



Visualisierung historischer Geodaten mit Google Earth

Anhang

KML 2.1 Tag Referenz für REGIS:GE

Bennet Schulte

Technische Fachhochschule Berlin

Berlin, Juli 2007



Inhaltsverzeichnis Anhang

Inhaltsverzeichnis Anhang	II
Abbildungsverzeichnis Anhang	III
Keyhole Markup Language Tags	IV
A.1 <Object>	IV
A.2 <kml>.....	IV
A.3 <Document>	IV
A.4 <Folder>	IV
A.5 <Feature>	IV
A.6 <Geometry>	V
A.7 <LineString>	VI
A.8 <LinearRing>	VI
A.9 <Polygon>.....	VI
A.10 <MultiGeometry>	VII
A.11 <styleUrl>.....	VII
A.12 <LabelStyle>	VII
A.13 <LineStyle>	VII
A.14 <PolyStyle>.....	VII
A.15 <StyleMap>.....	VII
A.16 Radio-Style-Ordner	VIII
A.17 <IconStyle>	VIII
A.18 <BalloonStyle >.....	VIII
A.19 <LookAt>	IX
A.20 <ScreenOverlay>	X
A.21 <TimeSpan>	XI
A.22 <Region>	XI
A.23 <NetworkLink>.....	XV
A.24 <Link>	XV
A.25 Daten-Updates über NetworkLinks laden.....	XVII
A.26 Super-Overlays.....	XVIII
A.27 CGI-Scripting für KML	XVIII
Literaturverzeichnis.....	XVIII

Abbildungsverzeichnis Anhang

Abb. A-1:	KML-Elemente (Quelle: GOOGLE EARTH 2007B)	V
Abb. A-2:	radio-Style (Quelle: GOOGLE EARTH 2007B).....	VIII
Abb. A-3:	heading in KML (Quelle: GOOGLE EARTH 2007B).....	IX
Abb. A-4:	range, tilt und altitude in KML (Quelle: GOOGLE EARTH 2007B).....	X
Abb. A-5:	LatLonAltBox in KML (Quelle: GOOGLE EARTH 2007B).....	XII
Abb. A-6:	LOD und FadeExtent in KML (Quelle: GOOGLE EARTH 2007B).....	XIII
Abb. A-7:	Regionen in KML (Quelle: GOOGLE EARTH 2007B).....	XIV
Abb. A-8:	Updates via Links (Quelle: GOOGLE EARTH 2007B).....	XVII

Keyhole Markup Language Tags

GML bietet rund 1000 Tags, KML in der Version 2.1 nur 140 Tags. Von diesen sind für die Verwendung bei REGIS:GE nicht alle interessant, daher sind im Folgenden nur Tags beschrieben, die wichtig sind oder sein könnten (aus dem Englischen übersetzt nach GOOGLE EARTH 2007B).

A.1 <Object>

- ist ein abstraktes Element und die Obergruppe aller Elemente
- Objects können für Updates eine ID (Identifikation) besitzen (siehe A.25)
- ID's werden auch bei Styles verwendet
- ID ist eine Standard-XML-ID

A.2 <kml>

- ist das Wurzelement eines KML-Dokuments
- ist zwingend notwendig und beinhaltet die XML-Deklaration am Beginn der Datei
- `<kml xmlns="http://earth.google.com/kml/2.1"> ... </kml>`

A.3 <Document>

- beinhaltet Folder

A.4 <Folder>

- beinhaltet alle Elemente, z.B.: Placemarks, NetworkLinks, Overlays etc. und bündelt diese zu einer Gruppe
- wird verwendet, um diese Elementen hierarchisch zu ordnen
- es kann selbst Features wie `<visibility>` besitzen (siehe `<Feature>`)

A.5 <Feature>

- ist ein abstraktes Element und beschreibt Placemarks, NetworkLinks, Overlays
- mögliche Werte sind:
 - `<name>` : freiwählbarer Name eines Placemarks, Folders, Networklinks oder Labels des Objekts im Viewer
 - `<visibility>` : [Boolean Wert, voreingestellt 1 (ja)] gibt an, ob das Objekt beim Öffnen der KML angezeigt wird. In Google Earth kann der Nutzer diesen Wert über die Checkbox selbst verändern
 - `<open>` : [Boolean Wert, voreingestellt 1 (ja)], gibt an, ob der Inhalt eines Folder beim Öffnen der KML kollabiert ist (0) oder angezeigt wird. In Google Earth kann der Nutzer diesen Wert über die Checkbox selbst verändern

- **<Snippet>** : Eine nicht HTML unterstützende Kurzbeschreibung, die unter dem Namen der Placemarkliste in Google Earth angezeigt wird. Wenn kein Snippet definiert wurde, werden die ersten beiden Zeilen der Description (Beschreibung) verwendet.
 - Mit `maxLines="x"` kann die Anzahl der Zeilen für das Snippet bestimmt werden (voreingestellt: 2)
- **<description>** : beinhaltet die Beschreibung des Objekts und wird in einem Ballon in der 3D-Ansicht angezeigt, wenn auf das Element geklickt wurde
 - kann normalen Text beinhalten und auch einige HTML-Funktionen, aber keine anderen Sprachen wie PHP, JSP, ASP, VBScript, Javascript, Java oder Python
 - wenn der Text kein HTML beinhaltet, versucht Google Earth selbständig den Text in HTML zu formatieren (Absätze in `
` URL's in Hyperlinks etc.)
 - mit `<![CDATA[Inhalt]]>` können auch selbst Formatierungen bzw. Sonderzeichen, Bilder etc. in HTML Codierung eingefügt werden.

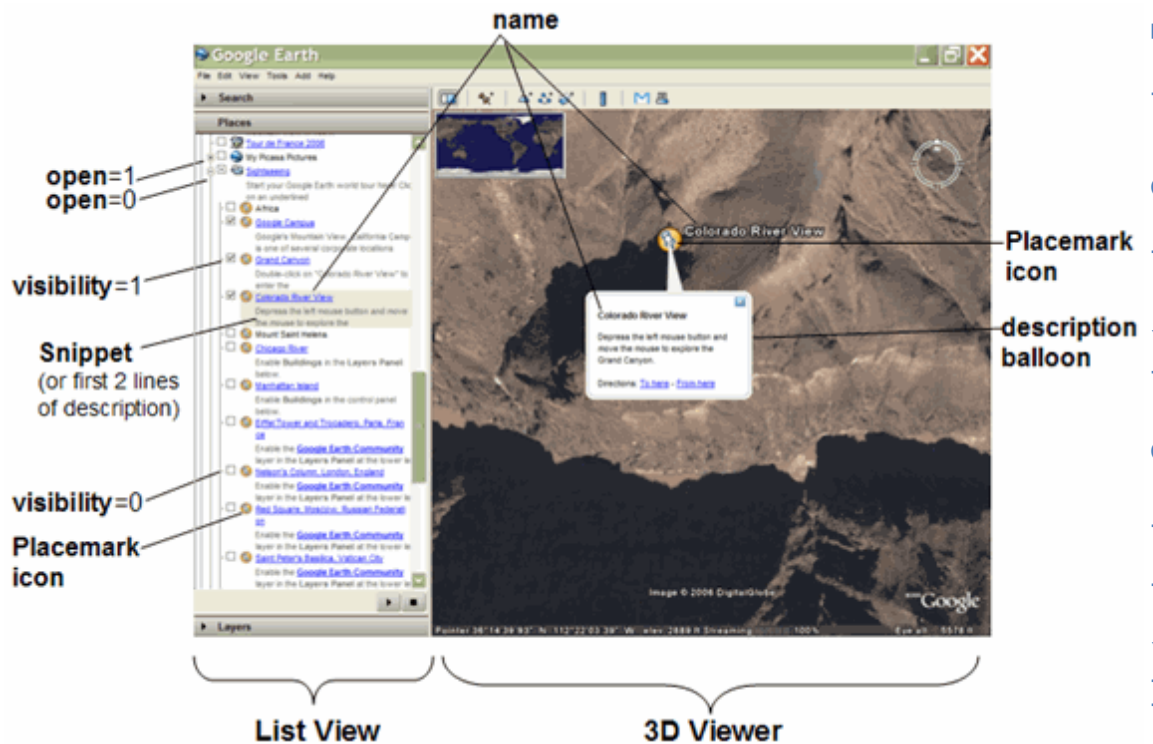


Abb. A-1: KML-Elemente (Quelle: GOOGLE EARTH 2007B)

A.6 <Geometry>

- ist ein abstraktes Element und beinhaltet alle Funktionen der Geometrie
- mögliche Werte sind:
 - **<extrude>** : [Boolean Wert, voreingestellt 0 (nein)], beschreibt, ob das geometrische Objekt mit dem Boden verbunden ist. Diese Funktion benö-

- tigt aber einen `<altitudeMode>` und verlängert die Grundfläche in den Himmel je nach angegebener Höhe und Modus.
- `<tessellate>` : [Boolean Wert, voreingestellt 0 (nein)], spezifiziert ob eine Linie oder Polygone dem Oberflächenmodell folgen darf (nur bei `<altitudeMode>` `clampToGround`). Besonders bei langen Linien macht es Sinn, damit sie der Oberfläche der Erde folgen und nicht teilweise unterirdisch liegen.
 - `<altitudeMode>` : spezifiziert wie Höhenangaben der `<coordinates>` interpretiert werden, mögliche Werte sind:
 - `clampToGround` (voreingestellt): ignoriert alle Höhenangaben in den `<coordinates>`, Feature befindet sich direkt auf dem Oberflächenmodell
 - `relativeToGround` : Feature wird immer in der definierten Höhe über dem Gelände dargestellt, egal wie hoch dieses ist
 - `absolute`: Feature wird immer mit Bezug auf den Meeresspiegel dargestellt, egal wie hoch das darunter befindliche Gelände ist
 - `<coordinates>` : Lageangabe aus zwei oder mehr Koordinaten. Diese werden als Gleitkommazahl für die geographische Länge, Breite und optional Höhe angegeben. Getrennt werden die Tupel durch Freizeichen (Länge,Breite,Höhe Länge,Breite,Höhe).

A.7 <LineString>

- definiert eine Linie
- über `<LineStyle>` können Farbe und Dicke der Linie definiert werden
- wenn ein `<LineString>` extrudiert entsteht eine Fläche bis zum Boden in Form einer Wand
- während die Linie den `<LineStyle>` (siehe A.13) verwendet, benutzt die entstandene Fläche einen `<PolyStyle>` (siehe A.14)

A.8 <LinearRing>

- definiert einen geschlossenen `<LineString>` und somit die Außenlinie eines Polygons. Mit ihm kann auch zur Erzeugung eines Lochs in einer Fläche ein innerer und äußerer Ring erzeugt werden (siehe A.9).
- mögliche Werte sind (siehe A.6):
 - `<extrude>`
 - `<tessellate>`
 - `<altitudeMode>`
 - `<coordinates>` (benötigt)

A.9 <Polygon>

- beinhaltet die Außengrenze eines geschlossenen Polygons (`<LinearRing>`)
- mögliche Werte sind:
 - `<outerBoundaryIs>` (benötigt) : beinhaltet einen `<LinearRing>`
 - `<innerBoundaryIs>` : beinhaltet einen `<LinearRing>`, welcher aus der Fläche des Polygons ausgeschnitten wird
 - mehrere Öffnungen möglich

A.10 <MultiGeometry>

- beinhaltet mehrere Arten Geometrien (Linie, Fläche, Punkt), die einem Feature zugehörig sind (nötig zum Erstellen von Features mit Exklaven oder Landflächen mit Inseln)

A.11 <styleUrl>

- gibt den zu verwendenden Style eines Objekts mittels einer Style ID an
- diese Style ID kann zu einem im Dokument definierten Style oder einer StyleMap führen oder auf eine externe Datei verweisen (URL+#ID=URI)
- z.B.
`<styleUrl>http://server.de/stylefile.xml#strasse</styleUrl>`

A.12 <LabelStyle>

- definiert wie das Element <name> eines Objekt in der 3D-Ansicht dargestellt wird
- mögliche Werte sind:
 - <color> : Farbe des Textes
 - <scale> : Größe des Textes, voreingestellt 1

A.13 <LineStyle>

- Definiert, wie ein Linienobjekt in der 3D-Ansicht dargestellt wird
- mögliche Werte sind
 - <color> : Farbe der Linie
 - <width> : Breite der Linie in Pixel, voreingestellt 1

A.14 <PolyStyle>

- definiert wie eine Fläche in der 3D-Ansicht dargestellt wird
- mögliche Werte sind
 - <fill> : [Boolean Wert, voreingestellt 1 (ja)], Füllung
 - <color> : Farbe der Füllung
 - <outline> : [Boolean Wert, voreingestellt 1 (ja)] → verwendet <LineStyle>

A.15 <StyleMap>

- verbindet zwei verschiedene Styles zu einem mit einer einzigen ID, z.B. für Mouse-Over-Effekte
- mögliche Werte sind:
 - <Pair> (benötigt): im Inneren werden durch <Key> die Modi normal oder highlight (Mouse-Over) einem Style - der durch die <styleUrl> definiert wird - zugewiesen

A.16 Radio-Style-Ordner

Es ist möglich den Nutzer in der freien Auswahl der Elemente der KML-Datei einzuschränken, die er gleichzeitig darstellen kann (siehe Abbildung 5-2). Dies kann über den `<listItemType> radioFolder` realisiert werden. So kann der Nutzer nur noch einen von ihm gewählten Ordner auf einmal darstellen. Alle anderen werden automatisch ausgeblendet.

```
<Style>
  <ListStyle>
    <listItemType>radioFolder</listItemType>
  </ListStyle>
</Style>
```

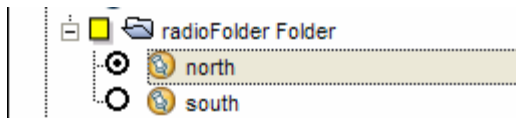


Abb. A-2: radio-Style (Quelle: GOOGLE EARTH 2007B)

A.17 <IconStyle>

- definiert wie Symbole für Punkt-Placemarks in der Liste und in der 3D-Ansicht aussehen
- mögliche Werte sind:
 - **<heading>** : Kompass-Richtungen in Grad (0 bis $\pm 180^\circ$), voreingestellt 0: Norden (siehe Abbildung 5-3)
 - **<scale>** : Größe des Symbols, voreingestellt 1
 - **<Icon>** : definiert eine Grafik für das Symbol, lokal oder im Netz
 - **<hotSpot>** : spezifiziert die Position, mit der das Symbol am `<Point>` verankert ist. Die x- und y-Werte können als Pixel (`pixels`), als Teile des Symbols (`fraction`) oder als Innenpixel (`insetPixels`) beschrieben werden

A.18 <BalloonStyle >

- Spezifiziert wie der Ballon, der die Beschreibung der Features im Viewer beinhaltet, aussieht (siehe Abbildung 5-1)
- mögliche Werte sind:
 - **<bgColor>** : Hintergrundfarbe (voreingestellt weiß: `ffffff`)
 - **<textColor>** : Textfarbe (voreingestellt Schwarz: `000000`)
 - **<text>**: beinhaltet den Text, der angezeigt werden soll, darüber hinaus können automatische Variablen eingefügt werden, die aus der Struktur (Name, Beschreibung etc.) des Elements den Inhalt ausgeben:
 - `$(name)`
 - `$(description)`
 - `$(id)`
 - `$(Snippet)`
 - `$(geDirections)` : erzeugt Google Earth-Navigationselement

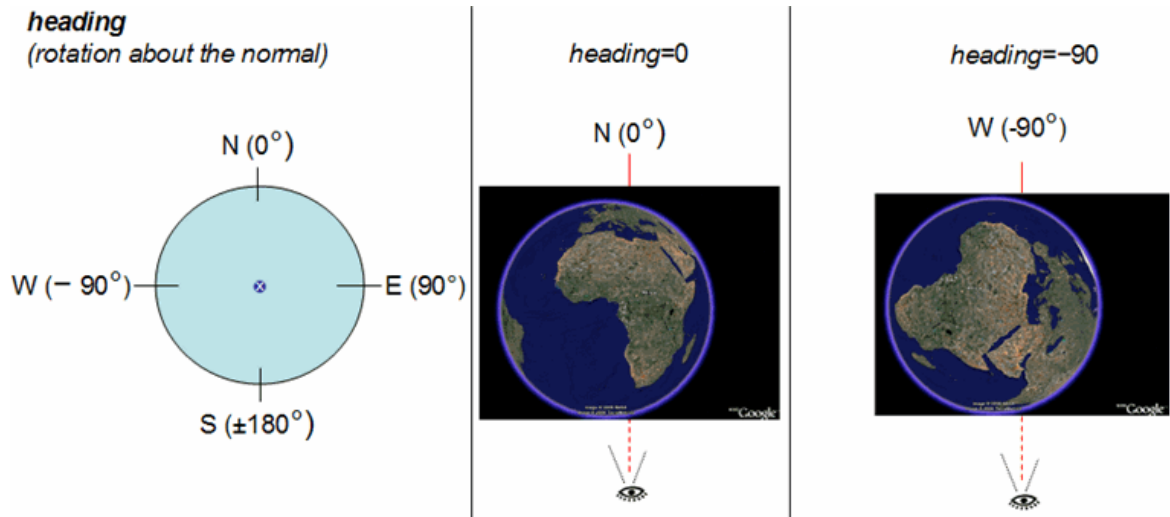


Abb. A-3: heading in KML (Quelle: GOOGLE EARTH 2007B)

A.19 <LookAt>

- Definiert eine Kameraposition, die zu einem Element gehört. Doppelklick auf ein Element löst einen Flug in die <LookAt> Kameraposition aus
- mögliche Werte sind:
 - <longitude> (benötigt) : Breitenangabe in Grad des Punktes, auf den die Kamera blickt, -180 bis 0 westlich und 0 bis 180 östlich des Mittelmeridians
 - <latitude> (benötigt) : Breitenangabe in Grad des Punktes, auf den die Kamera blickt -90° bis 90° (0=Äquator)
 - <altitude> : Höhe über der Erdoberfläche in Metern (voreingestellt = 0), siehe <altitudeMode> (5.3.4.6) für Interpretationsarten dieses Wertes
 - <range> (benötigt) : Entfernung in Metern zum Punkt, der durch <longitude>, <latitude>, und <altitude> angegeben ist (Siehe Abbildung 5-4)
 - <tilt> : Winkel (0-90°) zwischen der Richtung der <LookAt> Position und dem normalen Blick zur Erdoberfläche, 0 = Blick von Oben, 90 = Blick entlang des Horizonts (Siehe Abbildung 5-4)
 - <heading> : siehe A.17
 - <altitudeMode> : siehe A.6

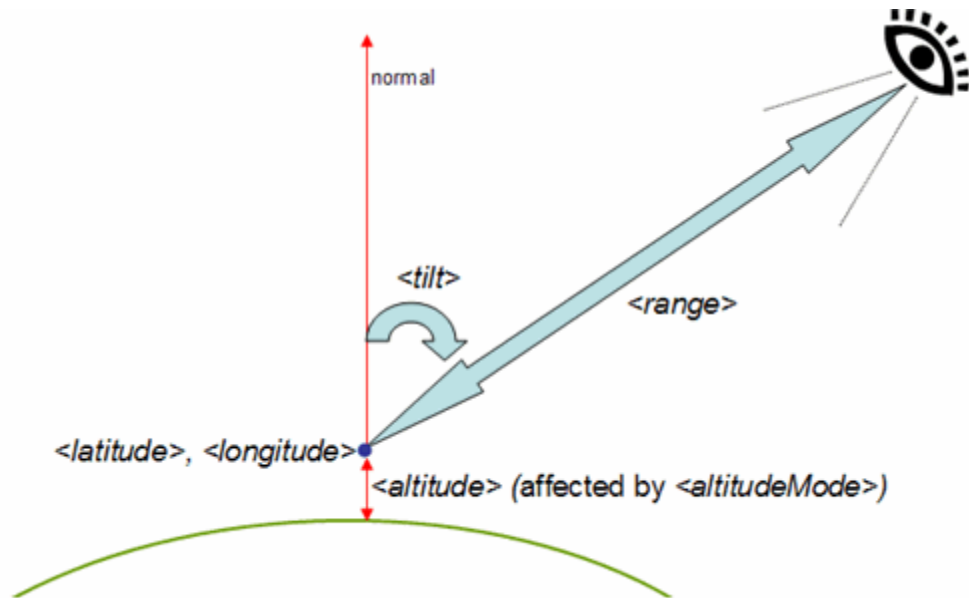


Abb. A-4: range, tilt und altitude in KML (Quelle: GOOGLE EARTH 2007B)

A.20 <ScreenOverlay>

- platziert eine fixierte Rastergrafik auf dem Bildschirm, z.B. ein Logo
- mögliche Werte sind:
 - **<overlayXY>** : spezifiziert einen Punkt auf oder außerhalb der Rastergrafik, an welchem sie an der Bildschirmkoordinatenposition (<screenXY>) dargestellt wird. Die x- und y-Werte können auf drei Weisen verwendet werden, als Pixel (`pixels`), als Abschnitt der Grafik (`fraction`), oder als Innenpixel (`insetPixels`), was den Abstand von der oberen rechten Ecke der Grafik beschreibt. Die x- und y-Lage kann auf verschiedene Weise spezifiziert werden, z.B. kann x in Pixel angegeben und y als Bildausschnitt. Der Koordinatenursprung ist die untere linke Ecke der Grafik
 - **<screenXY>** : spezifiziert einen Punkt relativ zum Ursprung des Bildschirms, an welchem die Grafik mit ihrem <overlayXY> platziert wird. Die x- und y-Werte können auf drei Weisen verwendet werden, als Pixel (`pixels`), als Abschnitt des Bildschirms (`fraction`), oder als Innenpixel (`insetPixels`), was den Abstand von der oberen rechten Ecke des Bildschirms beschreibt. Die x- und y-Lage kann auf verschiedene Weise spezifiziert werden, z.B. kann x in Pixel angegeben und y als Bildausschnitt. Der Koordinatenursprung ist die untere linke Ecke des Bildschirms.
 - **<rotationXY>** : ein Punkt relative zum Bildschirm an dem die Grafik rotiert wird
 - mögliche Werte sind:
 - **x,y** : Anzahl der Pixel
 - **xunits** : Einheit, in welcher der x-Wert dargestellt wird (voreingestellt: `fraction`)
 - ein Wert als Einheit `fraction` lässt den x-Wert einen Bildausschnitt des Monitors beschreiben (z.B. 0=links, 0.5=Mitte, 1=rechts) .

- ein Wert in `pixels` beschreibt die Anzahl der Pixel vom Koordinatenursprung unten links.
- ein Wert in `insetPixels` beschreibt die Anzahl der Pixel in der oberen rechten Ecke der Grafik oder des Bildschirms.
- **<size>** : Größe der Rastergrafik
- mögliche Werte sind
 - -1 Originalgröße
 - 0 Seitenverhältnis beibehalten
 - n Wert der Größe in Pixel
- Beispiele für `fraction` :

Zentrierte Grafik

```
<ScreenOverlay>
  <overlayXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
  <screenXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Grafik oben links

```
<ScreenOverlay>
  <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Grafik oben rechts

```
<ScreenOverlay>
  <overlayXY x="1" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="1" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

A.21 <TimeSpan>

- Beschreibt den Zeitraum, in dem das Feature angezeigt wird
- mögliche Werte sind:
 - **<begin>** : Beginn des Zeitraumes
 - **<end>** : Ende des Zeitraumes
 - fehlt einer der Werte ist das Ende bzw. der Beginn des Zeitraums offen
 - dabei wird als Datumsformat das des W3C XML-Schemas verwendet:
 - `yyyy-mm-ddThh:mm:sszzzzzz` :
 - T trennt Datum und Zeit
 - die Zeitzone ist entweder die koordinierte Weltzeitzone 0 (UTC) `Z` oder `zzzzzz`, was die Abweichung von UTC in `±hh:mm` angibt (W3C 2004)

A.22 <Region>

- Regionen sind ein mächtiges neues KML-Element, welches es gestattet sehr umfangreiche Datensätze in Google Earth darzustellen und Detaillevels zu defi-

nieren, um eine Generalisierung vornehmen zu können. Die Performance und das geladene Datenvolumen können dadurch begrenzt werden, dass Objekte nur angezeigt werden, wenn sie im Sichtbarkeitsbereich des Nutzers liegen oder einen bestimmten Teil der Bildschirmfläche einnehmen. Detaillevels können dazu dienen, detaillierte Objekte zu laden und anzuzeigen, wenn sie eine ausreichend große Fläche auf dem Bildschirm einnehmen und so in ihrem Detailgrad überhaupt sinnvoll nutzbar zu sein.

- mögliche Werte sind:
 - **Bounding Box**
 - Ein `<Region>` Element hat eine `<LatLonAltBox>`. Diese definiert eine Box, die nicht sichtbarer Teil eines Features ist. Sie hat eine Nord-, Süd-, Ost- und West-Ausdehnung sowie bei Bedarf (3D-Objekte oder 2D in der Luft) eine Höhe. Diese wird in ihrem Ausmaß in der Höhe durch eine Mindesthöhe `<minAltitude>` und eine Maximalhöhe `<maxAltitude>` festgelegt.
 - Objekte, die über eine Bounding Box verfügen, werden erst angezeigt, wenn ein Teil der Box den Sichtbereich des Bildschirms schneidet und die in `Level of Detail (LOD)` angegebene Quadratpixel-Anzahl für die Darstellung `LatLonAltBox` auf dem Bildschirm dem definierten Bereich entspricht. Erst wenn beides zutrifft wird die Region aktiviert.

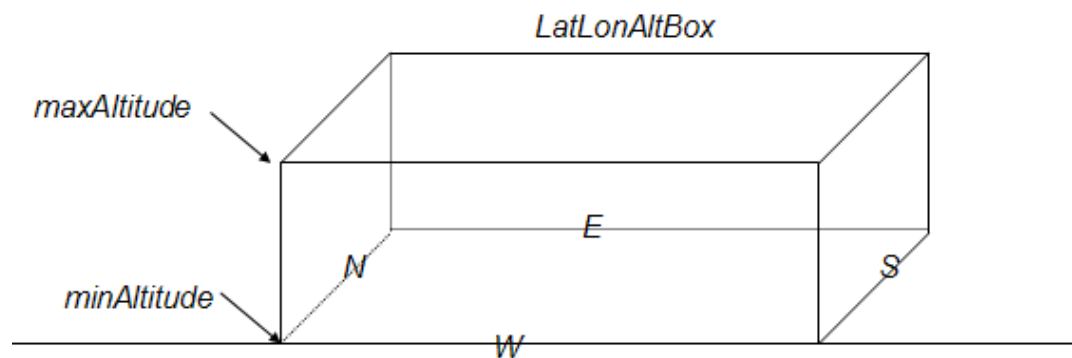


Abb. A-5: LatLonAltBox in KML (Quelle: GOOGLE EARTH 2007B)

- **Level of Detail (LOD):**
 - Das Detaillevel (LOD) dient dazu, Objekte nur anzuzeigen wenn sie eine bestimmte Größe auf dem Bildschirm einnehmen. Da der Monitor nur einen begrenzten sichtbaren Raum und eine Maximalauflösung anbietet ist dies sehr nützlich. Eine hohe Zahl von Elementen würden nicht viel nutzen, wenn sie sich beim Herauszoomen so überlagern, dass man sie nicht mehr erkennen kann. So können selbst große Datenmengen bewältigt werden, da diese erst visualisiert würden wenn sie groß genug dargestellt werden um sie erkennen zu können.
 - Es können auch gestaffelte Auflösungen realisiert werden, um die Ladedauer mit LOD zu optimieren. So können Rastergrafiken je nach Entfernung zur Kamera in höheren oder niedrigeren Auflösungen geladen werden. Wenn man z.B. eine Rastergrafik mit einer Flächenabdeckung von Berlin in einer Auflösung von 10000x10000 Pixel hat, wäre es sinnvoll, nicht die Grafik in ihrer gesamten Auflösung zu laden, wenn der Nutzer auf das Niveau von

ganz Europa heraus gezoomt ist. Stattdessen kann man eine niedrig aufgelöste Version einsetzen und mit der anderen austauschen sobald der Nutzer an Berlin heran zoomt. Dies geschieht über Angabe der Anzahl der Quadratpixel, welche die Grafik auf dem Schirm mindestens ausfüllen muss oder höchstens ausfüllen darf um angezeigt zu werden. Die Mindestanzahl wird über `<minLodPixels>` und die maximale Anzahl über `<maxLodPixels>` definiert. Dabei ist zu beachten, dass wenn die Kamera zum Objekt gekippt ist, die Quadratpixel gezählt werden, die aus dieser Position sichtbar sind. Somit kann sich die relative Höhe, in der es zur Einblendung bzw. Ausblendung kommt in Abhängigkeit vom Blickwinkel ändern.

- Für `<maxLodPixels>` ist unendlich (-1) voreingestellt.
- **Fade Extent:**
 - Es kann auch ein Fading-Bereich für eine Region angegeben werden. Mit dieser Funktion wird das zur Box gehörende Objekt sanft über einen opaken Zustand ein- und ausgeblendet, und das plötzliche Aufpoppen von Objekten vermieden. Google Earth nutzt `minFadeExtent` um den Bereich festzulegen, ab welchem das Objekt voll sichtbar ist, wobei bei `minLodPixels` das Einblenden beginnt. Von `maxFadeExtent` bis `maxLodPixels` ist der Abblendbereich in der anderen Zoomrichtung, wobei bei `maxFadeExtent` die Ausblendung beginnt und bei `maxLodPixels` abgeschlossen ist.
 - Fading ist sehr rechenintensiv und sollte möglichst nicht auf Rastergrafiken angewendet werden. Ansonsten können alle Objekte außer Icons (werden angezeigt sobald der Bereich von 0.5 überschritten ist) gefaded werden.

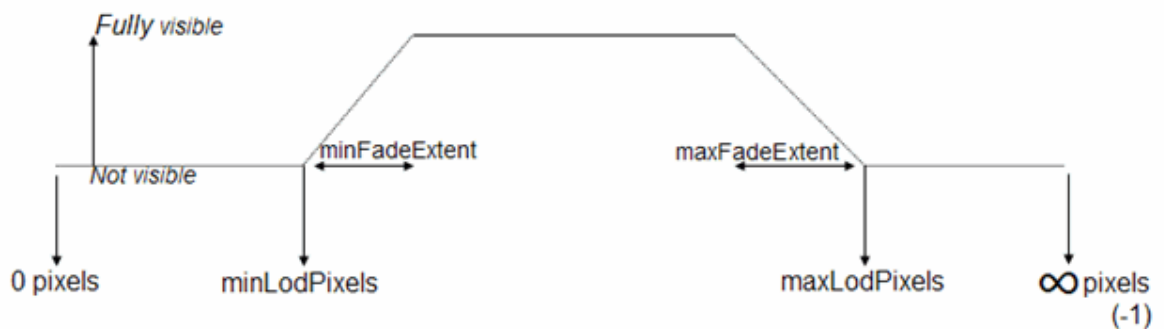


Abb. A-6: LOD und FadeExtent in KML (Quelle: GOOGLE EARTH 2007B)

- **Beispiel:**

```

<Region>
  <LatLonAltBox>
    <north>40.750683130314</north>
    <south>40.748162385230</south>
    <east>-73.966608428427</east>
    <west>-73.969476624071</west>
  
```

```

    <minAltitude>0</minAltitude>
    <maxAltitude>300</maxAltitude>
    <altitudeMode>absolute</altitudeMode>
  </LatLonAltBox>
  <Lod>
    <minLodPixels>64</minLodPixels>
    <maxLodPixels>-1</maxLodPixels>

    <minFadeExtent>128</minFadeExtent>
    <maxFadeExtent>512</maxFadeExtent>
  </Lod>
</Region>

```

- **Verschachtelte Regionen:**

- Man kann Regionen auch miteinander verschachteln, größere Regionen mit gröberen Details mit kleineren Regionen mit feineren Details. Jede Region hat ihr eigenes LOD, das definiert wie groß die Region in Pixel auf dem Bildschirm sein muss, bevor der mit ihr verbundene Inhalt sichtbar wird. Wenn die Kamera näher kommt, ergänzt oder ersetzt die detailliertere Box mit einem größeren LOD die bereits geladene(n) Box(en) mit niedrigeren LOD und gröberen Details. Dabei sollte die `<LatLonAltBox>` Childregion vollständig im Inneren der `<LatLonAltBox>` der Parentregion sein.
- Regionen werden durch die Ordner- und NetworkLink-Hierarchie vererbt. Alle Elemente in einem Ordner erben z.B. die definierte Region des Ordners, in dem sie sich befinden und der in der Hierarchie höher steht.

- **„intelligentes“ Laden von Region-basierten Network Links**

- die Verwendung von Regionen in Verbindung mit NetworkLinks erlaubt es eine Hierarchie zu erstellen, mit der eine Sub-Region definiert werden kann. Dies geschieht über den NetworkLink, der die entsprechende KML-Datei erst lädt, wenn die `<LatLonAltBox>` des Links aktiv wird. Der `<viewRefreshMode>` muss dafür den Auslöser `onRegion` erhalten. Mit diesem System können auch enorme Mengen an Daten zur Verfügung gestellt werden, da sie erst geladen werden wenn sich der Nutzer in ihrem Sichtbereich befindet.

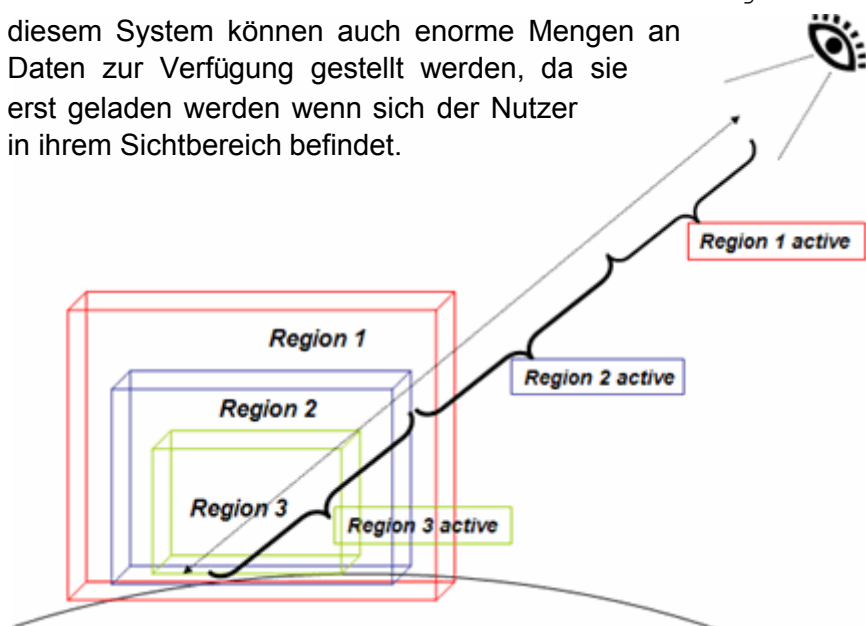


Abb. A-7: Regionen in KML (Quelle: GOOGLE EARTH 2007B)

A.23 <NetworkLink>

- verbindet eine KML- oder KMZ-Datei von der Festplatte oder aus dem Internet mit Google Earth
- können in Kombination mit Regionen benutzt werden um sehr große Datensets effektiv zu laden
- mögliche Werte sind:
 - **<Link>** (benötigt) : siehe Link
 - **<refreshVisibility>** : [Boolean Wert, voreingestellt 0 (Nutzer bestimmt)], Sichtbarkeitseinstellungen eines Features. Mit dem Wert 1 setzt bei einer Aktualisierung über den NetworkLink die Sichtbarkeit auf die Einstellung der **<visibility>** zurück
 - **<flyToView>** : [Boolean Wert, voreingestellt 0 (Nutzer bestimmt)], die Kameraposition eines Features wird bei einem Wert von 1 bei einer Aktualisierung immer auf den **<LookAt>** des die Elemente beinhaltenden Ordners zurückgesetzt

A.24 <Link>

- ein Link ist eine Weiterleitung und beschreibt die Verknüpfungen für:
 - KML-Dateien in NetworkLinks
 - Rastergrafiken von Icons, Iconstyles und Overlays
 - Model-Dateien im **<Model>** Element
- Verknüpfungen werden abhängig von Parametern geladen und aktualisiert
- Dabei gibt es die Möglichkeit Parameter in Abhängigkeit von der Zeit, **<refreshMode>** und **<refreshInterval>** und der Lage der Kamera, **<viewRefreshMode>** und **<viewRefreshTime>** zu definieren. Der Link bestimmt auch den Maßstab des Bounding Box-Elements, das an den Server gesendet wird (**<viewBoundScale>**) und kann eine Reihe optionaler Sichtparameter - und Parameter die Version und Sprachinformationen enthalten - an den Server senden (**<viewFormat>**).
- wenn eine Datei eingesetzt wurde, besteht die URL, die an den Server gesendet wird aus drei Informationen:
 - dem href (Hypertext Reference) welcher die zu ladende Datei spezifiziert
 - einen arbitrary format string, welcher durch den (a) Parameter der im **<viewFormat>** angegeben wurde spezifiziert wurde, oder den (b) Bounding Box Parameter wenn keine **<viewFormat>** Element vorhanden ist
 - einen zweiten format string der im **<httpQuery>** Element spezifiziert ist
- wenn die Datei, die im **<href>** angegeben ist, lokal gespeichert ist werden **<viewFormat>** und **<httpQuery>** nicht verwendet
- mögliche Werte sind:
 - **<href>** (benötigt) : eine URL (http-Adresse oder Pfad zur lokaler Datei). Wenn das Elternelement des **<Link>** ein NetworkLink ist, ist **<href>** eine KML-Datei. Ist das Elternelement ein Model, ist **<href>** eine COLLADA-Datei; ist es ein Overlay ist **<href>** eine Grafik. Relative URL's können benutzt werden und müssen relativ zur KML-Datei angegeben werden die den **<Link>** beinhaltet.
 - **<refreshMode>** : Gibt einen zeitbasierten Aktualisierungsmodus an

- mögliche Werte sind:
 - **onChange** : Aktualisierung, wenn die Datei geladen wird oder sich die Linkparameter ändern (voreingestellt)
 - **onInterval** : Aktualisierung alle n Sekunden (angegeben in `<refreshInterval>`)
 - **onExpire** : Aktualisierung wenn die Ablaufzeit erreicht ist
- `<refreshInterval>` : Gibt die Anzahl der Sekunden an, nach denen aktualisiert wird (voreingestellt = 4)
- `<viewRefreshMode>` : Spezifiziert, wie der Link aktualisiert wird, wenn sich die Kamera ändert
 - mögliche Werte sind:
 - **never** (voreingestellt) : ignoriert Änderungen in der Ansicht und `<viewFormat>` Parameter
 - **onStop** : Aktualisierung n Sekunden (in `<viewRefreshTime>`) nach dem Ende der Bewegung
 - **onRequest** : Aktualisierung nur auf Anweisung des Nutzer (z.B. durch Rechtsklick, Aktualisieren in Google Earth)
 - **onRegion** : Aktualisierung, wenn die Region aktiv wird (siehe A.22)
- `<viewRefreshTime>` : Gibt die Anzahl der Sekunden an, die nach Ende der Bewegung vergehen, bevor aktualisiert wird (voreingestellt = 4) (siehe `<viewRefreshMode>` → `onStop`)
- `<viewBoundScale>` : Gibt die Größe der BBOX-Parameter an bevor diese zum Server gesendet werden (voreingestellt = 1). Ein Wert von `<1` bedeutet, weniger als den gesamten Bildschirm zu benutzen, bei einem Wert von `>1` wird alles verwendet, was über den sichtbaren Bereich der aktuellen Ansicht hinaus geht.
- `<viewFormat>` : Gibt das Format des query string an, der mit den `<href>` des Links verbunden ist, bevor die Datei geladen wird (wenn im `<href>` eine lokale Datei angegeben ist wird dieses Element ignoriert)
- Wenn ein `<ViewRefreshMode>` mit `onStop` definiert wird und der `<viewFormat>` Tag nicht in der Datei vorhanden ist, wird automatisch folgende Information an den query string gehängt:

BBOX=[bboxWest] , [bboxSouth] , [bboxEast] , [bboxNorth]

Diese Information entspricht der Web Map Service (WMS) Bounding Box-Spezifikation. Wenn ein leerer `<viewFormat>` Tag erstellt wird, wird diese Information nicht dem query string hinzugefügt.

Es können auch eigene Sichtparameter dem query string hinzugefügt werden. Wenn ein format string erzeugt wird, wird dieser statt der BBOX-Information verwendet. Wenn die BBOX-Information zusätzlich angehängt werden soll, müssen diese Parameter mit den anderen eigenen Parametern hinzugefügt werden.

Es können folgende Parameter für den format string genutzt werden (Google Earth wird automatisch alle aktuellen Werte zum Zeitpunkt der Erstellung des query string hinzufügen):

- **[lookatLon]**, **[lookatLat]** : geographische Länge und Breite des Punktes zu dem die Kamera schaut

- **[lookatRange]**, **[lookatTilt]**, **[lookatHeading]** : Kameraparameter (siehe <range>, <tilt>, und <heading> in <LookAt> A.19)
 - **[horizFov]**, **[vertFov]** : horizontales und vertikales Sichtfeld der Kamera
 - **[horizPixels]**, **[vertPixels]** : Größe in Pixel in der 3D Ansicht
 - **[terrainEnabled]** : gibt an, ob das Oberflächenmodell angezeigt wird
- **<httpQuery>** : fügt die Information zum query string hinzu, basierend auf den angebenen Parametern (Google Earth wird automatisch alle aktuelle Werte zum Zeitpunkt der Erstellung des query string hinzufügen)
 - mögliche Werte sind:
 - [clientVersion]
 - [kmlVersion]
 - [clientName]
 - [language]

A.25 Daten-Updates über NetworkLinks laden

- Über das Update-Element der NetworkLinkControl können Daten verändert, gelöscht oder ergänzt werden.

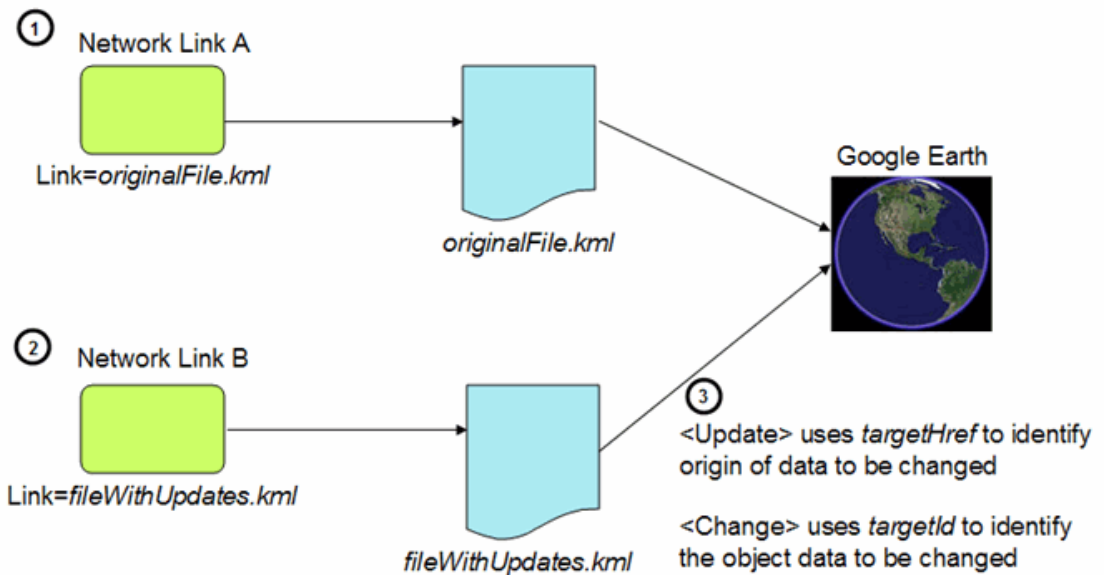


Abb. A-8: Updates via Links (Quelle: GOOGLE EARTH 2007B)

1. Ein NetworkLink lädt die original KML in Google Earth. Ein Element, das später aktualisiert werden soll, benötigt eine eindeutige ID.
2. Ein anderer NetworkLink lädt eine zweite KML-Datei, welche Updates enthält (jede Kombination aus Änderung, Erstellung und Löschung) zu den KML-Objekten, die bereits geladen wurden. Die Update-KML-Datei beinhaltet zwei Referenzen, um zu aktualisierende originale KML-Daten zu identifizieren:

3. `<Update>` benutzt das `targetHref` Element, um die zu ändernde original KML-Datei zu identifizieren und `<Change>` verwendet die eindeutige ID des Objekts. Ist das Objekt identifiziert, kann es nach Belieben verändert werden.

A.26 Super-Overlays

- Um z.B. eine 47MB große Rasterdatei in Google Earth effizient einzusetzen, macht es Sinn, die Rastergrafik in Kacheln zu zerlegen und in Auslösungsstufen - die nur geladen werden, wenn sie im Sichtbereich liegen - und auch nur in geeigneter Auslösung.
- Dies passiert wieder über Regionen und NetworkLinks, die erst aktiv werden, wenn sie im Sichtbereich sind und eine entsprechende Fläche des Bildschirms füllen und dann die entsprechende Kachel in entsprechender Auslösung laden. Beim herein- und herauszoomen werden diese dann durch andere Auflösungsstufen definiert und in der LOD (A.22) ersetzt

A.27 CGI-Scripting für KML

Zusätzlich zu Dateien mit statischen Daten kann das `<href>` eines NetworkLinks auch auf dynamisch genierte Daten, z.B. von einem CGI-Skript auf einen Server, verweisen. Mit Kenntnissen in Skriptsprachen wie PHP, Python oder Perl ist es möglich, ein Skript zu schreiben, dass KML-konforme Daten an jeden NetworkLink sendet.

Zwei Bedingungen sind notwendig, um KML durch ein CGI-Skript zu erzeugen:

Wenn eine Anfrage von Google Earth zum Server gesendet wird, muss erstens mit einem HTTP 200-Code geantwortet werden, und zweitens muss der Typ der Antwort `text/plain` oder `application/vnd.google-earth.kml+xml` sein.

Die Antwort muss eine gültige KML bilden.

Literaturverzeichnis

GOOGLE EARTH (2007B): KML Documentation. – Online in Internet:

<http://code.google.com/apis/kml/documentation> [Stand 2007-05-16].

W3C (2004): D ISO 8601 Date and Time Formats. – XML Schema Part 2: Datatypes

Second Edition. – Online in Internet: <http://www.w3.org/TR/xmlschema-2/#isoformats> [Stand 2007-05-15].