CONTENT-BASED TRANSCODING OF IMAGES IN THE INTERNET

John R. Smith, Rakesh Mohan and Chung-Sheng Li

IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
{jrsmith,rakesh,csli}@watson.ibm.com

ABSTRACT

We present a system for transcoding images in the Internet in order to improve their delivery to client devices with a wide range of communication, processing, storage and display capabilities. The content-based image transcoder analyzes the images in order to classify them into image type and image purpose classes. The system then utilizes transcoding policies based on the content classes to manipulate and transcode the images. We describe the image transcoding process for a variety of client devices, including PDAs, HHCs, TV browsers and color PCs, and demonstrate improvements in delivery speed and accessibility of the images in the Internet.

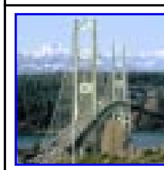
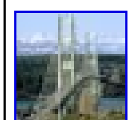
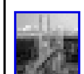
Workstation	Color PC	TV browser	HHC	PDA	Smart Phone
					"bridge"
size: 256 x 256	192 x 192	128 x 128	96 x 96	64 x 64	-
fidelity: 38 KB	23 KB	8 KB	4 KB	0.6 KB	100 B
color: 24 bit RGB	24 bit RGB	256 colors	4 bit gray	B/W	-

Figure 1: Image transcoding modifies the images along the dimensions of size, fidelity and color in order to adapt them to the client devices.

1. INTRODUCTION

The diversity of Internet client devices is growing at a rapid pace. However, the capabilities of these devices to access, process and display Internet content varies widely. Color workstations with high bandwidth Internet connections can readily access and display large colorful images. However, many hand-held computers (HHCs), PDAs, screen phones and smart phones can display only small images. Television-based Web browsers are constrained by the low-resolution interlaced display of the television screens. Even the color personal computers that make up the majority of Internet client devices typically have slow access to image-rich content. Given the variety of client devices, it is difficult for Internet content publishers to accommodate the wide spectrum of client capabilities.

In order to improve the accessibility of images in the Internet, we are developing a network-based Internet content transcoding system that analyzes, manipulates and transcodes images on-the-fly [1]. As depicted in Figure 1, the images are modified along the dimensions of image size, fidelity, and color in order to adapt them to the client devices. The transcoding policies are exercised according to the capabilities of the client devices and the results of image content analysis. In particular, the system classifies the images into image type and image purpose classes in order choose among the transcoding functions.

1.1. Related work

There is a growing need for processing content in the network in order to improve its accessibility. Recent transcoding efforts have focussed on compressing and caching images in the Internet in order to reduce the data transmission and speed-up delivery. Fox, et al.., developed a system for compressing Internet content lossily at a proxy in order to deal with client variability and improve end-to-end performance [2]. Ortega, et al., investigated a new image caching policy that reduces the resolution of infrequently accessed images in order to conserve storage space and bandwidth [3]. Several commercial systems such as Intel's Quick Web [4] and Spyglass' Prism [5] compress the images at the Internet service providers' proxy to speed-up download time.

We develop a more powerful image transcoding system that analyzes the images, the related text and Web document context in order to select policies for adapting the images to the client devices. The system transcodes the images along the dimensions of size, fidelity, and color in order to better adapt them to the client device's communication, processing, storage, and display capabilities.

1.2. Outline

In this paper, we present the content-based image transcoder system. In Section 2, we present the image content analysis system that classifies the images into image type and purpose classes. In Section 3, we describe the image transcoding functions and policies. Finally, in Section 4, we examine the potential improvement in accessibility of images for a growing diversity of client devices. We also demonstrate the potential for end-to-end speed up in image access via a network-based image transcoding proxy.

2. IMAGE CONTENT ANALYSIS

The image analysis system classifies the images based on their content into image type and purpose classes. We define the following image type classes: $\mathcal{T}$ = {BWG, BWP, GRG, GRP, SCG, CCG, and CP}, where

1. BWG – b/w graphic
2. BWP – b/w photo
3. GRG – gray graphic
4. GRP – gray photo
5. SCG – simple color graphic
6. CCG – complex color graphic
7. CP – color photo

The graphics *vs.* photographs distinction is loosely targeted for distinguishing between synthetic and natural images. Although the distinction is not always clear for images on the Web [6], the incentive for distinguishing between them is to use transcoding functions that are separately tuned for handling these types. We define also the following image purpose classes $\mathcal{P}$ = {ADV, DEC, BUL, RUL, MAP, INF, NAV, CON}, where

1. ADV – advertisement, i.e., banner ads
2. DEC – decoration, i.e., background textures
3. BUL – bullets, points, balls, dots
4. RUL – rules, lines, separators
5. MAP – maps, i.e., images with click focus
6. INF – information, i.e., icons, logos, mastheads
7. NAV – navigation, i.e., arrows
8. CON – content related, i.e., news photos

We also map the images into subject classes using related text. The semantic information potentially provides substitute text for the images for client devices that cannot handle images.

2.1. Image type classification

The image type classification system utilizes a decision tree classifier. The decision tree, depicted in Figure 2, classifies the images along the dimensions of color content (color, gray, b/w), and source (photographs, graphics). Distinguishing between b/w, gray and color is often not trivial because of artifacts introduced in the image production and compression. Examples of the seven image type classes are illustrated at the bottom of Figure 2.

The image type decision tree consists of five decision points, each of which utilizes a set of features extracted from the images. Keeping in mind the need for real-time, on-line transcoding, the features are extracted only as needed for the tests in order to minimize processing. The image features are derived from several color and texture measures computed from the images. We obtained the classification parameters for these measures from a training set of 1,282 images retrieved from the Web.

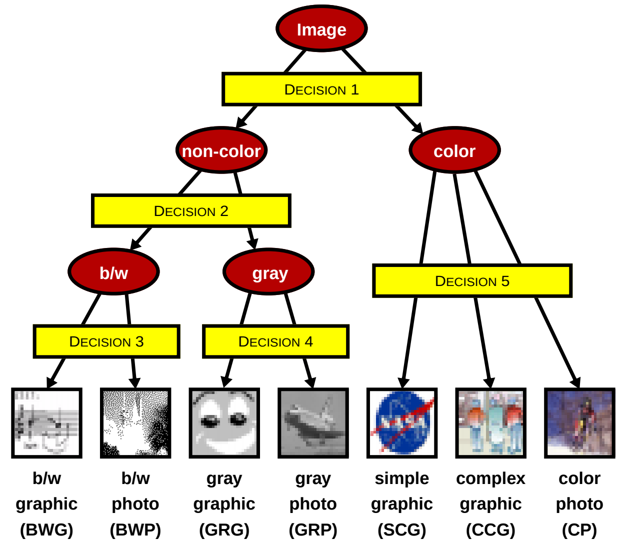


Figure 2: Image type decision tree consisting of five decision points for classifying the images into image type classes.

Each image $X[m,n]$ has three color components, corresponding to the RGB color channels as follows: $X_{rgb} = (x_r, x_g, x_b)$, where $x_r, x_g, x_b \in \{0, 255\}$. The decision tree performs the following tests for each image X:

1. **Color** *vs.* **non-color**. The first test distinguishes between color and non-color images using the measure of the mean saturation per pixel μ_s. The saturation channel y_s of the image is computed from X as follows:

$$y_s = \max(x_r, x_g, x_b) - \min(x_r, x_g, x_b).$$

Then, $\mu_s = \frac{1}{MN}\sum_{m,n} y_s[m,n]$ gives the mean saturation, where M, N are the image width and height, respectively. Table 1 shows the mean $E(\mu_s)$ and standard deviation $\sigma(\mu_s)$ of the saturation measure for the set of 1,282 images. The mean saturation μ_s discriminates well between color and non-color images since the presence of color requires $\mu_s > 0$, while strictly non-color images have $\mu_s = 0$. However, due to noise, a small number of saturated colors often appear in non-color images. For example, for the 464 non-color images, $E(\mu_s) = 2.0$.

Test 1	#	$E(\mu_s)$	$\sigma(\mu_s)$
Non-color	464	2.0	5.6
Color	818	63.0	46.2

Table 1: The color *vs.* non-color test uses mean saturation per pixel μ_s.

2. **B/W** *vs.* **Gray**. The second test distinguishes between b/w and gray images using the entropy P_v and variance V_v of the intensity channel y_v. The intensity channel of the image is computed as follows:

$y_v = 0.3x_r + 0.6x_g + 0.1x_b$. Then, the intensity entropy is given by $P_v = -\sum_{k=0}^{255} p[k] \log_2 p[k]$, where

$$p[k] = \frac{1}{MN} \sum_{m,n} \begin{cases} 1 & \text{if } k = y_v[m,n] \\ 0 & \text{otherwise.} \end{cases}$$

The intensity variance is given by

$$V_v = \frac{1}{MN} \sum_{m,n} (y_v[m,n] - \mu_v)^2,$$

where $\mu_v = \frac{1}{MN} \sum_{m,n} y_v[m,n]$. Table 2 shows the statistics of P_v and V_v for 464 non-color images. We can see that for b/w images the expected entropy P_v is low and expected variance V_v is high. The reverse is true for gray images.

Test 2	#	$E(P_v)$	$\sigma(P_v)$	$E(V_v)$	$\sigma(V_v)$
B/W	300	1.4	1.1	11,644	4,993
Gray	164	4.8	2.1	4,196	2,256

Table 2: The b/w *vs.* gray test uses intensity entropy P_v and variance V_v.

3. **BWG** *vs.* **BWP**. The third test distinguishes between b/w graphics and b/w photos using the minimum of the mean number of intensity switches in horizontal and vertical scans of the image. The mean number of intensity switches in the horizontal direction μ^h_{sw} is defined by

$$\mu^h_{sw} = \frac{1}{MN} \sum_{m,n} \begin{cases} 1 & \text{if } y_v[m-1,n] \neq y_v[m,n] \\ 0 & \text{otherwise.} \end{cases}$$

The vertical switches μ^v_{sw} are defined similarly from the transposed image y'_v. Then, the intensity switch measure is given by $W_v = \min(\mu^h_{sw}, \mu^v_{sw})$.

4. **GRG** *vs.* **GRP**. The fourth test distinguishes between gray graphics and gray photos using the intensity switch measure W_v and intensity entropy P_v. Table 3 shows the mean $E(W_v)$ and standard deviation $\sigma(W_v)$ of the intensity switch measure for 300 b/w and 164 gray images. The switch measure distinguishes well between b/w graphics and photos since it typically has a much lower value for b/w graphics. The gray graphics are found to have a lower switch measure and lower entropy than the gray photos.

Test 3	#	$E(W_v)$	$\sigma(W_v)$	-	-
BWG	90	0.09	0.07	-	-
BWP	210	0.47	0.14	-	-
Test 4	#	$E(W_v)$	$\sigma(W_v)$	$E(P_v)$	$\sigma(P_v)$
GRG	80	0.40	0.26	3.3	1.8
GRP	84	0.81	0.16	6.1	1.4

Table 3: The BWG *vs.* BWP test uses intensity switches W_v. The GRG *vs.* GRP uses W_v and intensity entropy P_v.

5. **SCG** *vs.* **CCG** *vs.* **CP**. The fifth test distinguishes between simple color graphics, complex color graphics and color photos. The images are transformed to HSV and vector quantized, as described in [7]. The process generates a 166-HSV color representation of the image y_{166}, where each pixel refers to an index in the HSV color look-up table.

Test 5	SCG	CCG	CP
#	492	116	210
$E(\mu_s)$	69.7	71.2	42.5
$\sigma(\mu_s)$	50.8	46.2	23.5
$E(P_{166})$	2.1	3.1	3.3
$\sigma(P_{166})$	0.8	1.0	0.7
$E(W_{166})$	0.24	0.36	0.38
$\sigma(W_{166})$	0.16	0.16	0.15

Table 4: The SCG *vs.* CCG *vs.* CP test uses mean saturation μ_s, HSV entropy P_{166} and HSV switches W_{166}.

We use the 166-HSV color entropy P_{166} and mean color switch per pixel W_{166} measures. In the computation of the 166-HSV color entropy, $p[k]$ gives the frequency of pixels with color index value k. The color switch measure is defined as in the test three measure, except that it is extracted from the 166-HSV color image y_{166}. We use also the measure of mean saturation per pixel μ_s. Table 4 shows the statistics for μ_s, P_{166}, and W_{166} for 818 color images. Color graphics have a higher expected saturation $E(\mu_s)$ than color photos. But, color photos and complex color graphics have higher expected entropies $E(P_{166})$ and switch measures $E(W_{166})$ in the quantized HSV color space.

2.2. Image purpose classification

Web documents often contain information related to each image that can be used to infer information about them [8, 9]. The system uses this information with the image type to classify the images into the image purpose classes $\mathcal{P}$. The system makes use of five contexts for the images in the Web documents: $\mathcal{C}$ = {BAK, INL, ISM, REF, LIN}, defined in terms of HTML code as follows:

1. BAK – background, i.e., <body backgr=...>
2. INL – inline, i.e., <img src=...>
3. ISM – ismap, i.e., <img src=... ismap>
4. REF – referenced, i.e.,
5. LIN – linked, i.e., <img src=...>

The system also uses a dictionary of terms extracted from the text related to the images. The terms are extracted from the 'alt' tag text, the image URL address strings, and the text nearby the images in the Web documents. The system makes use of terms such as $\mathcal{D}$ = {"ad", "texture", "bullet", "map", "logo", "icon"}. The system also extracts a number of image attributes, such as image width (w), height (h), and aspect ratio ($r = w/h$).

The system classifies the images into the purpose classes using a rule-based decision tree framework described in [10]. The rules map the values for image type $t \in \mathcal{T}$, context

$c \in \mathcal{C}$, terms $d \in \mathcal{D}$, and image attributes $a \in \{w, h, r\}$ into the purpose classes. The following examples illustrate some of the image purpose rules:

$$
\begin{aligned}
p = \text{ADV} &\leftarrow t = \text{SCG},\ c = \text{REF},\ d = \text{``ad''} \\
p = \text{DEC} &\leftarrow c = \text{BAK},\ d = \text{``texture''} \\
p = \text{MAP} &\leftarrow t = \text{SCG},\ c = \text{ISM},\ w > 256,\ h > 256 \\
p = \text{BUL} &\leftarrow t = \text{SCG},\ r > 0.9,\ r < 1.1,\ w < 12 \\
p = \text{RUL} &\leftarrow t = \text{SCG},\ r > 20, h\ < 12 \\
p = \text{INF} &\leftarrow t = \text{SCG},\ c = \text{INL},\ h < 96,\ w < 96
\end{aligned}
$$

2.3. Image summarizer

In order to provide feedback about the embedded images for text browsers, the system generates image summary information. The summary information contains the assigned image type and purpose, the Web document context, and related text. The system uses an image subject classification system that maps images into subjects categories (s) using key-terms (d), i.e., $d \rightarrow s$, which is described in [9]. The summary information is made available to the transcoding engine to allow the substitution of the image with text.

3. IMAGE TRANSCODING

The system transcodes the images using a set of transcoding policies. The policies apply the transcoding functions that are appropriate for the client devices.

3.1. Transcoding functions

The system provides a set of transcoding functions that manipulate the images along the dimensions of image size, fidelity, and color, and that substitute the images with text or HTML code. Example transcoding functions include

1. **Size**: minify, crop, and subsample.
2. **Fidelity**: JPEG compress, GIF compress, quantize, reduce resolution, enhance edges, contrast stretch, histogram equalize, gamma correct, smooth, sharpen, and de-noise.
3. **Color content**: reduce color, map to color table, convert to gray, convert to b/w, threshold, and dither.
4. **Substitution**: substitute attributes (a), text (d), type (t), purpose (p), and subject (s), and remove image.

3.2. Client device characteristics

The growing number of client devices that are gaining access to the Internet are varied in their communication, processing, storage and display capabilities. Table 5 illustrates some of the variability in device bandwidth, display size, display color and storage among devices

Since many devices are constrained in their capabilities, they cannot simply access image content as-is on the Internet. For example, many PDAs cannot handle JPEG images, regardless of size. The HHCs cannot easily display Web pages loaded with images because of screen size limitations. Color PCs often cannot access image content quickly over dial-up connections. The presence of fully saturated red or white images causes distortion on NTSC TV-browser displays. The transcoder framework allows the content providers to publish content at the highest fidelity, and the system manipulates the content to adapt to the unique characteristics of the devices.

Client device	Bandwidth (bps)	Display size	Display color	Device storage
PDA	14.4K	320 x 200	b/w	1MB
HHC	28.8K	640 x 480	gray	4MB
TV browser	56K	544 x 384	NTSC	1GB
Color PC	56K	1024 x 768	RGB	2-4GB
Workstation	10M	1280 x 1024	RGB	>4GB

Table 5: Summary of client device capabilities.

3.3. Transcoding policies

The transcoding system employs the transcoding functions in the transcoding policies. Consider the following example transcoding policies based upon image type and client device capabilities:

$$
\begin{aligned}
\text{minify}(X) &\leftarrow \text{type}(X) = \text{CP},\ \text{device} = \text{HHC} \\
\text{subsample}(X) &\leftarrow \text{type}(X) = \text{SCG},\ \text{device} = \text{HHC} \\
\text{dither}(X) &\leftarrow \text{type}(X) = \text{CP},\ \text{device} = \text{PDA} \\
\text{threshold}(X) &\leftarrow \text{type}(X) = \text{SCG},\ \text{device} = \text{PDA} \\
\text{JPEG}(X) &\leftarrow \text{type}(X) = \text{GRP},\ \text{bandwidth} \leq 28.8\text{K} \\
\text{GIF}(X) &\leftarrow \text{type}(X) = \text{GRG},\ \text{bandwidth} \leq 28.8\text{K}
\end{aligned}
$$

Notice that two methods of image size reduction are employed: minify and subsample. The difference is that minify performs anti-aliasing filtering and subsampling. Minifying graphics often generates false colors during filtering, which increases the size of the file. This can be avoided by subsampling directly. We also distinguish between graphics and photographs for compressing and reducing the color of the images. For compression, JPEG works well for gray photographs but not for graphics. For GIF, the reverse is true. When converting color images to b/w, dithering the photographs improves their appearance, while simply thresholding the graphics improves their readability. By performing the image type content analysis, the system is able to better select the appropriate transcoding functions.

The transcoding policies also make use of the image purpose analysis. Consider the following example transcoding policies:

$$
\begin{aligned}
\text{fullsize}(X) &\leftarrow \text{purpose}(X) = \text{MAP} \\
\text{remove}(X) &\leftarrow \text{purpose}(X) = \text{ADV} \\
&\ \text{bandwidth} \leq 14.4\text{K} \\
\text{substitute}(X, \text{``<li>''}) &\leftarrow \text{purpose}(X) = \text{BUL}, \\
&\ \text{device} = \text{PDA} \\
\text{substitute}(X, t) &\leftarrow \text{purpose}(X) = \text{INF}, \\
&\ \text{display size} = 320 \times 200
\end{aligned}
$$

The first policy makes sure that map images are not reduced in size in order to preserve the click focus translation. The second policy illustrates the removal of advertisement images if the bandwidth is low. The third policy substitutes the bullet images with the HTML code "<li>," which draws a bullet without requiring the image. A similar policy substitutes rule images with "<hr>". The last policy substitutes the information images, i.e., logos, icons, mastheads, with related text if the device screen is small.

3.4. Image transcoding proxy

The content-based image transcoder is part of a network-based transcoding proxy, see Figure 3. The transcoding proxy handles the requests from the client devices for Web documents and images. The proxy retrieves the documents and images, analyzes, manipulates and transcodes them, and delivers them to the devices.

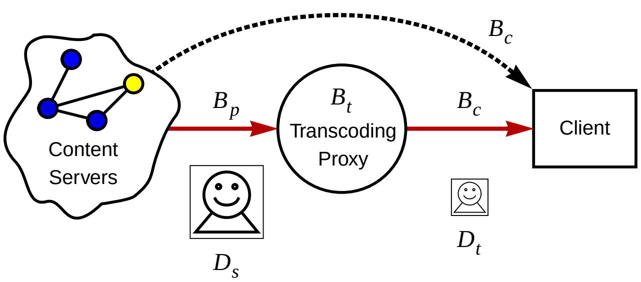


Figure 3: The content-based image transcoding proxy analyzes, manipulates and transcodes images, on-the-fly, to adapt them to the capabilities of the client devices.

4. EVALUATION

The transcoding proxy is designed to have a relatively high bandwidth between the proxy and the content server. In most cases, the proxy has a relatively low bandwidth to the client. Reducing the data sizes of the images at the transcoding proxy via image compression, size and color reduction can result in faster end-to-end delivery, even when accounting for the latencies introduced by the content analysis and transcoding.

The transcoding proxy system is illustrated in Figure 3, where B_p gives the proxy-to-server bandwidth, B_c gives the client-to-proxy bandwidth, and B_t gives the transcoder bandwidth. The terms D_s and D_t denote the data sizes of original and transcoded images, respectively. The latency in retrieving the image directly to the client is given by $L_c = D_s/B_c$. The latency in retrieving the image via the transcoding proxy is given by $L_t = D_s/B_p + D_s/B_t + D_t/B_c$. The transcoder results in a net speed-up by a factor $L_c/L_t \geq 1$ if the data compression ratio D_s/D_t is

$$\frac{D_s}{D_t} \geq \frac{B_p B_t}{B_p B_t - B_c B_t - B_p B_c}.$$

Consider a relatively high proxy-to-server bandwidth of $B_p = 1000$ Kbps, a client-to-proxy bandwidth of $B_c = 20$ Kbps, and a transcoder bandwidth of $B_p = 2400$ Kbps. A data compression ratio at the proxy of $D_s/D_t \geq 1.03$ results in a net end-to-end speed-up. If the data is compressed by a factor of $D_s/D_t = 8$, the speed-up is by a factor of $L_c/L_t \approx 6.5$. If $B_p = 50$ Kbps, the data compression ratio needs to be increased to $D_s/D_t \geq 1.8$ to have a speed-up in delivery. In this case, data compression of $D_s/D_t = 8$ speeds up delivery by a factor of $L_c/L_t \approx 1.9$.

5. SUMMARY

We presented a system for transcoding images in the Internet in order to adapt them to client devices with a wide range of communication, processing, storage and display capabilities. The content-based image transcoder analyzes the images and classifies them into image type and image purpose classes. The system then utilizes transcoding policies based on the content classes to manipulate, transcode, and adapt the images. The image transcoding system improves access of a variety of client devices, including PDAs, HHCs, TV browsers and color PCs to the images in the Internet.

6. REFERENCES

[1] J. R. Smith, R. Mohan, and C.-S. Li. Transcoding Internet content for heterogenous client devices. In *Proc. IEEE Inter. Symp. on Circuits and Syst. (ISCAS)*, June 1998. Special session on Next Generation Internet.

[2] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir. Adapting to network and client variability via on-demand dynamic distillation. In *ASPLOS-VII*, Cambridge, MA, October 1996.

[3] A. Ortega, F. Carignano, S. Ayer, and M. Vetterli. Soft caching: Web cache management techniques for images. In *Workshop on Multimedia Signal Processing*, pages 475 – 480, Princeton, NJ, June 1997. IEEE.

[4] Intel Quick Web. http://www.intel.com/quickweb.

[5] Spyglass-Prism. http://www.spyglass/products/prism.

[6] V. Athitsos, M. J. Swain, and C. Frankel. Distinguishing photographs and graphics on the World-Wide Web. In *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, June 1997.

[7] J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, volume 2670, pages 426 – 437, San Jose, CA, February 1996. IS&T/SPIE.

[8] N. C. Rowe and B. Frew. Finding photograph captions multimodally on the World Wide Web. Technical Report Code CS/Rp, Dept. of Computer Science, Naval Postgraduate School, 1997.

[9] J. R. Smith and S.-F. Chang. Visually searching the Web for content. *IEEE Multimedia Mag.*, 4(3):12 – 20, July–September 1997.

[10] S. Paek and J. R. Smith. Detecting image purpose in World-Wide Web documents. In *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology - Document Recognition*, San Jose, CA, January 1998.