系统工程师测试题目结果：

开发环境：VC++6.0 OPENCV SQLserver2005

效果图片如下：

各部分说明如下：

1.循环显示所提供的图像，并设置读图像的间隔时间。

1.1 添加picture控件，用于显示图片。

1.2 在工程的设置中，添加OPENCV系统头文件的路径。

1.3 在CDahengApp::InitInstance()里添加语句SetTimer(NULL,1,5000,NULL);设置时间间隔为5s。

1.4 在CDahengDlg中添加消息WM_TIMER，在函数OnTimer（）中填入如下内容：

```
int h;//h用于循环，将所用的图片放在某一个路径下，文件名为1.bmp，2.bmp。。。5.bmp
CString Open_Filename;
for (h=1;h<6;h++)
{
  switch (nIDEvent)
  {
 case 1:
    {
char c[10]=" ";
Open_Filename="D:\\系统工程师\\所用图片\\新建文件夹 (5)\\";
Open_Filename=Open_Filename+itoa(h,c,10);
Open_Filename=Open_Filename+".bmp";
IplImage* ipl = cvLoadImage(Open_Filename,-1); //读取图片，缓存到局部变量ipl中
  if(!ipl)
     return;
 if(TheImage)//TheImage为IplImage*类型的变量
cvZero(TheImage);
```

```
ResizeImage(ipl);//对读入的图片进行缩放，使其宽或高最大值者刚好等于预设的大小，
再复制到 TheImage 中
ShowImage(TheImage,IDC_picture);//显示图片，IDC_picture为picture控件的名称
SetTimer(1,5000,NULL); //定时，时间为5s
Invalidate(FALSE);
CDialog::OnTimer(1);//调用定时函数
cvReleaseImage(&ipl);//释放ipl
break;
}
default:
    break;
}
}
```

其中，函数 ResizeImage(ipl)和ShowImage(TheImage,IDC_picture)定义分别如下：

```
void CDahengDlg::ResizeImage(IplImage *img)
{
  int w = img->width;
  int h = img->height;
  int max = (w > h)?w: h;  // 找出宽和高中的较大值者
  float scale = (float) ( (float) max /500.0f );
  float scale2 = (float) ( (float) max / 500.0f ); // 计算将图片缩放到TheImage区
域所需的比例因子
  int nw = (int)( w/scale );
  int nh = (int)( h/scale2 );  // 缩放后图片的宽和高
   int tlx = (nw > nh)?0: (int)(256-nw)/2;
   int tly = (nw > nh)?(int)(256-nh)/2: 0; // 设置ROI 区域，用来存入图片 img
   cvSetImageROI( TheImage, cvRect( tlx, tly, nw, nh) );     // 对图片 img 进行缩
放，并存入到 TheImage 中
   cvResize( img, TheImage );
   cvResetImageROI( TheImage ); // 重置 TheImage 的 ROI 准备读入下一幅图片


void CDahengDlg::ShowImage(IplImage *img, UINT ID)
{
  CDC* pDC = GetDlgItem( ID ) ->GetDC();        // 获得显示控件的 DC
        HDC hDC = pDC ->GetSafeHdc();      // 获取 HDC(设备句柄) 来进行绘图操作
        CRect rect;
        GetDlgItem(ID) ->GetClientRect( &rect );
        int rw = rect.right - rect.left;            // 求出图片控件的宽和高
        int rh = rect.bottom - rect.top;
        int iw = img->width;                        // 读取图片的宽和高
        int ih = img->height;
        int tx = (int)(rw - iw)/2;          // 使图片的显示位置正好在控件的正中
```

```
            int ty = (int)(rh - ih)/2;
            SetRect( rect, tx, ty, tx+iw, ty+ih );
            CvvImage cimg;
            cimg.CopyOf( img );                          // 复制图片
            cimg.DrawToHDC( hDC, &rect );        // 将图片绘制到显示控件的指定区域内
 ReleaseDC( pDC );
 }
```

2.做一个dll检测动态库，实现图像反色。

2.1 创建一个Win32 Dynamic-Link Library的工程，在该工程里，添加一个C++源文件，其中编写的反色函数如下：

```
#include "cv.h"
#include "highgui.h"
_declspec(dllexport) void revcolor(IplImage * img)
{
    int step=img->widthStep; //取出图像每行所占的字节数
    int height=img->height;
    int width=img->width;
    uchar* data=(uchar*)img->imageData;
    for (int i=0;i<height;i++)
        for(int j=0;j<width;j++)
            for(int k=0;k<3;k++) //3为图像的通道
                data[i*step+j*3+k]=255-data[i*step+j*3+k];
}
```

2.2 Build，生成.dll文件和.lib文件

2.3 将上述两个文件复制到Daheng程序所在的目录下，并在Daheng工程设置的Link选项卡下，加入上述的lib文件。

2.4 在dahengDlg.cpp的前面加上extern void revcolor(IplImage * img)，对dll中的函数进行申明。

2.5 在daheng的对话框中再加入另外一个picture控件，名称为IDC_picture2。

2.6 在1.4的OnTimer（）函数体中加入如下语句：

```
IplImage* ipl2 = cvLoadImage(Open_Filename,-1);//ipl2和TheImage2均为反色图准备
if(!ipl2)
    return;
if(TheImage2)
cvZero(TheImage2);
revcolor(ipl2);
ResizeImage(ipl2);
ShowImage(TheImage2,IDC_picture2);//反色图显示在第二个控件中
cvReleaseImage(&ipl2);
```

3.连接数据库

3.1 在 SQL Server2005 中创建数据库 management，该数据库中有表 daheng 用来存放检测信息。daheng 表的结构如下：

属性名	类型	是否为主键	允许空
编号	int	是	
检测结果	char(10)	否	√
检测个数	int	否	√
废品个数	int	否	√
光电时差	int	否	√
检测速度	char(10)	否	√
检测产品	char(10)	否	√
产品图	image	否	√

3.2 在工程的 StdAfx.h 头文件里引入 ADO 库文件。代码如下所示：

```
#import "c:\program files\common files\system\ado\msado15.dll" no_namespace
rename("EOF","adoEOF")
```

3.3 定义 ADO 连接变量指针，在 daheng.h 文件的 class CDahengApp : public CwinApp 中加入

```
_ConnectionPtr      pConn;
```

在 dahengDlg.h 文件的 class CDahengDlg : public CDialog 中添加：

```
_RecordsetPtr m_pRecordset;              // 记录集接口
_ConnectionPtr pConn;
```

3.4 在 daheng.cpp 文件的 BOOL CDahengApp::InitInstance()里初始化 COM：

```
AfxOleInit();
pConn.CreateInstance(__uuidof(Connection));
try
{
    pConn->ConnectionString="Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=8878467;Initial Catalog=management;Data Source=IBPBVFNVQZAT8YI\\SQLEXPRESS";  //打开本地 Sql Server 库 management
    pConn->Open("","","",adConnectUnspecified);
}
catch (_com_error &e)
{
    AfxMessageBox(e.ErrorMessage());
    AfxMessageBox((LPCTSTR)e.Description());
}
```

3.5 在 dahengDlg.cpp 的#endif 下面添加：

```
extern CDahengApp theApp;          // 用 theApp 来获取库连接指针
```

在 BOOL CDahengDlg::OnInitDialog()函数中添加：

```
m_pRecordset.CreateInstance("ADODB.Recordset");
    try
    {
    m_pRecordset->Open("SELECT* FROM daheng", _variant_t((IDispatch*)theApp.pC
onn,true),adOpenStatic,adLockOptimistic,adCmdText);
    }
    catch(_com_error e)
    {
    AfxMessageBox("读取数据库失败!");
    }
```

3.6 在对话框中加入一个按钮，名称为“存入数据库”，为按钮添加相应的函数 OnSave,函数体如下：

```
pConn.CreateInstance(__uuidof(Connection));
CFile f;
CString  FilePathName;
CFileException e;
CString strSQL;
CString s1,s2,s3,s4,s5,s6;
GetDlgItemText(IDC_jiancejieguo,s1);
GetDlgItemText(IDC_jiancegeshu,s2);
GetDlgItemText(IDC_feipingeshu,s3);
GetDlgItemText(IDC_guangdianshicha,s4);
GetDlgItemText(IDC_jaincesudu,s5);
GetDlgItemText(IDC_jiancechanpin,s6); // 获取 6 个编辑框的内容  保存到 s1 到 s6
m_pRecordset->AddNew(); //添加新记录
m_pRecordset->PutCollect("检测结果",(_bstr_t)s1);
m_pRecordset->PutCollect("检测个数",(_bstr_t)s2);
m_pRecordset->PutCollect("废品个数",(_bstr_t)s3);
m_pRecordset->PutCollect("光电时差",(_bstr_t)s4);
m_pRecordset->PutCollect("检测速度",(_bstr_t)s5);
m_pRecordset->PutCollect("检测产品",(_bstr_t)s6);
if(TheImage)
{
    cvSaveImage("D:\\系统工程师\\所用图片\\新建文件夹 (5)\\10.bmp",TheImage);
//获取图片，保存在临时文件 10.bmp 中
    CFile f;
    CString FilePathName("D:\\系统工程师\\所用图片\\新建文件夹 (5)\\10.bmp");
```

```
CFileException e;
if(f.Open(FilePathName, CFile::modeRead | CFile::typeBinary, &e))
{
                int nSize = f.GetLength();              //先得到文件长度
    BYTE * pBuffer = new BYTE [nSize]; //按文件的大小在堆上申请一块内存
    if (f.Read(pBuffer, nSize) > 0 )      //把文件读到 pBuffer(堆上申请一块内存)
    {
        BYTE *pBuf = pBuffer;       ///下面这一大段是把 pBuffer 里的数据放到库
        VARIANT                 varBLOB;
        SAFEARRAY       *psa;
        SAFEARRAYBOUND      rgsabound[1];
        if(pBuf)
        {
            rgsabound[0].lLbound = 0;
            rgsabound[0].cElements = nSize;
            psa = SafeArrayCreate(VT_UI1, 1, rgsabound);
            for (long i = 0; i < (long)nSize; i++)
                SafeArrayPutElement (psa, &i, pBuf++);
            varBLOB.vt = VT_ARRAY | VT_UI1;
            varBLOB.parray = psa;
        m_pRecordset->GetFields()->GetItem("产品图")->AppendChunk(varBLOB);
        }
        delete [] pBuffer;      //删掉堆上申请的那一块内存
        pBuf=0;
    }
    f.Close();                  //关闭文件
}
    m_pRecordset->Update();
    m_pRecordset->Close();
}
```