Geo-Koordinaten in Access: Google Earth, Google Maps, ...

Demoanwendungen

DI Wilfried Baumann

Einzelunternehmer mit Schwerpunkten:

- Beratung, Datenbanken, Webtechnologien, Problemlösungen...

Seit vielen Jahren unter anderem in Access tätig.
Ein wichtiger Kunde ist das Naturhistorische Museum Wien:

- Erstellung von Sammlungsdatenbanken

Diese Präsentation wurde mit impress.js erstellt

Überblick des Vortrags

Geokoordinaten: Bedeutung, Definition, praktische Probleme

Kleine Demo-Access-Anwendungen:

Anw. I	Entfernungs-DB	sphärische Geometrie	
Anw. II	GPS List DB	KML (Placemarks, Overlays)	
Anw. III	Heat Map DB	Google Maps API Javacript	
Anw. IV	Get Coord DB	Google Maps API Webservice	
Anw. V	PLZ DB	alternativ: HTML5 Canvas	

Rechtliche Überlegungen

Geo-Koordinaten in Access

Begriffsklärung: Geokodierung, Verortung (Geotagging, Geocoding)

http://api.geonames.org/postalCodeSearch?postalcode=30159&maxRows=1&username=demo

52.3749839 9.7388153

PLZ					
30159	Hannover	Region Hannover	Niedersachsen		

Inverse Geokodierung ("reverse geocoding")

http://api.geonames.org/findNearbyPlaceName?lat=52.3749839&lng=9.7388153&username=demo

```
<geonames>
  <geoname>
    <toponymName>Hannover</toponymName>
    <name>Hanover</name>
    <lat>52.37052</lat>
    <lng>9.73322</lng>
    <geonameId>2910831</geonameId>
    <countryCode>DE</countryCode>
    <countryName>Germany</countryName>
    ...
    <distance>0.62632</distance>
  </geoname>
</geonames>
```

Bedeutung von Geo-Koordinaten

Bedeutung von Geo-Koordinaten

- GPS-Tracker, Smartphones, Kameras, Navigationsgeräte, externe Sensoren, GSM, WLAN ...
- Programme/Portale wie Google Earth, Google Maps, OpenStreetMap, GeoNames, ...
- Geo-Tag(Generator), HTML5 Geolocation API, Geotargeting, Geo URI (geo:37.786971,-122.399677;u=35), ...

siehe auch

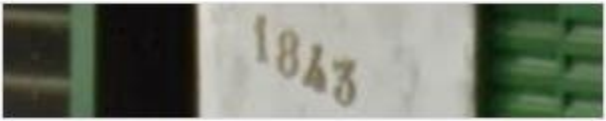
Freikarten für die Ideenwelt und Modellbaumesse Wien 2014

Mit diesem Formular kannst Du Dich für den Erhalt von Freikarten anmelden. Die Verteilung der Karten passiert nach dem "first come - first serve" Prinzip. Aus der Anmeldung ensteht kein Anspruch auf eine Freikarte. Die Freikarten liegen in ausgedruckter Form vor. Um Deine Freikarte zu erhalten, kannst Du sie entweder bei uns abholen oder wir senden sie Dir zu. Weitere Informationen dazu erhältst Du nach absenden des Formulars.

Name

E-Mail Adresse

gewünschte Anzahl an Karten



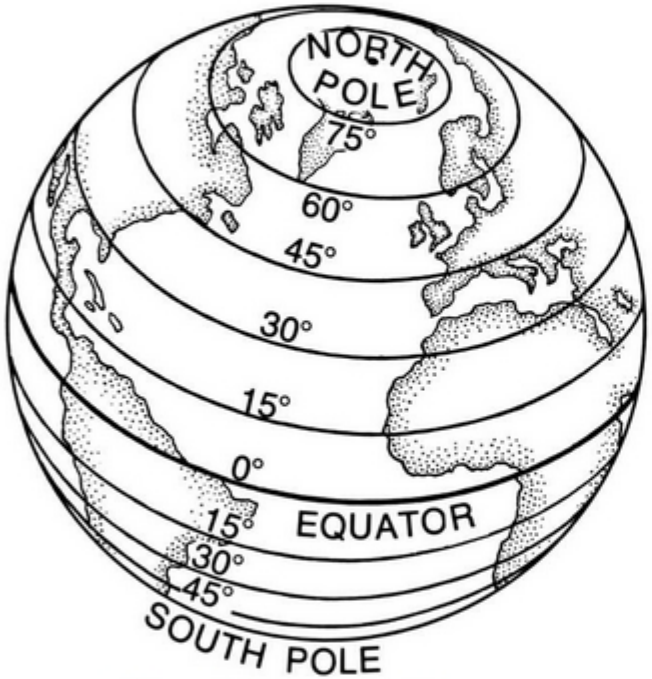
1843

reCAPTCHA™

Privacy & Terms

Ich freue mich schon auf die Ideenwelt

Geo-Koordinaten

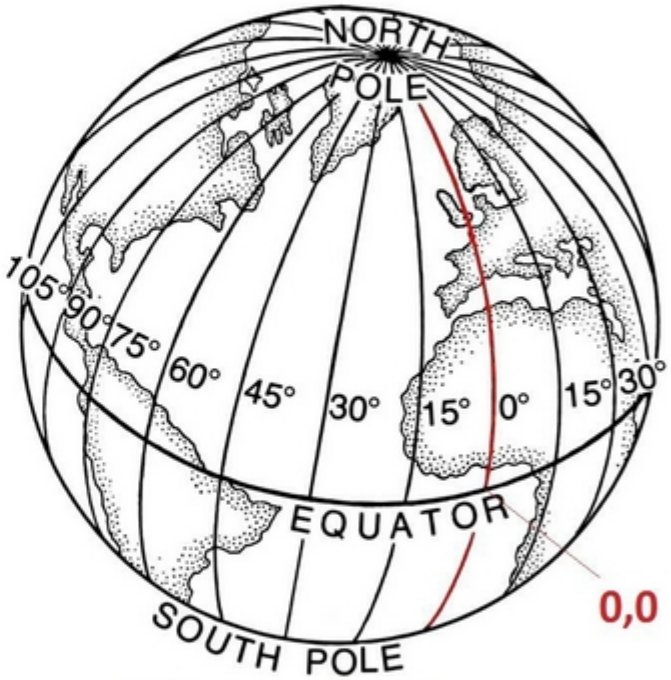


Breitenkreise

Nordpol

Äquator

Südpol



Längenkreise

Geo-Koordinaten in der Praxis

Greenwich (London)
51° 28' 52.5" N, 0° 0' 29.05" W
51° 28.875' N, 0° 0.48416' W (Grad, Dezimalminuten)
51.48125°, -0.008069° (Dezimalgrad)
Speicherung früher meist in Textfeldern:

- Tippfehler
- "." oder "," bzw "O" oder "E", welche Zeichen für ' und " ?
- Breite und Länge sind manchmal vertauscht
- Trennzeichen zw. Lat und Long?
- ...

regexp?: geolucidate

Access Anwendung I: Entfernungsberechnung

Welche Filiale ist die nächstgelegene? Wie weit ist sie entfernt?

Entfernung berechnen

10

34

Zurück

plz1

plz
1004
1010
1012
1013
1014
1015
1016
1020
1022
1023
1024
1025
1030
1032

plz2

plz
3400
3402
3403
3411
3412
3413
3420
3421
3422
3423
3424
3425
3430
3433

Berechnen

Microsoft Access

Entfernung 1010-3420: 15.66 km

OK

Entfernungsberechnung: Formel

www2.informatik.hu-berlin.de/~sanftleb/Kugelabstand.pdf

Abstand von P_1 und P_2 auf der Kugeloberfläche

$$d(P_1, P_2) = \frac{r \cdot \pi}{180} \cdot cos^{-1}(cos(\beta_1) \cdot cos(\beta_2) \cdot cos(\alpha_1 - \alpha_2) + sin(\beta_1) \cdot sin(\beta_2))$$

Beispielrechnung

$$\alpha_1 = 34°,\ \beta_1 = 23°,\ \alpha_2 = 55°,\ \beta_2 = 43°,\ r = 1 \Longrightarrow d(P_1, P_2) \approx 0.4624$$

Die "sphärische Trigonometrie" ist die Grundlage der Entfernungs- und Richtungsberechnung zwischen Koordinaten auf der Kugel.

Entfernungsberechnung: Code

```
Public Function Entfernung_Berechnen(dblLat1 As Double, _
                                     dblLon1 As Double, _
                                     dblLat2 As Double, _
                                     dblLon2 As Double) As Double
' Eingabe in Grad, Ergebnisse in km:
' Entfernung_Berechnen(0, 0, 1, 0)    >   111,319490772362
' Entfernung_Berechnen(0, 0, 0, 1)    >   111,319490772362
' Entfernung_Berechnen(45, 0, 46, 0)  >   111,319490772401
' Entfernung_Berechnen(45, 0, 45, 1)  >    78,7142672689772
Dim dblRes As Double
   '---
   dblLat1 = dblLat1 / 180 * PI
   dblLon1 = dblLon1 / 180 * PI
   dblLat2 = dblLat2 / 180 * PI
   dblLon2 = dblLon2 / 180 * PI
   '---
   dblRes = 6378.137 * Arkuskosinus(Sin(dblLat1) * Sin(dblLat2) + _
                                    Cos(dblLat1) * Cos(dblLat2) * _
                                    Cos(dblLon2 - dblLon1))
   Entfernung_Berechnen = dblRes
End Function
```

Geo-Koordinaten in Access

Google Earth

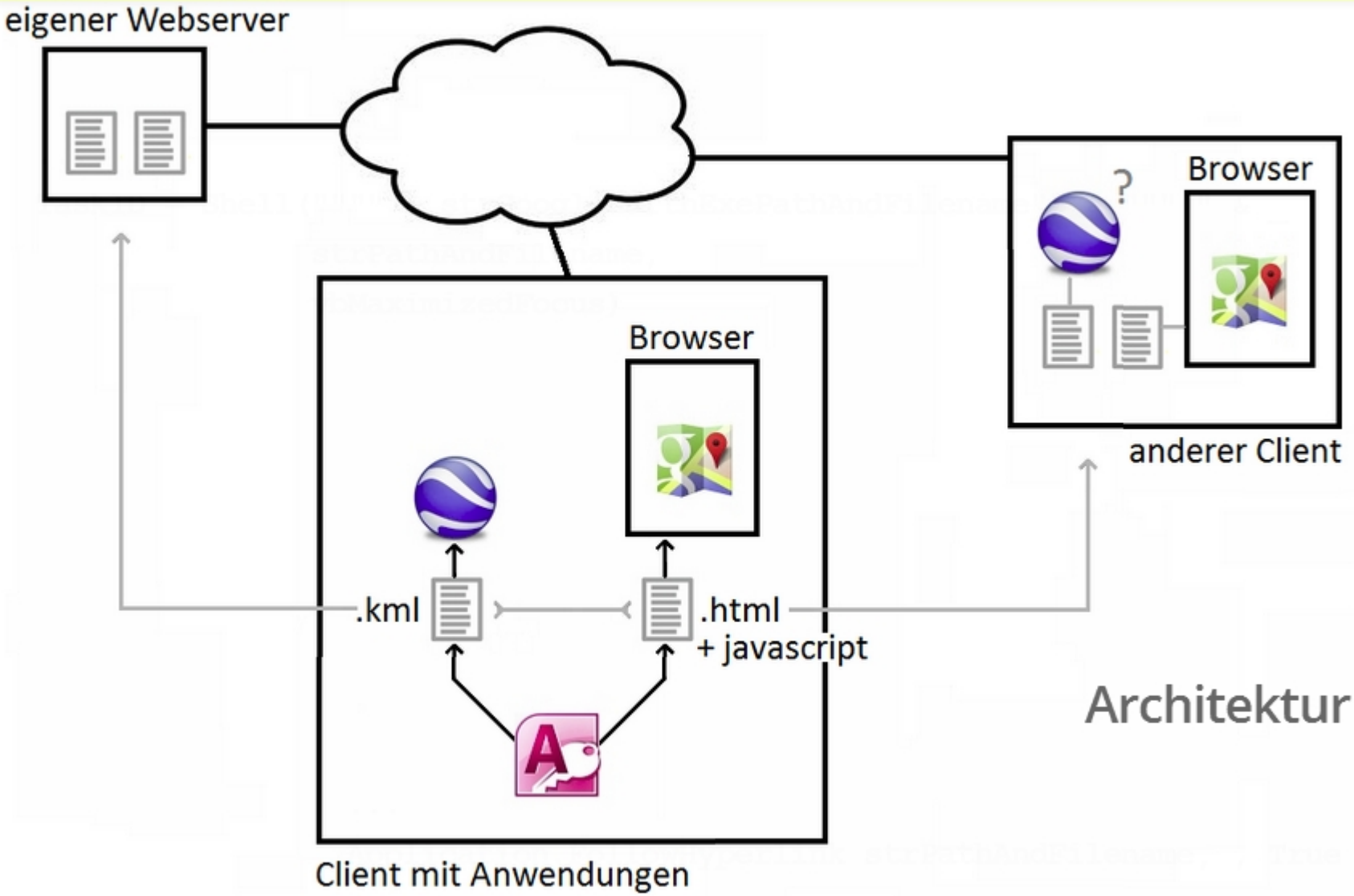
- Virtuelle Weltkugel: 3D-Ansichten, digitales Höhenmodell, ...
- Desktopanwendung
- Anzeige externer Daten oder Export mittels Keyhole Markup Language (KML)

Google Maps

- Suche von Orten und Anzeige auf 2D-Karte
- Browser-Anwendung
- Programmierschnittstelle: Google Maps APIs

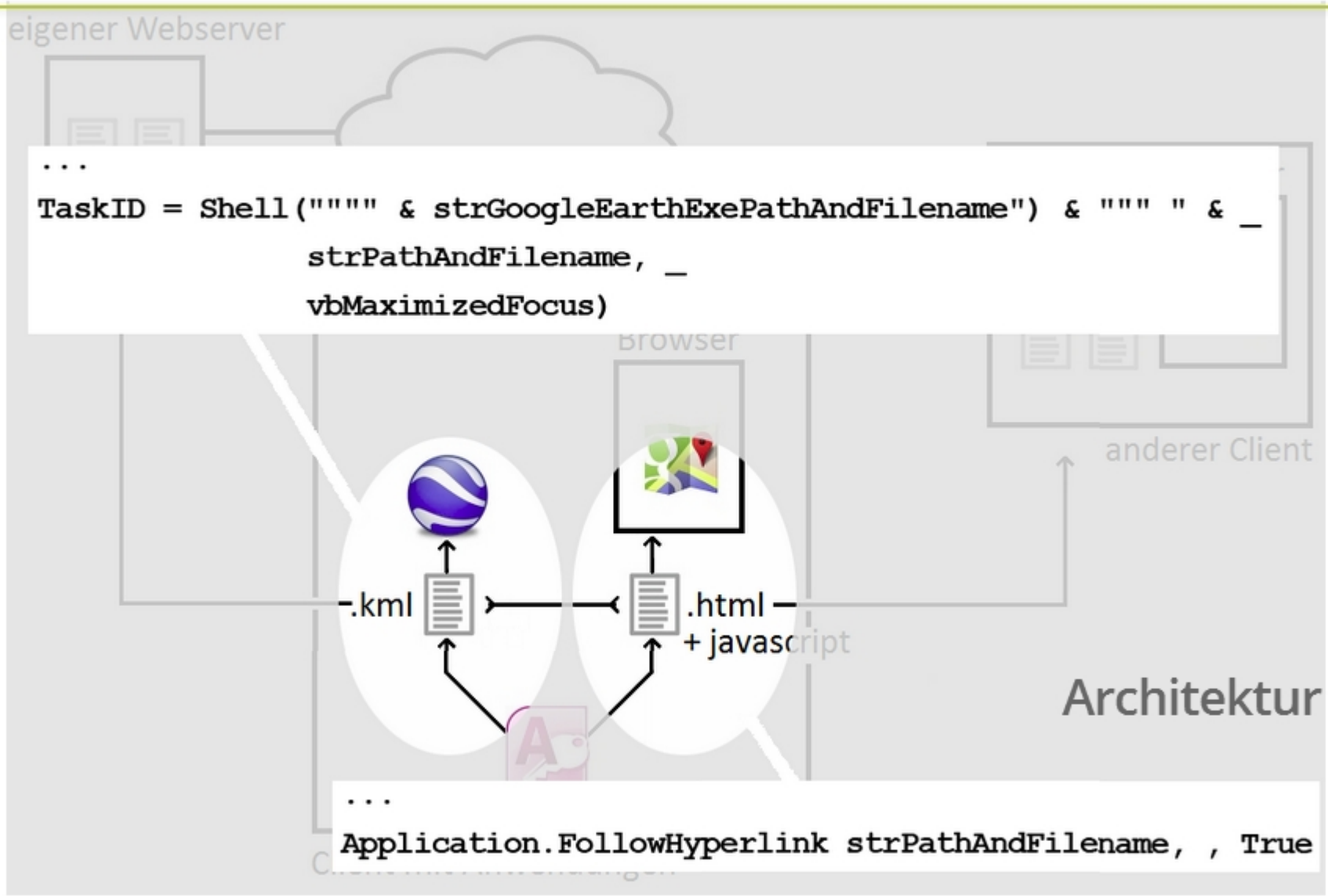
Beide Anwendungen greifen im wesentlichen auf die gleichen Datenbestände zu.

Geo-Koordinaten in Access



Architektur

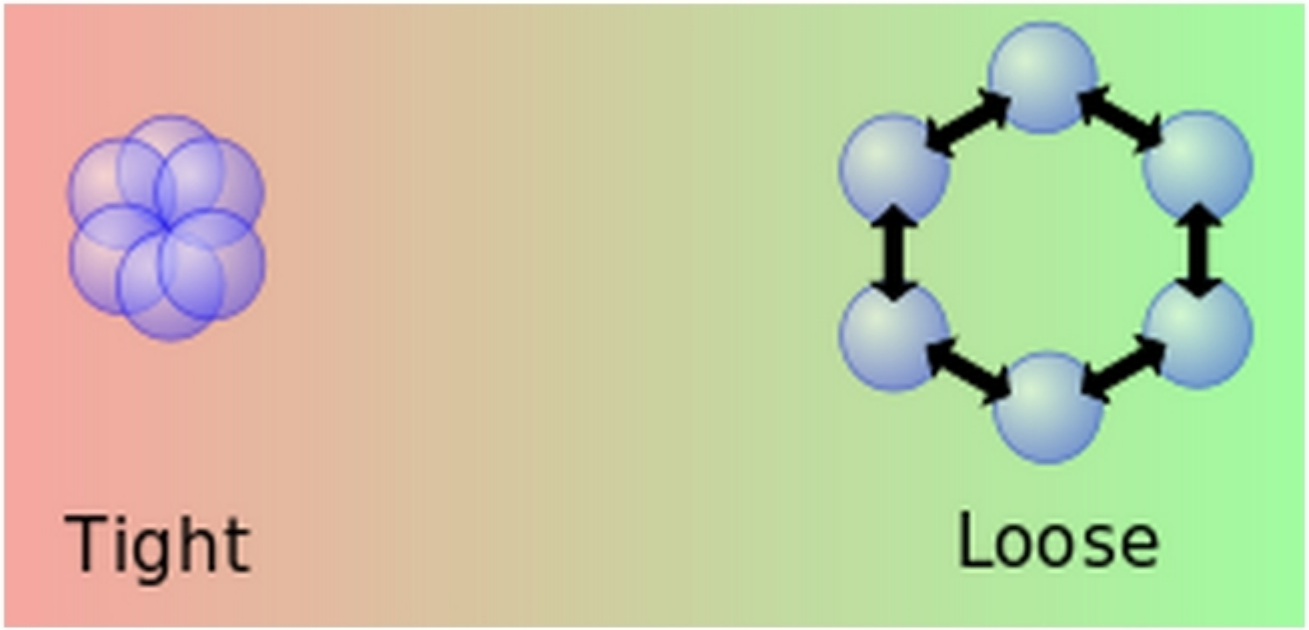
Geo-Koordinaten in Access




```
...
TaskID = Shell("""" & strGoogleEarthExePathAndFilename") & """ " & _
                strPathAndFilename, _
                vbMaximizedFocus)
```

```
...
Application.FollowHyperlink strPathAndFilename, , True
```

Enge Kopplung - Schwache Kopplung



Für schwache Kopplung anzustreben sind gekapselte, spezialisierte Module mit klar definierten, sparsamen Schnittstellen.

Keyhole Markup Language, KML (Samples)

Ortsmarke, Overlay, Pfad, Polygon, ..., Folder, ...
Google Earth kann KML-Daten sowohl importieren als auch exportieren.

Ortsmarke

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
       at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

vergleiche dazu

Keyhole Markup Language, KML (Samples)

Ortsmarke, Overlay, Pfad, Polygon, ..., Folder, ...
Google Earth kann KML-Daten sowohl importieren als auch exportieren.

Ortsmarke

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
      at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

vergleiche dazu

Keyhole Markup Language (KML)

Ground Overlay

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder>
    <name>Ground Overlays</name>
    <description>Examples of ground overlays</description>
    <GroundOverlay>
      <name>Large-scale overlay on terrain</name>
      <description>Overlay shows Mount Etna erupting
          on July 13th, 2001.</description>
      <Icon>
        <href>http://developers.google.com/kml/documentation/images/etna.jpg</href>
      </Icon>
      <LatLonBox>
        <north>37.91904192681665</north>
        <south>37.46543388598137</south>
        <east>15.35832653742206</east>
        <west>14.60128369746704</west>
        <rotation>-0.1556640799496235</rotation>
      </LatLonBox>
    </GroundOverlay>
  </Folder>
</kml>
```

Access Anwendung II: GPS List

frm_GPS_List

GPS List

- Führende "*" erlaubt zb *mann findet Bergmann und Bachmann etc
- "?" erlaubt zb B?rg findet Berger und Burger
- ' erlaubt wie in Côte d'Ivoire
- "Cot" findet automatisch auch Côt etc
- Eingabe von "?" findet Datensätze mit Werten in dieser Spalte

TODO | Zurück

sel. ID_... lat_orig lon_orig lat lon F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12

Quick — Use "Quick" for immediate filtering. Otherwise use search button above.

Blocks: Select, Deselect

selecte	ID_GPS_List	Lat_orig	Lon_orig	Lat	Lon	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
	1			47,777854	16,86991	1		6	9	2001			1	0	Fang
	2			47,76891	16,757957	2		11	7	1985			1	0	Fang
	3			47,753539	16,757081	3		30	3	1964			1	0	Fang
	4			47,769222	16,756156	4		5	4	1964			4	0	Fang
	5			47,934143	16,854383	5		22	9	1958			1	0	Fang
	6			47,934143	16,854383	5		26	10	1961			4	0	Fang
	7			47,934143	16,854383	5		19	6	1962			1	0	Fang
	8			47,934143	16,854383	5		5	11	1984			2	0	Fang
	9			47,934143	16,854383	5		16	5	1985			1	0	Fang

Datensatz: 1 von 355 | Kein Filter | Suchen

alle sichtbaren selektieren
alle sichtbaren de-selektieren
alle de-selektieren (auch die nicht sichtbaren)

Admin

Ortsmarkierung alle
Overlay alle

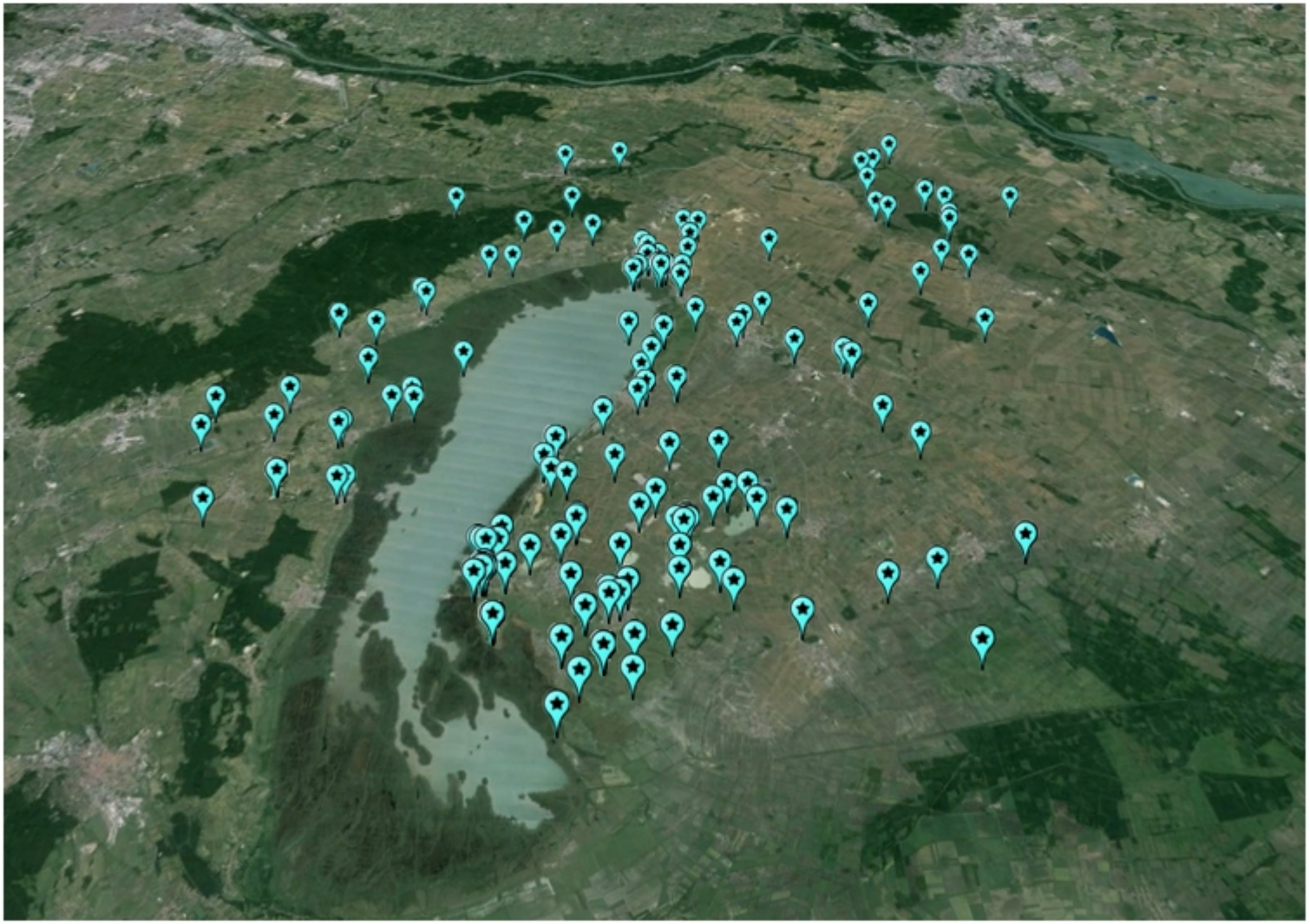
A: Update — Latitude — Longitude

B: Update — Latitude — Longitude

C: Update — Latitude — Longitude

zB 48°12'16.34"N, 16°21'38.89"E

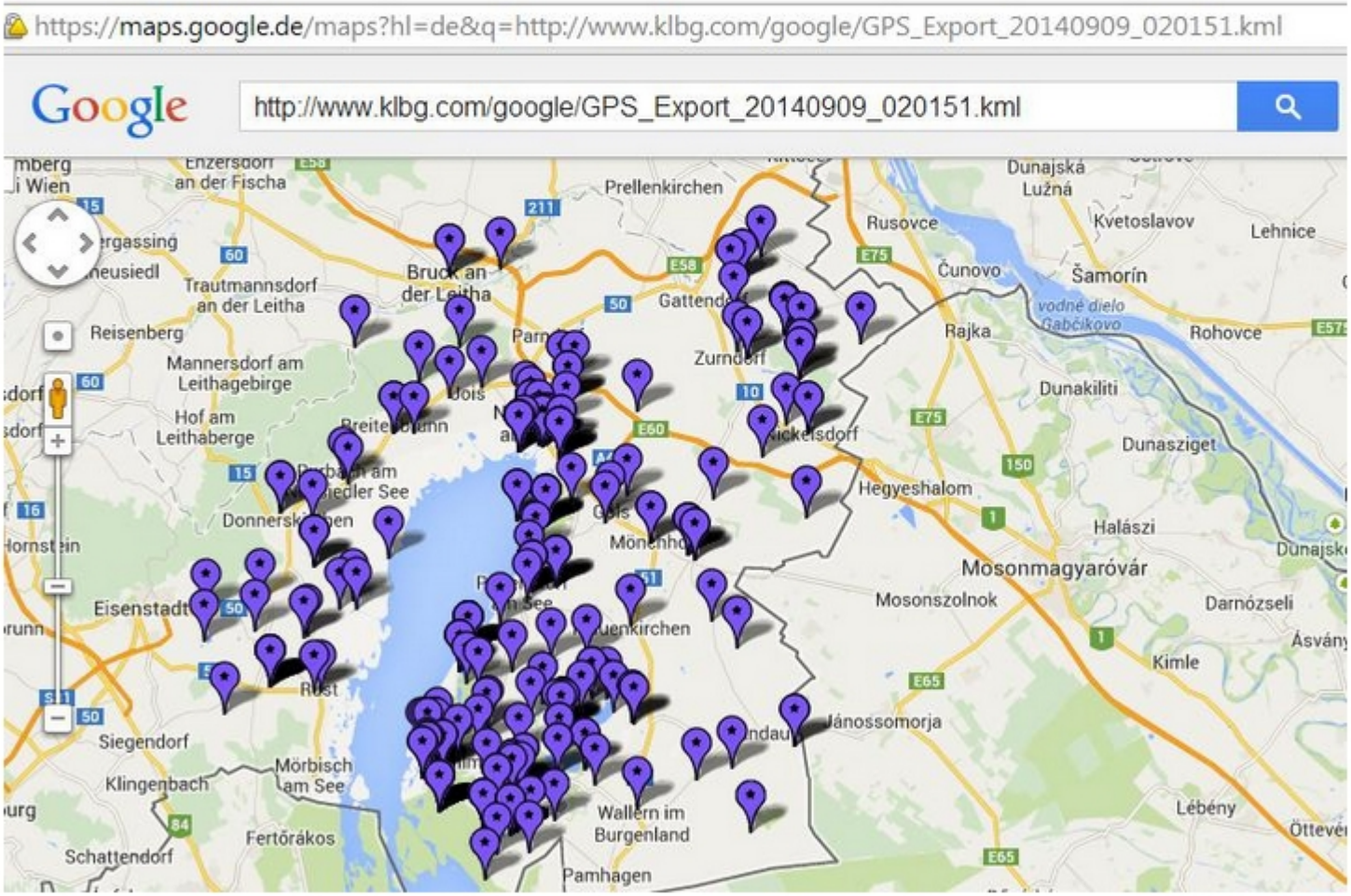
GPS List Ausgabe: Ortsmarken



Geo-Koordinaten in Access

Öffnen von KML-Dateien in Google Maps über URL-Parameter

https://maps.google.de/maps?hl=de&q=http://www.klbg.com/google/GPS_Export_20140909_020151.kml



GPS List Funktionsweise: Templates

Start

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
  <name><<NAME>></name>
  <Style id="MyStyle">
    <IconStyle id="MyIconStyle">
      <Icon>
        <href><<URL_PIN>></href>
      </Icon>
    </IconStyle>
  </Style>
  <Folder>
      <name><<FOLDERNAME>></name>
```

Block

```
        <Placemark>
           <name><<NAME>></name>
           <description><<INFO>></description>
           <styleUrl>#MyStyle</styleUrl>
           <Point>
              <coordinates><<LONGITUDE>>,<<LATITUDE>>,0</coordinates>
           </Point>
        </Placemark>
```

Ende

```
    </Folder>
</Document>
</kml>
```

GPS List Funktionsweise: Overlays

Nach dem gleichen Schema verfährt man bei Overlays, falls man auch diese automatisch einfügen will.
Beispiel: In KML gibt es keinen Kreis, manche behelfen sich mit regulären Polygonen siehe https://code.google.com/p/kmlcircle/ . Man kann aber auch ein Overlay als Kreis (Scheibe, Linie, radialer Verlauf, transparent oder nicht) verwenden. Dabei ergibt sich folgendes Problem:

Abstand der Meridiane de.wikipedia.org/wiki/Geographische_Länge

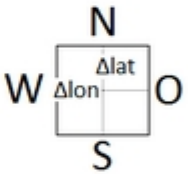
Der Abstand zwischen zwei Längenkreisen ist am Äquator am größten. Dort ist die Abweitung von einem Grad 111,32 km groß. Diese Distanz beträgt in etwa 60 Seemeilen, da die Seemeile ursprünglich einer Bogenminute entsprach.

An den Polen hingegen fallen alle Längengrade in einem Punkt zusammen. Der Abstand a zwischen zwei Längenkreisen lässt sich für einen Breitengrad φ und den Erdradius r näherungsweise mit folgender Formel berechnen:

$$a = \frac{2\pi \cdot r \cdot \cos\varphi}{360}$$

Mit r = 6.371 km erhält man $a \approx 111,2km \cdot \cos\varphi$

In Deutschland liegt dieser Wert zwischen 63,8 km (Flensburg, Sylt) und 75,5 km (Oberstdorf).



Für einen regional begrenzten Bereich kann man das einmal ausrechnen. Für grosse Bereiche muss man die korrekten Werte jedesmal ausrechnen.

GPS List Funktionsweise: Overlays

```
Public Function GetDeltaLat(dblKm As Double) As Double
'GetDeltaLat(2)
Const PI = 3.14159265358979
Const R = 6371
   GetDeltaLat = ((dblKm * 360) / (2 * PI * R))
End Function
```

```
Public Function GetDeltaLon(dblLat As Double, dblKm As Double) As Double
'GetDelta(0,2)
'GetDelta(90,2)
Const PI = 3.14159265358979
Const R = 6371
   If Abs(dblLat) > 85 Then
      GetDeltaLon = 0
   End If
   GetDeltaLon = ((dblKm * 360) / (2 * PI * R * Cos(Deg2Rad(dblLat))))
End Function
```

```
? GetDeltaLat(111.32)
 1.00112481170873

? GetDeltaLon(0,111.32)
 1.00112481170873
? GetDeltaLon(47.42,75.5)
 1.00350098125649
? GetDeltaLon(54.8,63.8)
 0.995376505586378
```

Abstand der Meridiane

... Äquator am größten. Dort ist die Abweitung von einem Grad 111,32 km

... und 75,5 km (Oberstdorf).

... zwischen 63,8 km (Flensburg, Sylt)

Die Funktion arbeitet aber in der Nähe der Pole nicht korrekt.

GPS List Ausgabe: Overlays

Google Maps APIs: Javascript, Webservices, ...

- Einbindung von Google-Maps-Karten in eigene Seiten
- Erstellung komplexer standortbasierender Anwendungen
  → Nutzung als Darstellungsschicht
  ← Holen von Geoinformationen
- Im Vergleich mächtiger und umfangreicher als KML

Google Maps APIs: Javascript, Ortsmarke

```
function initialize() {
  var myLatlng = new google.maps.LatLng(-25.363882,131.044922);
  var mapOptions = {
    zoom: 4,
    center: myLatlng
  }
  var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

  var marker = new google.maps.Marker({
      position: myLatlng,
      map: map,
      title: 'Hello World!'
  });
}
google.maps.event.addDomListener(window, 'load', initialize);
```

Google Maps APIs: Javascript, Polygon

```
// This example creates a simple polygon representing the Bermuda Triangle.
function initialize() {
  var mapOptions = {
    zoom: 5,
    center: new google.maps.LatLng(24.886436490787712, -70.2685546875),
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };
  var bermudaTriangle;
  var map = new google.maps.Map(document.getElementById('map-canvas'),
      mapOptions);
  // Define the LatLng coordinates for the polygon's path.
  var triangleCoords = [
    new google.maps.LatLng(25.774252, -80.190262),
    new google.maps.LatLng(18.466465, -66.118292),
    new google.maps.LatLng(32.321384, -64.75737),
    new google.maps.LatLng(25.774252, -80.190262)
  ];
  // Construct the polygon.
  bermudaTriangle = new google.maps.Polygon({
    paths: triangleCoords,
    strokeColor: '#FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 2,
    fillColor: '#FF0000',
    fillOpacity: 0.35
  });
  bermudaTriangle.setMap(map);
}
google.maps.event.addDomListener(window, 'load', initialize);
```

Access Awendung III: Google Maps API Javascript, Heat Map DB

Heat-Map aus Urlaubsbildern (Bewegungsmuster, Kundenverteilung)

Gewinnung der Koordinaten

Access Awendung III: Google Maps API Javascript, Heat Map DB

Heat-Map aus Urlaubsbildern (Bewegungsmuster, Kundenverteilung)

Gewinnung der Koordinaten

```
Public Function Get_Lat(strFile As String) As String
Dim wsh As New WshShell
Dim cmd As WshExec
   Set cmd = wsh.Exec("c:\exiv2\exiv2 -g Lat " & GetEnvironmentVariable("PicturePath") & strFile)
   Do While cmd.Status = WshRunning
      Sleep 10
   Loop
   Get_Lat = cmd.StdOut.ReadAll
End Function
```

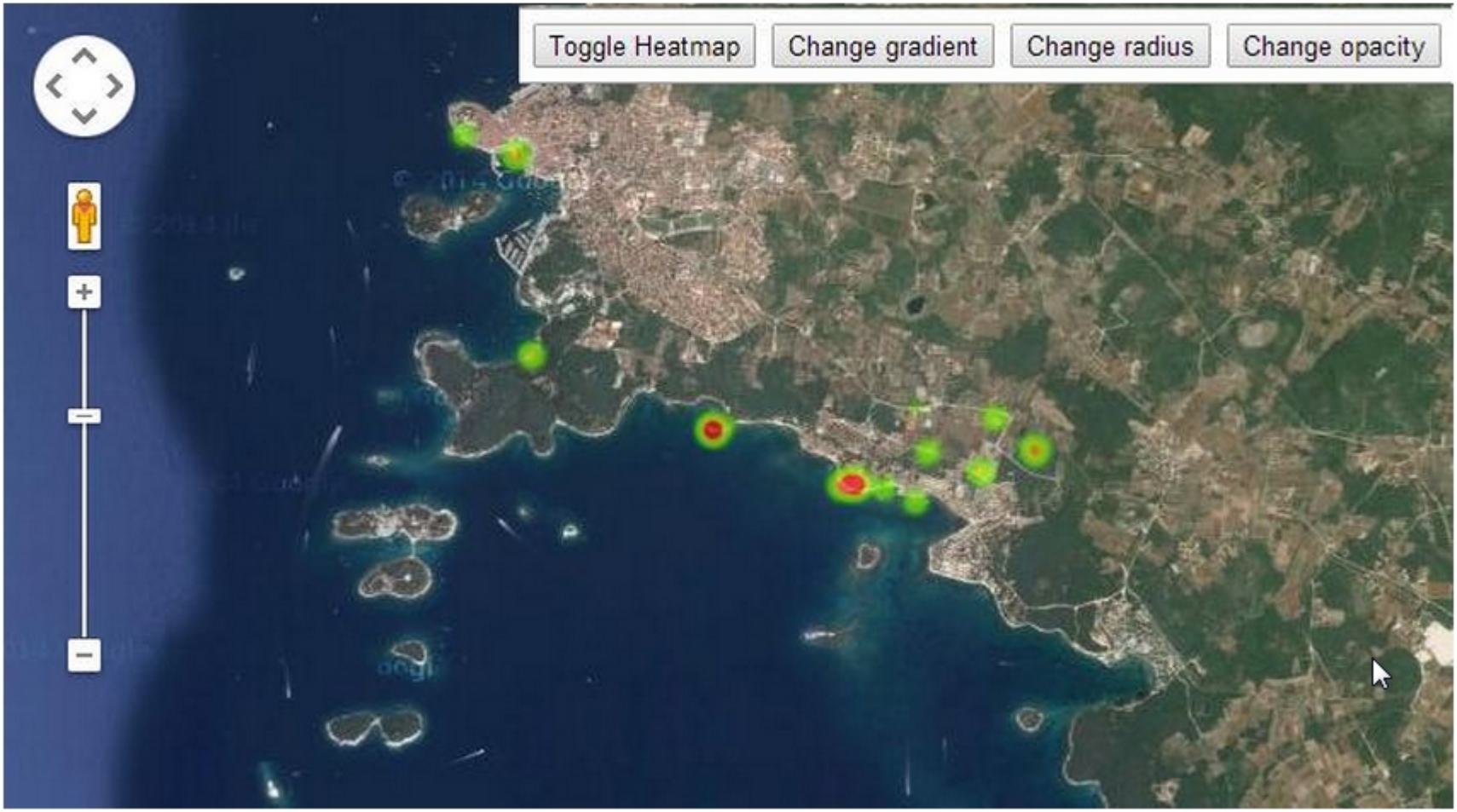
```
Public Function Get_Lon(strFile As String) As String
Dim wsh As New WshShell
Dim cmd As WshExec
   Set cmd = wsh.Exec("c:\exiv2\exiv2 -g Lon " & GetEnvironmentVariable("PicturePath") & strFile)
   Do While cmd.Status = WshRunning
      Sleep 10
   Loop
   Get_Lon = cmd.StdOut.ReadAll
End Function
```

```
Public Sub Get_GPS_Data()
Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim strFile As String
Dim strLat As String
Dim strLon As String
Dim l1, l2 As Long
On Error GoTo Err_Get_GPS_Data
   Set db = CurrentDb()
   Set rs = db.OpenRecordset("select * from tbl_GPS_Data")
    strFile = Dir(GetEnvironmentVariable("PicturePath") & "*.jpg")
    l1 = 0
    l2 = 0
    Do While Len(strFile) > 0
        l1 = l1 + 1
        Debug.Print strFile
        strLat = Get_Lat(strFile)
        strLon = Get_Lon(strFile)
        If (Len("" & strLat) * Len("" & strLon) > 0) Then
           l2 = l2 + 1
           rs.AddNew
           rs("File") = strFile
           rs("Lat") = Get_Lat(strFile)
           rs("Lon") = Get_Lon(strFile)
           rs.Update
        End If
        'Debug.Print Get_Lat(strFile)
        strFile = Dir
    Loop
    MsgBox ("fertig: " & l2 & " Bilder mit GPS-Daten von insgesamt " & l1 & " Bildern ...")
    Exit Sub
Err_Get_GPS_Data:
   MsgBox ("Error: " & Err.Number & ", " & Err.Description)
End Sub
```

Google Maps: Heat Map DB, Code

```
    ...
    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&libraries=visualization"></script>
    <script>
// Adding 500 Data Points
var map, pointarray, heatmap;

var taxiData = [
  new google.maps.LatLng(45.0654077777778, 13.6755325),
  new google.maps.LatLng(45.0622308333333, 13.6742502777778),
  ...
  ...
  ...
  new google.maps.LatLng(45.0612461111111, 13.666005)];

function initialize() {
  var mapOptions = {
    zoom: 13,
    center: new google.maps.LatLng(45.06, 13.65),
    mapTypeId: google.maps.MapTypeId.SATELLITE
  };
  map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
  var pointArray = new google.maps.MVCArray(taxiData);
  heatmap = new google.maps.visualization.HeatmapLayer({
    data: pointArray
  });
  heatmap.setMap(map);
}

function toggleHeatmap() {
  heatmap.setMap(heatmap.getMap() ? Null : map);
}

function changeGradient() {
  var gradient = [
    'rgba(0, 255, 255, 0)',
    ...
```

Google Maps: Heat Map DB, Ergebnis



Access Anwendung IV: Google Maps API Webservices, Get Coord

maps.googleapis.com/maps/api/geocode/xml?address=3420,AUSTRIA&sensor=false

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<GeocodeResponse>
  <status>OK</status>
 ▼<result>
   <type>postal_code</type>
   <formatted_address>3420, Österreich</formatted_address>
  ▼<address_component>
    <long_name>3420</long_name>
    <short_name>3420</short_name>
    <type>postal_code</type>
   </address_component>
  ▼<address_component>
    <long_name>Niederösterreich</long_name>
    <short_name>NÖ</short_name>
    <type>administrative_area_level_1</type>
    <type>political</type>
   </address_component>
  ▼<address_component>
    <long_name>Österreich</long_name>
    <short_name>AT</short_name>
    <type>country</type>
    <type>political</type>
   </address_component>
  ▼<geometry>
   ▼<location>
     <lat>48.3284087</lat>
     <lng>16.2979609</lng>
    </location>
    <location_type>APPROXIMATE</location_type>
   ▼<viewport>
     ...
```

```
   ▼<viewport>
    ▼<southwest>
      <lat>48.3197958</lat>
      <lng>16.2661710</lng>
     </southwest>
    ▼<northeast>
      <lat>48.3490029</lat>
      <lng>16.3251276</lng>
     </northeast>
    </viewport>
   ▼<bounds>
    ▼<southwest>
      <lat>48.3197958</lat>
      <lng>16.2661710</lng>
     </southwest>
    ▼<northeast>
      <lat>48.3490029</lat>
      <lng>16.3251276</lng>
     </northeast>
    </bounds>
   </geometry>
  </result>
</GeocodeResponse>
```

Access Anwendung IV: Google Maps API Webservices, Get Coord

Code
Dokumentation XML DOM
Dokumentation: SelectSingleNode

```
'-----------------------------------------------------------------------------------------
' Procedure : Geocode with Google Geocoding API v3
' Version   : 1.01
' DateTime  : 03/03/2011
' Author    : Philben
' Purpose   : converting addresses into geographic coordinates
' Parameter : No mandatory. string format or NULL
' Reference : http://code.google.com/intl/fr-FR/apis/maps/documentation/geocoding/index.html
' Remark    : Query limit of 2,500 geolocation requests per day
'           : A good accuracy is different of a good geocoding !!!
'           : Minimum delay between two queries : >= 200 ms
'-----------------------------------------------------------------------------------------
Public Function Geocode(Optional ByVal vAddress As Variant = Null, _
                        Optional ByVal vTown As Variant = Null, _
                        Optional ByVal vPostCode As Variant = Null, _
                        Optional ByVal vRegion As Variant = Null, _
                        Optional ByVal sCountry As String = "UNITED KINGDOM") As tGeocodeResult
   On Error GoTo catch
   Dim oXmlDoc As Object
   Dim sUrl As String, sFormatAddress As String
   If Not IsNull(vAddress) Then vAddress = Replace(vAddress, ",", " ")
   sFormatAddress = ("" & vAddress + ",") & _
                    ("" & vTown + ",") & _
                    ("" & vRegion + ",") & _
                    ("" & CStr(vPostCode) + ",") & _
                    "" & sCountry
   'To create the URL
   sUrl = "http://maps.googleapis.com/maps/api/geocode/xml?address=" & sFormatAddress & "&sensor=false"
   ''XMLDOM to get the XML response
   Set oXmlDoc = CreateObject("Microsoft.XMLDOM")
   With oXmlDoc
      .Async = False
      If .Load(sUrl) And Not .selectSingleNode("GeocodeResponse/status") Is Nothing Then
         'Status code
         Geocode.sStatus = .selectSingleNode("GeocodeResponse/status").Text
         'If a result is returned
         If Not .selectSingleNode("GeocodeResponse/result") Is Nothing Then
            'formatted_address
            Geocode.sRetAddress = .selectSingleNode("//formatted_address").Text
            'Accuracy
            Geocode.sAccuracy = .selectSingleNode("//location_type").Text
            'Latitude and longitude
            Geocode.dLatitude = Val(.selectSingleNode("//location/lat").Text)
            Geocode.dLongitude = Val(.selectSingleNode("//location/lng").Text)
         End If
      End If
   End With
   Set oXmlDoc = Nothing
   Exit Function
catch:
   Set oXmlDoc = Nothing
   Err.Raise Err.Number, , Err.Description
End Function
```

Rechtliche Aspekte

Copyright

First of all, it has to be noted that there is no legal protection for bare facts, only the form in which facts are presented can enjoy protection. For copyright protection to arise, it is necessary that a work meets two requirements:

- Originality (an intellectual creation)
- a personal feature (individuality)

Exceptions to Copyright: Facts and Data

"Copyright law does not apply to facts, data, or ideas." ... "However, copyright may protect a collection of data as contained in a database or compilation, but only if it meets certain requirements. Simply working really hard to gather the data is not enough." ... "It is important to remember that even if a database or compilation is arranged with sufficient originality to qualify for copyright protection, the facts and data within that database are still in the public domain. Anyone can take those facts and reuse or republish them, as long as that person arranges them in a new way. **Unless they are accessible only under a contract that conditions access on limiting how the facts and data may or may not be used**" ...

Copyright Protection of Databases

Rechtliche Aspekte: Google Maps API

https://developers.google.com/maps/terms#section_10_12

...
9.1 Free, Public Accessibility to Your Maps API Implementation.
...
(b) No Pre-Fetching, Caching, or Storage of Content. ... , except that you may store: (i) limited amounts of Content for the purpose of improving the performance ... (and in no event for more than 30 calendar days), ...
...
(c) No Mass Downloads or Bulk Feeds of Content. ... For example, you are not permitted to offer a batch geocoding service that uses Content contained in the Maps API(s).
...
(g) No Use of Content without a Google Map. You must not use or display the Content without a corresponding Google map, unless you are explicitly permitted to do so in the Maps APIs Documentation, ...

Geo-Koordinaten in Access

Rechtliche Aspekte: Google Maps API, Pro Version

Google Maps API für Unternehmen
https://developers.google.com/maps/licensing?hl=de

What are the differences between Google Maps API for Business and Google Earth Enterprise?
http://www.google.com.hk/enterprise/earthmaps/maps-faq.html

Google Maps API Premier
http://www.google.com/intl/de_ALL/enterprise/earthmaps/maps.html

Geo-Koordinaten in Access

Rechtliche Aspekte: Google Maps API, Pro Version

Google Maps API für Unternehmen
https://developers.google.com/maps/licensing?hl=de

What are the differences between Google Maps API for Business and Google Earth Enterprise?
http://www.google.com.hk/enterprise/earthmaps/maps-faq.html

Google Maps API Premier
http://www.google.com/intl/de_ALL/enterprise/earthmaps/maps.html

Alternative: Visualisierung mit HTML5 Canvas

Zuletzt noch eine Alternative zu Google Maps und Google Earth. Die Darstelllung erfolgt nur mittels HTML5-Canvas und Karte als Hintergrund. Die vorgestellte Lösung funktioniert nur für regional abgegrenzte Gebiete, die Erdkrümmung wurde nicht berücksichtigt.

Screenshots zur Umsetzung
Zur Anwendung

Visualisierung mit HTML5 Canvas: Code

```
<!DOCTYPE HTML>
<html><!-- wilfriedbaumann@gmx.at 2014 -->
  <head>
    <style>
      body {margin: 0px;padding: 0px;}
      #myimg{position:absolute;z-index:1;}
      #container{
          display:inline-block;
          width:1240px;
          height:690px;
          margin: 1em;
          position:relative;
          box-shadow: 0 5px 50px #333;
      }
      #mycanvas{
          position:relative;
          z-index:20;
      }
    </style>
    <script>
      var i = 0;
      function set_bg(){
          i = i + 1;
          if (i == 9) i = 1;
          var img =
          document.getElementById('myimg');
          var imageObj = new Image();
          imageObj.src = "./" + i + ".jpg";
      }//set_bg
```

```
      function draw_data() {
          var canvas = document.getElementById('mycanvas');
          var context = canvas.getContext('2d');
          context.beginPath();
context.moveTo(954, 515);
context.lineTo(954, 515-5);
................
context.moveTo(527, 543);
context.lineTo(527, 543-5);
context.moveTo(538, 565);
context.lineTo(538, 565-5);
          context.stroke();
      }//draw_data
    </script>
  </head>
  <body>
    <div id="container">
      <img id="myimg" alt="" />
      <canvas id="mycanvas" width="1240" height="690">
      </canvas>
      <button onclick="set_bg()">&gt;</button>
    </div>
    <script>
      set_bg();
      draw_data();
    </script>
  </body>
</html>
```

Zusammenfassung

- Geokoordinaten: Bedeutung, Definition, praktische Probleme
- Demo-Access-Anwendungen
  - sphärische Geometrie
  - KML
  - Google APIs (Javascript, Webservices)
  - HTML5 Canvas
- Rechtliche Überlegungen

Datenschutz und Privatsphäre!