Designation: E2955 – 13

Standard Practice for Simulating Profilograph Response to Longitudinal Profiles of Traveled Surfaces[1]

This standard is issued under the fixed designation E2955; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 These practices cover the calculation of profilograph response to longitudinal profiles of traveled surface roughness.

1.2 These practices utilize computer simulations to obtain profilograph responses including: surface sensing wheel (recording wheel) vertical displacement, and reference platform wheels (supporting wheels) motions, as a function of distance.

1.3 These practices present standard profilograph simulations for use in the calculations.

1.4 The values stated in SI units are to be regarded as standard. The values given in parentheses are for information only and are not considered standard.

1.5 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

2. Referenced Documents

2.1 *ASTM Standards:*[2]

E950 Test Method for Measuring the Longitudinal Profile of Traveled Surfaces with an Accelerometer Established Inertial Profiling Reference

E1274 Test Method for Measuring Pavement Roughness Using a Profilograph

3. Terminology

3.1 See Test Method E1274.

4. Summary of Practices

4.1 These practices use a measured profile (see Test Method E950) or a synthesized profile as part of a computer simulation to obtain the response of the surface sensing wheel of a profilograph. The first practice uses a kinematic profilograph simulation model to obtain typical profilograph profiles as a function of distance. The second practice obtains the vertical displacement of the surface sensing wheel using the surface sensing wheel size, supporting wheel locations, and the spacing between the supporting wheels. The output is the continuous vertical motion of the surface sensing wheel of the simulated profilograph for a predetermined distance. The units are relative vertical movement (millimetres (inches)) at longitudinal surface profile location (metres (feet)). Pertinent information affecting the results must be noted.

5. Significance and Use

5.1 These practices provide a means for evaluating traveled surface-roughness characteristics directly from a measured profile. The calculated output values represent profilograph response to traveled surface roughness.

5.2 These practices provide pavement surface profiles to evaluate pavement condition using the profilograph index.

6. Apparatus

6.1 *Computer*—The computer is used to calculate the surface sensing wheel vertical displacement of a profilograph with twelve reference platform wheels in response to a traveled surface profile using a synthesized profile or a profile obtained in accordance with Test Method E950 as the input. Filtering shall be provided to permit calculation (for example, butterworth low pass or moving average low pass filters).

6.2 *Data-Storage Device*—A data-storage device shall be provided for the reading of profiles and the recording and long-term storage of computed data. Profile data shall be scaled to maintain a resolution of 0.025 mm (0.001 in.) and to accommodate the full range of amplitudes encountered during normal profile-measuring operations. The device shall not contribute to the recorded data any noise amplitude larger than 0.025 mm (0.001 in.).

6.3 *Simulation Input*—Digital profile recordings of pavement surface profiles shall be obtained in accordance with Test Method E950 or synthesized. The profile must be recorded at intervals no greater than 30.48 mm (1.2 in.). When more than one path of a traveled surface is measured, the recorded profile

---

[1] This practice is under the jurisdiction of ASTM Committee E17 on Vehicle - Pavement Systems and is the direct responsibility of Subcommittee E17.33 on Methodology for Analyzing Pavement Roughness.

Current edition approved Dec. 1, 2013. Published January 2014. DOI: 10.1520/E2955-13.

[2] For referenced ASTM standards, visit the ASTM website, www.astm.org, or contact ASTM Customer Service at service@astm.org. For *Annual Book of ASTM Standards* volume information, refer to the standard's Document Summary page on the ASTM website.

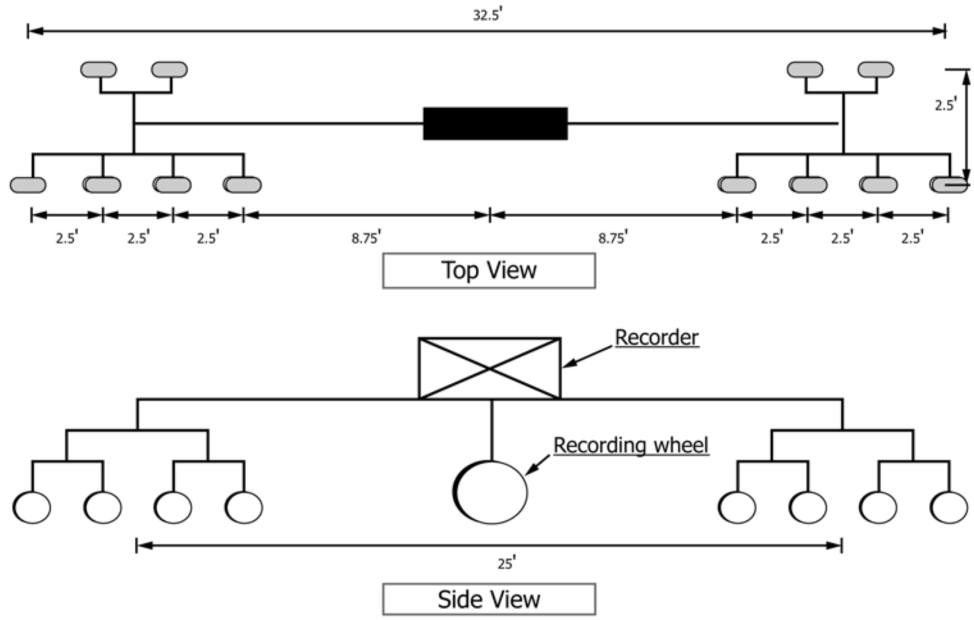

FIG. 1 A Typical California Profilograph Configuration

data for the paths shall be at the same longitudinal location along the measured profiles to avoid phase shift between the paths. The recorded profile shall include all of the noted field data described in the Procedure (Data Acquisition) and Report sections of Test Method E950. The length of the pavement surface profile must be reported with the results; however, caution must be exercised to ensure that transients in the simulation do not influence the results. It is recommended that the profile input to the simulation include a lead-in section at least 150m (500 ft) long to eliminate the effects of start-up transients.

7. Profilograph Simulation Program

7.1 These practices use a typical California profilograph model with twelve supporting wheels as shown in Fig. 1. The mathematical model is described in FHWA report.[3] The mathematical model was developed with the following assumptions:

7.1.1 All structural connections are perfectly rigid,

7.1.2 All hinge joints and wheel bearings are frictionless, and

7.1.3 All wheels are at a point-contact with the pavement surface at all times.

7.2 The calculation procedure to produce the profilograph vertical movement at position *x*, *R(x)*, from the recorded pavement surface profile is defined as:[4]

$$R(x) = \left(\sum_{i=1}^{N} C_i \, P_i \, (x - d_i) \right) - P_r(x - d_r) \qquad (1)$$

where:

N = the number of wheels in the left and right side of the support system,
P_I = the recorded profile, on which the i^{th} wheel is traveling (subscript *r* refer to those of the surface sensing wheel),
d_i = the offset distance from the location x for each wheel (subscript *r* refer to those of the surface sensing wheel), and
C_i = the influence coefficient corresponding to the i^{th} wheel (C_i = 1⁄16 for the 8 right side wheels and C_i = 1⁄8 for the 4 left side wheels).

7.2.1 As the California Profilograph shown in Fig. 1 travels on a pavement, 3 profile paths will be passed: 4 left wheels travel on one path, 8 right wheels travel on another and the third is under the recording wheel. Comparing the traces from the mathematical model using the same measured profile for all wheels with the real recorded traces using the profilograph on the same pavement in the study,[5] showed that both traces are similar to each other. Therefore, it can be assumed that all wheels travel on the same profile when computing the "Profilograph Index" in 8.2.

7.2.2 As defined in Test Method E1274, the blanking band is a band of uniform height with its longitudinal center positioned optimally between the highs and lows of the profilograph trace depicting at least 100 ft of pavement. In general, a band height between 0.0 mm (0.0 in.), 2.54 mm (0.1 in.), or 5.08 mm (0.2 in.) is used in computing the "Profilograph Index" in 8.2.

7.2.3 Scallops are the excursions of the trace above and below the blanking band as defined in Test Method E1274. The vertical maximum of a scallop must not be less than 0.762 mm

[3] Kulakowski, B. T. and Wambold, J. C., "Development of Procedures for the Calibration of Profilographs", Report No. FHWA-DP- 89-110, Federal Highway Administration, August 1989.

[4] See http://www.airporttech.tc.faa.gov/NAPTF/Download/Roughness/CA%20Profilograph%20Index%20Calculation%20Model.doc.

[5] Walker, P. S. and Lin, H. T., "Profilograph Correlation Study with Present Serviceability Index (PSI)", FHWA-88-072-002, Federal Highway Administration, 1988.

(0.03 in.) and the longitudinal length must be longer than 0.6 m (2 ft) as defined in Test Method E1274 and previous studies.[6,7]

8. Applications

8.1 *Excessive Height Template*—A clear plastic piece marked with a 25.0 ± 0.5 mm (1.00 ± 0.02 in.) line that is the stipulated cutoff height distance from a straightedge on the template. The cutoff height is defined as 7.614 mm (0.3 in.), 10.152 mm (0.4 in.), or 12.69 mm (0.5 in.).

8.1.1 The procedure for manually determining bump height from a template and a profilograph paper trace has been implemented using a simulated profilograph profile. A Bump Height Template Index (BHTI) is defined as:

$$BHTI = \frac{ComputedBumpHeight}{Bump\ HeightLimit} = \frac{h_B}{h_L} \quad (2)$$

8.1.2 Therefore, when BHTI exceeds a value of one the limit has been exceeded. Bump Height Limit, h_L, in the equation is equal to the cutoff height.

8.2 *Profilograph Index (PI)*—Profilograph Index is defined by the blanking band, scallop height, and total number of scallops in the segment, *m*. The equations are defined as:

$$PI_{segment} = \frac{\sum_{j=1}^{m} Smax,j}{L}(in./mile) \quad (3)$$

where:

$$PI = \frac{\sum_{k=1}^{s} PI_k \times L_k}{\sum_{k=1}^{s} L_k}(in./mile) \quad (4)$$

[6] American Concrete Pavement Association, "Constructing Smooth Concrete Pavements," ACPA TB-006.0-C, 1990.

[7] "Operation of California Prafilograph and Evaluation of Profiles", California Test 526, 1978.

where:

$PI_{segment}$ = profilograph index for a segment,
$Smax, j$ = the maximum of absolute height in the j^{th} scallop,
L = the segment length.

8.2.1 Subscript *k* indicates the k^{th} segment in the pavement divided into *s* segments. *L* is the segment length.

9. Calibration

9.1 There is no calibration involved in the use of these practices. The profile collection equipment should be calibrated in accordance with the manufacturer's requirements.

10. Report

10.1 Report at least the following information for this practice:

10.1.1 Data from profiles obtained in accordance with Test Method E950 including date, the time of day of the measurement, or the date of the synthesized profile,

10.1.2 Profilograph simulation program used,[8]

10.1.3 Parameter values used to calculate Profilograph Index,

10.1.4 Parameter values used to calculate Bump Height Template Index, and

10.1.5 Results of the analysis.

11. Keywords

11.1 bump height; ProFAA; profilograph index; profilograph simulation; ProVAL

[8] Currently two computer programs, ProFAA and ProVAL, provide profilograph simulations and analysis. "ProFAA" is the Federal Aviation Administration's computer program for computing pavement elevation profile roughness indexes (http://www.airporttech.tc.faa.gov). "ProVAL" is an engineering software application used to view and analyze pavement profiles (http://www.roadprofile.com).

APPENDIXES

(Nonmandatory Information)

X1. PROFILOGRAPH INDEX COMPUTER PROGRAM

X1.1 Included in this appendix is the coding in Visual Basic.NET language for a computer subroutine (ProFAA), SUBROUTINE PROFILOGRAPH INDEX (PI), (see Fig. X1.1), which calculates the Profilograph Index as prescribed by this practice. A sample subroutine is also included, which when executed, prompts the user for the name of a data file containing the profile data to be processed and the parameters needed by the subroutine to compute the PI. The subroutine is called and returns the computed PI values to the main program which then displays them.

X1.2 The sample program can process data files containing two profile tracks in either SI or inch-pound units. For example, "UnitsOut.ft" in Fig. X1.1 is utilized to convert inch-pound units (feet) to SI units (metres).

```
Partial Public Class MainWindow ' CAProfilograph

  ' Subroutines from ProFAA 2.0 for computing:
  ' 1. CA Profilograph measuring wheel displacement in response to an
  '    elevation profile sampled at 75 mm spacing.
  ' 2. CA Profilograph Profile Index (PI) from the measuring wheel
  '    displacement found in step 1.
  ' 3. CA Profilograph must grind bump template.

  ' The subroutines are implemented in Microsoft Visual Studio 2012,
  ' Visual Basic.NET.

  Public Shared CAProfilographDone As Boolean
  Public Shared BandPassDone As Boolean
  Public CADeltaX As Double
  Private CAProfilographProfile() As Single, CAProfilographIndex() As Single
  Private CAProfilographSamples As Integer, CAProfilographDeltaX As Double
  Private PIIndex() As Single, PIlength() As Single
  Private HalfBand As Double = 0.1          ' inch
  Private ScallopMin As Double = 0.03       ' inch,
  Private ScallopLmin As Double = 2.0       ' ft
  Private ScallopAbove, ScallopBelow As Integer
  Public Shared Lprofilograph As Double = 500.0    ' ft
  Private PIN, PINmin As Integer
  Private PICum As Double

  Private BTEdgeMaxX As Double, BTHeightLimit As Double
  Private BTProfile() As Single, BTSamples As Integer
  Private BTBumpHeight() As Single, BTBumpLength() As Single
  Private BTIndex() As Single, BTEdgeLength() As Single
  Private BTDone As Boolean, BTDeltaX As Double
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index

```
  Private Sub StartRunCAProfilograph(Index As Integer)
'   Preliminaries to computing measuring wheel and PI.

    If PlotCanvas(0) Is Nothing Then
      Ret = MsgBox("A profile has not been loaded. Please read a file" _
                   & vbCrLf & "to load the profile before running CA _
                   Profilograph.", , "Invalid Selection")
      Exit Sub
    End If

    sbrStatus.Items(0) = "Computing CA Profilograph" : DoEvents()
    Dim DecimateN As Integer = 3     ' Decimation rate, 75 mm spacing.

    ' Returns CAProfilographProfile as decimated Profile
    ' with length CAProfilographSamples.
    Call DecimateProfile(PlotCanvas(0).ArrayLength, CAProfilographProfile,
                         CAProfilographSamples, DecimateN)
    CAProfilographDeltaX = DeltaD * DecimateN
    Call WriteIndexLabel(Indexes.CAProfiler + 1, "")
    Call RunCAProfilograph(CAProfilographProfile, CAProfilographSamples,
                           CAProfilographDeltaX)


  Sub RunCAProfilograph(ByRef Profile() As Single, ByVal NSamples As Integer,
                        ByVal DX As Double)

    Dim I, II, PartN, IStart, IEnd As Integer
    Dim TempZ1, TempZN, ZAve, PIndexSect As Double
    Dim CenterX, TandemSpacing As Double
    Dim LW1, LW2, LW3, LW4 As Integer
    Dim WheelRad, ZW(), TotalLength As Double
    Dim J, NRad, NSegs As Integer
    Dim PIOffset, PITotal, PartLength() As Double
    Dim SA, SB, NSASB, Nloop, NSend(), AddNlast As Integer
    Dim FFreq As Double
    Dim BandA(), BandB() As Double
    Dim SingleBCL(), BCLL As Single
    Dim IBCL, IIBCL, NBCL As Integer
    ' The following are Public variables.
    ' Dim CAProfilographIndex() As Single
    ' Dim PIN, PINmin As Integer
    ' Dim ScallopMi, PICum, HalfBand As Double
    ' Dim ScallopAbove%, ScallopBelow%

    ' DX = sample spacing in feet.
    ' Profile() = an array containing the measuring wheel values in inches.
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
' Setup the profilograph geometry.
CenterX = 25 / 2      ' ft
TandemSpacing = 2.5   ' ft
LW1 = CInt((CenterX + TandemSpacing * 1.5) / DX)
LW2 = CInt((CenterX + TandemSpacing * 0.5) / DX)
LW3 = CInt((CenterX - TandemSpacing * 0.5) / DX)
LW4 = CInt((CenterX - TandemSpacing * 1.5) / DX)

' Setup to correct for measuring wheel radius. This can have a
' significant effect on rigid pavements because the wheel radius
' smooths dips and steps out.
WheelRad = 1.3    ' ft
' Check for contact over +/- 30 degrees.
NRad = CInt(Int(WheelRad / DX / 2.0))
ReDim ZW(NRad)
For I = 1 To NRad
  ZW(I) = (WheelRad - _
          Math.Sqrt(WheelRad * WheelRad - I * DX * I * DX)) * 12
Next I

ReDim CAProfilographIndex(NSamples - 1)
ReDim SingleArray(NSamples - 1 + 2 * (LW1 + NRad))

' All arrays in VB.NET are zero-based so extend the profile
' to include the wheels at both ends of the profile.
For I = LW1 + NRad To NSamples - 1 + LW1 + NRad
  SingleArray(I) = Profile(I - LW1 - NRad)
Next
TempZ1 = 2 * SingleArray(LW1 + NRad + 1)
TempZN = 2 * SingleArray(NSamples + LW1 + NRad - 2)
For I = 0 To LW1 + NRad
  SingleArray(LW1 + NRad - I) = CSng(TempZ1) - _
                                SingleArray(I + LW1 + NRad + 1)
  SingleArray(NSamples - 1 + LW1 + NRad + 1) = _
         CSng(TempZN) - SingleArray(NSamples - 1 + LW1 + NRad - I)
Next I

For I = LW1 To NSamples - 1 + LW1
  ZAve = SingleArray(I - LW1) + SingleArray(I - LW2) * 3
  ZAve = ZAve + SingleArray(I - LW3) * 3 + SingleArray(I - LW4)
  ZAve = ZAve + SingleArray(I + LW4) + SingleArray(I + LW3) * 3
  ZAve = ZAve + SingleArray(I + LW2) * 3 + SingleArray(I + LW1)
  TempZ1 = SingleArray(I)
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
    ' Correct for measuring wheel radius.
    For J = 1 To NRad     ' Disable with -ve sign.
      TempZN = SingleArray(I + J) - ZW(J)
      If TempZN > TempZ1 Then TempZ1 = TempZN
      TempZN = SingleArray(I - J) - ZW(J)
      If TempZN > TempZ1 Then TempZ1 = TempZN
    Next J
    ' The simulated measuring wheel displacement is stored in
    ' CAProfilographIndex()
    CAProfilographIndex(I - LW1) = CSng(TempZ1 - ZAve / 16)
    Next I

  ' Use a low-pass filter with the cuttoff frequency = 0.5 cy/ft to
  ' eliminate spikes.
  FFreq = 0.5 ' cy/ft
  Call FilterSubs.ButterWorth3(FFreq, DX, CAProfilographIndex,
                               NSamples, FilterBothWays)

  ' Plot or store the measuring wheel displacement here

  ' Calculate PI for all sections Lprofilograph ft long.
  PIndexSect = Lprofilograph
  NBCL = NSamples
  ReDim SingleBCL(NBCL)
  For I = 0 To NBCL - 1
    SingleBCL(I) = CAProfilographIndex(I)
  Next I

  PartN = CInt(PIndexSect / DX)  ' Number of samples for Lprofilograph ft.
  PINmin = CInt(ScallopLmin / DX)
  ScallopMin = 0.03
  HalfBand = 0.1 'Blanking Band = +/- 0.1 inches.
  PIOffset = 0.0
  NSegs = NBCL \ PartN
  MinAveragingDistance = PIndexSect / 10
  If (NBCL - NSegs * PartN) * DX > MinAveragingDistance Then
    NSegs = NSegs + 1
    AddNlast = 0
  Else
    AddNlast = NBCL - NSegs * PartN
  End If

  ReDim ZW(NSegs), PartLength(NSegs)
  Dim GRMSpartLength(NSegs) As Double
  ReDim BandA(NSegs), BandB(NSegs), NSend(NSegs)
  Dim A, B, C As Double
  Dim K As Integer
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
    For J = 1 To NSegs

      NSend(J) = 0
      TempZN = 0
      PIN = 0
      IStart = (J - 1) * PartN + 1
      IEnd = IStart + PartN
      If J = NSegs Then IEnd = IEnd + AddNlast
      If IEnd > NBCL Then IEnd = NBCL

      ' The position of the blanking band is obtained by a
      ' lease squares linear fit, Y=a*X+b.
      Call BlankingBandPosition(SingleBCL, DX, IStart, IEnd, A, B, J)

      ' Look for the best location of the blanking band in the section.
      ' The number of scallops above and below the blanking band will be
      ' approximately the same (<=2) at the position.

      SA = 0 : NSASB = 0 : Nloop = 0 : C = 0
      Do

        ' Calculate the number of scallops above the blanking band
        ' and store in ScallopAbove.
        ScallopAbove = 0
        ScallopBelow = 0
        Call Scallop(IStart, IEnd, A, B, C, DX, SingleBCL,
                     True, TempZN, False)
        If PIN > 0 Then NSend(J) = PIN
        ' A scallop above the Blanking-Band occurs in
        ' the Jth and (J+1)th segments.

        ' Calculate the number of scallops below the blanking band
        ' and store in ScallopBelow.
        TempZN = 0 : PIN = 0
        Call Scallop(IStart, IEnd, A, B, C, DX, SingleBCL,
                     False, TempZN, False)
        If PIN > 0 Then NSend(J) = -PIN
        ' A scallop below the Blanking-Band occurs in
        ' the Jth and (J+1)th segments.

        ' Adjust the position of the Blanking Bank
        ' If ABS(SB) = ABS(ScallopAbove - ScallopBelow) > 1.
        ' But if the number of ABS(SB) non-changes > 10
        ' or the number of loops > 50, then exit the loop.
        SB = ScallopAbove - ScallopBelow
        If Math.Abs(SB) <= 1 Or NSASB > 10 Or Nloop > 50 Then
          Exit Do
        Else
          If Math.Abs(SB) = Math.Abs(SA) Then
            NSASB = NSASB + 1
          Else
            NSASB = 0
          End If
          Select Case Math.Sign(SA * SB)
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
        Case 0, 1
          C = C + Math.Sign(SB) * 0.01
          K = 0
        Case -1
          K = K + 1
          C = C + Math.Sign(SB) * 0.01 / (2 ^ K)
      End Select
      SA = SB
    End If
    Nloop = Nloop + 1
    DoEvents()

  Loop

  BandA(J) = A : BandB(J) = B + C

Next J ' NSegs times.

For J = 1 To NSegs

  PICum = 0 : TempZN = 0 : PIN = 0
  IStart = (J - 1) * PartN + 1
  IEnd = IStart + PartN
  If J = NSegs Then IEnd = IEnd + AddNlast
  If IEnd > NBCL Then IEnd = NBCL
  ScallopAbove = 0

  A = BandA(J) : B = BandB(J)
  If J > 1 And NSend(J - 1) > 0 Then
    II = 0
    Do
      II = II + 1
      I = IStart - II
      PIOffset = A * (II - 1) * DX + B
      TempZ1 = SingleBCL(I) - PIOffset - HalfBand
      If TempZ1 > 0 Then
        PIN = PIN + 1
        If TempZ1 > TempZN Then TempZN = TempZ1
      Else
        Exit Do
      End If
    Loop
  End If

  Call Scallop(IStart, IEnd, A, B, C, DX, SingleBCL, True, TempZN, True)
  ZW(J) = PICum

Next J ' NSegs times.
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
For J = 1 To NSegs

  PICum = 0 : TempZN = 0 : PIN = 0
  IStart = (J - 1) * PartN + 1
  IEnd = IStart + PartN
  If J = NSegs Then IEnd = IEnd + AddNlast
  If IEnd > NBCL Then IEnd = NBCL
  ScallopBelow = 0

  A = BandA(J) : B = BandB(J)
  If J > 1 And NSend(J - 1) < 0 Then
    II = 0
    Do
      II = II + 1
      I = IStart - II
      PIOffset = A * (II - 1) * DX + B
      TempZ1 = -SingleBCL(I) + PIOffset - HalfBand
      If TempZ1 > 0 Then
        PIN = PIN + 1
        If TempZ1 > TempZN Then TempZN = TempZ1
      Else
        Exit Do
      End If
    Loop
  End If

  Call Scallop(IStart, IEnd, A, B, C, DX, SingleBCL, False, TempZN, True)
  ZW(J) = ZW(J) + PICum
  PITotal = PITotal + ZW(J)
  PartLength(J) = (IEnd - IStart) * DX
  ZW(J) = ZW(J) / (PartLength(J) / 5280)

Next J ' NSegs times.

' Calculate the PI over the full length of the profile.
PITotal = 0 : TotalLength = 0
IStart = 1 : IEnd = NSegs
For I = IStart To IEnd
  PITotal = PITotal + ZW(I) * PartLength(I)
  TotalLength = TotalLength + PartLength(I)
Next I

PITotal = PITotal / TotalLength

' Store the results for the output chart and document.
With IndexSummaries(Indexes.CAProfiler)
  .Name = CAProfilographName
  .Caption = PlotCanvas(Indexes.CAProfiler + 1).PlotCaption
  .DeltaX = DX
  .MinAveragingDistance = MinAveragingDistance
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
      .PartN = NSegs
      .TotalAverage = PITotal
      .TotalMax = 0.0
      .TotalMin = 0.0
      .TotalRMS = 0.0
      ReDim .PartLength(NSegs - 1), .PartAverage(NSegs - 1)
      ReDim .PartRMS(NSegs - 1)
      For I = 1 To NSegs
        .PartLength(I - 1) = PartLength(I)
        .PartAverage(I - 1) = ZW(I)
        .PartRMS(I - 1) = 0.0
      Next I
      .IndexUnitValue = CDbl(UnitsValue(UnitsEnum.inchpermile))
      .IndexUnitFormat = CStr(UnitsValue(UnitsEnum.inchpermileFormat))
      .IndexUnitName = CStr(UnitsValue(UnitsEnum.inchpermileName))
      S = Format(PITotal * .IndexUnitValue, .IndexUnitFormat) _
          & " " & .IndexUnitName
      ' Write the PI over the total profile length to the user interface.
      Call WriteIndexLabel(Indexes.CAProfiler + 1, S)
    End With

    CAProfilographDone = True

  End Sub

  Sub BlankingBandPosition(ByRef F() As Single, ByVal DX As Double,
                           ByVal NSTART As Integer, ByVal Nend As Integer,
                           ByRef A As Double, ByRef B As Double, ByVal II As Integer)

    Dim I, J, N As Integer
    Dim U As Double, V As Double

    N = Nend - NSTART + 1
    U = 0 : V = 0
    Select Case II
      Case 1
        For I = NSTART To Nend
          J = I - NSTART + 1
          U = U + (J - 1) * F(I)
          V = V + F(I)
        Next I
        A = 6 * (2 * U - (N - 1) * V) / N / (N - 1) / ((N + 1) * DX)
        B = 2 * ((2 * N - 1) * V - 3 * U) / N / (N + 1)
      Case Is > 1
        For I = NSTART To Nend
          J = I - NSTART + 1
          U = U + J * F(I)
          V = V + F(I)
        Next I
        A = 6 * (2 * U - (N + 1) * V) / N / (N - 1) / ((N + 1) * DX)
        B = 2 * ((2 * N + 1) * V - 3 * U) / N / (N - 1)
    End Select

  End Sub
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

```
Sub Scallop(ByVal IStart As Integer, ByVal IEnd As Integer,
            ByVal A As Double, ByVal B As Double,
            ByVal DX As Double, ByRef CAPI() As Single,
            ByVal Sabove As Boolean,
            ByRef TempZN As Double, ByVal Bcm As Boolean)

  Dim I As Integer
  Dim PIOffset, TempZ1 As Double

  ' Public ScallopMin As Double = 0.03         'inch,
  ' Public ScallopLmin As Double = 2.0         'ft
  ' Public PICum As Double
  ' Public PI, PINmin As Integer
  ' Public ScallopAbove, ScallopBelow As Integer

  PINmin = CInt(ScallopLmin / DX)
  If Bcum = True Then C = 0

  For I = IStart To IEnd
    PIOffset = A * (I - IStart + 1) * DX + B + C
    If Sabove = True Then
      TempZ1 = CAPI(I) - PIOffset - HalfBand
    Else
      TempZ1 = -CAPI(I) + PIOffset - HalfBand
    End If
    If TempZ1 > 0 Then
      PIN = PIN + 1
      If TempZ1 > TempZN Then TempZn = TempZ1
    Else
      If PIN > 0 Then
        If PIN >= PINmin And TempZn > ScallopMin Then
          If Sabove = True Then
            ScallopAbove = ScallopAbove + 1
          Else
            ScallopBelow = ScallopBelow + 1
          End If
          If Bcum = True Then PICum = PICum + TempZN
        End If
        TempZN = 0
        PIN = 0
      End If
    End If
  Next I

End Sub
```

FIG. X1.1 Sample Subroutines in Microsoft Visual Basic.NET to Compute Profilograph Index *(continued)*

X2. EXAMPLES OF PROFILOGRAPH SIMULATION AND PROFILOGRAPH INDEX COMPUTATION

X2.1 Included in this appendix is an example of the profilograph simulation using the coding in Visual Basic.NET language for the ProFAA computer subroutine as presented in the Appendix X1. The simulated profiles using an inertial profiler collected with 25 mm (1 in.) sample spacing are compared with measured ones by California type profilograph. The profile comparison and corresponding correlation coefficient (R^2) are provided (see Fig. X2.1).

X2.2 The two simulation programs (ProFAA and ProVAL) are run and the simulated profiles are compared (see Fig. X2.2). The computed profilograph indexes based on the simulated profiles using both airfield and highway profiles are compared with correlation coefficient (R^2) (see Fig. X2.3).

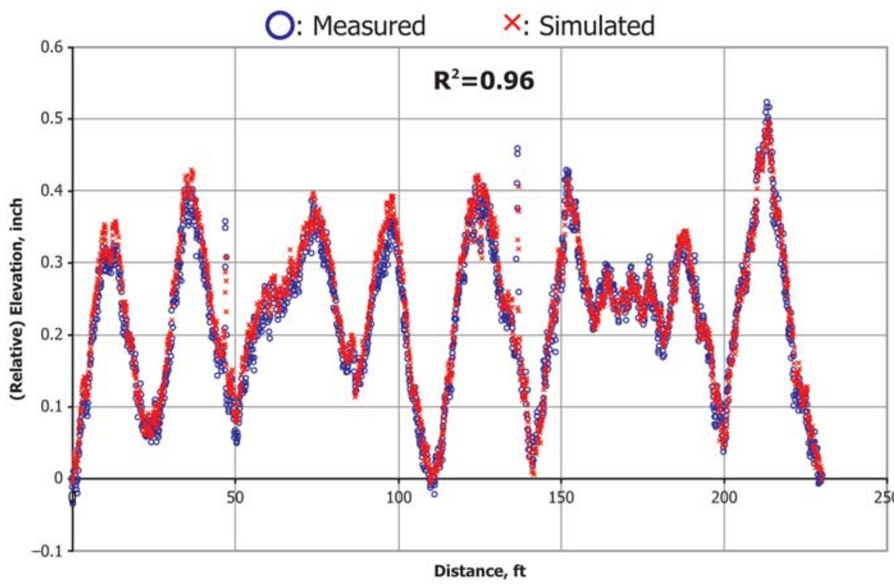

FIG. X2.1 Comparison of Profilograph Measured Profiles with ProFAA Simulated Profiles (Example)

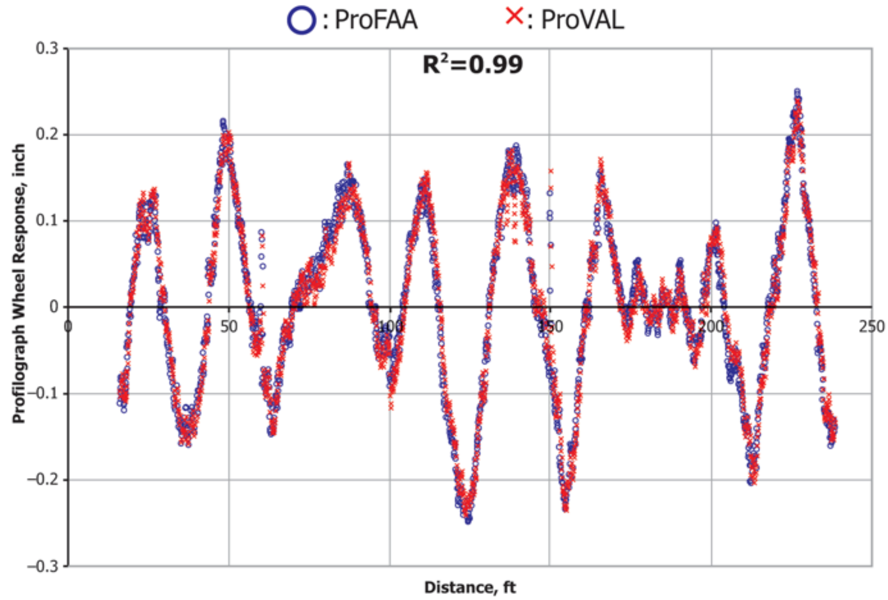

FIG. X2.2 Comparison of ProFAA Simulated Profilograph Profiles with ProVAL Simulated Profiles (Example)

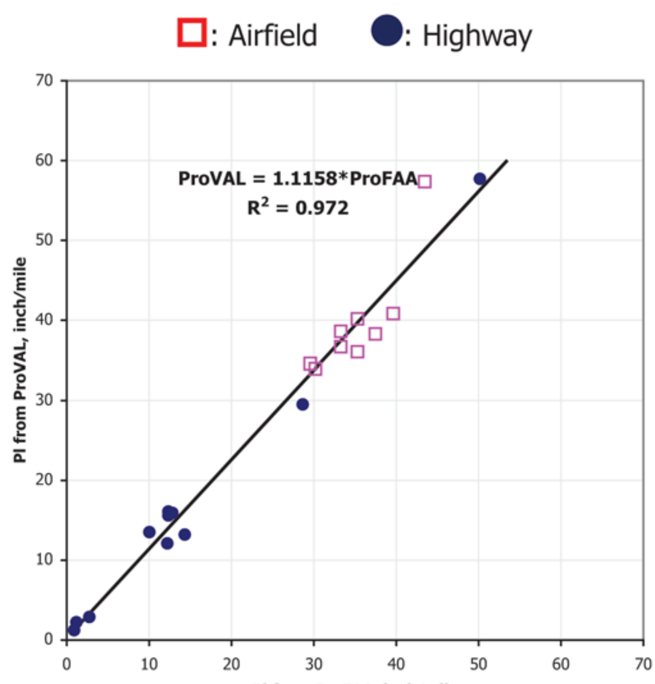

FIG. X2.3 Profilograph Index (PI) Computations by ProFAA and ProVAL (Example)