内含 出席國際學術會議心得報告及發表之論文各一份

多媒體訊號處理(I)

Multimedia Signal Processing (I)

計畫編號：NSC89-2213-E-002-092

執行期限：88年8月1日至89年7月31日

主持人：貝蘇章　　台灣大學電機系教授

一、中文摘要

結合台大電機系在訊號處理，影像處理及大型積體電路設計方面的研究，以研究群方式組成，將多媒體訊號處理及壓縮技術作系統的整合。

二、計畫緣由與目的

繼HDTV之後，國科會成立第二期“數位視訊推動小組”鼓勵各校成立“研究群”，形成群體合作研究，並與工業界，產業界進行研究與建教合作。

三、研究成果

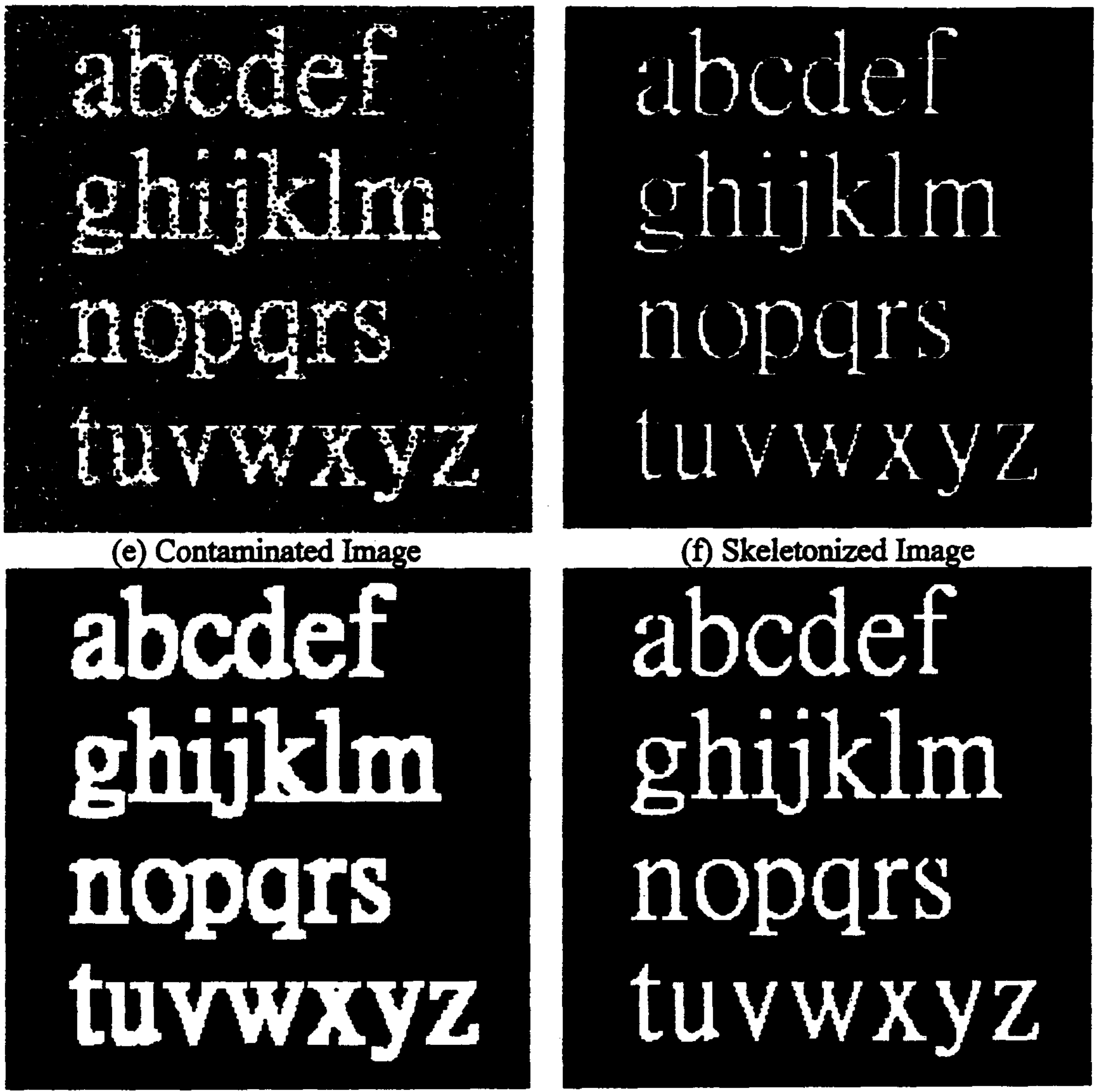
1. 開發有效率的音視訊數位浮水印技術。
2. 開發適用於多媒體的FIR filter副頻帶濾波器組理論與設計技術。
3. 研究高效率視訊編碼演算法及架構設計，並著手部分設計關鍵零組件IC。
4. 利用梯形架構和小波做有損耗及無損耗的影像壓縮。
5. 研發文字驅動或人臉動畫模擬技術，以其適用於極低位元率視訊會議。

四、結論與討論

所開發的“多媒體訊號”技術，理論模擬及硬體架構設計已完成，功能改進及系統整合仍待進一步研究。

五、參考文獻

本計畫研究成果所發表的10篇論文。



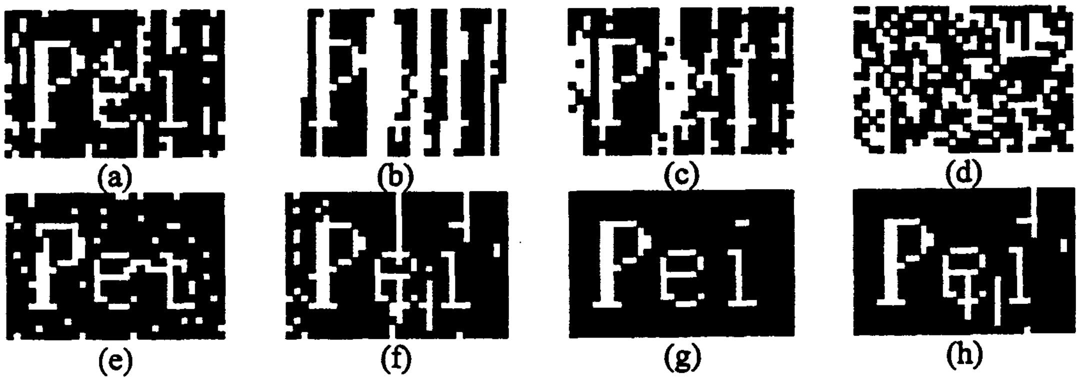
(e) Contaminated Image

(f) Skeletonized Image

(g) Dilated Image

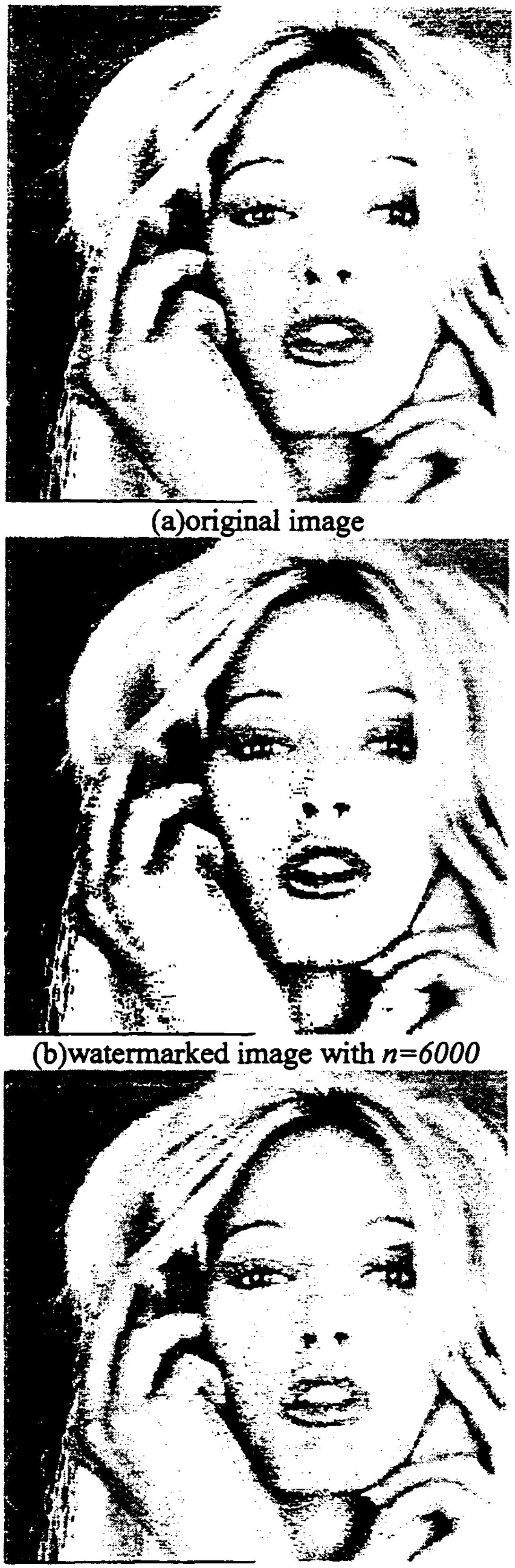
(h) Gaussian Blurred Image

Fig. 4-7 Attacked Watermarked Images


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. .4-8 Watermark Attracted from the Attacked Images

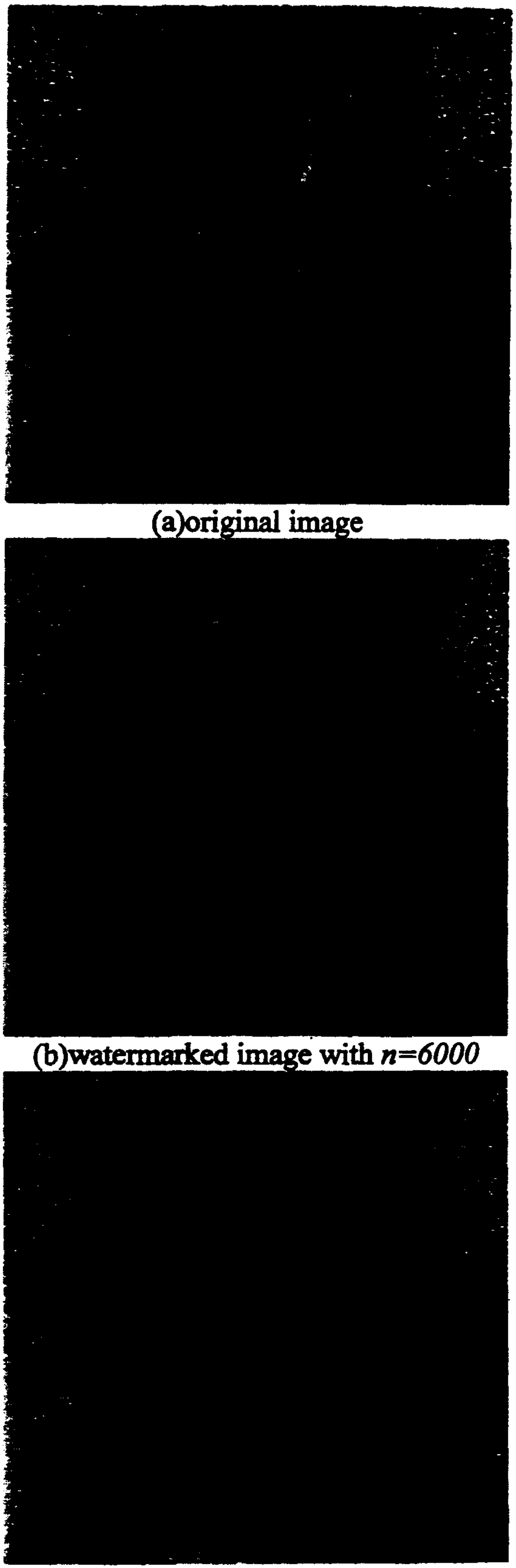


(a)original image

(b)watermarked image with *n=6000*

(c)watermarked image with *n=65536*

Fig. 3-5 The TIFFANY image



(a)original image

(b)watermarked image with *n=6000*

(c)watermarked image with *n=65536*

Fig. 3-6 The BABOON image

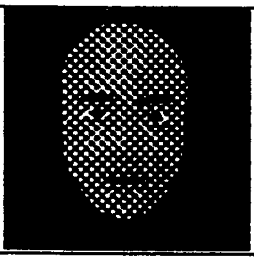
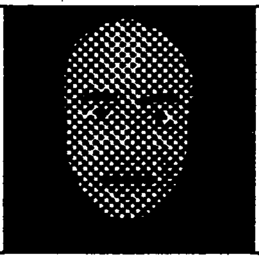
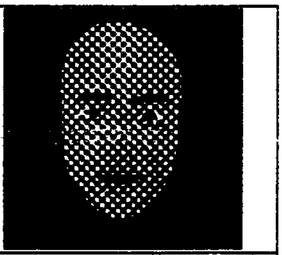
接著我們將以句子『中華民國』為例，說明如何產生國語音人臉。利用語音合成系統可產生如下的資料

	聲音編號	開始的時間 (0.1ms)	聲母長短 (0.1ms)	韻母長短 (0.1ms)
中	1374	40.202263	4.444161	13.236006
華	2301	0.285576	5.981871	13.953496
民	2277	-0.055003	6.937511	16.857878
國	2310	2.485189	1.890097	26.938824

『中』的聲音編號為 1374，其中 1 為國語音第一聲，
374 為注音符號ㄓㄨㄥ

經查詢得知　聲母為『ㄓ』，編號 16，視素為 v5
韻母為『ㄨㄥ』，編號 33，視素為 v14 +v11

因此中的臉部表情為

		
v5	**v4**	**v11**
0.4444161(ms)	1.3236006(ms)	

『華』的聲音編號為 2301，其中 2 為國語音第二聲，
301 為注音符號ㄏㄨㄚ

經查詢得知　聲母為『ㄏ』，編號 16，視素為 v3
韻母為『ㄨㄚ』，編號 26，視素為 v14+v7

因此中的臉部表情為

出席國際會議報告書

「2000國際電路與系統會議」(2000 IEEE Int'l Symposium on Circuits and Systems)假歐洲瑞士日內瓦市盛大舉行，筆者應邀出席該會，發表論文四篇，並獲取 IEEE Fellow Award 得獎證書，茲將參加會議的心得分述於下：

一、本會共有一仟五佰多位來自世界各國四十多個國家之學者、專家與會，約有943篇論文在大會發表，內容涵蓋電路、VLSI、CAD、數位訊號處理、影像壓縮與視訊處理、電力與系統、多媒體等五個內容非常豐富又具學術權威地位的重要會議。

二、此次大會的主辦單位為洛桑聯邦理工學院電機系及國際電機電子工程師學會電路與系統分會協

辦(IEEE Circuits and Systems Society)，其目的想藉此會交換各國最新之電路和系統之科技，並促進學術交流。

三. 日內瓦為瑞士的觀光及交流城市，會議地点為國際會議中心，交通便利，設備優良，是個理想的會議場所。

四. 會議的頭一天共有4個短期課程班(Tutorial)及Special的研討會舉行，讓與會者有再接受新科技之機會。大會的學術論文發表方式分成14個場地同時進行，另外有個較大的場地以海報方式舉行，並有書展及Special系統及產品展示。海報方式能和原作者親身相互討論及交換心得，效果十分良好。

P. 2

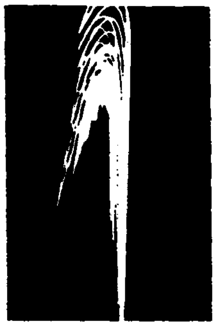
五大會在開幕式，邀請二位諾貝爾獎得主，發表學術專題演講，內容精彩，受益良多。此次大會本人受邀擔任數位控制器之Session Chair，並在星期二的晚宴接受IEEE Fellow Award的得獎證書，感到非常光榮。

六該七日內見的都市規劃及交通建設十分良好，還有的湖光山色及自然景觀令人留下深刻的印象。本人此次參加「教育部」補助旅費，得以順利成行，在此深表謝意，並攜帶大會論文集六冊，引進及吸收不少新知及科技發展，供國內參考，並達到學術交流目的。

台大電機系教授
貝蘇章 敬上
89/7/1

貝蘇章

P.3



ISCAS 2000 GENEVA

ISCAS 2000

THE 2000 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS

SUNDAY, MAY 28 — WEDNESDAY, MAY 31, 2000
INTERNATIONAL CONFERENCE CENTER OF GENEVA (CICG)
GENEVA, SWITZERLAND

CALL FOR PAPERS

IEEE

CAS

Organizing Committee

General Chair
Martin Hasler
Swiss Fed. Inst. of Techn., Lausanne
martin.hasler@epfl.ch

Vice Chair
George S. Moschytz
Swiss Fed. Inst. of Techn., Zürich
moschytz@isi.ee.ethz.ch

Technical Program Chair
Joos Vandewalle
Katholieke Univ. Leuven, Belgium
joos.vandewalle@esat.kuleuven.ac.be

Techn. Program Vice Chair
Georges Gielen
Katholieke Univ. Leuven, Belgium
georges.gielen@esat.kuleuven.ac.be

Special Sessions Chair
Maciej Ogorzalek
Univ. of Mining&Metallurgy, Krakow
maciej@zet.agh.edu.pl

Special Sessions Vice Chair
Eric Vittoz
CSEM, Neuchâtel
eric.vittoz@csem.ch

Tutorial Chair
Josef A. Nossek
University of Techn., Munich, Germany
nossek@nws.e-technik.tu-muenchen.de

Tutorial Vice Chair
Hans Peter Graf
AT&T Labs Research, Red Bank NJ
hpg@research.att.com

Publications Chair
Hervé Dedieu
Swiss Fed. Inst. of Techn., Lausanne
herve.dedieu@epfl.ch

Publicity Chair
Martin Hänggi
Swiss Fed. Inst. of Techn., Zürich
haenggi@isi.ee.ethz.ch

Exhibit Chair
Christian Enz
CSEM, Neuchâtel, Switzerland
christian.enz@csem.ch

Local Arrangements Chair
André Stauffer
Swiss Fed. Inst. of Techn., Lausanne
andre.stauffer@epfl.ch

Registration Chair
Christiane Good
Swiss Fed. Inst. of Techn., Lausanne
christiane.good@epfl.ch

Finance Chair
Jean-Louis Pfaeffli
Energie Ouest Suisse, Lausanne
elisabeth.schoenau@eos-gd.ch

Database Chair
Bertrand Dutoit
Swiss Fed. Inst. of Techn., Lausanne
bertrand.dutoit@epfl.ch

International Coordinators
Michael Peter Kennedy
University College Dublin, Ireland
peter.kennedy@ucd.ie
Shin'ichi Oishi
School of Science&Eng., Tokyo, Japan
oishi@oishi.info.waseda.ac.jp
Angel Rodríguez-Vázquez
Universidad de Sevilla, Spain
angel@cnm.us.es

Emerging Technologies for the 21st Century

The 2000 IEEE International Symposium on Circuits and Systems will be held in Geneva, Switzerland. The symposium is sponsored by the **IEEE Circuits and Systems Society**.

The symposium will include regular sessions on the topics listed below; special sessions on emerging circuits and systems topics; plenary sessions on selected advanced aspects of the theory, design and applications of circuits and systems, and short courses given by experts in specific state-of-the-art subject areas.

The symposium will be organized into parallel lecture sessions and poster sessions. Papers suitable for poster presentation are those that require interactive discussion; otherwise, poster and lecture presentations carry equal weight.

Prospective authors are invited to submit their papers reporting original work as well as tutorial overviews in all areas of circuits and systems.

Exciting new areas and problems for research in circuits and systems will be presented in special sessions.
Short courses will provide up-to-date knowledge on important topics.

Topics for regular sessions include, **but are not limited to**, the following:

1. **Analog Circuits and Signal Processing**
   - 1.1 Analog circuits and filters
   - 1.2 Switched capacitor/current techniques
   - 1.3 Analog and mixed signal processing
   - 1.4 Data conversion and Σ-Δ modulation
2. **General and Nonlinear Circuits and Systems**
   - 2.1 Linear and nonlinear circuits and systems
   - 2.2 Linear and nonlinear circuit theory
   - 2.3 Chaos, bifurcation and applications
   - 2.4 Distributed circuits and systems
3. **Digital Signal Processing**
   - 3.1 Digital filters and filter banks
   - 3.2 Wavelets and multirate signal processing
   - 3.3 Adaptive signal processing
   - 3.4 Multidimensional systems
   - 3.5 Fast computations for signal processing
4. **Multimedia and Communications**
   - 4.1 Speech processing and coding
   - 4.2 Image processing and coding
   - 4.3 Video and multimedia technology
   - 4.4 Signal processing for communications
   - 4.5 Computer communications
5. **VLSI Circuits and Systems**
   - 5.1 Analog and digital ICs
   - 5.2 Low power design
   - 5.3 VLSI physical design
   - 5.4 Testing: analog, digital and mixed
   - 5.5 High level synthesis & hardware/software codesign
   - 5.6 Logic synthesis and formal verification
   - 5.7 Fault tolerant systems
   - 5.8 Sensors and micromachining
6. **Computer Aided Design**
   - 6.1 Numerical and symbolic methods
   - 6.2 Linear and nonlinear optimization
   - 6.3 Graph theory, combinatorial optimization
   - 6.4 Modeling and simulation techniques
   - 6.5 CAD tools
7. **Neural Networks and Systems**
   - 7.1 Neural networks
   - 7.2 Cellular neural networks
   - 7.3 Fuzzy logic and circuits
   - 7.4 Learning and intelligent systems
8. **Power Systems and Industrial Applications**
   - 8.1 Sensors
   - 8.2 Robotics
   - 8.3 Banking and security systems
   - 8.4 Micromechatronics
   - 8.5 Power electronics and systems

Authors are invited to fill out the electronic form that enables the electronic paper submission at the address:

WWW address for submission: http://iscas.epfl.ch

The full four-page paper in double column format, including paper title, authors' names and affiliation, and short abstract is requested. Only electronic submissions (postscript format) are accepted. Those submitters who are unable to send in their contribution electronically are asked to contact the program chairs
(e-mail: iscas2000@esat.kuleuven.ac.be, fax +32/16/32.19.70).
Once accepted, authors will be asked to prepare the final four-page camera-ready paper for the symposium proceedings.

AUTHOR'S SCHEDULE

Deadline for Submission of Papers	October 1, 1999
Notification of Acceptance	December 23, 1999
Deadline for Submission of Camera-Ready Paper	January 31, 2000

Proposals for Special Sessions, Plenary Sessions, and half or full-day Short Courses may be submitted to the respective chair by September 15, 1999. Please contact them directly for further information. Check the symposium web site for up to date information: **http://iscas.epfl.ch**. Specific questions to the Organizing Committee members may be directed, as appropriate, to one of the listed e-mail addresses.

Limited Color Display for Compressed Video

Soo-Chang Pei, Ching-Min Cheng and Lung-Feng Ho

Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan, R. O. C. Email:pei@cc.ee.ntu.edu.tw

Abstract

Many display devices nowadays still allow a limited number of colors, called color palette, to be displayed simultaneously. Besides, images and videos in most world wide web (WWW) databases are in compressed formats. Therefore, it becomes an important issue to retrieve a suitable color palette from compressed domain in order to have fast and faithful color reproduction for these devices. In this paper, the color palette design methods for compressed videos are presented. The proposed approaches use the reduced image rather than the whole image for the color palette design to avoid the heavy computation in video decompression. Experimental results show that output image quality of proposed methods is acceptable to human eyes. In addition, empirical results show that the proposed shifting-window scheme can reduce the main problem of displaying quantized image sequences, screen flicker.

I. Introduction

With the prevalence of multimedia and internet, more and more digital images and videos are available for people to access. The digital image or video format is usually quantized with integer from 0 to 255 for each of three color components (e.g. red, green, blue). All possible combinations of three of these values gives 256^3 (or 16 million) distinct colors for full-color digital display. However, due to the costs of high speed video RAM, many current PCs and workstations generally have a single 8-bit frame buffer to allow only a limited number of colors, called color palette, to be simultaneously displayed. If an acceptable output image quality is desired, it is necessary to develop a useful procedure, called color quantization, for designing the color palette.

In the past, the color palette design focused on uncompressed data. For one single image, several color quantization algorithms have been proposed. Heckbert has proposed a median cut (MC) algorithm[1] where the color space is recursively divided into M rectangular regions with equal color occurrence and their centroids being representation colors. Recently, the popular vector quantization (VQ) technique[4] is applied to the color palette design, too. The colors in input image are used as training vectors in order to refine the initial rough palette. But VQ-type algorithms are usually computational intensive such that they are not suitable for real-time processing. Pei and Cheng have suggested a dependent scalar quantization (DSQ) algorithm[2], which exploits dependency of input colors and sequentially partitions the color space. The experimental results show that the DSQ can reduce the computation complexity and its output image quality is acceptable to human eyes. Also some color quantization researchers have focused on the processing of image sequences. Roytman and Gotsman[8] have proposed an algorithm for dynamic color quantization of image sequences, which quantizes each image independently to produce a different color palette for each image. Besides, they suggest the approach of color palette filling to prevent frequent switching of color palettes, which leads to the problem of screen flicker.

However, with the consideration of communication bandwidth and storage space, most image or video data nowadays comes in compressed format. Extra heavy computation of image or video decompression is required before any above mentioned color quantization is employed. How to design color palette directly from compressed data has thus become a significant issue. In this paper, we extend the DSQ to present some novel color palette design methods for compressed videos. The proposed methods use the reduced image in compressed domain to design color palette. we propose a shifting-window scheme to display longer sequences without screen In section II, we describe the proposed approach for compressed videos, which uses the reduced image rather than the whole image for the color palette design to avoid the heavy computation. In addition, the technique of extracting key color frames for compressed videos is presented and a shift-window scheme is proposed to solve the screen flicker problem. Section III reports the simulation results of proposed methods and some discussions are made.

II. Limited Color Display for Compressed Video

Now more and more video clips are able to be accessed in some World Wide Web databases. However, with the limit of bandwidth of internet, those video clips are usually compressed by using MPEG [3] format. In this condition, how to analyze the video clips to design a suitable color palette for those machines with limited color display becomes important. In this section, we will introduce methods of color palette design for MPEG compressed video.

The low-resolution DC images of MPEG video, called DC sequence, are first extracted for color palette design. Additionally, to avoid huge computation costs, we will only apply the DSQ to those DC images called key color frames, which contain the major color information of the entire DC sequence, for obtaining the color palette of MPEG video. From the observation of DC sequences, we noticed that color information inside a frame is unchanged for most of the sequence except key color frames, which consists of color changes in the sequence. Thus, detection of color changes is essential for finding key color frames. The detection of key color frames is based on two steps and introduced in the following subsection:

1. detection of potential key color frames

2. detection of key color frames and extraction of color palette

Besides, if only one fixed color palette is used, the degradation would get worse and worse as the sequence length gets longer. To overcome this problem, a multiple color palette scheme called shifting-window scheme is proposed to display compressed video with good quality even when the sequence length is getting longer.

A. Extraction of DC sequence

We will use MPEG-1 video as the example to illustrate the process of DC sequence extraction. In MPEG-1, the DC coefficient of the DCT block in I frame is the average of pixels in the block. The DC image for I frame can be easily formed by getting the DC coefficient in every block. However, the challenge exists in extracting DC images from P or B frames, which use motion compensation to exploit the temporal redundancy. A general case for P frame has been proposed by Yeo and Liu[7] and is showed in Figure 1.

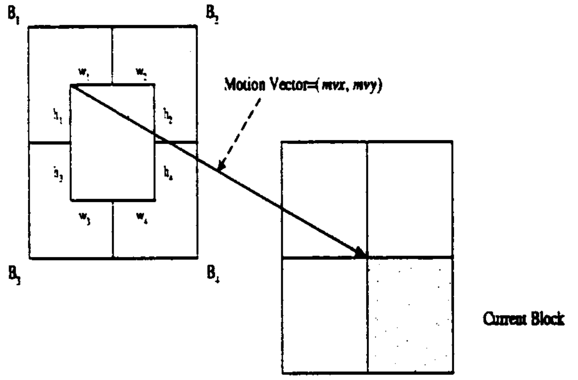


Figure 1: The extraction of DC image.

Here, B_{ref} is the block with motion vector (mvx, mvy) pointing to the current block of interest and B_1, B_2, B_3 and B_4 are the four neighboring blocks which derive the reference block B_{ref}. We can express the DC component of DCT coefficients of B_{ref} denoted as DCT(B_{ref}) in the following equation.

$$(\mathrm{DCT}(B_{ref}))_{00} = \sum_{i=1}^{4}\sum_{m=0}^{7}\sum_{l=0}^{7} w^i_{ml}(\mathrm{DCT}(B_i))_{ml} \quad (1)$$

where w^i_{ml} are weighting factors related to the motion vector.

This precise calculation of DCT coefficients is time-consuming. Since we are only interested in the DC component, the first-order approximation is used rather than precise calculation.

$$(\mathrm{DCT}(B_{ref}))_{00} = \sum_{i=1}^{4} \frac{w_i h_i}{64}(\mathrm{DCT}(B_i))_{00} \quad (2)$$

where w_i and h_i are the overlapping width and height of B_{ref} in block B_i. It can be shown that $\frac{w_i h_i}{64}$ corresponds to w^i_{00} in Eq. 1 and this is why it's called first-order approximation. Although the approximation error will accumulate, the error is acceptable in most cases because the GOP size is usually small and the error will reset to zero at every I frame. The other advantage of this approximation is that it requires only the motion vector information and the DC values in the reference frames. This approximation approach can also be applied to B frame where two reference frames might be needed. The problem of half-pixel-wise prediction can be solved by using the average of larger blocks depending on the motion vector. The 9x9 block is needed if half-pixel-wise prediction occurs in both x and y direction. The 8x9 block is used for half-pixel accuracy only in x direction and 9x8 block is for y direction only.

B. Key Color Frame Detection for Compressed Video

After the DC sequence is derived from a MPEG video, we use its DC images for key color frame detection. To represent color information of a frame, hue component of HSV space is adopted in this paper. Hue component has been widely accepted as a good candidate of representing color difference. Using this representation, we compare the normalized difference of hue histogram between consecutive frames in order to detect boundaries between consecutive color changes. The principle behind this is that two frames having a unchanging background and objects will show little difference in their corresponding hue histograms. The normalization procedure is utilized for reducing the impact of noise imposing on the hue histograms of consecutive frames. The normalized hue difference between the hue histograms of the lth and $(l-1)$th frame, S_l is given by the following equations:

$$S_l = \sum_{j=1}^{L} NHD_{l,l-1}(j) \quad (3)$$

where

$$NHD_{l,l-1}(j) = \begin{cases} H_l^3(j) & \text{for } H_{l-1}(j) = 0 \\ \left|\frac{H_l(j)-H_{l-1}(j)}{min(H_l(j),H_{l-1}(j))+1}\right| & \text{otherwise} \end{cases} \quad (4)$$

with $H_l(j)$ and $H_{l-1}(j)$ being hue histograms of the two consecutive frames respectively and L of Eq. 3 being the number of hue component bins in comparison. In Eq. 4, we choose the minimum value of $H_l(j)$ and $H_{l-1}(j)$ to normalize the hue difference. The condition, $H_{l-1}(j) = 0$, is set to reflect the situation when pixels with the certain hue value j exists in frame l, but not in frame $l-1$.

Similar to luminance change of the DC sequence, we have observed that color change is also a local activity which involves details regarding several neighboring frames. For scene change analysis, Yeo and Liu [5] has suggested to set the threshold of luminance change in order to match the local activity. We adopt this approach to detect color changes in this paper and choose a sliding window thresholding technique proposed by Yeo and Liu [5] to avoid false alarms which might occur in camera operations or object changes. In this technique, $2n-1$ frames with $2n-2$ hue differences are examined in a local range. Inside this local window, a color change from $(l-1)$th to lth image occurs if the following conditions are satisfied:

1. The difference is the maximum with the window, i.e., $S_j \le S_l, j = l-n+1, \cdots, l-1, l+1, \cdots, l+n-1$.
2. S_l is also m times of the second maximum in this window.

After examination of each window, the window is shifted one frame to prepare the next examination until the whole sequence is processed. In criterion 1, the parameter n is set to be smaller than the minimum duration between two color changes, but large enough to avoid false alarms. This is because as the window size gets smaller, the threshold is closer to be a global approach which is unfavorable for color change detection. If we set $n = 30$ for a 30 frame/sec video sequence, it means that there cannot be two color changes within a second. The parameter m in criterion 2 is imposed to guard against some camera operations such as fast panning or zooming. For these operations, the hue differences S_l would maintain consecutive peaks across several frames. From experimental results, we understand that the design of m depends on the tradeoff between increasing the detection rate and decreasing the false alarm rate. It has been found that the values of m varies from 2.0 to 4.0 give good results.

Through the above sliding-window thresholding scheme, the detected frames are called potential key color frames. From experimental results, we observed that there are redundant false-alarmed frames, which don't contain significant color information, inside these potential key color frames. Then, we adopt a coarse-to-fine strategy to eliminate those false-alarmed frames. In this strategy, these potential key color frames is processed one more time by the sliding-window thresholding scheme. After this examination, the detected frames are desired key color frames which are then used by the DSQ for the extraction of color palettes. We illustrate the proposed scheme of detecting key color frames for compressed video in Figure 2.

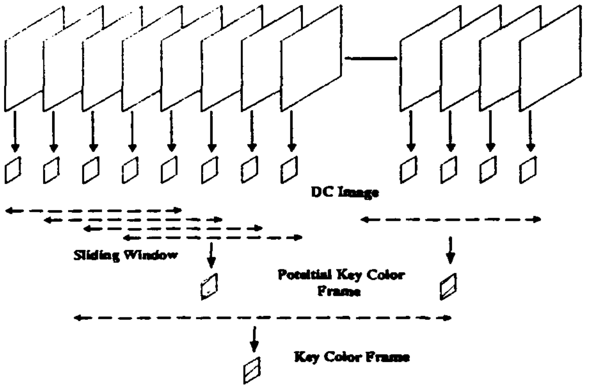


Figure 2: Detection of key color frame.

To do limited-color display with the color palette extracted by the proposed scheme, we employ the same procedures as for the compressed image. Using the well-organized boundaries of color cubes partitioned by the DSQ, pixels of each image of decoded MPEG sequence are mapped to their associated representative colors.

C. Shifting-Window Scheme

If only one fixed color palette is assigned to the whole sequence, the performance of color quantization is getting worse as the sequence length gets longer. On the other hand, if a color palette is designed for every key color frame, a serious visual artifact, screen flicker, may occur when the color palette is changed for these key color frames[8]. This problem happens because there is a sudden change of images colors. When the frame buffer of the display contains an image, a new color palette which belongs to the next image is already active. This phenomenon of screen flicker is sharp and unpleasant to human eyes.

To solve the screen flicker problem, we still use the DSQ to design color palettes for the sequence. But a color palette is designed for each fixed-length shifting-window in the sequence. The procedures of color palette design in the video sequence of each window are the same as mentioned above. The key color frames in each shifting-window are detected and applied to the DSQ for the color palette design as depicted in Figure 3. These windows contains overlapping frames, which causes the color distribution would not vary too much from window to window even if the color change occurs. As a result, the DSQ can generate smoothly varying color palettes for the sequence if the bit allocation procedure is fixed. In addition, since every entry of the color palette of the processed window would not differ significantly from that of the next window, screen flicker is greatly reduced.

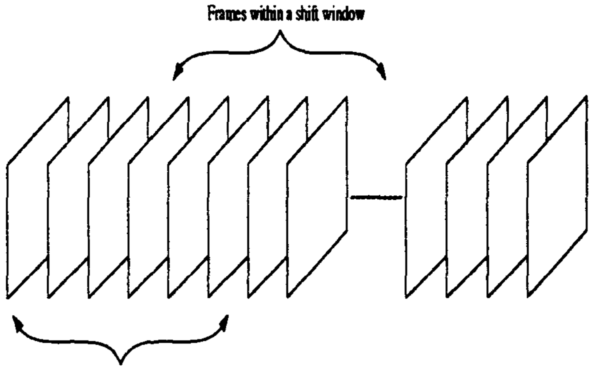


Figure 3: The proposed shifting-window scheme.

III. Experimental Results and Discussions

To evaluate the performance of the proposed method for compressed videos, a test sequence "News" which has 300 YUV frames with size 352x240 is used. The sequence consists of three main video segments which are news conference of a president candidate, a TV news reporter and news of a tropical storm. There are special effects of dissolving, fading in and fading out existed within each transition of two segments. This test sequence is first compressed by MPEG-1 compression. Then the proposed scheme is applied to design a color palette of 256 colors on the YUV color space. The color components were quantized in the order of Y(Luminance) first, then U and V last. The numbers of bits in each color component after bit allocation in the DSQ are 4(Y), 2(U) and 2(V).

The detected key color frames from the extracted DC sequence are frame 0, 3, 90, 195 and 210 which are shown in Figure 4. The parameters of criterion 1 and 2 used to find a color change are set to be 7 and 2, respectively. The sum of hue difference in the first step of scheme of detecting key color frames is plotted in Figure 5. As we can see, these frames indeed represent significant color difference. Frame 210 is detected because the yellow caption appears. And we plot in Figure 6 the PSNR distribution of luminance component of the decoded MPEG-1 sequence and the decoded sequence quantized by the proposed scheme. In this figure, the average PSNR of the decoded MPEG-1 sequence is 33.9805 dB and that of the proposed scheme is 32.4647 dB, which shows only about 1.5 dB lost on average in the color quantization. When we analyze Figure 6, it is noticed that some peaks or intra-frames among frame 90 and frame 210 have about 5 dB lost between PSNR values of the decoded MPEG-1 sequence and the proposed scheme. Normally, decoded intra-frames of MPEG-1 sequence are good quality since no motion estimation is involved. For these intra-frames, it shows that the matching of colors of decoded MPEG-1 frame with those of the original frame is better than

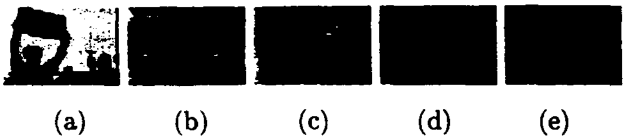


Figure 4: The DC images of detected key color frames: (a)frame 0 (b)frame 3 (c)frame 90 (d)frame 195 (e)frame 210

that of the representative colors obtained from the proposed scheme. However, since the PSNR values of the proposed scheme for these frames are around 35 dB, we don't perceive significant degradation of picture quality in experiments. Concerning the computation time, the proposed method took about 8.8 seconds to extract a color palette for the test sequence when using a SUN SPARC20 workstation.

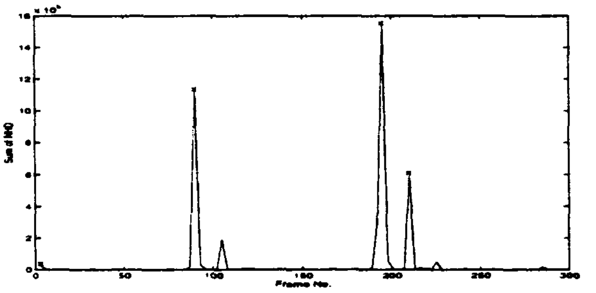


Figure 5: Sum of hue difference plot of "News".

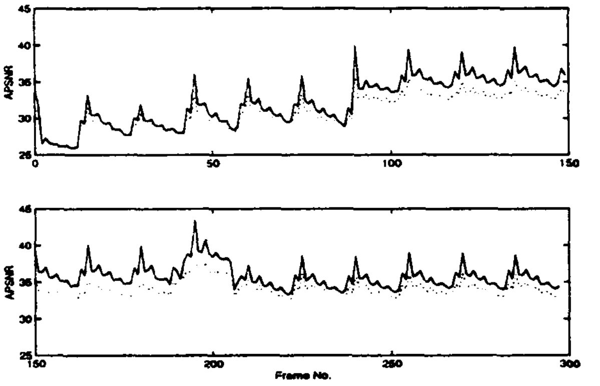


Figure 6: The PSNR in the sequence "News". The solid curve is for the decoded MPEG-1 sequence and the dotted curve corresponds to the proposed scheme.

We also executed the shifting-window scheme with the size of shifting-window being 150 and the size of overlapping frames between neighbouring windows being 75 on the test sequence. This configuration results in three shifting-windows to cover the test sequence. The designed color palettes for frames 0-149, 75-225 and 150-299 are shown in Figure 7(a), (b) and (c). As we can see, the color palette of frames 150-299 contains more dark colors which appear in the video clip of the tropical storm. And the gradual changes can be observed among these color palettes. When the corresponding quantized sequence is played back with frames 0-74 using palette of Figure 15(a), frames 75-224 using palette of Figure 15(b), and frames 225-299 using palette of Figure 15(c), we have seen that screen flicker phenomenon is insipid and acceptable to human eyes.

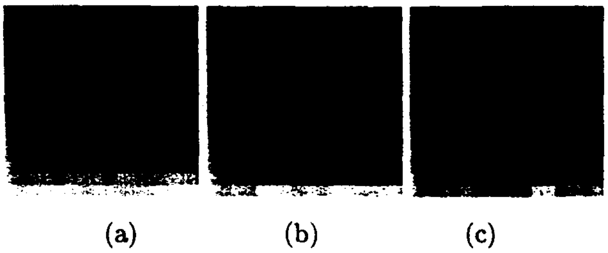


Figure 7: The color palettes for the sequence "News" when the shifting-windows scheme is employed.

References

[1] P. Heckbert, "Color image quantization for frame buffer display", *Comput. Graph.*, vol. 16, no. 3, pp. 297-397, Jul. 1982.

[2] S. C. Pei and C. M. Cheng, "Dependent scalar quantization of Color Images", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 5, no. 2, pp. 124-139, Apr.1995.

[3] "MPEG Video Committee Draft," ISO-IEC/JTC1/SC29/WE11/MPEG 90/176, Dec. 1990.

[4] R. Gray, "Vector quantization", *IEEE ASSP Mag.*, vol. 1, pp. 4-29, Apr. 1984.

[5] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Videos", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 533-544, Dec. 1995.

[6] Y. Nakajima, "a Video Browsing Using Fast Scene Cut Detection for an Efficient Networked Video Database Access", *IEICE Trans. Inf. and Syst.*, vol.E77-D, no. 12, pp. 1355-1364, Dec. 1994.

[7] B. L. Yeo and B. Liu, "On the extraction of DC sequences from MPEG compressed video", *International Conference on Image Processing*, vol. 2, pp. 260-263, Oct. 1995.

[8] E. Roytman and C. Gotsman, "Dynamic color quantization of video sequences", *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 3, pp. 274-286, Sep. 1995.

INTEGER DISCRETE FOURIER TRANSFORM AND ITS EXTENSION TO INTEGER TRIGOMATRIC TRANSFORMS

Soo-Chang Pei *Jian-Jiun Ding*

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C

Email address: pei@cc.ee.ntu.edu.tw

ABSTRACT

DFT has the good quality of performance and fast algorithm. But when we implement the DFT, we will require the floating-points multiplication. In this paper, we will introduce the *integer Fourier transform* (ITFT). ITFT is approximated to the DFT, but all the entries in the transform matrix are integer numbers. So it only requires the fixed-points multiplication, and the implementation can be much simplified, especially for VLSI. This new transform will work very similar as the original DFT, for example, the transform results are similar and the shifting-invariant property is also preserved for ITFT. Besides, we will also introduce the general method to derive the integer transform. By this approach, we can derive many types of integer transforms (such as integer cosine, sine, and Hartley transforms).

I. INTRODUCTION

Because of the direct relations with frequency spectrum and the fast algorithm, DFT is a very popular tool for signal processing. But when we implement the DFT, some floating-points multiplication operations are required. The implementation of floating-points multiplication is usually trouble and time-consuming. Besides, for the computer, it requires the floating-points processor, and will be more complex and expensive.

In this paper, we will derive the *integer Fourier transform* (ITFT). It approximates to the DFT, but all the entries in the transform matrix are integer numbers. We can implement the ITFT without any the floating-points multiplication operation because there are no non-integer entries. In section 2, we will introduce the general method to derive the integer transform from some real discrete transform. This method is the generalization and the modification of the method introduced by Cham [1] (He used his method to derive the integer cosine transform). In section 3, we will derive the 8-points ITFT. Then, in section 4, we will discuss the performance and the property of ITFT. We will show ITFT remains almost all the performance quality of DFT. In section 5, we make a conclusion.

II. THE GENERAL METHOD TO DERIVE THE INTEGER TRANSFORM

Integer transform is the discrete transform that all the entries in the transform matrix are integer numbers. When we try to find the integer transform analogous to some real transform, we can use following the steps described as below. Here we use A to denote the transform matrix of the original real transform, use B to denote the forward transform matrix of the integer transform, and use D to denote the inverse transform matrix of the integer transform.

(1) Find the *symmetry relation* and the *equality relation* for each row of the original real transform, and find the *sign* of the entries of the transform matrix.

(2) *Forming the prototype matrix of the forward integer transform* from the relations obtained in the step 1. The prototype must satisfy the following rules:

(a) If for the original transform matrix,

$A(m, a1) = A(m, a2)$ $A(m, b1) = -A(m, b2)$

$A(m, c1) = jA(m, c2)$ $A(m, d1) = -jA(m, d2)$

then for the forward integer transform,

$B(m, a1) = B(m, a2)$ $B(m, b1) = -B(m, b2)$

$B(m, c1) = jB(m, c2)$ $B(m, d1) = -jB(m, d2)$

(b) If for the original transform matrix,

$A(m1, n1) = 0$ $A(m2, n2) > 0$ $A(m3, n3) < 0$

then for the forward integer transform,

$B(m1, n1) = 0$ $B(m2, n2) > 0$ $B(m3, n3) < 0$

From these rules, we assign the unknowns in each entry of the prototype matrix. To be convenient, all the unknowns are constrained to be positive and real integers. If the prototype matrix has too many unknowns, we can try to make some unknowns be the same, or make some unknowns to be 1.

(3) *Forming the prototype matrix of the inverse integer transform.* The prototype of transform matrix of the inverse transform is of the same form as the forward transform, but we use another set of unknowns.

(4) *Constraints for orthogonality.* In this step, we search for the requirements to make the transform matrices of the forward and inverse transform to be orthogonal, and make the inner product of the same rows to be the values of 2^k where k is some integer. That is,

$$\sum_{k=0}^{N-1} B(m,k)\overline{D(n,k)} = R_m \delta_{mn} \quad \text{where } R_m = 2^k \quad (1)$$

From this, we obtain the *equality constraints* for the unknowns of the prototypes matrices.

(5) *Constraints for inequality.* In this step, we find the inequality relations among the unknowns of the prototypes matrices from the inequality relations for each row of the original non-integer transform matrix. That is, if for the original transform,

$A(m, n1) > A(m, n2)$

then for the forward and inverse integer transforms,

$B(m, n1) > B(m, n2)$ $D(m, n1) > D(m, n2)$

These will be the *inequality constraints* for the unknowns.

(6) ***Assign the values for all the unknowns.*** At last, we assign the values of unknowns. They must be real, positive integer numbers, and satisfy the constraints obtained in steps (4), (5).

We note, the transform matrix B of the forward transform and the transform matrix of the inverse transform D would be different. Thus, if X(m) is the forward integer transform of x(n),

$$X(m) = \sum_{n=0}^{N-1} B(m,n)\cdot x(n) \qquad (2)$$

Then we can recover x(n) from X(m) by

$$x(n) = \sum_{m=0}^{N-1} D^{*}(m,n)\cdot C_m^{-1}\cdot X(m) \qquad (3)$$

where * represents the conjugation. These can also be written as

$$X = B\cdot x \qquad x = D^H C^{-1} X \qquad (4)$$

where H is the Hermitian operation. The method introduced in [1] makes the transform matrix for the forward and the inverse integer transform to be the same, and can't avoid the floating-points multiplication for the inverse transform (since the values of R_m in Eq. (1) would not be the form of 2^k). Here we allow the forward and the inverse integer transform to be different. This enables us to fully avoid the floating-points multiplication operations, no matter for the forward or inverse transform. But since B, D are of the same forms, so the structures of the implementation of the forward and inverse transform are basically the same, except for the direction would be reversed and the parameters are different.

From the process introduced above, we can assure the integer transform we derived are very similar to the original non-integer transform, because (1) the symmetry, equality relations for each row, (2) the sign of each entry, (3) the orthogonality property, (4) the inequality relations for each row have been preserved. But sometimes, the integer transform is very hard to derive when we try to keep all the relations described above. In these cases, we will relax some of the relations.

III. THE 8-POINTS INTEGER FOURIER TRANSFORM

The original 8-points DFT is:

$$F(m,n) = \exp(-j\,m\,n\,\pi/4) \qquad m,\ n \in [\,0,\ 1,\ \ldots\ 7] \qquad (5)$$

To derive the 8-points integer Fourier transform (ITFT), we first try to form its prototype from the equality relation and the sign of each entry of the original 8-points DFT. We list the equality relations for each row of the 8-points DFT as below:

row	row 0	row 1	row 2	row 3	row 4	row 5	row 6	row 7
G=1	C=1		C=-j		C=-1		C=j	
G=2	C=1	C=-j	C=-1	C=j	C=1	C=-j	C=-1	C=j
G=4	C=1	C=-1	C=1	C=-1	C=1	C=-1	C=1	C=-1

Table 1 the equality relation of 8 points DFT

The values of G and C mean the m^{th} row of the integer Fourier transform prototype matrix will have the following relation

$$FI_P(m,n\oplus G) = C\cdot FI_P(m,n) \qquad \text{if } n\oplus G > n$$

$$FI_P(m,n\oplus G) = C^{-1}\cdot FI_P(m,n) \qquad \text{if } n\oplus G < n$$

where ⊕ is the exclusive-OR addition:

$$\sum_{i=1}^{K} a_i\cdot 2^i \oplus \sum_{i=1}^{K} b_i\cdot 2^i = \sum_{i=1}^{K}\left(a_i \,\mathrm{XOR}\, b_i\right)\cdot 2^i \qquad a_i,\ b_i = 0,\ 1$$

From these relations, and together with the sign of the entries in the 8-points DFT matrix, we can construct the prototype matrix of the 8-points forward ITFT as:

$$FI_P = \begin{bmatrix} e1 & e1 & e1 & e1 & e1 & e1 & e1 & e1 \\ a1 & a2-ja2 & -ja1 & -a2-ja2 & -a1 & -a2+ja2 & ja1 & a2+ja2 \\ e2 & -je2 & -e2 & je2 & e2 & -je2 & -e2 & je2 \\ b1 & -b2-jb2 & jb1 & b2-jb2 & -b1 & b2+jb2 & -jb1 & -b2+jb2 \\ e3 & -e3 & e3 & -e3 & e3 & -e3 & e3 & -e3 \\ c1 & -c2+jc2 & -jc1 & c2+jc2 & -c1 & c2-jc2 & jc1 & -c2-jc2 \\ e4 & je4 & -e4 & -je4 & e4 & je4 & -e4 & -je4 \\ d1 & d2+jd2 & jd1 & -d2+jd2 & -d1 & -d2-jd2 & -jd1 & d2-jd2 \end{bmatrix}$$

We assign the unknowns for the real part and image part of the entries separately to assure all the unknowns will be real numbers. In this prototype matrix, there are totally 12 unknowns. The amount of unknowns seems to be too much, so we will set some unknowns to be 1 and some unknowns to be equal. We set

$$e1 = e2 = e3 = e4 = 1, \qquad (6)$$

$$b1 = c1, \quad b2 = c2, \qquad d1 = a1, \quad d2 = a2 \qquad (7)$$

Then the prototype is simplified as:

$$FI_P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a1 & a2-ja2 & -ja1 & -a2-ja2 & -a1 & -a2+ja2 & ja1 & a2+ja2 \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ c1 & -c2-jc2 & jc1 & c2-jc2 & -c1 & c2+jc2 & -jc1 & -c2+jc2 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ c1 & -c2+jc2 & -jc1 & c2+jc2 & -c1 & c2-jc2 & jc1 & -c2-jc2 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ a1 & a2+ja2 & ja1 & -a2+ja2 & -a1 & -a2-ja2 & -ja1 & a2-ja2 \end{bmatrix} \qquad (8)$$

and there are only 4 unknowns. But we must assure the ITFT can still be derived after the simplification, otherwise we must remove some of the equality relations Eqs. (6), (7). We note, in Eq. (8), the m^{th} row of the prototype matrix will be the conjugation of the $(7\text{-}m)^{th}$ row, as the original DFT.

Then we form the prototype for the inverse ITFT. The prototype is of the same form as Eq. (8), but we change the unknowns {a1, a2, c1, c2} as {a3, a4, c3, c4}:

$$IFI_P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a3 & a4-ja4 & -ja3 & -a4-ja4 & -a3 & -a4+ja4 & ja3 & a4+ja4 \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ c3 & -c4-jc4 & jc3 & c4-jc4 & -c3 & c4+jc4 & -jc3 & -c4+jc4 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ c3 & -c4+jc4 & -jc3 & c4+jc4 & -c3 & c4-jc4 & jc3 & -c4-jc4 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ a3 & a4+ja4 & ja3 & -a4+ja4 & -a3 & -a4-ja4 & -ja3 & a4-ja4 \end{bmatrix} \qquad (9)$$

There are 4 unknowns for the inverse ITFT, so there are totally 8 unknowns for ITFT.

Then we search the constraints for the unknowns. First, we try to find the orthogonality constraints to make each row of the forward, inverse ITFT to satisfy the Eq. (1).

From the prototype of the forward, inverse ITFT, we find except for {row 1, 5}, {row 5, 1}, {row 3, 7}, and {row 7, 3}, all pairs of different rows will be orthogonal to each other. And calculating the inner product of {row 1, 5}, {row 5, 1}, {row 3, 7}, and {row 7, 3} directly, we find if we want these pairs to be orthogonal, then the following constraints must be satisfied:

(1) $a1c3 = 2a2c4$ (2) $a3c1 = 2a4c2$

Then, from the requirement that the inner product of the same row must be the value of the power of 2, we also obtain the constraint as:

(3) $a1a3 + 2a2a4 = 2^k$ (4) $c1c3 + 2c2c4 = 2^h$

where k, h are integer numbers. The constraints of (1), (2), (3), (4) will be the equality constraints of the unknowns.

Then, from the inequality relations for each row of original DFT matrix, we obtain the inequality constraints for the 8-points integer DFT as:

(5) $a1 \geq a2$ (6) $c1 \geq c2$ (7) $d1 \geq d2$ (8) $f1 \geq f2$

These are the inequality constraints. Thus we have obtained all the constraints for unknowns.

Then we assign the values of unknowns. Since there are 8 unknowns and only 4 equality constraints, so there are infinite choices for the values of unknowns. But to search the values of unknowns to satisfy all the constraints, especially to satisfy the constraints (3), (4), is a difficult task. We introduce a process as below. This process will make the work of searching for the values of unknowns to be more efficient.

(a) Choose the values of a1, a2 such that a1, a2 are integer numbers and

$$2 \cdot a2 \geq a1 \geq a2 \qquad (10)$$

(b) Find the integer values of c1, c2 such that c1 ≥ c2, and

$$a1 \cdot c2 + a2 \cdot c1 = 2^n \qquad \text{n is integer} \qquad (11)$$

(c) Set the value of a3, a4, c3, c4 as

$$c3 = 2 \cdot a2, \quad c4 = a1, \quad a3 = 2 \cdot c2, \quad a4 = c1$$

Then the values of unknowns are all obtained. We list some possible choices of the unknowns as below:

(1) a1=2, a2=1, c1=2, c2=1, a3=2, a4=2, c3=2, c4=2
This is the smallest, simplest integer solution for the unknowns. And in this case, the value of R_m in Eq. (1) is:

$$R_0 = R_2 = R_4 = R_6 = 2^3, \quad R_1 = R_3 = R_5 = R_7 = 2^4.$$

(2) a1=7, a2=5, c1=13, c2=9, a3=18, a4=13, c3=10, c4=7
We note, when we choose the parameters as the values listed above, the ratios of a1:a2, c1:c2, a3:a4, c3:c4, are all near to 1.414:1, which is ratio of the original discrete Fourier transform. If R_m is defined as Eq. (1), then in this case

$$R_0 = R_2 = R_4 = R_6 = 2^3, \quad R_1 = R_3 = R_5 = R_7 = 2^{10}.$$

We can implement the forward/inverse 8-points integer Fourier transform (8-points ITFT) in the following ways:

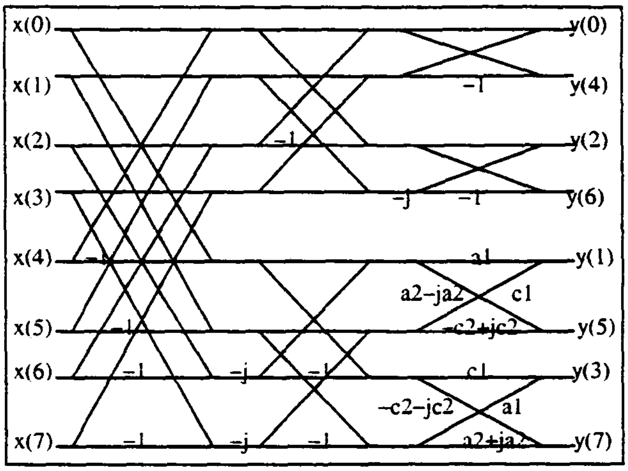


Fig. 1 Decimation-in-frequency implementation for the forward 8-point integer Fourier transform (ITFT)

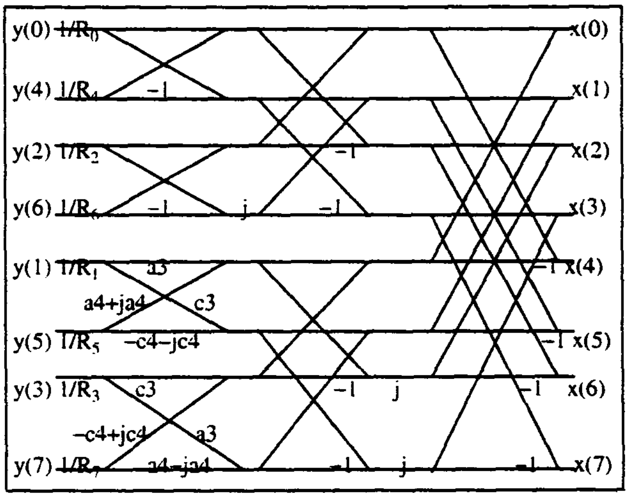


Fig. 2 Decimation-in-frequency implementation for the inverse 8-point integer Fourier transform

We find, for the 8-points ITFT, the number of the multiplication operations required is:

- forward/inverse transform: 8 fixed-points multi. operations
- normalization: 8 fixed-points multi. operations
- total: 24 fixed-points multiplication operations

And for the original 8-points DFT,

- forward/inverse transform: 2 floating-points multi. ops.
- normalization: 8 fixed-points multi. operations
- total: 4 floating-points, 8 fixed-points multi. operations

That is, we replace 4 floating-points multiplication operations with 16 fixed-points multiplication operations. The costs for doing 16 fixed-points multiplication operations is much less than the costs for doing 4 floating-points multiplication operation. Thus, the 8-points ITFT will be much faster than the original 8-points DFT. Besides, we can further reduce the number of fixed-points multiplication operations by setting a1=c1, or a2=c2, or a3=c3, or a4=c4, and each of them will save 2 fixed-points multiplication operations.

IV. THE PERFORMANCE OF THE INTEGER FOURIER TRANSFORM

We will see the performance of the 8-points integer Fourier transform (8-points ITFT). Here the parameters we choose are a1=7, a2=5, c1=13, c2=9, a3=18, a4=13, c3=10, c4=7.

We first see the transform result. We choose 2 inputs:
x1 = [2, 3, 4, 5, 4, 5, 2, 3]. (12)
x2 = [2.8, 4.3-0.6i, 3.7+0.9i, 3.1-0.6i, 4.6, 3.1+0.6i, 3.7-0.9i, 4.3+0.6i] (13)

We plot them in Fig. 3(a), 3(b). Then we do the original 8- points DFT, and plot the transform results in Fig. 3(c), 3(d). And then, we do the 8-points ITFT. To facilitate the comparison, we will normalize the transform results. That is, for the transform result of the 8-points ITFT

$$X(m)=\sum_{n=0}^{7} FI(m,n)\cdot x(n) \qquad (14)$$

We will normalize X(m) (the transform results of the 8-points ITFT) by the first column of the forward transform matrix:

$$\tilde{X}(m)=X(m)/FI(m,0). \qquad (15)$$

And plot them in Fig. 3(e), 3(f).

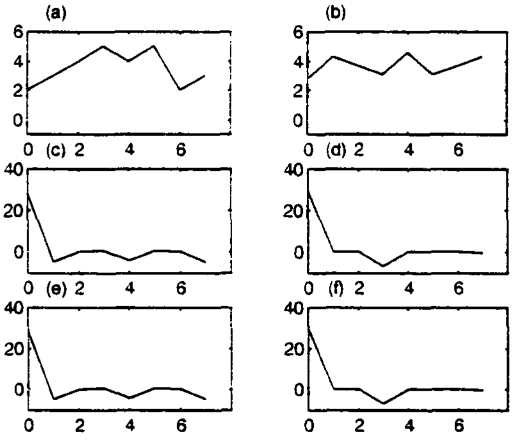


Fig. 3 The transform results of the original DFT and ITFT. (a), (b): Original signals. (c) (d): Results for original DFT. (e), (f): Results for ITFT. The real part and imaginary part are plotted in separable.

We find, the normalized transform results for the 8-points ITFT are very similar as the transform results for the original 8-points DFT. Also, we note that input x2 is the combination of a low frequency component and a high frequency component. As these 2 components can be separated by both the original DFT and the IDFT. So the IDFT can also be used for the filter design.

Then, we see the displacement property of the IDFT. Here we use the displacement of input x2[n] (defined in Eq. (15)) as the input:

$x3[n] = x2[((n+1))_8]$ $x4[n] = x2[((n+2))_8]$ (16)

And we plot x2, x3, x4 in Fig. 4(a), 4(c), 4(e). Then we do the 8-points ITFT for x2, x3, x4, and plot the amplitudes of the normalized results in Fig. 4(b), 4(d), 4(f).

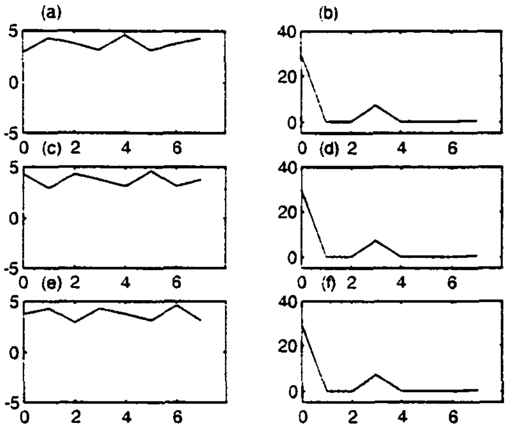


Fig. 4 The space-invariant property for the ITFT. (a)(c)(e): Shifted input. (b)(d)(f): Amplitude of the transform results of ITFT.

We find, for the ITFT, the displacement property also keeps. The signal after displacement will have the same transform amplitude as the original signal. When we compare the phase, there is a interesting phenomenon. We find, if X2(m), X3(m), X4(m) are the transform results of x2(n), x3(n), x4(n), then

angle(X3)−angle(X2)
= [0 −45 −90 −135 0 135 −270 −315] (17)
angle(X4)−angle(X2)
= [0 −90 −180 −270 0 −90 −180 −270] (18)

This is exactly the same as the original DFT.

V. CONCLUSION

In section 3, we have used the method introduced in section 2 to derive the 8 points integer Fourier transform (8 points ITFT). In fact, we can also derive the ITFT for some other number of points (such as the 6-points ITFT). We also can derive other types of integer transforms, such as the integer cosine, sine, and Hartley transforms, etc.

For the integer transform, we can replace all the floating-points multiplication operations with the fixed-points multiplication operations, so the integer transform is very efficient. If the datum we want to process are all integer number, using the integer transforms is especially efficient. The concept of the integer transform is very new, and there are only a few researches about it until now. Because they are convenient for implementation, and almost remain the quality of original non-integer transform, so they can compete with the original discrete transform in many applications. We believe the integer transform will be very popular in the future.

REFERENCE

[1] W. K. Cham, 'Development of integer cosine transform by the principles of dyadic symmetry', IEE Proceeding, Aug. 1989, vol. 136, no. 4, p 276-282

[2] A. V. Oppenheim, and R. W. Schafer, "Discrete-Time Signal Processing", Prentice-Hall, Inc., 1989

CLOSED-FORM DESIGN OF MAXIMALLY FLAT R-REGULAR MTH-BAND FIR FILTERS

Soo-Chang Pei and Peng-Hua Wang

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
Email address: pei@cc.ee.ntu.edu.tw

ABSTRACT

In this paper, we derive some properties of maximally flat R-regular Mth-band FIR filters. We show that the R-regularity implies maximally flat frequency response at $\omega = 0$. The R-regular constraints are a set of linear equations with complex coefficients. We can convert these complex-value equations to equivalent ones with only real coefficients. We also show that it is possible to completely determine the filter coefficients by R-regularity. Design examples are presented to illustrate the R-regularity properties and the effectiveness of the proposed approach.

1. INTRODUCTION

Mth-band filters are often used to design efficient digital sampling rate conversion systems for real time operations [1]. The impulse response $h(n)$ of an Mth-band FIR filter of order N has the property that

$$h(L + M\ell) = h(L)\,\delta(\ell), \quad \ell = 0, \pm 1, \pm 2, \dots \tag{1}$$

where L is the center of symmetry. That is, one of the M polyphase components is equal to $h(L)z^{-L}$ which is just a delay. An Mth-band filter satisfying the constraint of Eq. (1) in time domain has the following equivalent property in frequency domain [3, 4, 2]

$$\sum_{k=0}^{M-1} e^{j(2\pi k/M)L} H(e^{j(\omega+2\pi k/M)}) = Mh(L)e^{-j\omega L} \tag{2}$$

where $H(e^{j\omega})$ is the frequency response of the filter. Usually, $h(L)$ is assigned to be $1/M$ for normalization. In [3], constraint of the frequency response and bounds of passband and stopband ripples were derived. In [5], a linear-phase FIR Mth-band filter is decomposed into cascaded several FIR subfilters which is designed simultaneously by Remez-type algorithm. In [4], nonlinear-phase Mth-band FIR filters with reduced group delay were investigated and designed by using the eigenfilter approach.

Mth-band filter with R-regularity implies that there are R zeros at $\omega_k = 2\pi k/M$, $1 \le k \le M-1$ on its frequency response. In [4], the R-regularity is transformed into the linear constraints in the impulse response. In this paper, we derive several properties of the R-regular Mth-band FIR filter. We will show that the normalization in Eq. (2) is achieved if the frequency response at $\omega = 0$ is unity. Moreover, the R-regularity also implies maximally flat frequency response at $\omega = 0$. Based on the properties, the linear equations constraining the impulse response can be split into several polyphase equations with fewer coefficients involved.

A lowpass FIR filter with frequency response $F(e^{j\omega})$ is maximally flat if its impulse response is determined by the flatness at $\omega = 0$ and $\omega = \pi$, in which the flatness is defined as the number of zeros of $dF(e^{j\omega})/d\omega$ [2]. Generally speaking, the maximally flat Mth-band FIR filter cannot be completely determined by the flatness at ω_k since the number of flatness does not match the filter coefficients in general . In this paper, we apply the concept of the maximally flatness to design of R-regular Mth-band FIR filters, and give a situation under which the impulse response of Mth-band FIR filter is solved by the flatness at ω_k.

2. PROPERTIES OF R-REGULAR MTH-BAND FIR FILTERS

Let $H(z)$ be the transfer function of an causal Nth order Mth-band FIR filter. In this paper, we consider the general case in which N and L can be arbitrary integers. Suppose $((L))_M = r$ where $((L))_M$ denotes L modulo M, the transfer function $H(z)$ can be expressed as

$$H(z) = \sum_{\substack{n=0 \\ ((n))_M \neq r}}^{N} h(n)z^{-n} + h(L)z^{-L}. \tag{3}$$

If the frequency response $H(e^{j\omega})$ is R-regular with R zeros at $\omega_k = 2\pi k/M$ for $1 \le k \le M-1$, then the transfer function $H(z)$ has R zeros at $z = W^k$, $1 \le k \le M-1$, where $W = e^{-j2\pi/M}$ is a Mth root of unity. The location of these zeros implies that $H(z)$ can be factored as $H(z) = H_1(z)[1 + z^{-1} + z^{-2} + \cdots + z^{-(M-1)}]^R$.

Since $H(e^{j\omega})$ has R zeros at $\omega_k = 2\pi k/M$, $1 \le k \le M-1$, the following equations have to be satisfied:

$$\sum_{\substack{n=0 \\ ((n))_M \neq r}}^{N} h(n)n^q W^{kn} + h(L)L^q W^{kn} = 0$$

or, specifically,

$$\begin{aligned} &\sum_{\substack{n=0 \\ ((n))_M = 0}}^{N} h(n)n^q W^{kn} + \cdots + \sum_{\substack{n=0 \\ ((n))_M = r-1}}^{N} h(n)n^q W^{kn} \\ + &\sum_{\substack{n=0 \\ ((n))_M = r+1}}^{N} h(n)n^q W^{kn} + \cdots + \sum_{\substack{n=0 \\ ((n))_M = M-1}}^{N} h(n)n^q W^{kn} \\ + &\ h(L)L^q W^{kn} = 0 \end{aligned} \tag{4}$$

for $1 \le k \le M-1$ and $0 \le q \le R-1$. There are $R(M-1)$ equations in Eq. (4). In additional, the frequency response at $\omega =$

unknowns (since $p = 0, 1, \ldots, r-1, r+1, \ldots, M-1$). That is, $A_p(q)$ may be solved exactly. To find the values of $A_p(q)$, we rewrite Eq. (6) in matrix form as follow

$$\mathbf{Ax} = \mathbf{b} \tag{12}$$

where $\mathbf{A} =$

$$\begin{bmatrix} 1 & W & \ldots & W^{r-1} & W^{r+1} & \ldots & W^{M-1} \\ 1 & W^2 & \ldots & W^{2(r-1)} & W^{2(r+1)} & \ldots & W^{2(M-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{M-1} & \ldots & W^{(M-1)(r-1)} & W^{(M-1)(r+1)} & \ldots & W^{(M-1)^2} \end{bmatrix},$$

$$\mathbf{b} = \left[-\frac{1}{M}W^r L^q, -\frac{1}{M}W^{2r}L^q, \ldots, -\frac{1}{M}W^{(M-1)r}L^q\right]^t,$$

and

$$\mathbf{x} = [A_0(q), A_1(q), \ldots, A_{r-1}(q), A_{r+1}(q), \ldots, A_{M-1}(q)]^t,$$

for $1 \le k \le M-1; 0 \le q \le R-1$ where the superscript ^{t} denotes the transpose. It is easy to find out a solution for Eq. (12) by inspiration. In fact, $A_p(q) = L^q/M$ is an unique solution to Eq. (6) and (12). Note that $\mathbf{A}$ is nonsingular because $\det(\mathbf{A})$ is a *Vandermonde determinant* generated by distinct elements. Therefore, the solution to Eq. (12) is unique [6]. Since the above derivation is based on Property 1 which is deduced from Eq. (5), we obtain the following property.

Property 3. *The $R(M-1)+1$ equations expressed by Eq. (4) as well as Eq. (5) can be simplified to $R(M-1)$ equations represented as*

$$\sum_{\substack{n=0 \\ ((n))_M = p}}^{N} h(n)n^q = \frac{L^q}{M} \tag{13}$$

for $p = 0, 1, \ldots, r-1, r+1, \ldots, M-1$ and $0 \le q \le R-1$.

In summary, Eq. (4) and (5) can be reduced to Eq. (13) as well as $h(L) = 1/M$. However, the coefficients in Eq. (13) are real numbers while the coefficients in Eq. (4) are complex. The computation with real numbers involved needs much less memory than that with complex numbers.

3. DESIGN OF MAXIMALLY FLAT R-REGULAR MTH-BAND FIR FILTERS

In Eqs. (4) and (5), or equivalently, Eq. (13), it is obvious that the number of equations is determined by R and M. For a given M, we have more equations if R is increased. If the number of zeros at W^k is all the same, the number of unknowns is generally not equal to the number of equations, and consequently $h(n)$ can not be solved by Eqs. (4) and (5) only. However, it is possible to solve the impulse response exactly form Eqs. (4) and (5) for some specific M, N and L. That is, it is possible to completely determine the R-regular Mth-band FIR filter by the zeros at W^k for some M, N and L with a suitable choice of R. This design may be regarded as a generalization of the traditional maximally flat filter. In this section, we will indicate the situations to achieve the maximally flat design.

Let N_c denote the number of coefficients to be determined in Eq. (3) and N_p denote the number of filter coefficients in $A_p(q)$ defined by Eq. (7). Let $\lfloor x \rfloor$ denote the largest integer less than x. It is easy to show that

$$\begin{aligned} N_c &= N - \lfloor \frac{L}{M} \rfloor - \lfloor \frac{N-L}{M} \rfloor + 1, & (14) \\ N_p &= \lfloor \frac{N-p}{M} \rfloor + 1, 0 \le p \le M-1, p \ne r. & (15) \end{aligned}$$

The following property indicates the sufficient condition for the design of the maximally flat R-regular Mth-band filters.

Property 4. *Eqs. (4) and (5) can be solved exactly if*

$$((N))_M = M-2, ((L))_M = M-1,$$

and

$$R = \lfloor \frac{N}{M} \rfloor + 1.$$

In fact, based on the Property 4, the impulse response can be solved not only by Eqs. (4) and (5), but also by Eqs. (13). The closed form of the impulse response is

$$h(Mm+p) = \frac{(-1)^m \Pi_{i=0}^{R-1}(Mi+p-L)}{(Mm+p-L)m!(R-1-m)!}$$

for $0 \le m \le R-1$ and $0 \le p \le M-1, p \ne r = ((L))_M$.

4. ILLUSTRATIVE EXAMPLES

In this section, we gives some design examples to illustrate the properties derived in Sec. 2 and 3.

Example 1. In the example, we will design the 14th order R-regular fourth-band FIR filter. The index of symmetry is 7. Thus, we have $M = 4$, $N = 14$, and $L = 7$. According Eq. 14, there is $N_c = 14 - \lfloor 7/4 \rfloor - \lfloor 7/4 \rfloor + 1 = 13$ coefficients to be solved which are

$$\{h(0), h(4), h(8), h(12)\}, \quad \{h(1), h(5), h(9), h(13)\},$$
$$\{h(2), h(6), h(10), h(14)\}, \quad \text{and } \{h(7)\}$$

where each group denotes the unknown coefficients in corresponding polyphase components.

Since $((14))_4 = 4-2$ and $((7))_4 = 4-1$, according to Property 4 these impulse response can be solved by Eqs. (4) and (5) if $R = \lfloor 14/4 \rfloor + 1 = 4$. In fact, the coefficients of each polyphase components can be separated and solved by $(M-1)$ subequations of $A_p(q) = L^q/M$ in Eq. (13). The resulting transfer function is

$$\begin{aligned} H(z) &= \frac{1}{512}\left(-5 - 8z^{-1} - 7z^{-2} + 35z^{-4} + 72z^{-5}\right. \\ &+ 105z^{-6} + 128z^{-7} + 105z^{-8} + 72z^{-9} + 35z^{-10} \\ &+ \left.-7z^{-12} - 8z^{-13} - 5z^{-14}\right). \end{aligned}$$

It is easy to shown that

$$H(z) = \frac{-1}{512}\left(5 - 12z^{-1} + 5z^{-2}\right)\left(1 + z^{-1} + z^{-2} + z^{-3}\right)^4,$$

and $H(z) - z^{-L} =$

$$\frac{-1}{512}\left(5 + 28z^{-1} + 89z^{-2} + 208z^{-3} + 370z^{-4} + 488z^{-5}\right.$$
$$\left.+370z^{-6} + 208z^{-7} + 89z^{-8} + 28z^{-9} + 5z^{-10}\right)\left(1 - z^{-1}\right)^4$$

The designed filter has symmetric impulse and thus has linear phase response. Fig. 1(a) and (b) show the impulse response and magnitude response, respectively.

Example 2. In the example, we also design the R-regular fourth-band FIR filter of 14th order. But the index of symmetry is reduced to be 5. Thus, we have $M = 4$, $N = 14$, and $L = 5$. According Eq. 14, there is $N_c = 14 - \lfloor 5/4 \rfloor - \lfloor 9/4 \rfloor + 1 = 12$ coefficients to be solved which are

$$\{h(0), h(4), h(8), h(12)\}, \quad \{h(2), h(6), h(10), h(14)\},$$
$$\{h(3), h(7), h(11)\}, \quad \text{and } \{h(5)\}$$

where each group denotes the unknown coefficients in corresponding polyphase components.

We assign one degrees of freedom to be the unity DC gain constraint. Since the reset 11 constraints cannot be assigned to the 3 frequencies of $2\pi/4$, $4\pi/4$ and $6\pi/4$ evenly, the maximally flat design cannot be solved. However, if let $R = 3$ in Eqs. (4) and (5) and put the additional 2 degrees of freedom on the flatness at $\omega = \pi$, we can solve the impulse response. The resulting transfer function is

$$\begin{aligned} H(z) &= \frac{1}{2048}\big(-9 + 73z^{-2} + 192z^{-3} + 363z^{-4} + 512z^{-5} \\ &+ 501z^{-6} + 384z^{-7} + 197z^{-8} - 69z^{-10} - 64z^{-11} \\ &+ -39z^{-12} + 7z^{-14}\big). \end{aligned}$$

It is easy to shown that

$$\begin{aligned} H(z) &= \frac{-1}{2048}\left(3 - z^{-1}\right)\left(3 - 14z^{-1} + 7z^{-2}\right) \\ &\times \left(1 + z^{-1}\right)^2 \left(1 + z^{-1} + z^{-2} + z^{-3}\right)^3, \end{aligned}$$

and $H(z) - z^{-L} =$

$$\begin{aligned} &\frac{1}{2048}\big(9 + 27z^{-1} - 19z^{-2} - 321z^{-3} - 1242z^{-4} - 1246z^{-5} \\ &-834z^{-6} - 390z^{-7} - 111z^{-8} + 3z^{-9} + 21z^{-10} + 7z^{-11}\big) \\ &\left(1 - z^{-1}\right)^3 \end{aligned}$$

The impulse response is not symmetric. Fig. 2(a) and (b) show the impulse response and magnitude response, respectively.

5. CONCLUSIONS

In this paper, several properties of the R-regular Mth-band FIR filter are derived. We show that $h(L) = 1/M$ if the frequency response at $\omega = 0$ is unity. Moreover, if the Mth-band FIR filter is R-regular, then we can show that its frequency response have R degree of flatness at $\omega = 0$. Based on these properties, the linear equations constraining the impulse response can be split into several sub-group equations with fewer real polyphase coefficients involved. Generally speaking, the Mth-band FIR filter cannot be completely determined by the flatness at ω_k. In this paper, we apply the concept of the maximally flatness to the design of R-regular Mth-band FIR filters, and give a situation under which the impulse response of Mth-band FIR filter can be completely solved by the flatness at ω_k. Design examples are presented to verify these properties.

6. REFERENCES

[1] G. STRANG and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall Inc., 1993.

[3] F. Mintzer, "On half-band, third-band, and Nth-band FIR filters and their design," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, no. 5, pp. 734-738, October 1982.

[4] Y. Wisutmethangoon and T. Q. Nguyen, "A method for design of Mth-band filters," *IEEE Trans. Signal Processing*, vol. 47, no. 6, pp. 1669-1678, June 1999.

[5] T. Saramäki and Y. Neuvo, "A class of FIR Nyquist (Nth-band) filters with zero intersymbol interference," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 10, pp. 1182-1190, October 1987.

[6] B. Noble and I. W. Daniel, *Applied Linear Algebra*, Prentice-Hall International Inc., 1988.

[7] D. K. Faddeev and I. S. Sominskii, *Problems in Higher Algebra*, translated by J. L. Brenner, W. H. Freeman and Company, 1965.

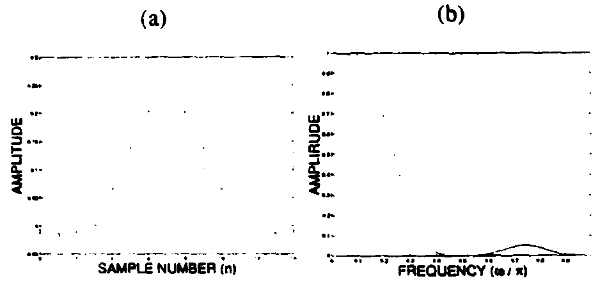


Figure 1: Design of the maximally flat 14th order R-regular fourth-band FIR filter with symmetry index $L = 7$. (a) Impulse response and (b) magnitude response.

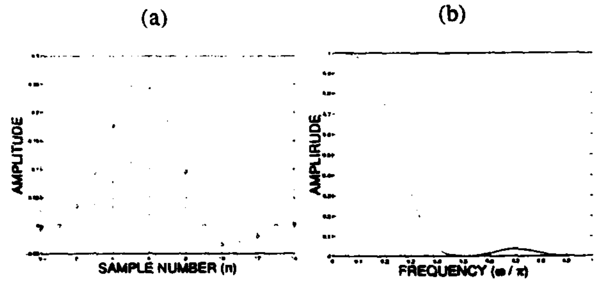


Figure 2: Design of the maximally flat 14th order R-regular fourth-band FIR filter with symmetry index $L = 5$. (a) Impulse response, and (b) magnitude response.