

ParaView によるポスト処理結果の VRML ファイル出力方法

本文書では、ParaView によるポスト処理結果を PDF3D ReportGen を利用して 3D PDF ファイルに変換するための VRML データの生成方法について紹介します。

ParaView に 3D PDF ファイルの出力機能を組み込んだ PDF3D PV+ という製品もあります。

PV+ では、ParavView から直接 3D PDF ファイルを作成することができます。(ReportGen と同等の機能を持っています。)

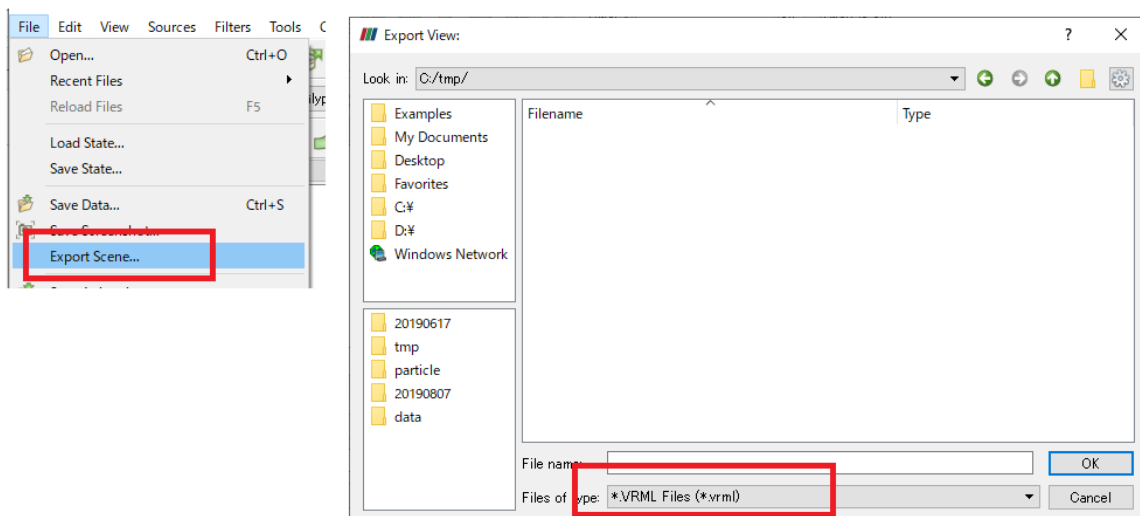
また、ReportGen の場合は、VRML ファイルから色情報の変換も行うため、色がそのまま再現できず、色に関するパラメータの設定が必要な場合があります(本ドキュメントの後半で説明)。PV+ では直接変換を行うため、表示している色がそのまま出力されます。その他、一部制限はありますが、PV+ では、可視化時に作成している色の凡例(カラー・レジェンド)も一緒に PDF に埋め込むことができます。

3D PDF ファイルを作成する対象ソフトが ParaView のみの場合は、PV+ の方が便利です。

VRML ファイルの作成

ParaView で可視化した結果を ReportGen で変換するには、VRML 形式のファイルを利用します。具体的には、[File] メニューの [Export Scene] メニューを選択します。保存のファイル・ダイアログが開きますので、[Files of type] から [*.VRML Files (*.vrm)] を選択し、ファイル名を指定してください。

現在表示されている状態が VRML ファイルに出力されます。



アニメーションの出力

時系列データのアニメーションを 3D PDF ファイルに変換するには、連番の VRML ファイルを作成してください。

ParaView で連番の VRML ファイルを作成するには、Python スクリプトを利用します。以下にスクリプトのサンプルを示します（拡張子 .py の名前のファイルを作成します）。

```
# render view
renderView1 = GetActiveViewOrCreate('RenderView')

# output dir
outdir = 'c:/tmp/'
prefix = 'para'
suffix = '.vrmf'
digit = 4

# time steps
tk = GetTimeKeeper()
timesteps = tk.TimestepValues
numTimesteps = len(timesteps)

# get animation
animationScene1 = GetAnimationScene()

# loop and output vrmf files
for i in range(numTimesteps):
    # filename
    outno = str(i)
    out_filename = outdir + prefix + outno.zfill(digit) + suffix
    print(out_filename)

    # move step
    animationScene1.AnimationTime = timesteps[i]

    # output command
    ExportView(out_filename, view=renderView1)
```

上記のスクリプトでは、時間ステップを進めながら、VRML ファイルの出力を行っています。

以下の 4 点に注目してください。

a) レンダー・ビューの取得と出力時の指定

まず、1 行目の `GetActiveViewOrCreate` メソッドで、今のレンダー・ビューを取得しています。
このビューを最後の行の `ExportView` メソッドの VRML ファイルの出力時に指定しています。

b) 時間ステップの取得とステップの移動

タイムステップを取得するには、`GetTimeKeeper` メソッドと `TimestepValues` プロパティを利用します。その配列の長さからステップ数 (`numTimesteps`) を取得しています。
このステップ数分のループを `for` 文で作成し、`timesteps` の配列の値をアニメーションのシーンの `AnimationTime` に設定することでステップを移動しています。

あるタイムステップ間のみ出力したい場合には、以下のように `for` 文を記述することもできます。

```
for i in range(25, numTimesteps):
```

例えば、上記の例では、25 ステップ目以降を出力します。

c) アニメーションの操作

アニメーションの操作を行うには、`GetAnimationScene` メソッドで取得したアニメーションのシーンを操作します。

この例では、ステップの番号を `timesteps` の配列で指定しています。

d) ファイル名の設定と VRML ファイルの出力

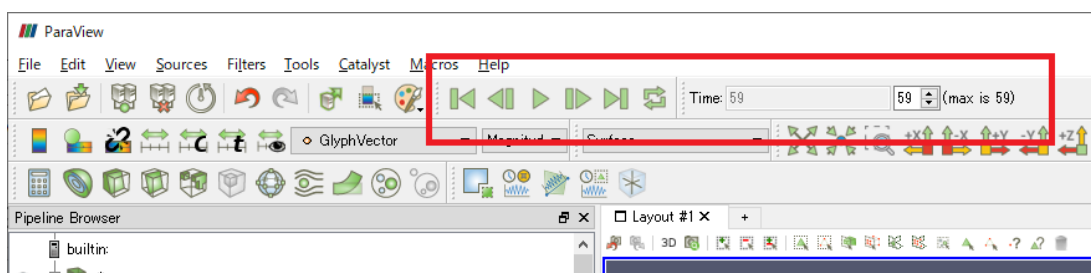
このスクリプトを実行すると、`c:/tmp/` フォルダに、`para0000.vrml`, `para0001.vrml`, ... の 4 桁のゼロ埋めされた連番ファイルが作成されます。

出力フォルダ (`outdir = "c:/tmp/"`)、ファイルの接頭子 (`prefix = "para"`)、ループ番号を利用したゼロ埋めの連番 (`outno.zfill(digit)`)、ファイルの拡張子 (`suffix = ".vrml"`) から、ファイル名 (`out_filename`) を作成しています。

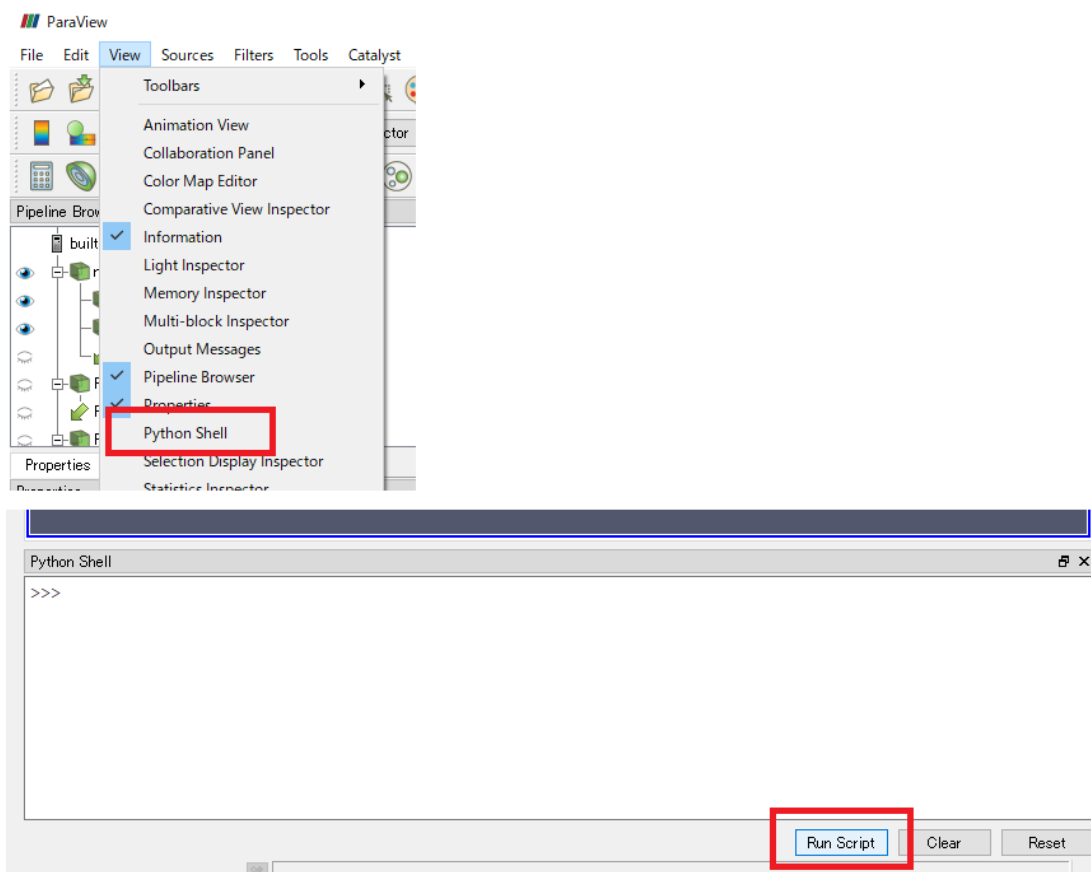
VRML ファイルの出力は `ExportView` メソッドで行います。

出力するファイルのタイプは、拡張子で決まります。

このスクリプトは、時系列データの可視化を行った後に実行します。
例えば、下図では、0 ～ 59 ステップのデータがあります。



スクリプトを実行するには、[View] メニューから [Python Shell] を選択します。
次に、開いた Python Shell にある [Run Script] ボタンをクリックします。



ファイル・ダイアログが開きますので、作成したスクリプト・ファイル (.py) を選択してください。

画面上でアニメーションのステップが変化します。また、そのステップ毎に、指定した VRML ファイルが作成されます。
この例では、c:/tmp/para0000.vrml ～ c:/tmp/para0059.vrml のファイルが作成されます。

ReportGen 変換時の注意

ParaView で作成した VRML ファイルを ReportGen で変換する場合には、以下の色の設定が必要です。
ParaView 上の色と変換後の 3D PDF 上の色が異なって表示される場合は、以下の設定を行ってください。
(ReportGen チュートリアルガイドの「カラー・コンター表示の改善」もあわせてご参照ください。)

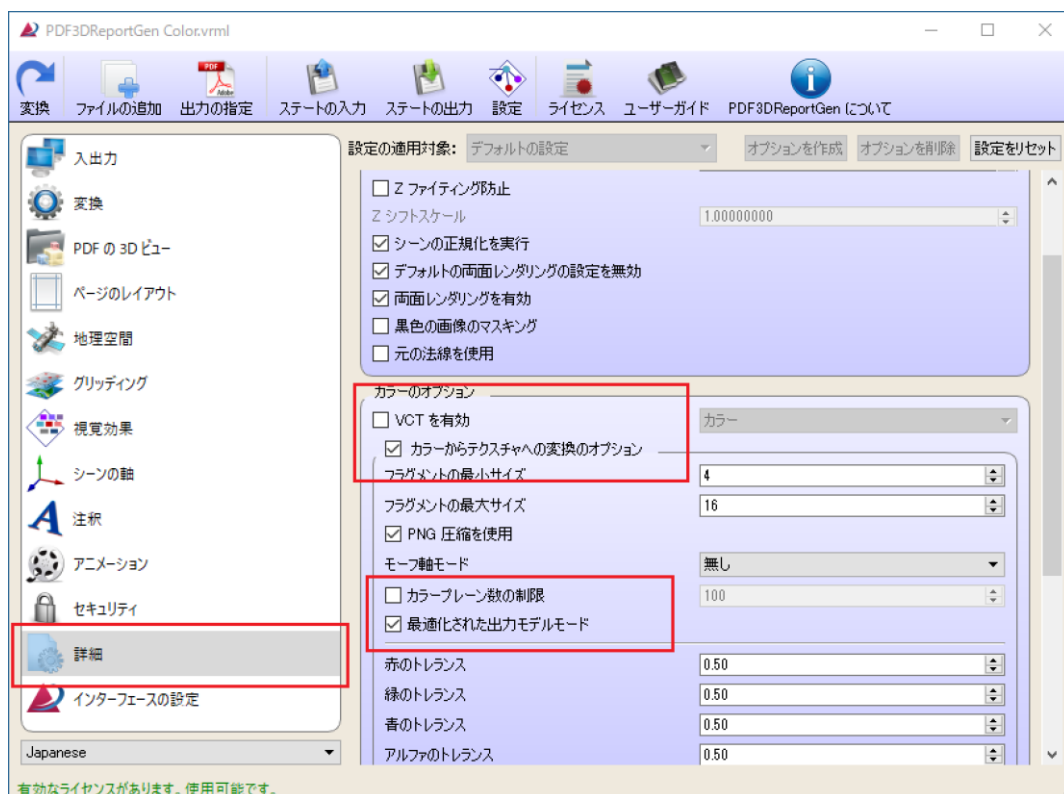
ParaView 上で、HSV の Hue (色彩) で変化するカラー・マップを利用 (ColorSpace に HSV を利用) している場合には、デフォルトのまま (以下の[VCT を有効]がオンのまま)、変換できる場合もあります。

ReportGen の [詳細] タブを選択します。

[カラーのオプション] から [VCT を有効] をオフにします。

次に [カラーからテクスチャへの変換のオプション] をオンにし、[カラープレース数の制限] をオフに、[最適化された出力モデルモード] をオンにします。

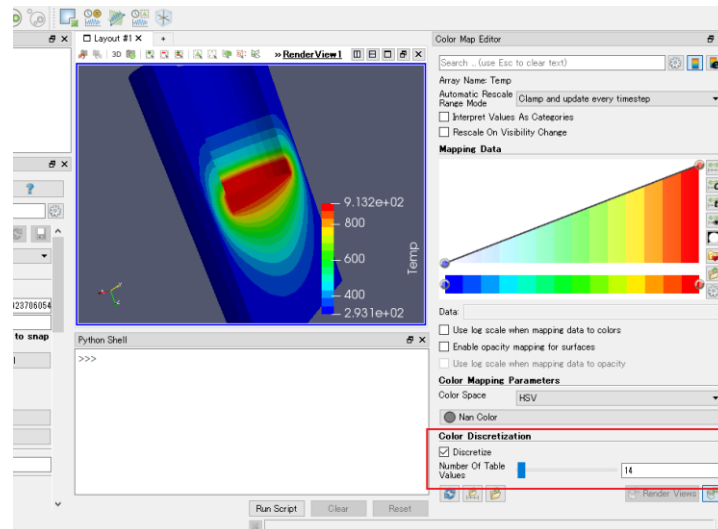
この設定を行った後に変換処理を実行します。



制限事項など

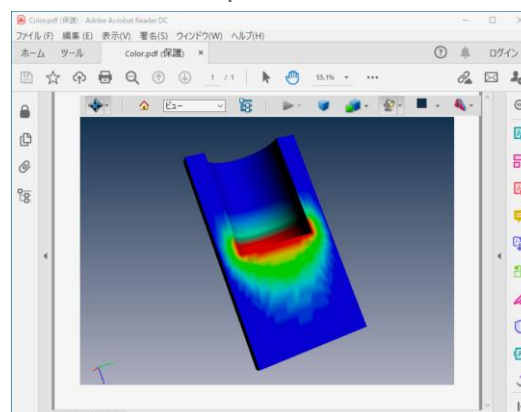
1. カラー分割数 (くっきりコンター図)

ParaView から出力される VRML ファイルには、頂点カラーが設定されます。また、その変換時には、各ポリゴンは、その頂点に設定された色で塗りつぶされます。ParaView 上では、下図に示すように色の分割数を少なくすると、等高線のようにその境界がはっきりしたコンター図を作成できますが、このデータを VRML ファイルに出力し、3D PDF ファイルに変換した場合には、このような境界がくっきりしたコンター図を作成することはできません。



ParaView 上でカラーの分割数を設定した様子

VRML ファイルに出力し、PDF3D ReportGen で変換した結果を以下に示します。

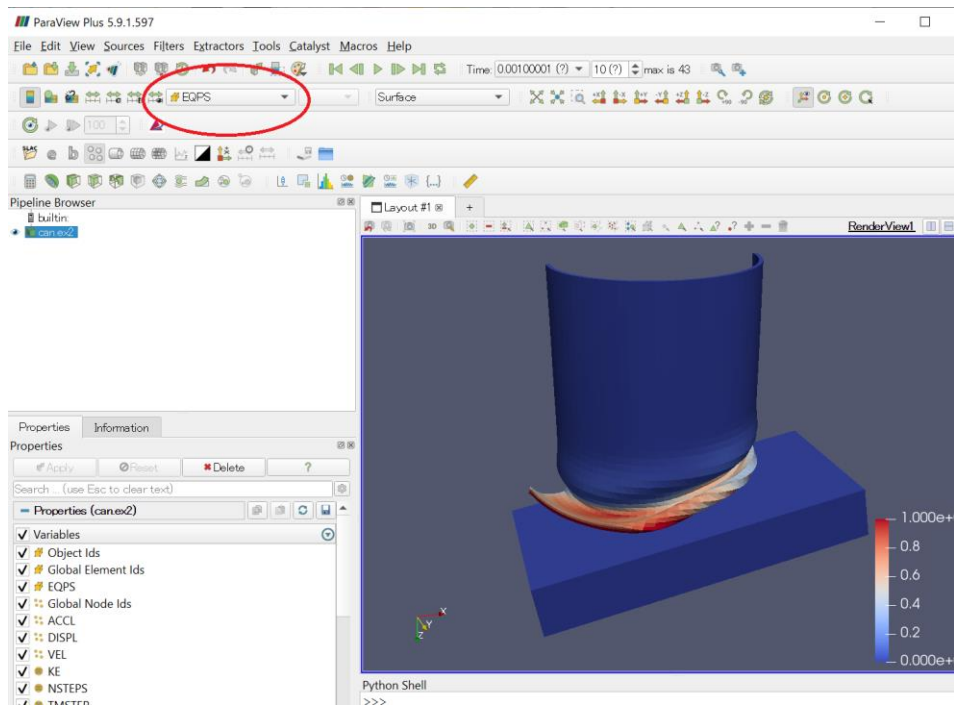


姉妹ソフトの PV+ を利用した場合には、ParaView 中のレンダリング情報から 3D PDF ファイルを作成しますので、このくっきりしたコンター図を再現できます。

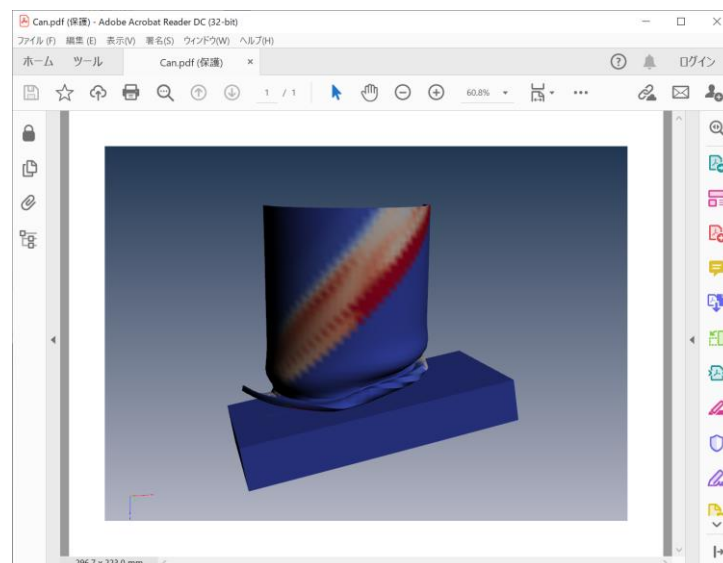
2. 要素カラーの扱い

要素の塗りつぶしによるコンター図（要素データによるコンター図）を表示している場合には、先の「ReportGen 変換時の注意」に述べた色のパラメータの設定を行っても、色が正しく変換されません。VRML 出力が頂点カラーに対応しているため、その要素のカラーが出力されないことが原因です。

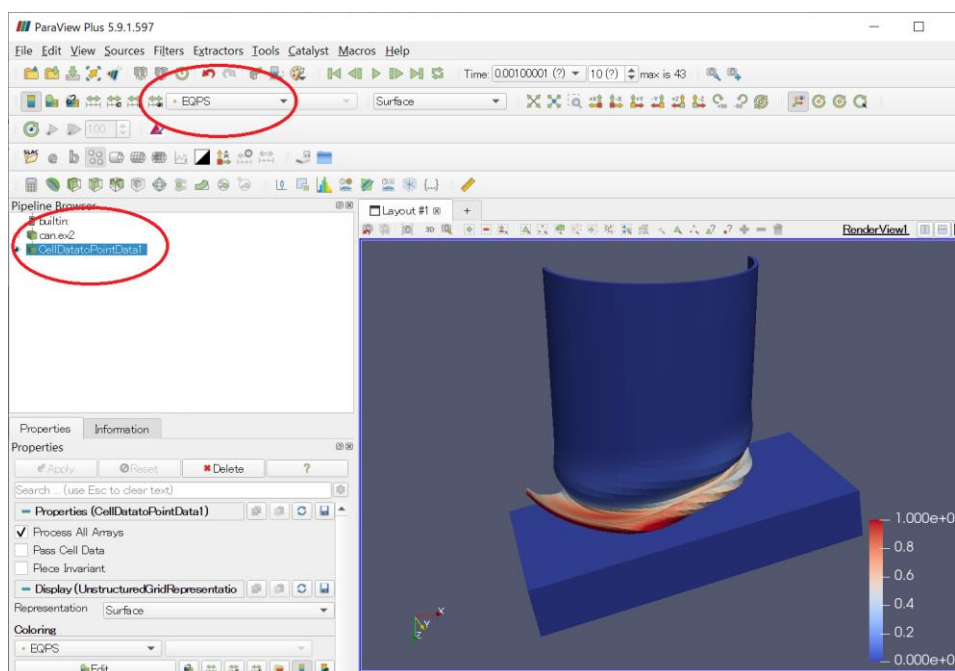
例えば、以下に例を示します。ParaView に含まれる Example を可視化した例です。読み込んだデータを要素データ（Cell データ）で色付けしています。



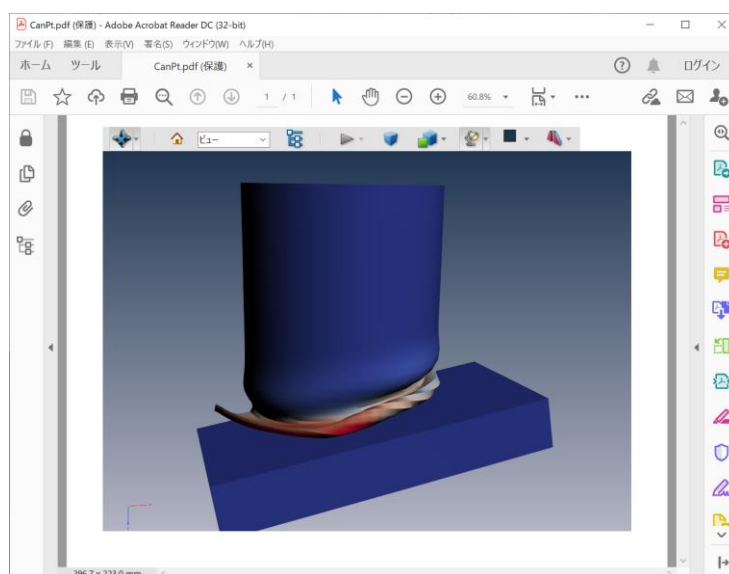
VRML ファイルに出力し、PDF に変換した例を以下に示します。色のパラメータを設定しても、色の分布が正しくありません。



ParaView から VRML ファイルに出力する場合には、要素データを頂点データに変換してください。データ変換には、[Cell Data to Point Data]フィルターを利用します。下図は、その変換を行った後の状態を示しています。



この VRML ファイルを変換した例を以下に示します。



姉妹ソフトの PV+ を利用した場合には、そのままレンダリング情報から変換を行いますので、要素データのまま扱うことができます。

3. ファイル・サイズ

VRML ファイルの変換時に、色のパラメータの設定（先の「ReportGen 変換時の注意」に述べた色の設定）を行った場合には、PV+で直接作成した PDF ファイルよりも、結果のファイル・サイズが大きくなります。VRML ファイルからの変換では、色のパラメータの設定を行うと 2 次元のカラー・テクスチャを作成しています。PV+では、直接レンダリングするデータを処理できるため、内部的にそのテクスチャを 1 次元で生成し、変換しています。その処理構造の違いによるものです。PV+で出力された PDF ファイルのサイズに比べ、ReportGen で変換されたファイルは、3 倍から 4 倍のサイズとなります。

以上