALTA RESOLUCIÓN

Daniela Rodríguez

Resumen

Las pantallas de alta resolución no son algo nuevo —hay prototipos documentados ya en los años ochenta— pero al parecer ha llegado el momento de implantar esta tecnología: en junio de 2010 Steve Jobs presentó el iPhone 4 con la pantalla Retina como principal novedad: una superficie de 3,5 pulgadas con una densidad de 336 píxeles por pulgada. El ojo humano es incapaz de distinguir los píxeles en una imagen de esa resolución, por lo que el equipo directivo de Apple decidió bautizarla como Retina y utilizarla también en su nuevo iPad, presentado en marzo de 2012. Esta vez se trata de una superficie de algo menos de 10 pulgadas (9,7) con una densidad de 264 ppp —2048×1536 píxeles; para hacernos una idea, una televisión de alta definición Full HD tiene 1920×1080 píxeles, un tercio menos—. El siguiente paso lógico era dar el salto a la gama alta de portátiles y equipos de sobremesa, y Apple presentó en junio los MacBook Pro con pantalla Retina. Acer y Asus también han anunciado portátiles con 1920×1080 píxeles para el primer semestre de este año, lo que hace sospechar de una próxima guerra por la alta resolución, que tendrá que ir acompañada de un aumento de potencia en los procesadores gráficos.

Ya hemos vivido antes revoluciones de este calibre: durante un tiempo viviremos una transición en la que los tipos que diseñemos se verán en pantallas de baja resolución, como las actuales, que rondan los 100 ppp, mientras aumenta el uso de pantallas de alta resolución, como las Retina de Apple, que rondan los 300 ppp. Cuando las últimas se conviertan en el estándar, los diseñadores de tipos se verán enfrentados a un cambio metodológico que afectará al proceso de hinting de sus creaciones, pues apenas existirán pantallas de baja resolución que deformen su trabajo —deformando las astas, cambiando los grosores, alterando la simetría de sus dibujos y cerrando o abriendo sus contragrafismos—. El trabajo del hinting se centrará en los cuerpos más pequeños, los que tienen menos píxeles por cuadratín, y puede que modifique también el uso del formato OpenType TrueType.

Bajo este nuevo paradigma es preciso averiguar el alcance de las deformaciones de los glifos en las nuevas pantallas y determinar qué nuevos procesos de hinting serán necesarios. En este trabajo se harán pruebas de pantalla a diferentes cuerpos y con distintas tipografías (con y sin hinting) para comenzar a perfilar el alcance y la profundidad del nuevo sistema, y se hará una propuesta concreta que pueda servir de base para futuros estudios sobre el tema.

CATEGORÍA

Comunicación

TÓPICO

Tipografía y tecnologías

PALABRAS CLAVE

Hinting, pantallas de alta definición, pantalla retina, proceso de producción tipográfica, anti-alias, rastreo

Hinting, high-definition displays, Retina Display, Typographic production process, anti-alias, rasterizing

INTRODUCCIÓN

El problema con las pantallas de baja resolución es el siguiente: actualmente las tipografías se producen en formato vectorial; sin embargo los dispositivos de salida (*output*) trabajan única y exclusivamente en formato mapa de bits (*bitmap*), lo que obliga a los sistemas operativos a realizar continuas traducciones entre un sistema y otro. Como ya sabemos en las traducciones siempre se pierde o se pervierte parte de la información, sin excepciones.

En el caso que nos ocupa, la tipografía en pantalla, hemos de añadir otra dificultad al proceso: la baja resolución de la mayoría de las pantallas. Siguiendo con la analogía, sería como si la traducción tuviera que hacerse por teléfono, y con interferencias o cortes.

A continuación describiré el proceso de rastreo (*rasterizing*) y los problemas que presenta, cómo los ingenieros de Adobe primero y de Apple después crearon un sistema para corregir las

Convergências: Revista de Investigaçã o e Ensino das Artes, n.º 9

desviaciones típicas del proceso: el *hinting*, y por último expondré cómo cambia el escenario con la irrupción de las pantallas de alta resolución. El objetivo es vislumbrar, a medio plazo, cómo va a afectar este cambio en los dispositivos de salida en el proceso de diseño y producción de una tipografía.

EL PROCESO DE RASTREO

Se coloca la imagen vectorial, y sobre ella la computadora coloca una rejilla, cuyo tamaño y densidad han sido calculados previamente. A continuación el RIP¹ comienza a rastrear o barrer la imagen horizontalmente, línea por línea, píxel a píxel, hasta detectar el vector; si más del 50% de la celda está ocupado por el perfil, se procede a rellenar de negro el píxel correspondiente. Para hacer el cálculo más rápido (pues el cálculo de superficies conlleva el empleo de mucha memoria), se dibuja el centro de cada píxel, y si está dentro del perfil, se deduce que este ocupa más del 50%, debido a que normalmente los perfiles son continuos, sin saltos ni interferencias. Cuando tiene la imagen completa, la transfiere a la pantalla o a la impresora.

Por debajo de 300 ppp y en los cuerpos más pequeños (ver imagen nº 1) la traducción comienza a presentar complicaciones: grosores de asta variables, deformaciones en curvas y diagonales, pérdida del contraste original, asimetría, etc.

9 puntos a 300 ppp

Begoñita

9 puntos a 600 ppp

Begoñita

6 puntos a 300 ppp

Begoñita

6 puntos a 600 ppp

Begoñita

4 puntos a 300 ppp

Begoñita

4 puntos a 600 ppp

Begoñita

Imagen 1: la palabra Begoñita compuesta en Adobe Jenson, a 4, 6 y 9 puntos respectivamente, escalados al mismo tamaño para comparar diferencias y con el anti-alias desactivado. Pantalla de baja resolución.

La explicación la tenemos en el proceso de escalado: las coordenadas de los vectores tienen por lo general valores enteros,² pero al multiplicarlos por un factor de escalado racional, por ejemplo

¹ Siglas de *Raster Image Processor*, es el programa que “pinta” las imágenes en cada dispositivo de salida.

² Al diseñar un nuevo alfabeto hay que tener cuidado y cumplir esta sencilla regla. Por esa razón los editores de tipos disponen de opciones como “ajustar los puntos a la rejilla” (Fontographer).

$x=a/b$, donde a y b son enteros también, el resultado es otro número racional, y para decidir qué píxeles encender o apagar hay que redondear la cifra obtenida, pues los píxeles no se pueden subdividir: se necesita una coordenada de números enteros. Este redondeo, a veces al alza, otras a la baja, es el principal cusante de que el escalado no sea lineal, y que por lo tanto exista el riesgo de que dos astas del mismo ancho se dibujen distintas —se pierde el ritmo y la simetría— o por el contrario que sutiles diferencias aparezcan exageradas brutalmente.

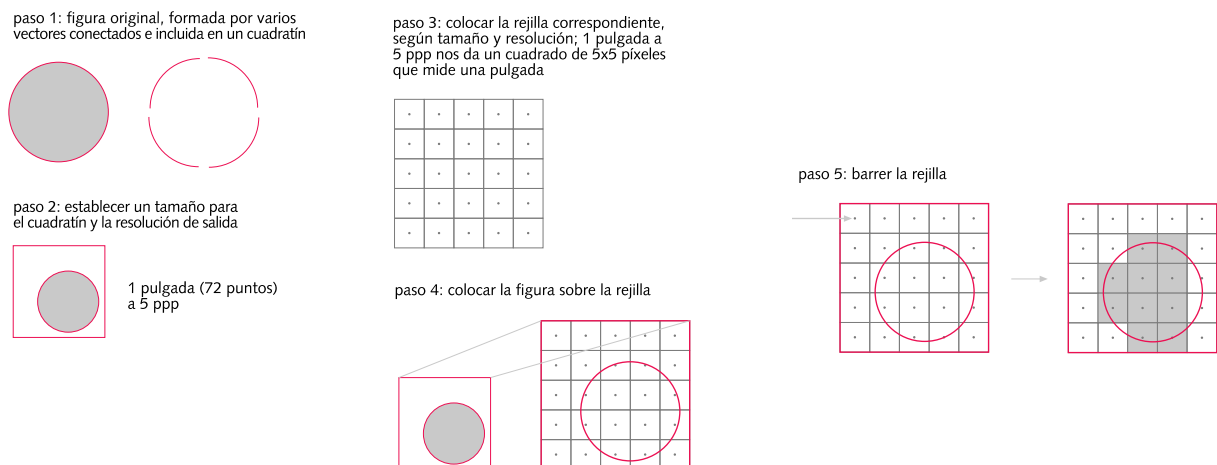


Imagen 2: el proceso de rastreo representado gráficamente

El rastreo se convierte en un problema en el momento que hay diversos dispositivos de salida con diferentes resoluciones y un formato vectorial que traducir a mapas de bit, porque se multiplica de forma exponencial el número de mapas posibles que construir.

Estos son los problemas habituales:

Desaparición de algún elemento sobre todo los nexos entre astas y algunos remates (imagen 3.a)

Pérdida de simetría (imagen 3.b)

Diagonales irregulares (imagen 3.c)

Desequilibrio en el grosor de astas y en los contrapunzones internos (imagen 3.d)

Exageración de los ajustes ópticos, como la diferencia de grosores en las astas de caja alta y baja, o la diferencia de altura entre figuras de forma rectangular, triangular y circular (imagen 3.e).

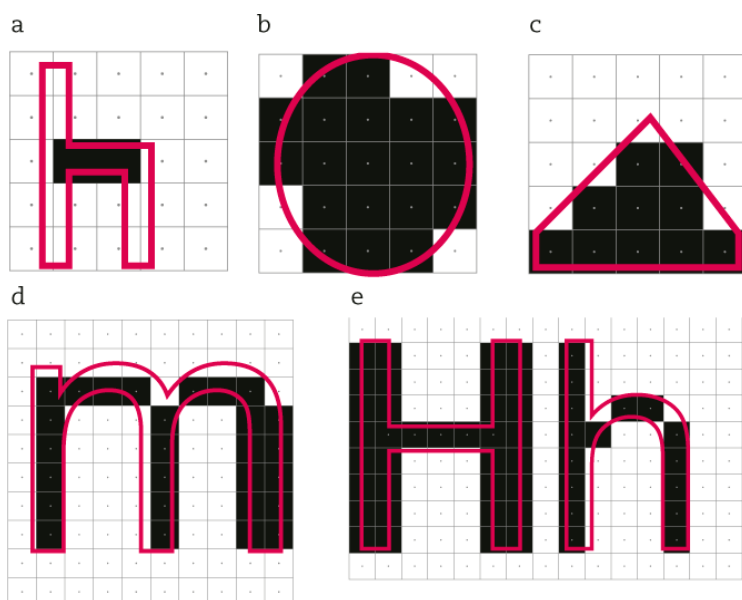


Imagen 3: muestras de los problemas habituales que se producen durante el rastreo

Dado que la linealidad o continuidad del escalado sólo es posible en un sistema analógico, en tipografía digital existen varias técnicas que persiguen el mismo objetivo: que el escalado parezca lineal, que se mantengan las características, regularidad y sutilezas del diseño original. De todas ellas nos interesa una en concreto:

EL PROCESO DE HINTING

Todos estos problemas, que aparecieron con la tipografía vectorial, se solucionan acompañando a cada letra de un pequeño programa o instrucción (*hint*) que permite deformar el perfil o la rejilla durante el rastreo y sólo durante el rastreo, de manera que se consigue construir mapas de bit adecuados al alfabeto en cuestión sin alterar el original —porque el original hace falta cuando se utiliza en grandes tamaños y altas resoluciones—.

Hay dos tipos de hinting: el PostScript y el TrueType. El primer sistema, original de Adobe, emplea una técnica llamada **deformación de la rejilla** (*grid warping*), patentada por Bitstream.

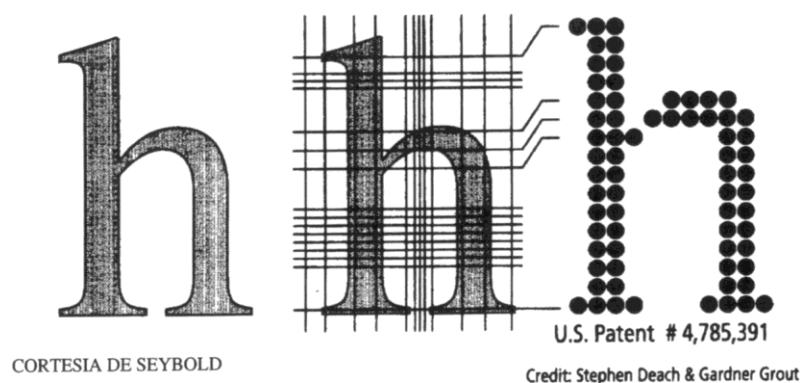


Imagen 4: esquema de funcionamiento del hinting PostScript

Deformar la rejilla permite efectuar compensaciones en bandas horizontales o verticales. Por ejemplo, forzar que dos astas tengan el mismo grosor. Aquí todo el trabajo lo hace el intérprete —si trabajaste con macs en la década de los noventa, recordarás el Adobe Type Manager (ATM)— por lo que se trata de un proceso relativamente rápido para el diseñador.

Si imaginamos que la fundición y el intérprete PostScript son personas como nosotros, su diálogo sería algo así como:

—Fundición (F): Oye, te paso la n de la Univers, tiene estas coordenadas.

—Intérprete (I): Vale, ¿a qué tamaño lo quieres? ¿y a qué resolución?

—F: Cuerpo 12, a 72 ppp

—I: Eso significa un PPC de 12 píxeles ¿cuántas UC tiene?

—F: 1.000, como siempre

—I: De acuerdo, ¿alguna instrucción en especial?

—F: Sí, el grosor de las astas principal es de 100 UC

—I: Muy bien, como el factor de escalado es 0,0012 (PPC/UC) las astas medirán 1,2 píxeles, redondeado a 1, aunque el dibujo diga otra cosa (un asta mide 97 UC y la otra 105 UC)

—F: Gracias

—I: A mandar, para eso estamos

Apple, por el contrario, decidió apostar por la mejora de la tecnología de las instrucciones invirtiendo el proceso, eliminando el papel del intérprete y creando un complejo lenguaje compilado que deforma una copia temporal del perfil, acomodándolo a la rejilla. De esa apuesta nació TrueType. La calidad de los tipos dependería a partir de entonces de cada diseñador, no del rastreador “inteligente”.

Si imaginamos que la fundición y el intérprete son personas como nosotros, su diálogo sería algo así como:

—Fundición (F): Oye, te paso la n de la Univers, tiene estas coordenadas.

—Intérprete (I): Vale, ¿a qué tamaño lo quieres? ¿y a qué resolución?

—F: Cuerpo 12, a 72 ppp

—I: Eso significa un PPC de 12 píxeles ¿cuántas UC tiene?

—F: 2048

—I: Muy bien, ve preparando el perfil acomodado para la rejilla que te he dicho, cuando lo tengas deformado me lo mandas.

—F: Vale, siempre igual, tengo que hacerlo todo yo sola

—I: Si tienes alguna queja, habla con Bill o con Steve...

—F: No hace falta, aquí la tienes, deformada

—I: Gracias, voy a rellenarla ahora mismo

—F: Bueno hasta la próxima, y recuerda, si encuentras algún agujero lo rellenas

—I: No te preocupes, lo haré gustoso; hasta otra

El proceso de hinting TrueType es mucho más laborioso y complejo, pues hay que efectuar todas las compensaciones para cada glifo y en varias resoluciones. Una locura. Como dijo Jelle Bosma, de Monotype, instruir en lenguaje TrueType es como jugar al Tetris... siempre puedes mejorar tu

puntuación, ¡pero nunca puedes ganar!.

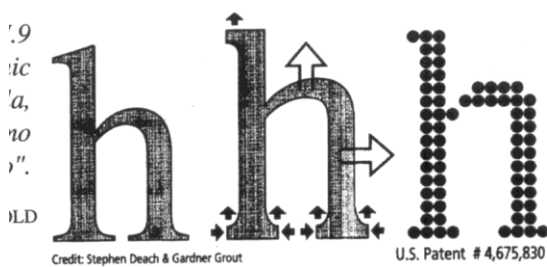


Imagen 5: esquema de funcionamiento del hinting TrueType

PANTALLAS DE ALTA RESOLUCIÓN

En junio de 2010 Steve Jobs presentó el iPhone 4 con la pantalla Retina como principal novedad: una superficie de 3,5 pulgadas con una densidad de 336 píxeles por pulgada. No se trataba de la primera pantalla de alta resolución, pero sí se convirtió en la primera en el mercado de masas en superar la barrera de los 300 ppp. El ojo humano es incapaz de distinguir los píxeles en una imagen de esa resolución, por lo que el equipo directivo de Apple decidió bautizarla como Retina y utilizarla también en su nuevo iPad, presentado en marzo de 2012. Esta vez se trata de una superficie de algo menos de 10 pulgadas (9,7) con una densidad de 264 ppp —2048×1536 píxeles; para hacernos una idea, una televisión de alta definición Full HD tiene 1920×1080 píxeles, un tercio menos—. El siguiente paso lógico es dar el salto a la gama alta de portátiles y equipos de sobremesa, lo que desatará una guerra por la alta resolución (Acer y Asus ya han anunciado portátiles con 1920×1080 píxeles para el primer semestre de este año) que tendrá que ir acompañada de un aumento de potencia en los procesadores gráficos.

El panorama que nos deja esta irrupción tecnológica es el siguiente: durante un tiempo viviremos una transición en la que los tipos que diseñemos se verán en pantallas de baja resolución, como las actuales, que rondan los 75/100 ppp, mientras aumenta el uso de pantallas de alta resolución, como las Retina de Apple, que rondan los 300 ppp. Cuando las últimas se conviertan en el estándar, los diseñadores de tipos se verán enfrentados a un cambio metodológico que afectará al proceso de hinting de sus creaciones, pues apenas existirán pantallas de baja resolución que deformen su trabajo —deformando las astas, cambiando los grosores, alterando la simetría de sus dibujos y cerrando o abriendo sus contragrafismos—.

El trabajo del hinting perderá su sentido o se centrará en los cuerpos más pequeños, los que tienen menos píxeles por cuadratín, y puede que modifique también el uso del formato OpenType TrueType, pues el hinting es uno de sus puntos fuertes.

Bajo este nuevo paradigma es preciso averiguar el alcance de las deformaciones de los glifos en las nuevas pantallas y determinar qué nuevos procesos de hinting serán necesarios. Para ello he planteado un pequeño experimento:

EL EXPERIMENTO

He creado un documento pdf que contiene una escala de cuerpos comprendidos entre 8 y 48 puntos compuestos en Verdana y Georgia —sin duda, las tipografías que mejor se ven en pantallas normales, de baja definición—. Cada escala aparece dos veces porque la columna de la izquierda está hecha con tipografía —utiliza hinting— y la de la derecha está trazada —no utiliza hinting.



Imagen 6: éste es el aspecto del documento; como aquí se muestra impreso —a partir de un archivo vectorial— no se aprecia diferencias entre ambas columnas.

A continuación cargué el documento en mi ordenador —un MacBook Pro de 13 pulgadas con Snow Leopard—, desactivé el anti-alias e hice una captura de pantalla al 100% de tamaño. El resultado es éste:

6 pt Begoñika
 8 pt Begoñika
 10 pt Begoñika
 12 pt Begoñika
 14 pt Begoñika
 16 pt Begoñika
 18 pt Begoñika
 24 pt Begoñika
 30 pt Begoñika
 48 pt Begoñika

6 pt Begoñika
 8 pt Begoñika
 10 pt Begoñika
 12 pt Begoñika
 14 pt Begoñika
 16 pt Begoñika
 18 pt Begoñika
 24 pt Begoñika
 30 pt Begoñika
 48 pt Begoñika

6 pt Begoñika
 8 pt Begoñika
 10 pt Begoñika
 12 pt Begoñika
 14 pt Begoñika
 16 pt Begoñika
 18 pt Begoñika
 24 pt Begoñika
 30 pt Begoñika
 48 pt Begoñika

6 pt Begoñika
 8 pt Begoñika
 10 pt Begoñika
 12 pt Begoñika
 14 pt Begoñika
 16 pt Begoñika
 18 pt Begoñika
 24 pt Begoñika
 30 pt Begoñika
 48 pt Begoñika

Imagen 7: captura de pantalla en baja resolución; recuerda que la columna de la izquierda utiliza hinting, la de la derecha no.

Como se puede apreciar en la columna de la izquierda, cuando el hinting está activado mejora muchísimo la visualización. Por debajo de 10 puntos no existe espacio suficiente para dibujar correctamente las letras y los contragrafismos se colapsan. En diseño web está prácticamente prohibido componer texto por debajo de 12, y el estándar es el cuerpo 16.

Está claro que el hinting es imprescindible para conseguir una visualización digna en pantallas normales, de baja resolución. Pero los programas tienen más herramienta para maquillar estos resultados: *anti-alias* y *sub-pixel rendering*. La primera es la más conocida, consiste en intercalar píxeles grises que desenfocan ligeramente el tipo dándole una apariencia más suave. La segunda

permite activar por separado cada uno de los tres componentes del píxel por separado, obteniendo el triple de precisión. Veamos el efecto aplicado en nuestra muestra:



Imagen 8: captura de pantalla en Acrobat Reader (baja resolución), con el *anti-alias* y el *sub-pixel rendering* activados. El segundo método sólo se utiliza en combinación con el hinting, por lo que sólo está actuando en la columna de la izquierda (si la publicación es en blanco y negro se verán los dos iguales).

Con las operaciones de maquillaje las formas mejoran en apariencia aunque molestan por debajo de 14 puntos (el desenfoque se hace evidente). Mejora especialmente la columna de la derecha, la que no tiene *hinting*. Si fuera posible impedir la desactivación del *anti-alias* podríamos incluso defender que el hinting no es demasiado necesario, y que con una herramienta automática sería suficiente. Pero no es el caso, y el resultado de la imagen 7 es inadmisibile.

Pero veamos ahora qué sucede en un iPad de tercera generación, donde no se puede desactivar el

Convergências: Revista de Investigaçã o e Ensino das Artes, n.º 9

anti-alias:

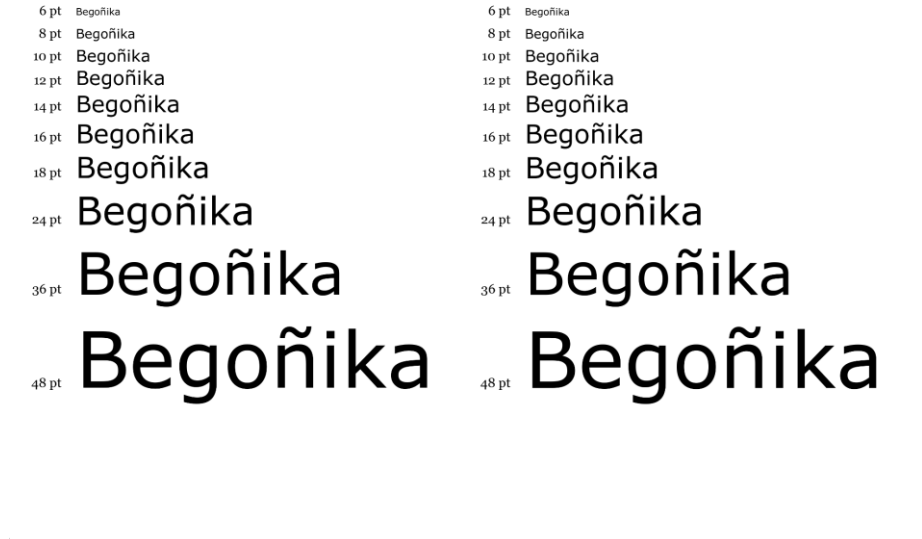


Imagen 9: captura de pantalla de alta resolución, con el anti-alias activado, de un pdf con las muestras anteriormente usadas (la columna de la izquierda es tipografía y por tanto utiliza hinting, la columna de la derecha no)

El resultado es totalmente sorprendente para mí. Confieso que esperaba peor comportamiento en cuerpos pequeños, pero el *anti-alias* a esa resolución compensa por completo el trabajo del *hinting*. Sólomente cuando los ordenadores reciban pantallas con esa densidad será necesario cierto nivel de *hinting* y sólo en ciertos cuerpos que apenas se usan en pantalla, donde todo lo que baje de 16 puntos es un cuerpo demasiado pequeño para la mayoría de los usuarios.

CONCLUSIONES

Este pequeño experimento demuestra que a medio plazo, si las nuevas pantallas de alta resolución se imponen, el hinting acabará siendo un rastro de la tecnología del pasado, un proceso lento, laborioso e innecesario para producir una fundición digital.

El siguiente paso es comprobar, en cuanto sea posible, el comportamiento en una pantalla de alta resolución conectada a un ordenador personal, donde se puede desactivar el *anti-alias*. De ahí saldrá el rango de cuerpos que necesiten *hinting* en alta resolución.

También habrá que comprobar si un proceso automático de *hinting* será suficiente para cubrir ese rango de cuerpos, con lo que crear tipografía será más fácil y accesible que nunca.

Por supuesto, y por último, será necesario medir hasta qué punto será necesario el formato OpenType TrueType en un escenario semejante, comprobando el rendimiento de cálculo de las curvas de uno y otro formato.

BIBLIOGRAFÍA

Haralambous, Yannis. *Fonts & Encodings*. O' Reilly: 2007.

Hertz, Jacky y Hersch, Roger D. «Towards a universal auto-hinting system for typographic shapes». *Electronic Publishing*, diciembre de 1994, vol. 7, núm. 4, pág. 251-260.

Horn, Berthold K.P. «Hinting of scalable outline fonts». *TUGboat*, 2201, vol. 0, núm. 0, pág. 1001-1002.

Rodríguez Valero, Daniel. *Tipografía Digital*. Valencia: Campgràfic, 2012.

Bil'ak, Peter. *Font Hinting*. Typotheque 2010 (en línea)

<<http://www.typotheque.com/articles/hinting>> [consultado el 3 de mayo de 2012]

Tennant, Bnonn. *16 PIXELS For Body Copy. Anything Less Is A Costly Mistake*. Smashing Magazine 2001 (en línea) <<http://www.smashingmagazine.com/2011/10/07/16-pixels-body-copy-anything-less-costly-mistake/>> [consultado el 3 de mayo de 2012]