

인쇄 관련업계 종사자를 위한 출력의 이해

1

인쇄라는 과정은 다양한 분야로 구성되어 있으며 서로 다른 분야에서 어떤 작업이 이루어지는지 모르는 경우가 대부분이다. 가장 보편적인 출판을 위한 종이 인쇄 과정은 크게 디자인(Design), 프리프레스(Prepress), 프레스(Press), 포스트프레스(Postpress)로 구분할 수 있다. 이 중에서 프리프레스(Prepress)는 인쇄될 내용(Contents)을 인쇄판에 기록하는 과정을 말한다.

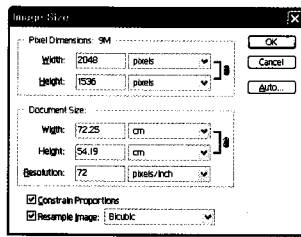
김호문 <칼리편집&출력>저자, 출판제작(편집&디자인) 모인 회장
fashionda@daum.net / 010-3807-2569

출력이란 인쇄용으로 작성한 내용을 인쇄에 사용하는 인쇄판에 기록하는 과정까지를 말한다. 인쇄용 데이터는 인쇄용 필름에 기록한 뒤, 다시 인쇄판에 기록하거나, 직접 인쇄판에 기록하는 두가지 방법을 사용한다. 이전과는 달리 요즘은 인쇄용 데이터를 컴퓨터로 제어하는 경우가 많기 때문에 출력을 제대로 이해하기 위해서는 컴퓨터 관련 용어나 개념을 정확히 알고 있어야 한다.

Pixel과 Resolution

임의의 사각 격자를 만들고 각각의 격자에 특정한 색을 입혀 만들어진 요소(또는 이미지)를 래스터(Raster)라고 한다. 래스터를 구성하고 있는 각각의 격자를 픽셀(Pixel)이라고 하는데 픽셀은 Picture Element를 합쳐서 만든 단어이다. 컴퓨터용 화면 표시 장치인 모니터는 각각의 픽셀에 Red, Green, Blue에 해당하는 빛을 발산하여 색을 구현한다. 픽셀은 독립적인 존재이나 서로 붙어있고 매우 작기 때문에 착시 현상을 일으켜 사람의 눈에는 연속적인 것으로 보이게 된다. 픽셀 하나하나마다 색상 정보를 가지기 때문에 픽셀이 많을수록(해상도가 높을수록, 이미지가 커질수록) 차지하는 용량도 커진다. 픽셀은 이미지 파일안에 데이터의 형태로 저장되는 것으로 개수 정보만 있을 뿐 크기 정보는 가지고 있지 않다.

래스터 형식의 가장 큰 특징은 해상도라는 품질 개념을 가진다. 해상도는 단위 면적(인치)당 픽셀 개수인 ppi(pixel per inch)라는 용어로 표현된다. 이미지의 해상도가 높으면 모니터 색을 표현하는 최소 단위인 하나의 셀에서 픽셀을 축소하여(여러 개의 픽셀을 하나의 셀에) 표현한다. 반면 이미지의 해상도가 낮으면 픽셀을 확대하여(하나의 픽셀을 여러 개의 셀에) 표현한다.



포토샵에서 해상도 확인하기

bitmap

컴퓨터 모니터에서 색을 표현할 때 각각의 픽셀에 비트로 표현된 RGB값을 사용한다. 즉, 각각의 픽셀은 RGB값을 가질 수 있다. R, G, B에 얼마만큼의 비트값을 부여하느냐에 따라서 표현 가능한 색상 종류가 달라진다. 반면 1비트와 같이 색이 있고(on, 1) 없음(off, 0)으로 래스터 형식을 표현할 수 있다. 이처럼 1비트 색상값을 갖는 형식을 Bitmap(비트맵)이라고 한다. 흔히 그림을 비트맵 이미지라 표현하는데 래스터 이미지가 맞는 말이다. 래스터 형식안에 비트맵

형식이 포함된다.

vector와 object

벡터란 포인트를 지정하고, 공식에 의해 이들을 연결한 선으로 정의되는데, 점과 곡선을 이용해서 외곽선을 만들고, 그 내부를 규정된 색상이나 패턴으로 채우는 방식으로 이미지를 만들어 낸다. 벡터 형식을 구성하는 기본 단위를 오브젝트(Object)라고 한다. 즉, 하나 이상의 오브젝트가 모여 벡터 이미지를 만든다.

텍스트는 벡터 형식을 사용하는가?

래스터 형식을 사용하는가?

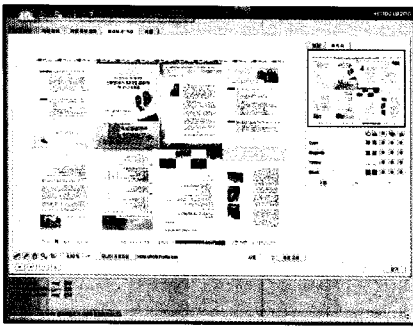
벡터 환경을 지원하는 페이지 레이아웃 소프트웨어(QuarkXPress, InDesign)이나 벡터 그래픽 소프트웨어(Illustrator, CorelDRAW)에서 입력된 텍스트는 벡터 형식을 사용하지만 래스터 그래픽 소프트웨어(Photoshop)에서 입력된 텍스트는 래스터 형식을 사용한다. 텍스트는 원칙적으로 벡터 형식이어야 한다. 물론 래스터 형식의 텍스트도 출력은 되지만 인쇄 품질은 벡터 형식에 비해서 떨어진다.

1.1 다양한 해상도

1) 다항식의 연산에 대한 성질
세 다항식 A, B, C에 대하여
① 교환법칙 : $A + B = B + A$, $AB = BA$
② 결합법칙 : $(A + B) + C = A + (B + C)$, $(AB)C = A(BC)$
③ 분배법칙 : $A(B + C) = AB + AC$, $(A + B)C = AC + BC$

2) 다항식의 나눗셈
다항식 A를 다항식 B(B ≠ 0)로 나누었을 때의 몫을 Q, 나머지를 R라 하면
 $A = BQ + R$

벡터 텍스트



래스터 텍스트

dot와 dpi

도트(dot)는 잉크가 묻는 최소 면적(점)을 가리킨다. 픽셀과 도트는 엄연히 다른 개념이며 하나의 픽셀이 하나의 도트로 치환되지는 않는다. 따라서 ppi와 dpi(dot per inch)는 다른 개념이지만 대부분 혼용하여 사용하는 경우가 많다.

rasterise

벡터 형식은 그대로 출력할 수 없으며, 출력 가능한 래스터 형식으로 변환해 주어야 한다. 이처럼 벡터형식으로 된 요소를 래스터 형식으로 변환하는 것을 rasterise(래스터라이즈)라고 한다. 벡터 그래픽 프로그램내에서 벡터 형식을 래스터 형식으로 변환하는 것은 object rendering(오브젝트 렌더링)이라고 한다. 래스터라이즈 작업을 거치게 되면, 벡터 요소가 1200ppi, 240ppi 등의 픽셀 형태로 변환된다.

출력의 표준 기술 - 포스트스크립트

포스트스크립트(PostScript)는 PDL(Page Description Language)의 일종이다. 즉, 출력(인쇄)되는 내용을 서술해 놓은 문장이라 할 수 있다. 포스트스크립트는 벡터 형식을 기반으로 한 이미지를 표현하는 Programming Language의 일종이다. 컴퓨터에서 작동되는 소프트웨어는 Basic, C/C++ 등과 같은 프로그램 언어를 사용하여 만든다. 뿐만 아니라 인터넷 브라우저(Browser) 상에서 작동하는 프로그램들도 이러한 프로그램 언어를 사용하여 만들어 낸다. 포스트스크립트는 인쇄를 위한 프로그램 언어라

고 이해하면 된다. 즉, 정해진 문법에 맞는 문장을 만들면(프로그래밍하면), 포스트스크립트 문장을 해석할 수 있는 출력기들을 이 문장들을 이용하여 출력물(이미지)를 만들어 낸다.

일반인에게 프로그래밍 작업은 생소하니 인터넷 홈페이지를 만드는 것을 예로 들어 보겠다. 홈페이지는 HTML이라는 프로그래밍 문법에 적합하게 만들어진 파일이다. 이 내용을 HTML 코드를 이해하고 해석할 수 있는 프로그램(Browser)들이 받아서 화면에 표시해주는 것이다. 요즘은 나모나 드림위버와 같은 편리한 툴을 이용하여 홈페이지(HTML 파일)를 만든다. 하지만, HTML 문법을 완벽히 이해하고 있다면, 한글 윈도우에서 제공하는 메모장과 같은 Text Editor만 가지고도 HTML 파일(홈페이지)을 만들 수 있다.

포스트스크립트도 마찬가지이다. 포스트스크립트 문법을 완벽하게 이해하고 있다면 Text Editor 만으로도 포스트스크립트 파일을 만들 수 있다. 하지만, 포스트스크립트 파일을 직접 만들기에는 기술적으로 모르는 내용이 많고, 특히 결과를 확인하기 곤란하기 때문에, 별도의 유틸리티 등을 이용하여 포스트스크립트 파일을 만들게 된다. 이처럼 포스트스크립트를 지원하는(그래픽) 프로그램을 사용하면, 해당 프로그램들이 디자인된 내용을 포스트스크립트 파일로 만들어 준다.

포스트스크립트 파일은 출력할 용도로 사용되므로(프린트 파일이므로) 대부분 파일 형태로 저장되지 않고, 인쇄 명령을 실행하는 순간 포스트스크립트 내용을 프린터를 비롯한 출력기로 전송하게 된다.

그러면 포스트스크립트를 지원하는 프린터가 내용을 해석하여 이미지를 만들어 낸다. 이처럼 포스트스크립트는 출력을 위하여 만들어진 컴퓨터 언어라고 보면 된다.

포스트스크립트를 지원한다는 의미

포스트스크립트를 지원한다는 의미는 소프트웨어(프로그램)와 하드웨어(프린터)에 따라 다르다. 소프트웨어에서 포스트스크립트를 지원한다는 것은 입력된 글자나 만들어진 그림과 같은 작업한 내용을 포스트스크립트 문법에 맞는 텍스트 파일로 만들 수 있다는 것을 말한다. 하드웨어에서 포스트스크립트를 지원한다는 것은 컴퓨터로부터 전달받는 포스트스크립트 형식의 데이터를 처리하여 출력 가능한 래스터 형태로 변환 가능하다는 것을 말한다. 따라서 포스트스크립트를 지원하는 하드웨어인 경우에는 PostScript Interpreter를 담당하는 연산장치가 내장되어 있어야 한다.

RIP(Raster Image Processing)

벡터 형식을 래스터 형식으로 변환하는 작업을 말한다. Rasterise라는 용어와 비슷한 개념이다. 즉, Rasterise하는 과정을 일컫는 말이기도 한다. 일반적으로 하드웨어에 포함된 PostScript Interpreter가 RIP의 역할을 수행한다. 리핑(Ripping)이라는 용어는 래스터라이즈 작업중이라는 것을 가리킨다.

Rasterising이라는 의미와 동일한데 이것보다는 리핑이라는 용어를 더 많이 사용한다.

RIP 프로그램

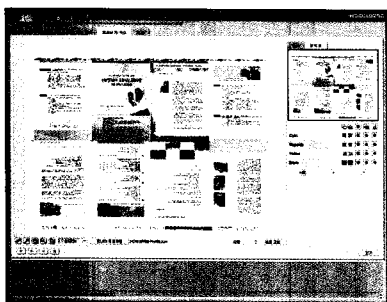
리핑 작업은 소프트웨어나 하드웨어를 이용하여 수행할 수 있다. 리핑된 결과물은 그냥 출력해도 되지만, 혹시 있을 오류를 대비하여 한번더 점검할 수 있다. 그러기 위해서는 리핑된 데이터를 래스터 형식의 이미지 파일로 저장해야 한다. RIP 프로그램은 소프트웨어 RIP의 일종으로 컴퓨터상에서 리핑을 해주고 그 결과를 확인할 수 있도록 해주는 역할을 한다. RIP 프로그램은 미리보기 기능이 있기 때문에 만들어진 데이터를 사전에 확인할 수 있다. 따라서 혹시 발생할지 모르는 각종 오류를 사전에 점검할 수

있으므로 출력 오류로 인한 종이(또는 필름)의 손실을 예방할 수 있다.

RIP 프로그램은 최적화된 PostScript Interpreter라고 이해하면 된다. 소프트웨어가 포스트스크립트를 지원한다고 해서 완벽한 것은 아니며 버그 등으로 인하여 오류가 발생할 수 있다. 이처럼 포스트스크립트를 지원 프로그램에서 만든 포스트스크립트 내용을 출력전에 한번 더 점검하는 역할을 하는 것이 RIP 프로그램이다. 만약 포스트스크립트를 지원하는 프로그램에서는 문제가 없어도 RIP 프로그램에서 문제가 발생한다면 원하는 출력 결과를 얻을 수 없다. 이 경우에는 오류 원인을 찾아내어 작업 과정에 반영해야 한다. (경우에 따라서는 오류를 수정한다고 해서 포스트스크립트 지원 프로그램에서 보이는 내용에는 변화가 없을 수 있다.)

RIP 프로그램은 상당히 많은 데이터를 처리해야 하기 때문에 설치되어 있는 컴퓨터가 고성능이어야 한다. 대부분 CPU를 두개 이상 장착 가능한 서버급 컴퓨터에 장착되어 있다.

RIP 프로그램은 별도로 구매할 수는 있지만, 대부분 필름 출력기와 같은 고해상도 출력기와 함께 판매되는 경우가 대부분이다. RIP 프로그램은 핵심 엔진의 제조회사에 따라 구분되는데 국내에서는 할로퀸(Halroquin)사, 어도비(Adobe)사 제품을 많이 사용한다.



하이델베르크의 RIP 프로그램인 Prinect Meta Dimension

포스트스크립트 파일/프린트 파일

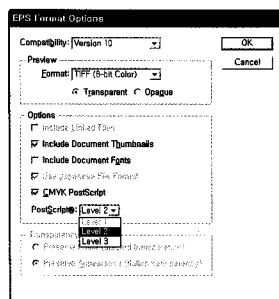
이것은 포스트스크립트 형식으로 된 파일을 말한다. 포스트스크립트를 지원하는 프로그램에서 만든 편집물을 출력하면 해당 내용을 포스트스크립트로 변

경하는 동시에 포스트스크립트를 지원하는 프린터로 전송하게 된다. 포스트스크립트 파일은 프린터로 전송되는 데이터를 컴퓨터안에 파일로 저장해 둔 것을 말하는데, *.PS나 *.PRN이라는 확장자를 사용한다.

이렇게 만들어진 포스트스크립트 파일은 해당 파일을 만든 원본 데이터 파일과는 독립적인 존재로 특정 소프트웨어가 없이 포스트스크립트를 지원하는 출력기로 출력할 수 있다는 장점이 있다. 포스트스크립트 파일은 과거에는 프린터로 전송되는 데이터를 남아채는 방식으로 파일을 만들었으나 요즘은 일부 프로그램의 경우에는 메뉴에서 제공하는 다른 이름으로 저장(Save As) 명령으로도 만들 수 있다. 포스트스크립트 파일은 그 용량이 너무 크기 때문에 잘 활용되지 않고, 페이지 개념이 없는 EPS 파일을 주로 사용한다.

EPS(Encapsulated PostScript)

벡터 형식으로 구성되는 포스트스크립트의 특성을 이용하여 포스트스크립트 형식의 텍스트 파일을 그림 파일로 사용할 수 있도록 만든 파일 형식이다. 포스트스크립트 파일은 텍스트 형식으로 되어 있기 때문에 이미지로 표현하기 위해서는 RIP 기능이 필요하다. EPS 파일은 그 형식에 래스터 형식의 미리보기 이미지를 포함시켜 포스트스크립트 파일을 래스터라이즈시키지 않아도 모니터상에서 볼 수 있도록 만든 파일 형식이다. EPS는 이미지로 사용되는 경우가 많기 때문에 포스트스크립트 파일과는 달리 페이지 개념이 없다. 즉 하나의 파일에 하나의 이미지(또는 한 페이지 내용)만 저장할 수 있다.



EPS 저장 옵션과 포스트스크립트 버전

일반적인 필름 출력 해상도와 선수

인쇄용 필름을 출력하고자 할 때 1도 인쇄물인 경우에는 1200dpi, 2도 인쇄물인 경우에는 1800dpi, 4도 인쇄물인 경우에는 2400dpi의 비트맵 이미지로 변환한다. 인쇄 도수에 따라 출력 품질이 달라지는 것은 인쇄에 사용되는 종이질 때문이다.

대부분 컬러(4도) 인쇄물에서는 인쇄 적성이 좋은 고급 종이를 사용하기 때문에 도수가 높을수록 출력 해상도도 높아진다. 만들어진 비트맵 이미지는 필름에 기록되는 과정에서 스크리닝(Screening) 기법을 사용하여 선수(lpi, line per inch) 단위로 변환된다. 선수는 망점(halftone dot)을 표현하기 위하여 사용하는 개념으로, 16dot를 1lpi로 변환하는 것이 가장 이상적이지만 여러 가지 사항을 고려하여 1도 인쇄물은 133선, 2도 인쇄물은 150선, 4도 인쇄물은 175선으로 변환하여 출력한다.

분판(Separation)

출력에 있어서 가장 중요한 두가지 요소는 리핑(Ripping)과 분판(Separation)이다. 일반적인 오프셋 인쇄 방식에서는 Cyan, Magenta, Yellow, Black이라는 프로세스 컬러(Process Color)를 사용한다. 분판이란, 인쇄용으로 작성된 내용을 색별로 분리해 내는 작업을 말한다. 분판 작업은 컬러(4도)로 인쇄할 때 사용하며 흑백(1도) 인쇄 작업에서는 모든 내용을 검정색 계조(Gradation)로 변환된다. 인쇄/출판용 프로그램들은 CMYK 색상 형식을 지원하므로 분판 과정은 복잡한 과정은 아니다. 다만 CMYK가 아닌 다른 색상 형식을 사용한 경우에는 정해진 규칙에 따라서 자동으로 CMYK 색상 형식으로 변경된다.

리핑된 결과는 1비트 형식의 이미지이므로 색을 갖지 않는다. 따라서 실제로 필름이나 인쇄판에 기록될 내용들은 다음과 같이 흑백 계조를 갖는다.



분판 전 원래 이미지



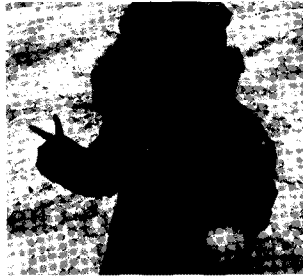
분판된 이미지-Magenta



분판된 이미지-Black



흑백 계조로 변환된 이미지



필름에 기록되는 내용-Magenta



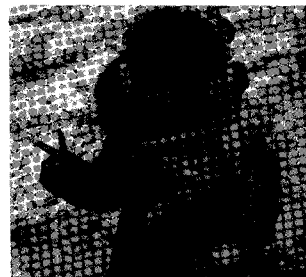
필름에 기록되는 내용-Black



분판된 이미지-Cyan



분판된 이미지-Yellow



필름에 기록되는 내용-Cyan



필름에 기록되는 내용-Yellow

A4로 편집된 내용의 출력용파일 크기는?

DTP 프로그램으로 만들어진 인쇄용 데이터는 최종적으로 래스터라이즈된 이미지 파일로 만들어져 매체(인쇄용 필름)에 기록된다. 따라서 출력용 파일의 크기를 예상할 수 있다. 가장 많이 사용하는 A4 크기(210mm×297mm)의 종이 면적은 623.7cm²며 해상도의 단위인 dpi(dot per inch)와 비교하기 위하여 인치로 환산하면 623.7/2.54=245.55 inch²가 된다. dpi는 인치당 점의 수를 의미하므로 면적인 경우에는 dpi값을 두 번 곱해야 한다. dot당 bit수를 계산하면 해상도에 따른 출력용 파일의 크기를 예상할 수 있다. 1도(흑백) 편집물이면 dot당 1bit로, 4도(컬러) 편집물이면 dot당 4bit로 계산하면 된다. 파일의 크기는 byte 단위이므로 최종 값을 8로 나눠야 한다(1byte=8bit).

이 계산 결과는 이론상의 최대 크기로 출력에 사용되는 1bit tif 파일의 크기와 동일하다. A4 크기로 편집된 내용을 2400dpi 해상도의 4도 출력용 파일로 만들면 약 3GB 크기가 된다. 출력물이 A4 한 장짜리인 것은 거의 없으므로 실제로 출력 과정에서 다루어야 하는 데이터의 양이 엄청나게 크다는 것을 알 수 있다. ☞

출력 해상도	파일 크기	출력 해상도	파일 크기
300dpi 1도	2.762MB	300dpi 4도	11.049MB
600dpi 1도	11.049MB	600dpi 4도	44.199MB
1200dpi 1도	44.199MB	1200dpi 4도	176.796MB
1800dpi 1도	176.796MB	1800dpi 4도	707.184MB
2400dpi 1도	707.184MB	2400dpi 4도	2828.736MB