

$$g[x_, y_] := \frac{1}{2 * \pi * \sigma^2} * e^{-\frac{x^2+y^2}{2*\sigma^2}};$$

$$f[x_, y_] := \left(g[x, y] + \sum_{i=1}^{10} g[x + i, y] + \sum_{i=1}^{10} g[x, y + i] + g[x + 10, y + 10] \right) /.$$

{σ → 1};

$$\text{Hessian2D}[f_] := \left(\begin{array}{cc} D[D[f[x, y], \{x\}], \{x\}] & D[D[f[x, y], \{x\}], \{y\}] \\ D[D[f[x, y], \{y\}], \{x\}] & D[D[f[x, y], \{y\}], \{y\}] \end{array} \right);$$

xLeft = -12;

xRight = 5;

yLeft = -12;

yRight = 5;

max1 = FindMaximum[Det[Hessian2D[f]] // Evaluate,
 {{x, 0}, {y, 0}}][[2, ;;, 2]]

max2 = FindMaximum[Det[Hessian2D[f]] // Evaluate,
 {{x, 0}, {y, -10}}][[2, ;;, 2]]

max3 = FindMaximum[Det[Hessian2D[f]] // Evaluate,
 {{x, -10}, {y, 0}}][[2, ;;, 2]]

max4 = FindMaximum[Det[Hessian2D[f]] // Evaluate,
 {{x, -10}, {y, -10}}][[2, ;;, 2]]

{-0.240217, -0.240217}

{1.73267 × 10⁻¹², -9.42336}

{-9.42336, 1.44454 × 10⁻¹¹}

FindMaximum: Encountered a gradient that is effectively zero. The result returned may not be a maximum; it may be a minimum or a saddle point.

{-10., -10.}

colorExtrema = Red;

Show[

Plot3D[f[x, y], {x, xLeft, xRight}, {y, yLeft, yRight},
 AspectRatio → Automatic,
 ImageSize → 490,
 PlotRange → {{xLeft - 1, xRight + 1}, {yLeft - 1, yRight + 1},
 Automatic},
 AxesLabel → {"x", "y", "L(x,y)"},
 PlotLegends → {"Gaussian landscape"},
 PlotLabel → "det (H_{Gaussian landscape})",

```

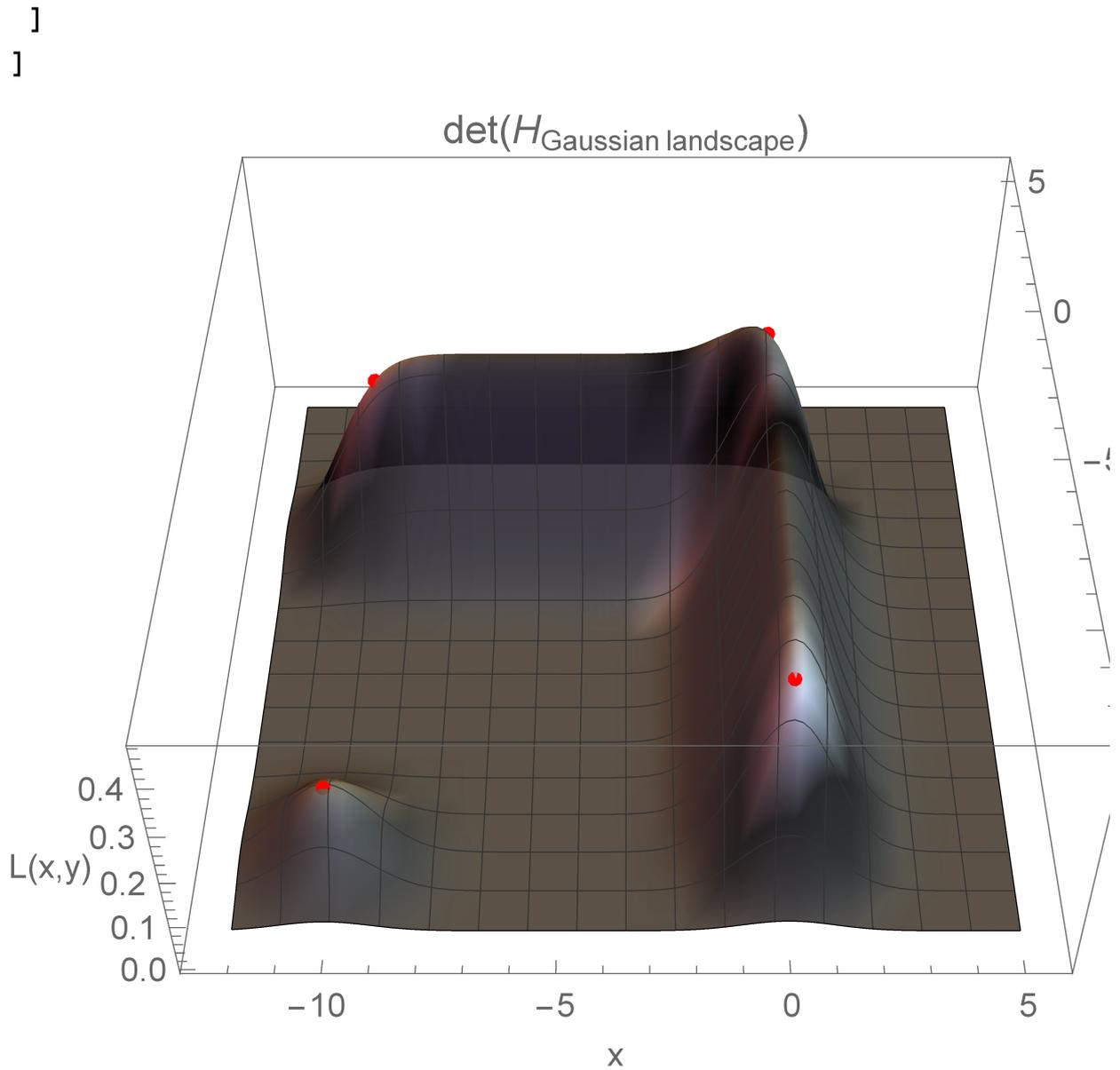
PlotStyle → Directive[Texture[
  DensityPlot[Det[Hessian2D[f]] // Evaluate, {x, xLeft, xRight}
    {y, yLeft, yRight},
  ColorFunction → GrayLevel,
  PlotRange → Full,
  Frame → None,
  PlotRangePadding → 0
]
], Opacity[0.9]],
PlotPoints → 100,
PerformanceGoal → "Quality",
BaseStyle → {FontSize → 14},
ViewPoint → {0, -2, 2}
],

```

```

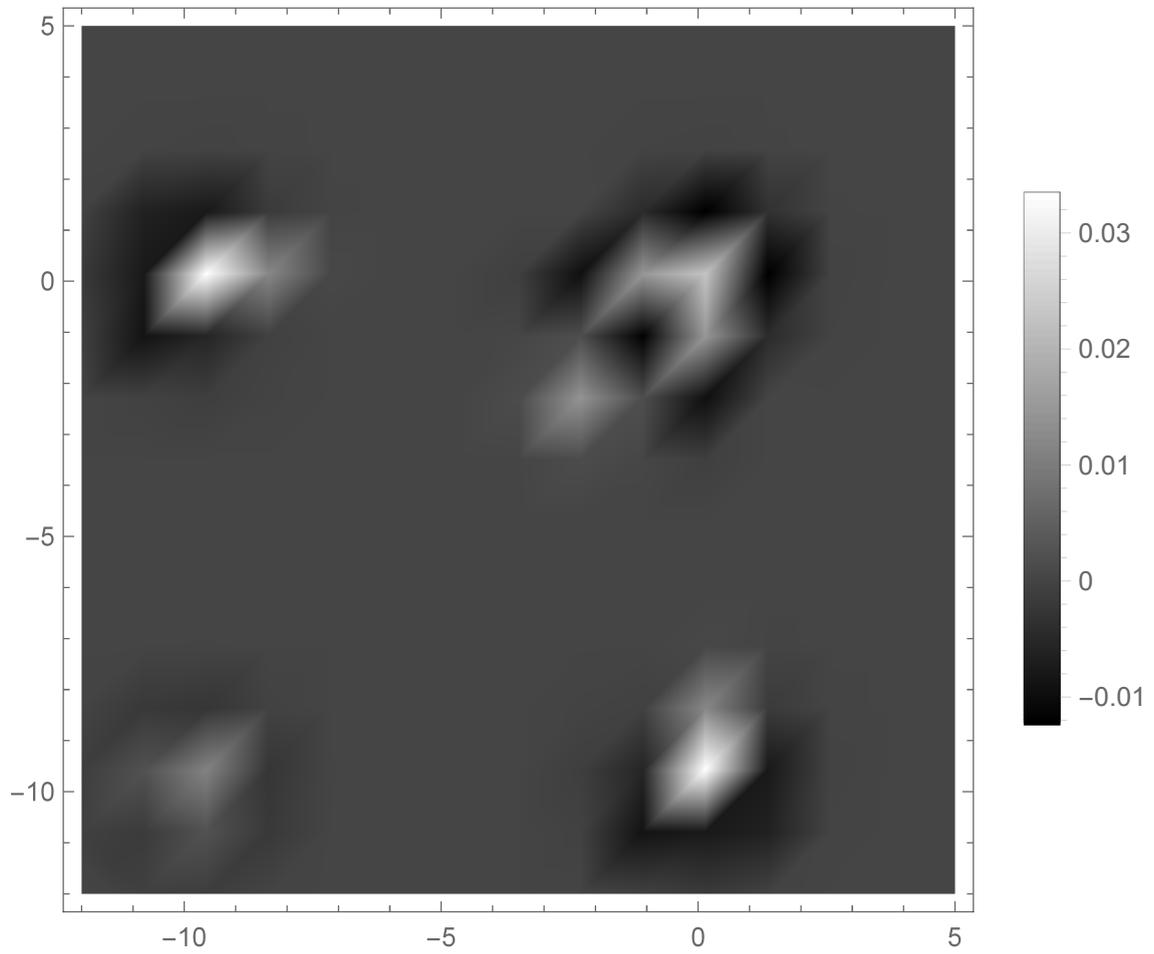
Legended[
Graphics3D[{
  {colorExtrema, PointSize[0.015],
    Point[{max1[[1]], max1[[2]], f[max1[[1]], max1[[2]]]}]},
  {colorExtrema, PointSize[0.015],
    Point[{max2[[1]], max2[[2]], f[max2[[1]], max2[[2]]]}]},
  {colorExtrema, PointSize[0.015],
    Point[{max3[[1]], max3[[2]], f[max3[[1]], max3[[2]]]}]},
  {colorExtrema, PointSize[0.015],
    Point[{max4[[1]], max4[[2]], f[max4[[1]], max4[[2]]]}]}
]],
{
PointLegend[{colorExtrema}, {"Local maxima"}],
BarLegend[{
  GrayLevel,
  (* Creates a density plot with the color bar and
  extracts the determinant response range. The plot
  itself is not used further *)
  InputForm[DensityPlot[Det[Hessian2D[f]] // Evaluate,
    {x, xLeft, xRight}, {y, yLeft, yRight},
    PlotLegends → Automatic, ColorFunction → GrayLevel,
    PlotRange → Full]] [[1, 2, 1, 1, 2]]
}],
LegendLayout → "Column"
]
}

```

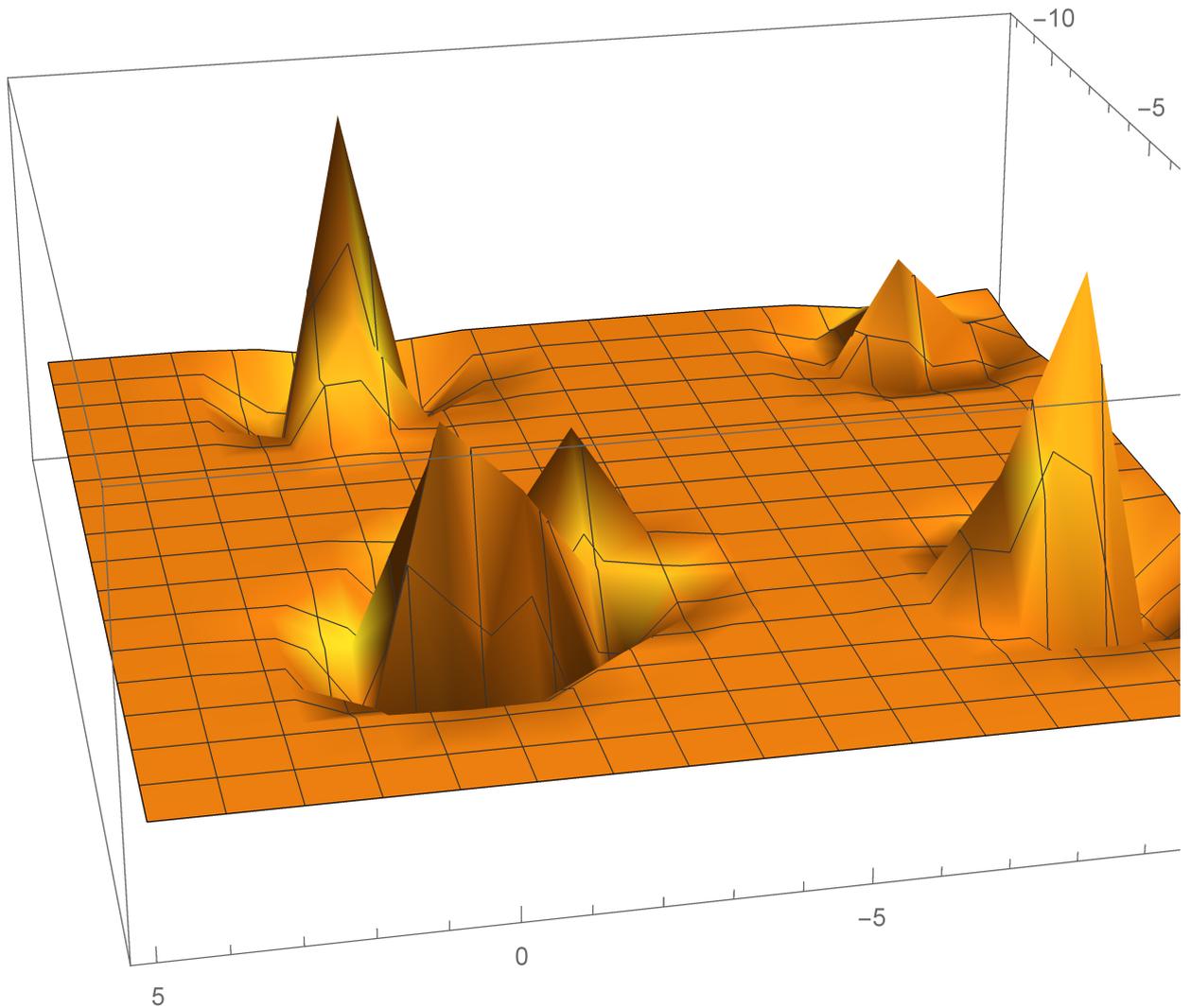


Different plots for the determinant response.

```
DensityPlot[Det[Hessian2D[f]] // Evaluate, {x, xLeft, xRight},
  {y, yLeft, yRight}, PlotLegends -> Automatic,
  ColorFunction -> GrayLevel, PlotRange -> Full]
```



```
Plot3D[Det[Hessian2D[f]] // Evaluate, {x, xLeft, xRight},  
{y, yLeft, yRight}, PlotRange → All, ImageSize → Large]
```



```
(* https://mathematica.stackexchange.com/questions/83556/compute-a-hessian-matrix-on-a-image *)
(* Gaussian filter as derivatives: https://www.swarthmore.edu/NatSci/mzucker1/e27\_s2016/filter-slides.pdf *)
```

```
img =
```



```
(* https://commons.wikimedia.org/wiki/File:Toulouse\_7412m.jpg *)
imgGray = ColorConvert[img, "Grayscale"];
```

```

maxScaleSpaceRadius = 50;
detImages = ConstantArray[0, maxScaleSpaceRadius];
gaussImages = ConstantArray[0, maxScaleSpaceRadius];
Do[
  {h[xx], h[xy], h[yy]} =
    GaussianFilter[ImageData[imgGray], r, #] & /@
    {{0, 2}, {1, 1}, {2, 0}};
  detImages[[r]] =
    FullSimplify@Det[{{m[xx], m[xy]}, {m[xy], m[yy]}}] /. m -> h;
  gaussImages[[r]] = GaussianFilter[ImageData[imgGray], r];

, {r, 1, maxScaleSpaceRadius}]

```

Mathematica uses a different coordinate system, y-axis starts at the bottom left corner and goes up. But row-access is from top to bottom (totally confusing).

```

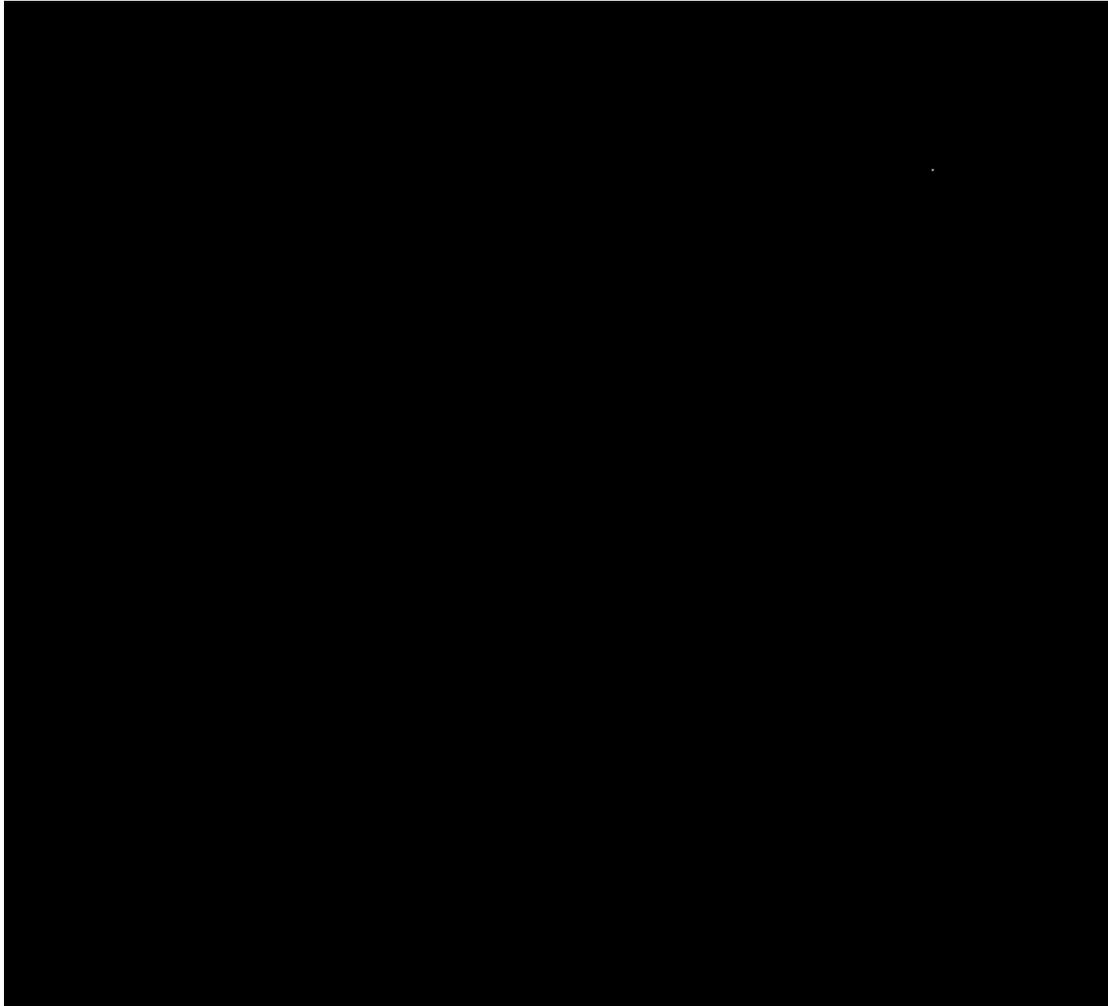
targetColumn = 532
targetRow = (ImageDimensions[img][[2]]) - 481

532

98

detTest = detImages[[24]];
detTest[[targetRow, targetColumn]] = 1000;
detTest // Image // ImageAdjust

```



```
Show[img,  
Graphics[{Red, Opacity[0.5], PointSize[0.015],  
Point[{targetColumn, 481}]}]]
```



```

detRespTarget = ConstantArray[0, maxScaleSpaceRadius];
detRespTargetNormalized = ConstantArray[0, maxScaleSpaceRadius];
detMins = ConstantArray[0, maxScaleSpaceRadius];
detMaxs = ConstantArray[0, maxScaleSpaceRadius];
gaussMins = ConstantArray[0, maxScaleSpaceRadius];
gaussMaxs = ConstantArray[0, maxScaleSpaceRadius];
Do[
  detRespTarget[[r]] =
    detImages[[r]][[targetRow, targetColumn]];

  (* Convert the radius back to the sigma value *)
  detRespTargetNormalized[[r]] =
    (2 * r)4 * detImages[[r]][[targetRow, targetColumn]];

  detMins[[r]] = Min[detImages[[r]]];
  detMaxs[[r]] = Max[detImages[[r]]];
  gaussMins[[r]] = Min[gaussImages[[r]]];
  gaussMaxs[[r]] = Max[gaussImages[[r]]];

, {r, 1, maxScaleSpaceRadius}]

prefactors = Table[(2 * r)4, {r, 1, maxScaleSpaceRadius}]
{16, 256, 1296, 4096, 10000, 20736, 38416, 65536, 104976,
160000, 234256, 331776, 456976, 614656, 810000, 1048576,
1336336, 1679616, 2085136, 2560000, 3111696, 3748096,
4477456, 5308416, 6250000, 7311616, 8503056, 9834496,
11316496, 12960000, 14776336, 16777216, 18974736, 21381376,
24010000, 26873856, 29986576, 33362176, 37015056, 40960000,
45212176, 49787136, 54700816, 59969536, 65610000,
71639296, 78074896, 84934656, 92236816, 100000000}

```

Decreased range of intensities if the image gets blurred with a Gaussian.

```

ListLinePlot[{gaussMaxs, gaussMins},
  DataRange → {1, 2 * maxScaleSpaceRadius},
  AxesLabel → {"σ", "I * Gσi"},
  PlotLegends → {"max response", "min response"},
  ImageSize → 500,
  BaseStyle → {FontSize → 14}]

```



```

    RescalingTransform[{{0, 1}, grange}, {{0, 1}, frange}]],
    s],
  Axes → False,
  Frame → True,
  FrameTicks → {{fticks, gticks}, {Automatic, Automatic}},
  (* The coordinates are in the coordinate system of
    the left graph, so the point must be mapped from
    coordinates from the right to ones to the left *)
  Epilog → {
    {PointSize[0.015], EdgeForm[{Thick, #}], FaceForm[Black],
    Disk[{Ordering[detMaxs * prefactors, -1][[1]] * 2,
      
$$\frac{\text{Max}[\text{detMaxs} * \text{prefactors}]}{\text{grange}[[2]]} * \text{frange}[[2]]$$

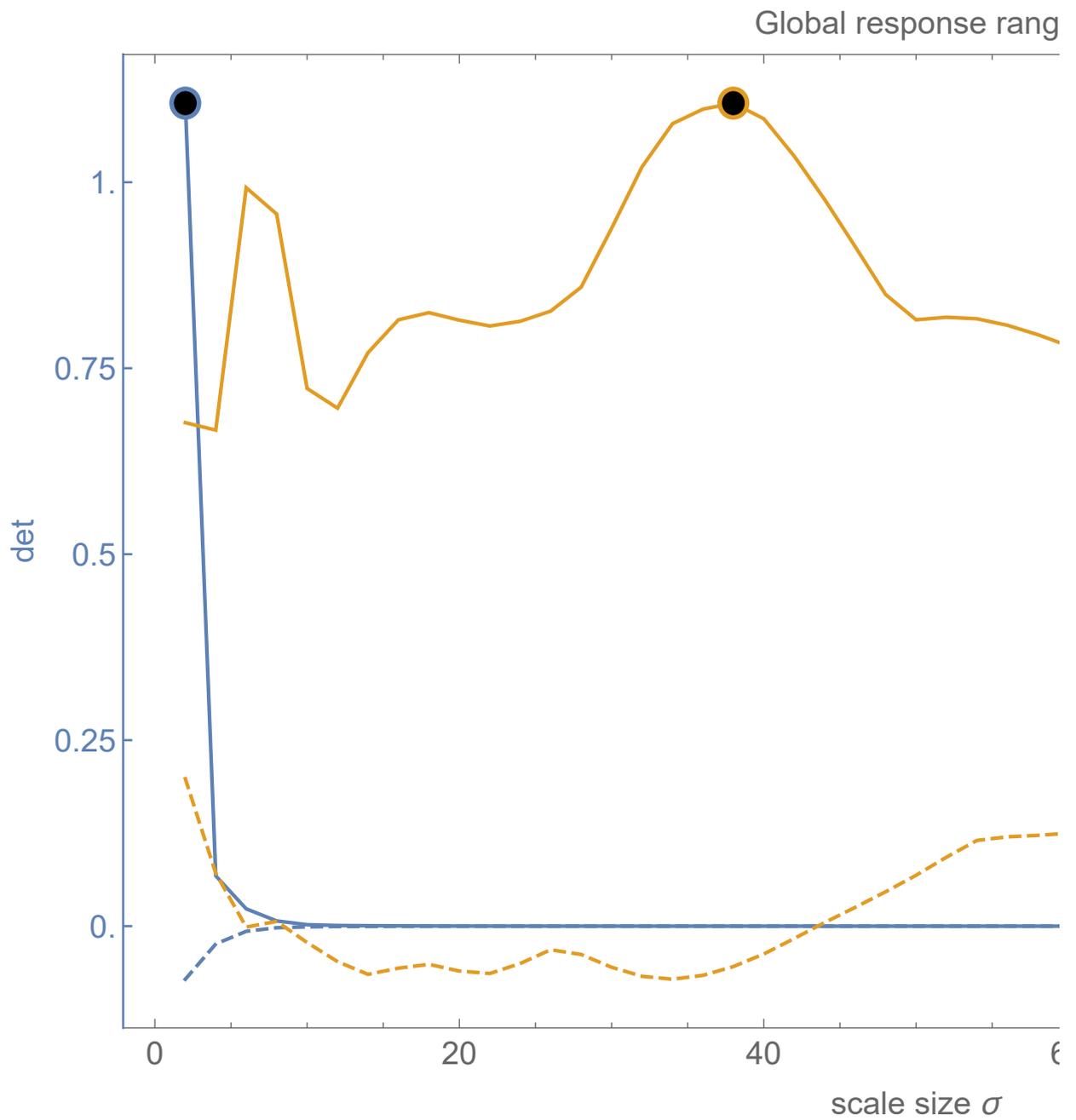
      }],
    {maxScaleSpaceRadius * pointScale,
      (frange[[2]] - frange[[1]]) * pointScale - 0.002}}],
    {PointSize[0.015], EdgeForm[{Thick, #}], FaceForm[Black],
    Disk[{Ordering[detMaxs, -1][[1]] * 2, Max[detMaxs]},
      {maxScaleSpaceRadius * pointScale,
        (frange[[2]] - frange[[1]]) * pointScale - 0.002}]]}
  }
]
]

```

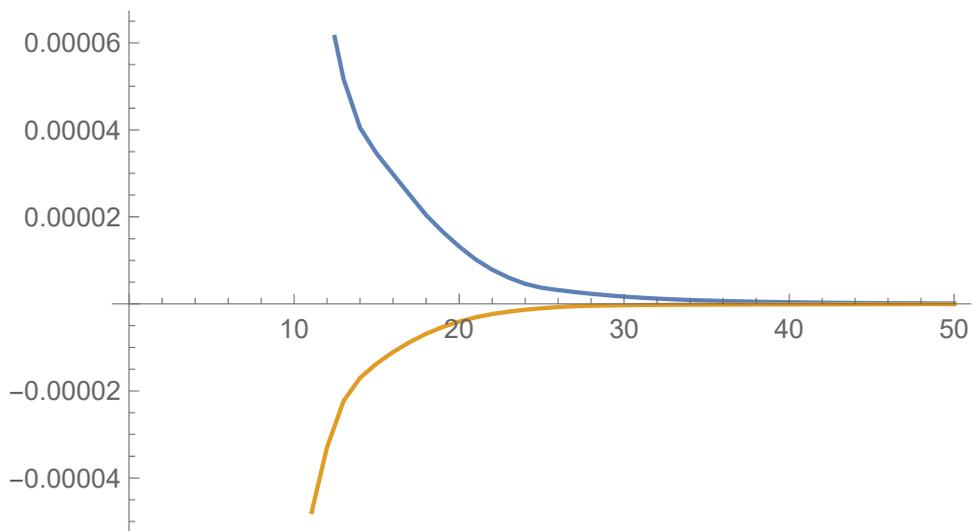
```

TwoAxisListPlotResp[ {
  {detMaxs, detMins},
  {detMaxs * prefactors, detMins * prefactors}
},
Joined → True,
FrameStyle → {Automatic, #, Automatic, #},
FrameLabel → {"scale size  $\sigma$ ", "det", "Global response range",
  " $\sigma^4 \text{det}$ "},
DataRange → {2, 2 * maxScaleSpaceRadius},
PlotRange → Full,
ImageSize → 800,
LabelStyle → Directive[FontSize → 14]
]

```



ListLinePlot[{detMaxs, detMins}]



```
maxIdxRespTarget = Ordering[detRespTargetNormalized, -1][[1]]
```

```
minIdxRespTarget = Ordering[detRespTargetNormalized, 1][[1]]
```

```
16
```

```
4
```

```
maxRespTarget = detRespTargetNormalized[[maxIdxRespTarget]]
```

```
minRespTarget = detRespTargetNormalized[[minIdxRespTarget]]
```

```
15.3165
```

```
-0.00102067
```

```
TwoAxisListPlotRespTarget[{f_, g_}, opts___] :=
```

```
Module[{fgraph, ggraph, frange, grange, fticks, gticks},
```

```
  fgraph = ListPlot[f, Axes → True, PlotStyle → ■, opts,
    PlotLegends → {"unnormalized"}];
```

```
  ggraph = ListPlot[g, Axes → True, PlotStyle → ■, opts,
    PlotLegends → {"normalized"}];
```

```
{frange, grange} =
  (PlotRange /. AbsoluteOptions[#, PlotRange])[[2]] & /@
  {fgraph, ggraph};
```

```
fticks = N@FindDivisions[frange, 5];
```

```
gticks =
```

```
  Quiet@
```

```
  Transpose@
```

```
    {fticks, ToString[NumberForm[#, 2], StandardForm] & /@
      Rescale[fticks, frange, grange]}];
```

```
Show[fgraph,
```

```
  ggraph /. Graphics[graph_, s___] :=>
```

```
  Graphics[GeometricTransformation[graph,
    RescalingTransform[{{0, 1}, grange}, {{0, 1}, frange}]],
    s],
```

```
  Axes → False,
```

```
  Frame → True,
```

```
  FrameTicks → {{fticks, gticks}, {Automatic, Automatic}},
```

```
(* The coordinates are in the coordinate system of
  the left graph, so the point must be mapped from
  coordinates from the right to ones to the left *)
```

```
  Epilog → {
```

```

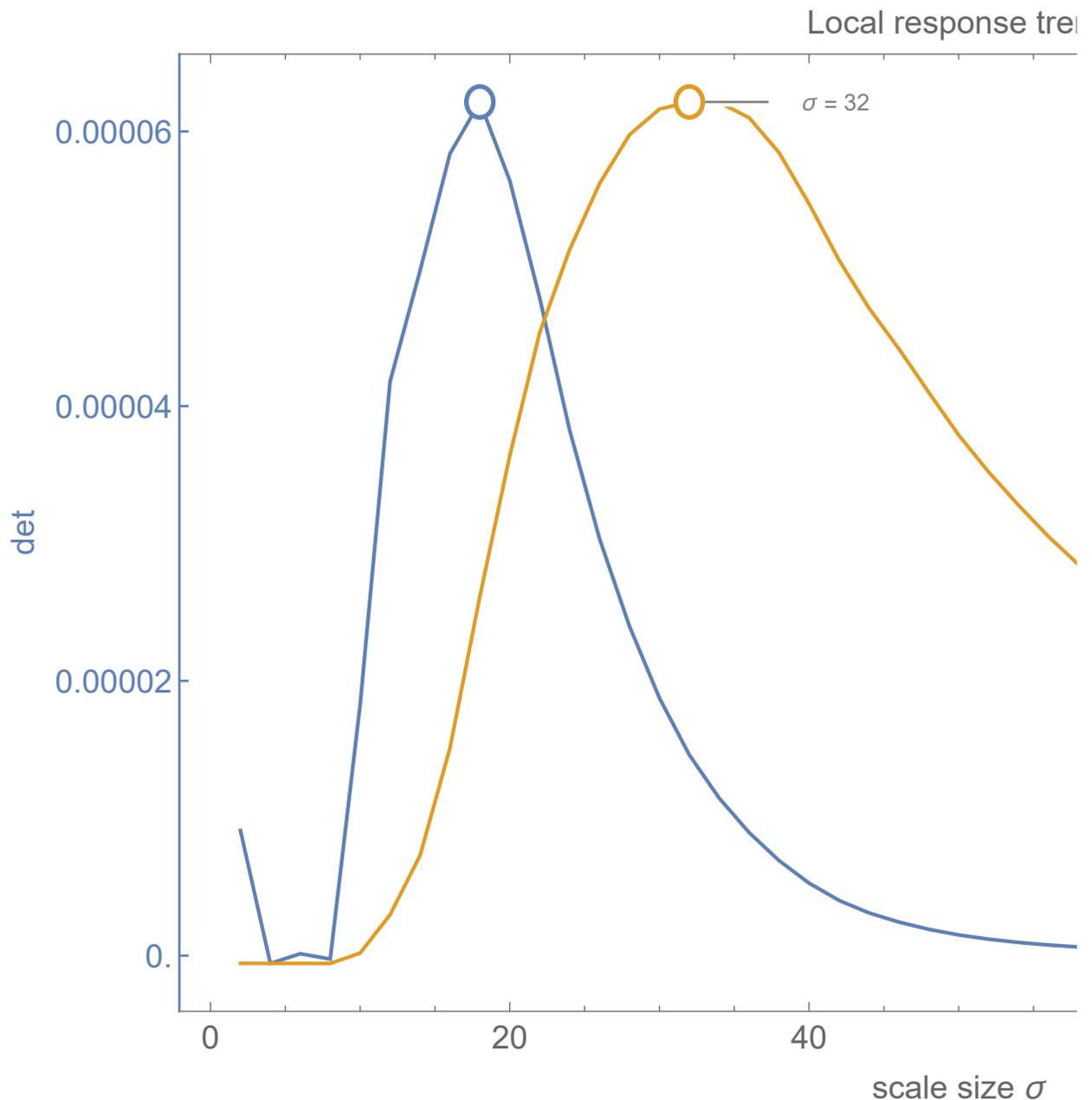
    {PointSize[0.015], EdgeForm[{Thick, #}], FaceForm[White],
      Disk[{maxIdxRespTarget * 2,  $\frac{\text{maxRespTarget}}{\text{grange}[[2]]} * \text{frange}[[2]]$ },
        {maxScaleSpaceRadius * pointScale,
          (frange[[2]] - frange[[1]]) * pointScale}]},
    {PointSize[0.015], EdgeForm[{Thick, #}], FaceForm[White],
      Disk[{Ordering[detRespTarget, -1][[1]] * 2,
        Max[detRespTarget]},
        {maxScaleSpaceRadius * pointScale,
          (frange[[2]] - frange[[1]]) * pointScale}]}]
  ]
]

```

```

TwoAxisListPlotRespTarget[{
  detRespTarget,
  ReplacePart[
    detRespTargetNormalized,
    maxIdxRespTarget → Callout[maxRespTarget,
      "  $\sigma =$ " <> ToString[maxIdxRespTarget * 2], After,
      LeaderSize → 30, LabelStyle → Gray]
  ]
},
Joined → True,
FrameStyle → {Automatic, #, Automatic, #},
FrameLabel → {"scale size  $\sigma$ ", "det", "Local response trend",
  " $\sigma^4 \text{det}$ "},
PlotRange → Full,
DataRange → {2, 2 * maxScaleSpaceRadius},
ImageSize → 800,
LabelStyle → Directive[FontSize → 14]
]

```



Compare the maximum response for the target location with the maximum response over all locations in all scale levels. This should be an indicator of how significant the response value is.

Max [detRespTarget]

Max [detMaxs]

Max [detRespTarget]

Max [detMaxs]

0.0000621447

1.10636

0.0000561703

```

Max[detRespTargetNormalized]
Max[detMaxs * prefactors]
Max[detRespTargetNormalized]
  Max[detMaxs * prefactors]

```

15.3165

34.6083

0.442567

```

maxDetResp = Max[detMaxs * prefactors]
minDetResp = Min[detMins * prefactors]

```

34.6083

-11.7458

```

plotDetResp[r_] := 
$$\frac{\text{detImages}[[r]] * (2 * r)^4 - \text{minDetResp}}{\text{maxDetResp} - \text{minDetResp}}$$
 // Image;

```

```

Manipulate[
  plotDetResp[r]
  , {r, 1, Length[detImages], 1}]

```



```
(*Export[
  FileNameJoin[{NotebookDirectory[], "frames/sigma=00.png"}],
  Table[plotDetResp[r], {r, 1, maxScaleSpaceRadius}],
  "VideoFrames", Antialiasing→True];*)
(*Export[
  FileNameJoin[{NotebookDirectory[],
    "HessianDetector_AirportImageDeterminantResponseScale2.png"}
  ], plotDetResp[2], "VideoFrames", Antialiasing→True];
Export[
  FileNameJoin[{NotebookDirectory[],
    "HessianDetector_AirportImageDeterminantResponseScale"<>
    ToString[maxIdxRespTarget]<>".png"}],
  plotDetResp[maxIdxRespTarget], "VideoFrames",
  Antialiasing→True];*)
```