

```
Dim myDevices As ZiDevices
Dim myStage As ZiFocusDevice
Dim myAF As clsAF09
Public DebStr As String
```

```
Private Sub cmdStart_Click()
```

```
Dim Z As Double
Dim Dist As Double
Dim Speed As Double
Dim Folder As String
Dim Offset As Double
txtReport.Text = ""
```

```
' check input data in text fields
```

```
If IsNumeric(txtOffset.Text) Then
    Offset = CDb1(txtOffset.Text)
```

```
Else
    MsgBox "Offset must be a number"
    Exit Sub
```

```
End If
```

```
If IsNumeric(txtDistance.Text) And CDb1(txtDistance.Text) > 0 Then
    Dist = CDb1(txtDistance.Text)
```

```
Else
    MsgBox "Distance must be a number > 0"
    Exit Sub
```

```
End If
```

```
If IsNumeric(txtSpeed.Text) And CDb1(txtSpeed.Text) > 0 Then
    Speed = CDb1(txtSpeed.Text)
```

```
Else
    MsgBox "Speed must be a number > 0"
    Exit Sub
```

```
End If
```

```
Dim myAlgo As AFAlgorithm
```

```
If optZeder.value = True Then myAlgo = Zeder_Weighted_Histogram_Sum
```

```
If optBrenner.value = True Then myAlgo = Brenner_Gradient
```

```
If optGroenYeo.value = True Then myAlgo = Groen_Yeo_Normalized_Variance
```

```
' select the actual stage z - position
```

```
Dim myZ As Double
myZ = 0
Z = myStage.Position
```

```
cmdStart.Enabled = False
```

```
' select the actual autofocus routine
```

```
' (the value 30000 sets the upper stage limit - adjust it to your microscope setting)
```

```
myAF.AF Z + Dist, Speed, Dist * 2, 30000, 1, Fluorescence, myZ, False, 0, myAlgo, chkAFMultiSpot.value
```

```
cmdStart.Enabled = True
```

```
txtReport.Text = myAF.FocusErrorReport & vbCrLf & myAF.DebugReport
```

```
' move the stage to the determined focus position
```

```
myZ = myAF.AF_FL_Z + Offset
myStage.Position = myZ
myStage.UpdateParameters False
```

```
End Sub
```

```
Private Sub txtReport_Change()
```

```
End Sub
```

```
Private Sub UserForm_Initialize()
```

```
' initialize the stage
```

```
Set myDevices = ZiApplication.Devices
```

```
Set myStage = myDevices.Open("Focus")
```

```
myStage.Calibrate False
```

```
' initialize the autofocus
```

```
Set myAF = New clsAF09
```

```
End Sub
```

start autofocus run

Distance +/- current z position [microns]

10

Speed [microns per second]:

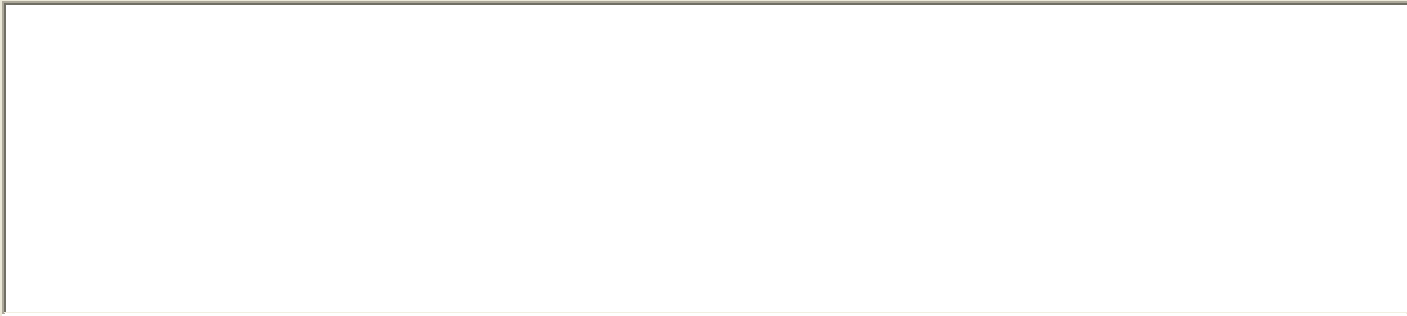
5

Offset [microns]:

0

Measure focus parameter on multiple spots

- Zeder - weighted histogram sum
- Brenner - Brenner gradient
- Groen, Yeo - normalized variance



modMain - 1

```
Public Sub RunAutofocus()  
    frmAF09.Show False  
End Sub
```

```
' class for interpolation calculations
```

```
Public ImgInFocus_I As Double
Public ImgInFocus_T As Double
Public ImgPreFocus_T As Double
Public ImgPostFocus_T As Double
Public ImgInFocus_Para As Double
Public ImgPreFocus_Para As Double
Public ImgPostFocus_Para As Double
Public InFocusTime_T As Double
Public InFocusTime_Para As Double
Public PreTime_Z As Double
Public PostTime_Z As Double
Public PreTime_T As Double
Public PostTime_T As Double
Public InFocus_Z As Double
```

```
' I = index, T = time, Z = z - poition
```

```
Public Sub CalcInFocusTime() ' calculate the in focus time using the focus parameters from the live
image collection
```

```
  ' quadratic interpolation
  ' approximation of the real focus time by a polynome 2nd order  $y = Ax^2+Bx+C$ 
  ' the polynomial is defined by the maximum point (x2,y2) and its neighbouring two points (x1,y1), (x3,y3)
```

```
  Dim x1 As Double: Dim x2 As Double: Dim x3 As Double
  Dim y1 As Double: Dim y2 As Double: Dim y3 As Double
  Dim a As Double: Dim B As Double: Dim C As Double
  x1 = ImgPreFocus_T: y1 = ImgPreFocus_Para
  x2 = ImgInFocus_T: y2 = ImgInFocus_Para
  x3 = ImgPostFocus_T: y3 = ImgPostFocus_Para
```

```
  If x1 = 0 Or x2 = 0 Or x3 = 0 Or y1 = 0 Or y2 = 0 Or y3 = 0 Then
    InFocusTime_T = 0
```

```
  Else
```

```
    a = (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / ((x1 - x2) * (x1 - x3) * (x3 - x2))
```

```
    B = (x1 * x1 * (y2 - y3) + x2 * x2 * (y3 - y1) + x3 * x3 * (y1 - y2)) / ((x1 - x2) * (x1 - x3) * (x2 - x3))
```

```
    C = (x1 * x1 * (x2 * y3 - x3 * y2) + x1 * (x3 * x3 * y2 - x2 * x2 * y3) + x2 * x3 * y1 * (x2 - x3)) / ((x1 - x2) * (x1 - x3) * (x2 - x3))
```

```
    If a <> 0 Then
```

```
      InFocusTime_T = -B / (2 * a)
```

```
      InFocusTime_Para = C - ((B * B) / (4 * a))
```

```
    Else
```

```
      InFocusTime_T = 0
```

```
    End If
```

```
  End If
```

```
End Sub
```

```
Public Sub CalcInFocusZ() ' calculate the in focus z - position using the in focus time and the z -
position collection
```

```
  ' linear interpolation
```

```
  If PreTime_T = PostTime_T Then
```

```
    InFocus_Z = 0
```

```
  Else
```

```
    InFocus_Z = PreTime_Z - ((PreTime_Z - PostTime_Z) * (PreTime_T - InFocusTime_T) / (PreTime_T - PostTime_T))
```

```
  End If
```

```
End Sub
```

```
' class for live image event analysis
```

```
Public Img As New ZiImage
```

```
Public T As Double
```

```
Private MIZpara(10) As Double ' focus paramater for the 9 spots & entire image calculated using the  
MIZ-methode
```

```
Private Declare Function timeBeginPeriod Lib "winmm.dll" (ByVal uPeriod As Long) As Long
```

```
Private Declare Function timeGetTime Lib "winmm.dll" () As Long
```

```
Public Property Get MIZParameter(index As Integer) As Double
```

```
    MIZParameter = MIZpara(index)
```

```
End Property
```

```
Public Sub AnalyzeImage(ByRef algo As AFAlgorithm, ByRef MultiSpot As Boolean)
```

```
    timeBeginPeriod 1
```

```
    Dim myMean As Double
```

```
    Dim myMeas As ZiMeasure: Set myMeas = New ZiMeasure ' initialize a measure object
```

```
    Dim myAcc As ZiImageAccessor: Set myAcc = New ZiImageAccessor
```

```
    Dim myReg As ZiGeomRegion: Set myReg = New ZiGeomRegion ' initialize a region object
```

```
    Dim myW As Integer ' width of subimage
```

```
    Dim myH As Integer ' height of subimage
```

```
    myW = Round(Img.Info.Width / 3) - 1
```

```
    myH = Round(Img.Info.Height / 3) - 1
```

```
    Dim i As Integer: i = 0
```

```
    Dim m As Integer: m = 0
```

```
    Dim n As Integer: n = 0
```

```
    Dim myHisto() As Long: ReDim myHisto(0 To 4095)
```

```
    Dim h As Long: Dim MeanGrayValue As Double
```

```
    Dim PixelArr() As Double
```

```
    Dim ST As Double
```

```
    Dim ST2 As Double
```

```
    myMeas.FeatureParameter(ziFeatureParaHistoBitCount) = 16
```

```
    myMeas.InputImage = Img
```

```
' measure the 9 spots
```

```
For m = 1 To 3
```

```
    For n = 1 To 3
```

```
        i = i + 1
```

```
        myReg.SetRectangle (n - 1) * myW, (m - 1) * myH, n * myW, m * myH
```

```
        myMeas.Region = myReg
```

```
        If algo = Zeder_Weighted_Histogram_Sum Then
```

```
            h = myMeas.FeaturesLong(ziFeaturesLongTypeDensHisto, myHisto)
```

```
            ST = timeGetTime
```

```
            MIZpara(i) = CalcMIZparam(myHisto)
```

```
            ST2 = timeGetTime
```

```
            myAcc.Detach
```

```
        End If
```

```
        If algo = Brenner_Gradient Then
```

```
            myAcc.Attach Img
```

```
            myAcc.SetRectangle (n - 1) * myW, (m - 1) * myH, n * myW, m * myH
```

```
            myAcc.GetArrayDouble PixelArr, CLng(myW), CLng(myH)
```

```
            ST = timeGetTime
```

```
            MIZpara(i) = CalcBrennerParam(PixelArr, myW, myH)
```

```
            myAcc.Detach
```

```
            ST2 = timeGetTime
```

```
        End If
```

```
        If algo = Groen_Yeo_Normalized_Variance Then
```

```
            myAcc.Attach Img
```

```
            myAcc.SetRectangle (n - 1) * myW, (m - 1) * myH, n * myW, m * myH
```

```
            myAcc.GetArrayDouble PixelArr, CLng(myW), CLng(myH)
```

```
            ST = timeGetTime
```

```
            MIZpara(i) = CalcVarParameter(PixelArr, myW, myH)
```

```
            ST2 = timeGetTime
```

```
            myAcc.Detach
```

```
        End If
```

```
    Next n
```

```
Next m
```

```
If MultiSpot = False Then
```

```
    myMean = 0: i = 0
```

```
    For m = 1 To 3
```

```
        For n = 1 To 3
```

```
            i = i + 1
```

```

        myMean = myMean + MIZpara(i)
    Next n
Next m
myMean = myMean / 9
For i = 1 To 9
    MIZpara(i) = myMean
Next i
End If

Set myMeas = Nothing
Set myReg = Nothing
Set myAcc = Nothing
End Sub

Private Function CalcBrennerParam(PixArr() As Double, myW As Integer, myH As Integer) As Double
    Dim i As Integer: Dim j As Integer: Dim Sum As Double
    Sum = 0
    For i = 0 To myW - 1
        For j = 0 To myH - 3
            Sum = Sum + (PixArr(i, j) - PixArr(i, j + 2)) * (PixArr(i, j) - PixArr(i, j + 2))
        Next j
    Next i
    CalcBrennerParam = Sum
End Function

Private Function CalcVarParameter(PixArr() As Double, myW As Integer, myH As Integer) As Double
    Dim i As Integer: Dim j As Integer: Dim Sum As Double
    Dim Prod As Double
    Sum = 0
    Dim meanGV As Double
    meanGV = 0
    For i = 0 To myW - 1
        For j = 0 To myH - 1
            meanGV = meanGV + PixArr(i, j)
        Next j
    Next i
    meanGV = meanGV / myW / myH

    Prod = CDBl(myW) * CDBl(myH) * CDBl(meanGV)
    For i = 0 To myW - 1
        For j = 0 To myH - 1
            Sum = Sum + (PixArr(i, j) - meanGV) * (PixArr(i, j) - meanGV)
        Next j
    Next i
    Sum = Sum / Prod
    CalcVarParameter = Sum
End Function

Private Function CalcMIZparam(histo() As Long) As Double
    Dim HistogramSum As Double
    HistogramSum = 0
    For i = 0 To 4095
        HistogramSum = HistogramSum + ((histo(i) ^ (1 / 5)) * i * i * i * i * i / 1E+15)
    Next i
    CalcMIZparam = HistogramSum
End Function

Public Sub SaveImage(FileName As String)
    On Error Resume Next
    Img.SaveAs FileName, False
End Sub

```

clsAF_ZPos - 1

' class for saving z - position events

Public Z As Double

Public T As Double

```
' ***** Michael Zeder 2009, Limnological Station, Institute of Plant Biology, University of Zürich
' ***** zeder_michael@hotmail.com - www.technobiology.ch
' *****
' ***** Autofocus routine
' *****
' ***** available under general public licence
' ***** published in: Cytometry A, 2009
```

Option Explicit

```
Public DebStr As String
' IMPORTANT - THE FOLLOWING CONSTANTS HAVE TO BE ADAPTED TO THE USERS SYSTEM

' maximal allowed time for one autofocus run in milliseconds
Const MaximalFocusTime As Double = 50000
' lowest possible stage position allowed in the autofocus run
Const LowesPossibleZPosition As Double = 10000

' option to save focus data in a text file / an image as jpg
' files are saved in the folder C:\FocusData
Const SaveFolder As String = "C:\FocusData"
Const SaveFocusData As Boolean = True ' save the FocusScore of each image into a text file
Const SaveFocusImage As Boolean = True ' save the focused image of each autofocus run as a jpg
Const SaveHistoData As Boolean = False ' save the histogram of each live image into a text file
Public SaveAllImages As Boolean ' save all live images during autofocus run
Public SaveSmallReport As Boolean
Public SaveDir As String
Public DebugReport As String

Dim F1 As Integer ' file number for SaveHistogramData file

Dim FocusZPos As Double ' optimal focus position

Dim strAcquisition As String
Dim Devices As ZiDevices
Dim AFCamera As ZiAxioCamDevice
Dim ZiIpUtilityFunction As New ZiIPUtilityFunctionPool
Dim ZiIPStatisticsFunction As ZiStatisticsFunction
Dim Statistics As Variant
Dim Histogram As Variant

Public AFReport As String

Dim FocusDevice As ZiFocusDevice
Dim Parameters As ZiParameters

' time measure variables
Dim TimeStartAcq As Double
Dim TimeEndAcq As Double
Dim TimeStartCalc As Double
Dim TimeEndCalc As Double
Dim TimeStartIA As Double
Dim TimeEndIA As Double

' collection for images
Dim IArr As Collection
Dim ZArr As Collection
Dim CArr As Collection
Dim LI As clsAF_LiveImage
Dim ZP As clsAF_ZPos
Dim Calc As clsAF_Calc

Dim i As Integer
Dim k As Integer

Dim WithEvents AF_I As ZiImage
Dim WithEvents AF_Z As ZiParameter

' API-functions for time measurement - timer with 1 ms resolution (requires winm.dll)
Private Declare Function timeBeginPeriod Lib "winmm.dll" (ByVal uPeriod As Long) As Long
Private Declare Function timeGetTime Lib "winmm.dll" () As Long

Public Enum AFMode
    Brightfield = 0
    Fluorescence = 1
```



```
End Enum
```

```
Public Enum AFAlgorithm
    Zeder_Weighted_Histogram_Sum = 0
    Brenner_Gradient = 1
    Groen_Yeo_Normalized_Variance = 2
End Enum
```

```
Public AFBFreport As String
Public AFFLreport As String
Public FocusError As Boolean
Public FocusErrorReport As String
Public AFBF_Z As Double
Public AFFL_Z As Double
Public AFBFrange As Double
Public AFFLrange As Double
Public TimeStamp As String
```

```
Private Sub AF_I_DataChanged(ByVal Hint As Long, ByVal Param1 As Long, ByVal Param2 As Long)
    ' event handler for camera live images
    Set LI = New clsAF_LiveImage
    Set LI.Img = AFCamera.Live.Image.Clone
    LI.T = timeGetTime - TimeStartAcq
    IArr.Add LI
End Sub
```

```
Private Sub AF_Z_ValueChanged(ByVal Key As String, ByVal TagId As Long, ByVal TagPath As String, ByVal value As Variant, ByVal ValueStatus As Long)
    ' event handler for stage - z positions
    Set ZP = New clsAF_ZPos
    ZP.Z = value
    ZP.T = timeGetTime - TimeStartAcq
    ZArr.Add ZP
End Sub
```

```
Private Sub Class_Initialize()
    ' *** Constructor ***

    ' set the devices that are required:
    ' the camera (AFCamera / ActiveAcquisition) and the stage (FocusDevice)
    Set Devices = ZiApplication.Devices
    strAcquisition = ZiApplication.ActiveAcquisition
    Set AFCamera = Devices.Open(strAcquisition)
    Set ZiIPStatisticsFunction = ZiIpUtilityFunction.CreateStatisticsFunction

    Set FocusDevice = Devices.Open("Focus")
    Set Parameters = FocusDevice.Parameters

    ' set the timer resolution to 1 ms
    timeBeginPeriod 1

    SaveAllImages = False ' save all live images during autofocus run
    SaveSmallReport = False
    SaveDir = "C:\AF_I\"
End Sub
```

```
Private Sub Class_Terminate()
    Set Devices = Nothing
    Set AFCamera = Nothing
    Set ZiIPStatisticsFunction = Nothing
    Set Parameters = Nothing
    Set FocusDevice = Nothing
End Sub
```

```
Public Sub AF(ByVal Startpos As Double, ByVal AbsSpeed As Double, ByVal Dist As Double, _
    ByVal Z_Max As Double, Stringency As Double, Mode As AFMode, ByRef FocusZPos As Double, ByRef FocusError As Boolean, ByRef AFRange As Double, AutofocusAlgorithm As AFAlgorithm, MultiSpot As Boolean)
```

```
    ' set the variables for focus error to false at the beginning
    AFReport = ""
```

```

FocusErrorReport = ""
DebugReport = ""
FocusError = False
Dim i As Integer: Dim j As Integer
' check if stage is calibrated
If FocusDevice.Calibrated = False Then
    MsgBox "please calibrate the stage!"
    Exit Sub
End If

Dim EndPos As Double: EndPos = Startpos - Dist ' EndPosition

' reset the collections for live image and z - position
Set IArr = New Collection
Set ZArr = New Collection
Set CArr = New Collection

' check if the StartPos is higher than the highest possible startposition defined by Z_Max
' if so, then set the StartPos to Z_Max in order not to damage the objective
If Startpos > Z_Max Then Startpos = Z_Max

' drive the stage to the startposition
FocusDevice.MoveTo Startpos, False

' set the event handlers for live image and stage
Set AF_I = AFCamera.Live.Image
Set AF_Z = Parameters.Item("Position")
AFCamera.Live.Speed = 1 ' CameraLive Speed - 0:slow - 1:medium - 2:fast
AFCamera.UpdateParameters False
' measure start time
TimeStartAcq = timeGetTime
' start to drive the stage downwards with the given speed (AbsSpeed)
FocusDevice.Go -AbsSpeed, True

Do ' collect live images and z - positions until z - endposition was reached or time is up
    DoEvents ' wait for other processes
    If FocusDevice.Position < EndPos Then Exit Do
    If timeGetTime - TimeStartAcq > MaximalFocusTime Then ' error-check
        Debug.Print "Unexpected Error in focus routine: focussing for more then " & MaximalFocusTime & " milliseconds!"
        FocusError = True
        FocusErrorReport = FocusErrorReport & " FErr: FocusTime overflow "
        Exit Do
    End If
Loop

Set AF_I = Nothing ' destroy the event handler for live image
Set AF_Z = Nothing ' destroy the event handler for live z - position

FocusDevice.Stop True ' the stage is stopped
TimeEndAcq = timeGetTime
TimeStartIA = timeGetTime

If IArr.Count < 10 Then ' if there are less than 10 images in the collection, abort!
    FocusError = True
    FocusErrorReport = FocusErrorReport & " FErr: less than 10 liveimages "
    Exit Sub
End If

For i = 1 To IArr.Count
    IArr(i).AnalyzeImage AutofocusAlgorithm, MultiSpot
    ' analyze the live images and calculate the focus measures
Next i

TimeEndIA = timeGetTime
TimeStartCalc = timeGetTime
For i = 1 To 9
    Set Calc = New clsAF_Calc ' create an entity for data calculation
    CArr.Add Calc
Next i

For i = 1 To 9
    CArr(i).ImgInFocus_I = 2
    CArr(i).ImgInFocus_Para = IArr(2).MIZParameter(i) ' assign the data of the second object
Next i

```

```

For i = 2 To IArr.Count - 1 ' find the in focus image spots in the live image collection
  For j = 1 To 9
    If CArr(j).ImgInFocus_Para < IArr(i).MIZParameter(j) Then
      CArr(j).ImgInFocus_I = i
      CArr(j).ImgInFocus_T = IArr(i).T
      CArr(j).ImgInFocus_Para = IArr(i).MIZParameter(j)
      CArr(j).ImgPreFocus_T = IArr(i - 1).T
      CArr(j).ImgPreFocus_Para = IArr(i - 1).MIZParameter(j)
      CArr(j).ImgPostFocus_T = IArr(i + 1).T
      CArr(j).ImgPostFocus_Para = IArr(i + 1).MIZParameter(j)
    End If
  Next j
Next i

For i = 1 To 9 ' interpolate the in focus time of the spots
  CArr(i).CalcInFocusTime
  If CArr(i).InFocusTime_T = 0 Then
    FocusError = True
    FocusErrorReport = FocusErrorReport & " FErr: FocusTime = 0 "
    Exit Sub
  End If
Next i

For i = 1 To ZArr.Count - 1 ' find the z - position before and after the in focus time of each
spot
  For j = 1 To 9
    If ZArr(i).T <= CArr(j).InFocusTime_T Then
      If ZArr(i + 1).T > CArr(j).InFocusTime_T Then
        CArr(j).PreTime_Z = ZArr(i).Z
        CArr(j).PreTime_T = ZArr(i).T
        CArr(j).PostTime_Z = ZArr(i + 1).Z
        CArr(j).PostTime_T = ZArr(i + 1).T
      End If
    End If
  Next j
Next i

For i = 1 To 9 ' interpolate the in focus time of the spots
  CArr(i).CalcInFocusZ
  If CArr(i).InFocus_Z = 0 Then FocusError = True
Next i
TimeEndCalc = timeGetTime

' determine overall focus position
Dim zpos As Double
Dim FocusMeasureMaxIndex As Double
Dim FocusMeasureMax As Double
FocusMeasureMaxIndex = 0
FocusMeasureMax = 0
zpos = 0
j = 0
For i = 1 To 9
  If CArr(i).ImgInFocus_Para > FocusMeasureMax Then
    FocusMeasureMaxIndex = i
    FocusMeasureMax = CArr(i).ImgInFocus_Para
  End If
  If CArr(i).ImgInFocus_Para > 20 Then ' this is an absolute threshold for MIZ-function
    zpos = zpos + CArr(i).InFocus_Z
    j = j + 1
  End If
Next i
If j > 0 Then
  FocusZPos = zpos / j
Else
  FocusZPos = CArr(FocusMeasureMaxIndex).InFocus_Z
End If

' determine focus range
Dim Min As Double
Dim Max As Double
Min = CArr(1).InFocus_Z
Max = CArr(1).InFocus_Z
For i = 2 To 9
  If Min < CArr(i).InFocus_Z Then Min = CArr(i).InFocus_Z
  If Max > CArr(i).InFocus_Z Then Max = CArr(i).InFocus_Z
Next i
AFRange = Max - Min

```

```

TimeStamp = Format(Date, "yymmdd") & "_" & Format(Time, "hhmmss")
If Mode = Brightfield Then
    AFBF_Z = FocusZPos
    AFBFrange = AFRange
Else
    AFFL_Z = FocusZPos
    AFFLrange = AFRange
End If

If SaveAllImages = True Then
    For i = 1 To IArr.Count
        If Mode = Brightfield Then
            IArr(i).SaveImage (SaveDir & "I_BF_" & TimeStamp & "-" & CStr(i) & ".jpg")
        Else
            IArr(i).SaveImage (SaveDir & "I_FL_" & TimeStamp & "-" & CStr(i) & ".jpg")
        End If
    Next i
End If

Dim str As String

' debug report
DebugReport = DebugReport & "Z Pos: " & vbTab & FocusZPos & " Range: " & vbTab & AFRange & vbCrLf & vbCrLf
str = "TimeStamp" & vbTab & "AcqTime" & vbTab & "IATime" & vbTab & "CalcTime" & vbTab & "Speed" & vbTab & "Dist" & vbTab & "NrImages" & vbTab & "NrZPos" & vbTab & "ErrRep"
DebugReport = DebugReport & str & vbCrLf
str = TimeStamp & vbTab & TimeEndAcq - TimeStartAcq & vbTab & TimeEndIA - TimeStartIA & vbTab & TimeEndCalc - TimeStartCalc & vbTab & AbsSpeed & vbTab & Dist & vbTab & IArr.Count & vbTab & ZArr.Count & vbTab & FocusErrorReport
DebugReport = DebugReport & str & vbCrLf & vbCrLf
str = "in-focus parameters for z, time, index, para for 9 subimages" & vbCrLf & "Subimage" & vbTab & "z" & vbTab & "time" & vbTab & "index" & vbTab & "para" & vbCrLf
For i = 1 To 9
    str = str & i & vbTab & Round(CArr(i).InFocus_Z, 2) & vbTab & Round(CArr(i).InFocusTime_T, 0) & vbTab & Round(CArr(i).ImgInFocus_I, 1) & vbTab & Round(CArr(i).ImgInFocus_Para, 1) & vbCrLf
Next i
DebugReport = DebugReport & str & vbCrLf & vbCrLf
str = "Time, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10"
DebugReport = DebugReport & str & vbCrLf
For i = 1 To IArr.Count
    str = IArr.Item(i).T
    For j = 1 To 9
        str = str & "," & CStr(Round(IArr.Item(i).MIZParameter(j), 1))
    Next j
    DebugReport = DebugReport & str & vbCrLf
Next i
DebugReport = DebugReport & vbCrLf
str = "Time,Zposition"
DebugReport = DebugReport & str & vbCrLf
For i = 1 To ZArr.Count
    str = ZArr.Item(i).T & "," & Round(ZArr.Item(i).Z, 2)
    DebugReport = DebugReport & str & vbCrLf
Next i
DebugReport = DebugReport & "-----" & vbCrLf & DebS
tr

' autofocus report
AFReport = CStr(Round(FocusZPos, 2)) & ","
For i = 1 To 9
    AFReport = AFReport & Round(CArr(i).InFocus_Z, 2) & ","
Next i
For i = 1 To 9
    AFReport = AFReport & Round(CArr(i).ImgInFocus_Para, 0) & ","
Next i
End Sub

```

