

รายการอ้างอิง

1. Pennebaker, William B. and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
2. Wallace, Gregory K. The JPEG Still Picture Compression Standard, *Communications of the ACM*, vol. 34, no. 4, April 1991, pp. 30-44.
3. Ahmed, N., Natarajan, T., and Rao, K. R. Discrete Cosine Transform, *IEEE Trans. Computers*, vol. C-23, Jan. 1974, pp. 90-93.
4. Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.
5. Rabbani, M., and Jones, P. Digital Image Compression Techniques, *Tutorial Texts in Optical Engineering*, vol. TT7, SPIE Press, 1991.
6. Hudson, G., Yasuda, H. and Sebestyen, I. The International Standardization of Still Picture Compression Technique, *Proceedings of the IEEE Global Telecommunications Conference*, November 1988, pp. 1016-1021.
7. Nelson, M. *The Data Compression Book*, M&T Books, Redwood City, CA. 1991.
8. Fraenkel, A. S., and Klein S. T. Novel Compression of Sparse Bit-Strings- Preliminary Report, in A. Apostolico and Z. Galil, eds., *Combinatorial Algorithms on Words*, Vol. 12, NATO ASI Series F:169-183, New York, Springer-Verlag, 1985.
9. Anedda, C., and Felician L. P-Compressed Quadrees for Image Storing. *The Computer Journal*, 31, 4 (1988), 353-357.
10. Ziv, J., and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, 1977, pp. 337-343.
11. Ziv, J., and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, September 1978.
12. Dwayne, P. "LZW Data Compression," *The Computer Applications Journal*, Circuit Cellar Ink, vol. 27, June/July 1992, pp. 36-48.
13. Montgomery, B. *LZW Compression Used to Encode/Decode a GIF File*, manuscript in the public domain, 1988.

14. Rodriguez, K. "Graphics file format patent Unisys seeks royalties from GIF developers,"
InfoWorld, vol. 17, January 9, 1995, p. 3.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

รายละเอียดซอฟต์แวร์

```

unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls, Kitkernel, Gauges, JRZIP;

type
  TForm1 = class(TForm)
    StatusBar1: TStatusBar;
    lbFileName: TLabel;
    lbHTNName: TLabel;
    OpenFileDialog1: TOpenDialog;
    PanelStack: TPanel;
    Bevel1: TBevel;
    lbStatus: TLabel;
    OpenFileDialog2: TOpenDialog;
    Gauge1: TGauge;
    Image3: TImage;
    OpenFileDialog: TOpenDialog;
    SaveDialog: TSaveDialog;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Memo1: TMemo;
    Image1: TImage;
    Image2: TImage;
    ImageBlank: TImage;
    imgDump: TImage;
    Label2: TLabel;
    Label3: TLabel;
    memoLog: TMemo;
    btnLoad: TButton;
    btnEncode: TButton;
    btnDecode: TButton;
    btnCancel: TButton;
    Label4: TLabel;
    GroupBox2: TGroupBox;
    GroupBox1: TGroupBox;
    cbSavePattern: TCheckBox;
    cbSaveLog: TCheckBox;
    cbZIP: TCheckBox;
    cbInclude: TCheckBox;
    procedure btnLoadClick(Sender: TObject);
    procedure btnEncodeClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnDecodeClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    hh,mm,ss,ms :word;
    EndLoop :boolean;

    HTsize :integer;
  
```

```

PatternSize :integer;
PatternMax :integer;
PatternCount :integer;

PatternData :PByteArray;
PatternDataCount :integer;

PatternStack :PByteArray;
StackCount :integer;
StackPos :int64;

Output:TFileStream;

BmpLineCount:integer;
BmpLine:PByteArray;

procedure PutStack(SrcBMP:TBitmap;var Count:integer);
function SameBMP(Bmp1,Bmp2:TBitmap):boolean;
function SameBMP8(Bmp1,Bmp2:TBitmap):boolean;

Procedure SaveStack;
Procedure DumpStack;
procedure GetHT(SrcBMP,DesBMP:TBitmap;x,y:integer);
procedure PutHT(SrcBMP,DesBMP:TBitmap;x,y:integer);
end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

#####
#####
#####

procedure TForm1.GetHT(SrcBMP,DesBMP:TBitmap;x,y:integer);
var RectTarget,RectSource:TRect;
begin
  RectTarget := Rect(0,0,HTsize,HTsize);
  RectSource := Rect(x,y,x+HTsize,y+HTsize);
  DesBMP.Canvas.CopyRect(RectTarget,SrcBMP.Canvas,RectSource );
end;

procedure TForm1.PutHT(SrcBMP,DesBMP:TBitmap;x,y:integer);
var RectTarget,RectSource:TRect;
begin
  RectTarget := Rect(x,y,x+HTsize,y+HTsize);
  RectSource := Rect(0,0,HTsize,HTsize);
  DesBMP.Canvas.CopyRect(RectTarget,SrcBMP.Canvas,RectSource );
end;

procedure TForm1.PutStack(SrcBMP:TBitmap;var Count:integer);
begin
  Inc(Count);
  With TImage.Create(PanelStack) do
  begin
    Parent:=PanelStack;
    Name:=Format('Stack%d',[Count]);
    Picture.Bitmap.Assign(SrcBMP);
    Top:=0; Left:=0;
    Tag:=0;
  end;
end;
end;

```



```

Function TForm1.SameBMP(Bmp1,Bmp2:TBitmap):Boolean;
var i,j:integer;
begin
  Result:=True;
  for i:=0 to Bmp1.Height-1 do
    for j:=0 to Bmp1.Width-1 do
      begin
        if Bmp1.Canvas.Pixels[i,j]<>Bmp2.Canvas.Pixels[i,j] then
          begin
            Result:=False;
            exit;
          end
        end;
      end;
    end;
  end;
end;

```

```

Function TForm1.SameBMP8(Bmp1,Bmp2:TBitmap):Boolean;
var i,j:integer;
    BmpLine1,BmpLine2:PByteArray;
begin
  Result:=True;
  for i:=0 to Bmp1.Height-1 do
    for j:=0 to (Bmp1.Width div 8)-1 do
      begin
        BmpLine1:=Bmp1.ScanLine[i];
        BmpLine2:=Bmp2.ScanLine[i];
        if BmpLine1[j] <> BmpLine2[j] then
          begin
            Result:=False;
            exit;
          end;
        end;
      end;
    end;
  end;
end;

```

```

Procedure TForm1.SaveStack;
var i,j,k:integer;
    BmpLine1:PByteArray;
    bmpCount:integer;
    bTemp:Byte;
begin
  bmpCount:=0;
  For i:=0 to PanelStack.ControlCount-1 do
    begin
      With TImage(PanelStack.Controls[i]).Picture do
        begin
          for j:=0 to Bitmap.Height-1 do
            begin
              BmpLine1:=Bitmap.ScanLine[j];
              for k:=0 to (Bitmap.Width div 8)-1 do
                begin
                  bTemp:=BmpLine1[0];
                  Output.Write(bTemp,1);
                end;
              inc(bmpCount);
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

Procedure TForm1.DumpStack;
var RectTarget,RectSource:TRect;
    i:integer;
begin
  imgDump.Picture.Bitmap.Width:=HTsize*PanelStack.ControlCount;
  imgDump.Picture.Bitmap.Height:=HTsize;

```

```

For i:=0 to PanelStack.ControlCount-1 do
begin
  RectTarget := Rect((HTsize*i),0,HTsize+(HTsize*i),HTsize);
  RectSource := Rect(0,0,HTsize,HTsize);
  imgDump.Picture.Bitmap.Canvas.CopyRect(RectTarget,
    TImage(PanelStack.Controls[i]).Picture.Bitmap.Canvas,RectSource );
end;
end;

#####
#####
#####

procedure TForm1.btnLoadClick(Sender: TObject);
begin
  OpenFileDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
  if OpenFileDialog1.Execute then
  begin
    lbFileName.Caption:=OpenDialog1.FileName;
    Image1.Picture.LoadFromFile(lbFileName.Caption);
    if (Image1.Picture.Bitmap.Width mod 8 <>0) or (Image1.Picture.Bitmap.Width mod 8 <>0) then
    begin
      Memo1.Text:='Error : Size is not proper.'#13#10;
      lbFileName.Caption:='';
      Image1.Picture.Bitmap.Assign(ImageBlank.Picture.Bitmap);
      Exit;
    end;
    Memo1.Text:='File loaded'#13#10;
    StatusBar1.Panels[0].Text:=ExtractFileName(lbFileName.Caption);
  end;
end;

procedure TForm1.btnEncodeClick(Sender: TObject);
var
  StartTime:TDateTime;
  OrgHeight,OrgWidth: integer;
  i,j,k,m,Count,LastIndex: integer;
  PByte1,PByte2: ^Byte;
  Byte1,Byte2: Byte;
  BMPBuff:TBitmap;
  ChildControl:TControl;
  FlagPutStack,Err:boolean;
  PyramidUp,PyramidDown:integer;
  TargetName,Text:String;
  buff:array[0..127] of byte;
  buff2:array[0..127] of char;
  iTemp:integer;
  iTemp64:int64;
  LastByte:Byte;
begin
  if lbFileName.Caption="" then exit;
  StartTime:=Now;
  Memo1.Text:='';

  //***** Prepare Bitmap *****
  With Image1.Picture do
  begin
    OrgHeight:=Bitmap.Height;
    OrgWidth :=Bitmap.Width;
    HTsize :=8;
    PatternCount :=0;
    PatternSize :=HTsize*HTsize; // 8*8 = 64 bits
    PatternMax :=HTsize*HTsize; // 8*8 = 64 patterns
  end;
  Memo1.Text:='Half Tone Encode Algorithm'#13#10;
  Memo1.Text:=Memo1.Text+Format('Size = %d x %d'+#13#10,[OrgHeight,OrgWidth]);
  Memo1.Text:=Memo1.Text+Format('HTsize=%d'#13#10,[HTsize]);

```

```

//---- Clear All Stack Buffer -----
For i:=PanelStack.ControlCount-1 downto 0 do
  PanelStack.Controls[i].Destroy;

//---- Create Buffer-----
BMPBuff:=TBitmap.Create;
BMPBuff.Assign(ImageBlank.Picture.Bitmap);
BMPBuff.Width:=HTsize; BMPBuff.Height:=HTsize;

//---- Create Output File -----
StatusBar1.Panels[1].Text:='>>';
StatusBar1.Panels[2].Text:=ChangeFileExt(ExtractFileName(lbFileName.Caption),'.htm');
lbHTNName.Caption:=ExtractFilePath(lbFileName.Caption)+StatusBar1.Panels[2].Text;
TargetName:=lbHTNName.Caption;
Output:=TFileStream.Create(TargetName,fmCreate);

//---- Header -----
Text:='HTN V1.0b';StrPCopy(buff2,Text);
Output.Write(buff2,Length(Text));
Output.Seek(32,soFromBeginning);
iTemp:=OrgHeight;Output.Write(iTemp,4);
iTemp:=OrgWidth; Output.Write(iTemp,4);
iTemp:=HTSize; Output.Write(iTemp,4);

//---- Data -----
Output.Seek(64,soFromBeginning);

Gauge1.MaxValue:=Image1.Picture.Bitmap.Height -1;
StackCount:=0; LastIndex:=0; Err:=False; EndLoop:=False;
j:=0;
while (j<Image1.Picture.Bitmap.Height-1) and (EndLoop=False) do
EndLoop
begin
  i:=0 ;
  while i<Image1.Picture.Bitmap.Width-1 do
  begin
    Form1.GetHT(Image1.Picture.Bitmap,BMPBuff,i,j);

  With PanelStack do
  begin
    if ControlCount=0 then
      FlagPutStack:=True
    else
      begin
        FlagPutStack:=True;
        m:=0;

#####
// 3 Options for make Stack
#####
//---- 1 Direct Search -----
//for k:=0 to ControlCount-1 do
//begin
// if Form1.SameBMP8(BMPBuff,TImage(Controls[k]).Picture.Bitmap)=True then
// begin FlagPutStack:=False; end;
//end;

//---- 2 Last Index Search -----
//K:=LastIndex;
//repeat
// if Form1.SameBMP8(BMPBuff,TImage(Controls[k]).Picture.Bitmap)=True then
// begin FlagPutStack:=False; LastIndex:=k; end;
// k:=k+1;
// if k>ControlCount-1 then k:=0;

```



```

//until (k= LastIndex) or (FlagPutStack=False);

//----- 3 Last Index Pyramid Search -----
K:=LastIndex; PyramidUp:=LastIndex; PyramidDown:=LastIndex;
If Form1.SameBMP8(BMPBuff,TImage(Controls[k]).Picture.Bitmap)=True
then
begin FlagPutStack:=False; LastIndex:=k; end
else
repeat
inc(PyramidUp);
if PyramidUp<=ControlCount-1 then
If Form1.SameBMP8(BMPBuff,TImage(Controls[PyramidUp]).Picture.Bitmap)=True then
begin FlagPutStack:=False; LastIndex:=PyramidUp; end;
dec(PyramidDown);
if (PyramidDown>=0) and (FlagPutStack=True) then
If Form1.SameBMP8(BMPBuff,TImage(Controls[PyramidDown]).Picture.Bitmap)=True then
begin FlagPutStack:=False; LastIndex:=PyramidDown; end;
until (FlagPutStack=False) or ( (PyramidUp>ControlCount-1) and (PyramidDown<0) );
//-----

end;

if FlagPutStack then begin Form1.PutStack(BMPBuff,StackCount); LastIndex:=StackCount-1; end;

TImage(PanelStack.Controls[LastIndex]).Tag:=TImage(PanelStack.Controls[LastIndex]).Tag+1;
LastByte:=LastIndex;
Output.Write(LastByte,1); // Write when LastByte is completed

#####
// Write to disk here
#####

end;
i:=i+HTSize;
end;
j:=j+HTsize;
Gauge1.Progress:=j;
if StackCount>=128 then begin Err:=True;EndLoop:=True; Memo1.Text:=Memo1.Text+'Error: Stack
Overflow#13#10; end;
Application.ProcessMessages;
lbStatus.Caption:=Format('Row=%d Stack=%d Last=%d',[j,StackCount,LastIndex]);
end;

// --- Close Output File -----
StackPos:=Output.Position;
Form1.SaveStack;
Output.Seek(48,soFromBeginning);
iTemp:=StackCount;Output.Write(iTemp,4);
iTemp64:=StackPos;Output.Write(iTemp64,8);
Output.Free;

BMPBuff.Free;
Gauge1.Progress:=Gauge1.MaxValue;
lbStatus.Caption:=";
Memo1.Text:=Memo1.Text+Format('Stack Count=%d#13#10,[StackCount]);

if Endloop then
begin
DeleteFile(lbHTNName.Caption);
lbHTNName.Caption:=";
StatusBar1.Panels[2].Text:=";
if Err=True then
lbStatus.Caption:='Error: Stack Overflow'

```

```

else
  lbStatus.Caption:='Error: Cancelled';
end;
// ทำการ Save Log file
if cbSaveLog.Checked then
begin
memoLog.Text:='Stack History'#13#10;
for i:=0 to PanelStack.ControlCount-1 do
  memoLog.Lines.Add(Format('P%0.3d = %d',[i,TImage(PanelStack.Controls[i]).Tag]));
memoLog.Lines.SaveToFile(ExtractFilePath(lbFileName.Caption)+'log.txt');
end;

if cbSavePattern.Checked then
begin
  Form1.DumpStack;
  imgDump.Picture.SaveToFile(ExtractFilePath(lbFileName.Caption)+'pattern.bmp');
end;

//----- Create ZIP output -----
if cbZIP.Checked then
begin
  TargetName:=ChangeFileExt(lbHTNName.Caption, '.ZIP');
  MakeZIP (TargetName,ExtractFilePath(lbHTNName.Caption));
  AddZIP (lbHTNName.Caption);
  if cbInclude.Checked then
  begin
    AddZIP(Application.ExeName);
    Memo1.Text:=Memo1.Text+'Extractor included'#13#10;
  end;
  CloseZIP;
  DeleteFile(lbHTNName.Caption);
  Memo1.Text:=Memo1.Text+'ZIP generated'#13#10;
end;

DecodeTime(Now-StartTime, hh, mm, ss, ms);
Memo1.Text:=Memo1.Text+Format('Time used %.2d:%.2d:%.2d'#13#10,[hh,mm,ss]);

end;

procedure TForm1.btnDecodeClick(Sender: TObject);
var buff2:array[0..127] of char;
  TargetName,Text:String;
  i,j,k,N:integer;
  Row,Col:integer;
  StartTime:TDateTime;
  iTemp:integer;
  iTemp64:int64;
  BMPBuff:TBitmap;
  BmpLine1:PByteArray;
  bTemp:Byte;
begin
  StartTime:=Now;

  if (StatusBar1.Panels[2].Text='') or (Not FileExists(lbHTNName.Caption)) then
  begin
    OpenFileDialog2.InitialDir:=ExtractFilePath(Application.ExeName);
    if not OpenFileDialog2.Execute then
    begin
      exit;
    end;
    lbHTNName.Caption:=OpenFileDialog2.FileName;
  end;

  //***** Open Half Tone Compressed File *****

```

```

TargetName:=lbHTNName.Caption;
Output:=TFileStream.Create(TargetName,fmOpenRead);

//----- Header -----
Output.Seek(00,soFromBeginning);
Output.Read(buff2,3);
if AnsiStrLComp(buff2,'HTN',3)=0 then Memo1.Text:='HTN File=OK'+#13#10 else
begin
Memo1.Text:='Error: Not HTN file.'+#13#10;
Output.Free;
exit;
end;

StatusBar1.Panels[2].Text:=ExtractFileName(TargetName);
StatusBar1.Panels[3].Text:='>>';
if FileExists(TargetName) then
TargetName:=ChangeFileExt(TargetName,'_x.bmp');
StatusBar1.Panels[4].Text:=ExtractFileName(TargetName);

Gauge1.Progress:=0;
Gauge1.MaxValue:=Image1.Picture.Bitmap.Height -1;
With Image2.Picture do
begin
Output.Seek(32,soFromBeginning);
Output.Read(iTemp,4);Bitmap.Height:=iTemp;
Output.Read(iTemp,4);Bitmap.Width:=iTemp;
Output.Read(iTemp,4);HTsize:=iTemp;

// --- Load Header ---
Output.Seek(48,soFromBeginning);
Output.Read(iTemp,4);StackCount:=iTemp;
Output.Read(iTemp*64,8);StackPos:=iTemp*64;
Memo1.Text:=Memo1.Text+'Half Tone Decode Algorithm'+#13#10;
Memo1.Text:=Memo1.Text+Format('Size = %d x %d'+#13#10,[Bitmap.Height,Bitmap.Width]);
Memo1.Text:=Memo1.Text+Format('HTsize=%d , Stack Count=%d'+#13#10,[HTsize,StackCount]);

//----- Create Buffer-----
BMPBuff:=TBitmap.Create;
BMPBuff.Assign(ImageBlank.Picture.Bitmap); // Initialize bitmap
BMPBuff.Width:=HTsize; BMPBuff.Height:=HTsize;

//----- Clear All Stack Buffer -----
For i:=PanelStack.ControlCount-1 downto 0 do
PanelStack.Controls[i].Destroy;

// --- Load Pattern to Stack ---
N:=0;
Output.Seek(StackPos,soFromBeginning);
For j:=0 to StackCount-1 do
begin
For i:=0 to BMPBuff.Height-1 do
begin
For k:=0 to (BMPBuff.Width div 8)-1 do
begin
BmpLine1:=BMPBuff.ScanLine[i];
Output.Read(bTemp,1);
BmpLine1[k]:=bTemp;
end;
end;
Form1.PutStack(BMPBuff,N);
end;

// --- Load Data ---
Output.Seek(64,soFromBeginning);

```

```

j:=0;
while (j<Image2.Picture.Bitmap.Height-1) do
begin
i:=0 ;
while i<Image2.Picture.Bitmap.Width-1 do
begin
Output.Read(bTemp,1);
Form1.PutHT(TImage(PanelStack.Controls[bTemp]).Picture.Bitmap,Image2.Picture.Bitmap,i,j);
i:=i+HTsize;
end;
j:=j+HTsize;
Gauge1.Progress:=j;
Application.ProcessMessages;
end;
end;

```

```
//----- End Decode -----
```

```
Gauge1.Progress:=Gauge1.MaxValue;
Output.Free;
```

```
//***** Create Bitmap Output File *****
```

```
Image2.Picture.SaveToFile(TargetName);
DecodeTime(Now-StartTime, hh, mm, ss, ms);
Memo1.Text:=Memo1.Text+Format('Time used %.2d:%.2d:%.2d'#13#10,[hh,mm,ss]);
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
BmpLineCount:=-1;
lbFileName.Caption:="";
lbHTNName.Caption:="";
StackCount:=0;
lbStatus.Caption:="";
```

```
end;
```

```
#####
#####
#####
```

```
procedure TForm1.btnCancelClick(Sender: TObject);
```

```
begin
```

```
EndLoop:=True;
```

```
end;
```

```
end.
```


ประวัติผู้เขียนวิทยานิพนธ์

นาย วุฒิชัย เจริญบุรี เกิดวันที่ 11 มกราคม 2519 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต ภาควิชาระบบควบคุมและเครื่องมือวัด คณะวิศวกรรมศาสตร์ มหาวิทยาลัยพระจอมเกล้าธนบุรี ปีการศึกษา 2540 เข้าศึกษาต่อที่ภาควิชาวิทยาศาสตร์ทางภาพถ่ายและเทคโนโลยีทางการพิมพ์ คณะวิทยาศาสตร์ จุฬาลงกรณ์



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย