

# Master Thesis

im Rahmen des  
Universitätslehrganges „Geographical Information Science & Systems“  
(UNIGIS MSc) am Zentrum für GeoInformatik (Z\_GIS)  
der Paris Lodron-Universität Salzburg

zum Thema

## „Ökomorphologie Seeufer“ GIS-gestützte Voranalysen

vorgelegt von

**Anita Bertiller**

U1466, UNIGIS MSc Jahrgang 2011

Zur Erlangung des Grades  
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Gutachter:  
Ao. Univ. Prof. Dr. Josef Strobl

Köniz, 30.06.2014

## **Erklärung der eigenständigen Abfassung der Arbeit**

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit die wörtlich oder sinngemäss übernommen wurden sind entsprechend gekennzeichnet.

Köniz, 30. Juni 2014      Anita Bertiller

## Zusammenfassung

Im Jahr 2011 wurde das Schweizerische Gewässerschutzrecht revidiert. Aufgrund der Änderungen sind die Kantone unter anderem verpflichtet, bis Ende 2018 eine strategische Planung zur Revitalisierung von stehenden Gewässern zu erstellen. Dazu werden gute Kenntnisse der Seen vorausgesetzt. Ein wichtiger Bestandteil ist der ökomorphologische Zustand des Gewässers. Während mit Ökomorphologie Stufe F für Fließgewässer schon seit gut 15 Jahren eine bewährte Methode vorliegt, existiert derzeit noch kein anerkanntes Vorgehen zur Kartierung von stehenden Gewässern.

Teiber et al. (2013) prüften bisherige Kartierungen von Seeufern auf ihre Methode und Übertragbarkeit auf die Schweizer Seen. Dabei wurden geeignete Kriterien zur Beurteilung der Gewässerstruktur ausgearbeitet. Auf dieser Basis wird zurzeit in einem gemeinsamen Projekt vom Bundesamt für Umwelt (BAFU) und der Eidgenössischen Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz (Eawag) eine Richtlinie zur Uferkartierung von Seen erarbeitet. 18 Merkmale sollen den Zustand der rund 170 grössten Seen beschreiben. Da der Zeit- und Materialaufwand bei Feldkartierungen enorm ist, sind keine Vorortaufnahmen vorgesehen. Stattdessen soll die Erfassung der Merkmale am Computer mit Hilfe von Ortho- und Schrägbildern erfolgen. Der Aufwand bleibt trotzdem gross.

Die vorliegende Master Thesis untersucht die Möglichkeiten der GIS-gestützten automatisierten Auswertung bestehender Geodaten. Für jede Kennzahl wurde geprüft, welche Ausprägungen aus flächendeckend verfügbaren, homogenen Datensätzen extrahiert werden können. Als Testgebiet diente das Seeland mit Bieler-, Murten- und Neuenburgersee. Trotz geografischer Nähe unterscheiden sie sich stark bezüglich Grösse, Tiefe, Uferneigung und Siedlungsdruck.

Die Analysen zeigen, dass zwei der insgesamt 18 Attribute gut geeignet sind für eine automatisierte Extraktion und die Resultate plausibel sind. Weitere sieben Merkmale können zumindest teilweise automatisiert gewonnen werden und verringern den Aufwand bei der manuellen Erfassung am Bildschirm.

## Summary

In 2011, the Swiss water protection was revised, and the resulting changes oblige the cantons to draw up a strategic plan for the revitalisation of stagnant waters by the end of 2018. Therefore a thorough knowledge of the lakes is needed. An important part concerns the ecomorphological condition of the water bodies. While a proven method for rivers has existed for 15 years using the ecomorphology Level F, there is no accepted procedure for the mapping of standing waters.

Teiber et al. (2013) examined former field mappings of lakeshores on their method and transferability to the Swiss lakes. Appropriate criteria for the assessment of water structure have been elaborated. Currently, a method to determine the ecomorphology of lakeshores is being developed by the Federal Office for the Environment (FOEN) and the Swiss Federal Institute of Aquatic Science and Technology (Eawag). 18 attributes will describe the status of the 170 largest lakes. Due to the enormous amount of time and material involved, there will be no field mapping. Instead, the attributes will be gathered with the help of orthophotos and oblique images. Nevertheless, this effort remains quite challenging.

This master thesis investigates the possibilities of GIS-based automated evaluation of existing spatial data. It has been tested for each attribute whose characteristics can be extracted from homogeneous data sets that are available all over the country. The area tested was the region Seeland, comprised of the lakes of Biel, Murten and Neuchâtel. In spite of their geographical proximity, they differ greatly in terms of size, depth, shore slopes and settlement pressure.

Analyses show that two out of 18 attributes are well suited to automated extraction and the results are plausible. Seven other attributes can be at least partially gathered in an automated way, thus reducing the effort of manual recording at the screen.

## Verzeichnisse

### Inhaltsverzeichnis

<b>Erklärung der eigenständigen Abfassung der Arbeit</b>	<b>I</b>
<b>Zusammenfassung</b>	<b>II</b>
<b>Summary</b>	<b>III</b>
<b>Verzeichnisse</b>	<b>IV</b>
Inhaltsverzeichnis	IV
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	IX
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangslage	1
1.2 Motivation	1
1.3 Aufgabenstellung und Abgrenzung	2
1.4 Aufbau der Arbeit	3
<b>2 Grundlagen</b>	<b>4</b>
2.1 Seen	4
2.2 Gesetzgebung	5
2.3 Ökomorphologie	6
2.3.1 Modul-Stufen-Konzept	6
2.3.2 Ökomorphologie Stufe F	7
2.3.3 Ökomorphologie Seeufer	9
<b>3 Methode</b>	<b>12</b>
3.1 Untersuchungsgebiet	13
3.2 Datengrundlage, Hard- und Software	15
3.2.1 Topografisches Landschaftsmodell swissTLM3D	15
3.2.2 Höhenmodell swissALTI3D	16
3.2.3 Digitales Oberflächenmodell DOM	17
3.2.4 Orthofoto SWISSIMAGE FCIR	17
3.2.5 Bundesinventare	17
3.2.6 Ökomorphologie Stufe F	18
3.2.7 Eingesetzte Hard- und Software	18
3.3 Uferlinie	19
3.4 Beurteilungsräume zur Bestimmung der Attribute	21
3.5 Übergeordnete Attribute	23
3.5.1 Seentyp	23
3.5.2 Wasserspiegelregulierung	23
3.6 Beurteilungsraum Ufer	24
3.6.1 Ufertyp	24
3.6.2 Standorttypische Ufervegetation	25
3.6.3 Ufersaum-Vegetation	27
3.6.4 Uferverbau und künstliche Morphologie	28
3.6.5 Fließgewässeranbindung (Mündung / Abfluss)	29

3.6.6	Land- und forstwirtschaftliche Nutzung im Uferstreifen	30
3.6.7	Freizeit- und Privatnutzung im Uferstreifen	31
3.6.8	Siedlung, Gewerbe, Industrie im Uferstreifen	32
3.6.9	Verkehrswege und –flächen im Uferstreifen	33
3.7	Beurteilungsraum Flachwasser	35
3.7.1	Ausdehnung der Flachwasserzone	35
3.7.2	Wellenexposition	36
3.7.3	Standorttypische emerse Vegetation der Flachwasserzone	36
3.7.4	Strukturen / Anlagen in der Flachwasserzone	37
3.7.5	Sohlenveränderung in der Flachwasserzone	38
3.8	Beurteilungsraum Hinterland	39
3.8.1	Hinterland-Übergangsvegetation	39
3.8.2	Hinterlandnutzung	40
3.9	Aggregation	41
<b>4</b>	<b>Ergebnisse</b>	<b>42</b>
4.1	Ufertyp	42
4.2	Standorttypische Ufervegetation	43
4.3	Ufersaum-Vegetation	44
4.4	Uferverbau und künstliche Morphologie	44
4.5	Fliessgewässeranbindung (Mündung / Abfluss)	45
4.6	Land- und forstwirtschaftliche Nutzung im Uferstreifen	46
4.7	Siedlung, Gewerbe, Industrie im Uferstreifen	46
4.8	Verkehrswege und –flächen im Uferstreifen	47
4.9	Ausdehnung der Flachwasserzone	48
4.10	Strukturen / Anlagen der Flachwasserzone	48
4.11	Hinterland-Übergangsvegetation	49
<b>5</b>	<b>Diskussion</b>	<b>50</b>
<b>6</b>	<b>Schlussfolgerung und Ausblick</b>	<b>52</b>
	<b>Literaturverzeichnis</b>	<b>53</b>
	<b>Anhang 1: Minimale Attributliste Ökomorphologie Seeufer</b>	<b>55</b>
	<b>Anhang 2: Erstellung Beurteilungsräume (Python-Skript)</b>	<b>58</b>
	<b>Anhang 3: Auswertung der einzelnen Attribute (Python-Skript)</b>	<b>61</b>
	Anhang 3.01: Ufertyp	61
	Anhang 3.02: Standorttypische Ufervegetation	63
	Anhang 3.03: Ufersaum-Vegetation	66
	Anhang 3.04: Uferverbau und künstliche Morphologie	69
	Anhang 3.05: Fliessgewässeranbindung (Mündung / Abfluss)	71
	Anhang 3.06: Land- und forstwirtschaftliche Nutzung im Uferstreifen	73
	Anhang 3.07: Siedlung, Gewerbe, Industrie im Uferstreifen	76
	Anhang 3.08: Verkehrswege und –flächen im Uferstreifen	81
	Anhang 3.09: Ausdehnung der Flachwasserzone	87
	Anhang 3.10: Strukturen / Anlagen in der Flachwasserzone	89
	Anhang 3.11: Hinterland-Übergangsvegetation	91

**Abbildungsverzeichnis**

<b>Abb. 1</b>	Aufbau der Master Thesis.....	3
<b>Abb. 2</b>	Skizze der verschiedenen Bereiche im See (nach Spreafico & Weingartner 2005) .....	4
<b>Abb. 3</b>	Schritte der GIS-gestützten Analyse.....	12
<b>Abb. 4</b>	Die drei Jurarandseen: Bielersee (frz. Lac de Biemme), Murtensee (frz. Lac de Morat) und Neuenburgersee (frz. Lac de Neuchâtel) PK1000 im Massstab 1:500'000 (Quelle: Bundesamt für Landestopografie).....	13
<b>Abb. 5</b>	Die Vergletscherung der Schweiz zur letzten Eiszeit im Massstab 1:500'000 (Quelle: Bini et al. 2009) .....	13
<b>Abb. 6</b>	Verwendete Grundlagedaten der Region Bielersee im Massstab 1:85'000 (Quelle Geodaten: Bundesamt für Landestopografie).....	15
<b>Abb. 7</b>	Die aus dem swissTLM3D (oben) automatisch extrahierte Uferlinie folgt Bauten im Flachwasser (unten links: rote Linie). Sie wird manuell vereinfacht (unten rechts: rote Linie). (Quelle Geodaten: Bundesamt für Landestopografie).....	20
<b>Abb. 8</b>	Beurteilungsräume zur Bestimmung der Attribute (nach BAFU 2014a) (Quelle Geodaten: Bundesamt für Landestopografie) .....	21
<b>Abb. 9</b>	Umgesetzte Beurteilungsräume zur Bestimmung der Attribute (Quelle Geodaten: Bundesamt für Landestopografie) .....	22
<b>Abb. 10</b>	Vorgehen bei der Bestimmung des Ufertyps am Beispiel des Bielersees (Quelle Geodaten: Bundesamt für Landestopografie).....	25
<b>Abb. 11</b>	Damit einem Uferabschnitt der Typ 7.02 bzw. 7.03 zugewiesen wird, müssen die Begrenzungslinie des ausgestanzten Polygons (Wald bzw. Flachmoor / Feuchtgebiet) sowohl mit der Uferlinie als auch mit der 15 m-Linie ab Ufer identisch sein. Andernfalls weist die Ufervegetation nicht die geforderte Breite auf. (Quelle Geodaten: Bundesamt für Landestopografie).....	26
<b>Abb. 12</b>	Das swissTLM3D weicht vor allem bei eingedolten Abschnitten stark vom kantonalen Gewässernetz ab (Beispiel Bielersee). Mit Hilfe von unterschiedlich grossen Puffern werden bei eingedolten Abschnitten auch weiter vom kantonalen Datensatz entfernte Gewässer gelöscht und nicht als zusätzliche Mündung erkannt. (Quelle Geodaten: Bundesamt für Landestopografie).....	30
<b>Abb. 13</b>	Für einen Uferabschnitt können mehrere Kategorien von Verkehrsanlagen zutreffen (Beispiel Bielersee). Obwohl ein Fussweg näher beim Seeufer liegt als die Bahnlinie, wird die Kategorie übernommen, welche zu einer schlechteren Bewertung führt, d.h. die Ausprägung ist 14.11 und nicht 14.03. (Quelle Geodaten: Bundesamt für Landestopografie).....	34
<b>Abb. 14</b>	Die für die Überprüfung der automatisiert erfassten Attribute gewählten Uferbereiche sind rot dargestellt. PK1000 im Massstab 1:300'000 (Quelle: Bundesamt für Landestopografie) .....	42

**Tabellenverzeichnis**

<b>Tab. 1</b>	Bearbeitungsstufen des Modul-Stufen-Konzepts (nach Eawag 2014).....	6
<b>Tab. 2</b>	Wichtigste Merkmale der Ökomorphologie Stufe F (nach BUWAL 1998b).....	8
<b>Tab. 3</b>	Zustandsklassen Ökomorphologie Stufe F (nach BUWAL 1998b) .....	8

<b>Tab. 4</b>	Minimale Attributliste zur Seeuferbeurteilung (nach BAFU 2014a). Der Wirkungstyp beschreibt die Verwendbarkeit des Attributs für die Beurteilung: Zusatzinformation (z), Hauptattribut für Bewertung (b <sub>1</sub> ), Bonus-Malus-Attribut (b <sub>2</sub> ) oder übergeordneter Einfluss (e). Eine detaillierte Auflistung mit den Attributausprägungen ist in den Kapiteln 3.5 bis 3.8 sowie im Anhang 1 zu finden. ....	11
<b>Tab. 5</b>	Kennzahlen der drei Seen, die als Beispiel für die GIS-gestützte Analyse verwendet werden (nach BFS 2009b). ....	14
<b>Tab. 6</b>	Topics in swissTLM3D (nach swisstopo 2014a). ....	16
<b>Tab. 7</b>	Der Neuenburgersee ist in swissTLM3D mit gut doppelt so vielen Stützpunkten pro km erfasst als der Bieler- und Murtensee. Bei VECTOR25 waren es deutlich weniger. ....	19
<b>Tab. 8</b>	Ausprägungen des Merkmals „Seentyp“ und dessen Klassifizierung (nach BAFU 2014a). ....	23
<b>Tab. 9</b>	Ausprägungen des Merkmals „Wasserspiegelregulierung“ und dessen Klassifizierung (nach BAFU 2014a). ....	23
<b>Tab. 10</b>	Ausprägungen des Merkmals „Ufertyp“ und dessen Klassifizierung (nach BAFU 2014a). ....	24
<b>Tab. 11</b>	Ausprägungen des Merkmals „Standorttypische Ufervegetation“ und dessen Klassifizierung (nach BAFU 2014a). ....	25
<b>Tab. 12</b>	Ausprägungen des Merkmals „Ufersaum-Vegetation“ und dessen Klassifizierung (nach BAFU 2014a, abgeändert). ....	27
<b>Tab. 13</b>	Ausprägungen des Merkmals „Uferverbau und künstliche Morphologie“ und dessen Klassifizierung (nach BAFU 2014a). ....	28
<b>Tab. 14</b>	Ausprägungen des Merkmals „Fließgewässeranbindung“ und dessen Klassifizierung (nach BAFU 2014a). ....	29
<b>Tab. 15</b>	Ausprägungen des Merkmals „Land- und forstwirtschaftliche Nutzung im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a). ....	30
<b>Tab. 16</b>	Ausprägungen des Merkmals „Freizeit- und Privatnutzung im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a). ....	31
<b>Tab. 17</b>	Ausprägungen des Merkmals „Siedlung, Gewerbe, Industrie im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a). ....	32
<b>Tab. 18</b>	Ausprägungen des Merkmals „Verkehrswege und –flächen im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a). ....	33
<b>Tab. 19</b>	Ausprägungen des Merkmals „Ausdehnung der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a). ....	35
<b>Tab. 20</b>	Ausprägungen des Merkmals „Wellenexposition“ und dessen Klassifizierung (nach BAFU 2014a). ....	36
<b>Tab. 21</b>	Ausprägungen des Merkmals „Standorttypische emerse Vegetation der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a). ....	36
<b>Tab. 22</b>	Ausprägungen des Merkmals „Strukturen / Anlagen in der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a). ....	37
<b>Tab. 23</b>	Ausprägungen des Merkmals „Sohlenveränderung in der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a). ....	38
<b>Tab. 24</b>	Ausprägungen des Merkmals „Hinterland-Übergangsvegetation“ und dessen Klassifizierung (nach BAFU 2014a). ....	39
<b>Tab. 25</b>	Ausprägungen des Merkmals „Hinterlandnutzung“ und dessen Klassifizierung (nach BAFU 2014a). ....	40

<b>Tab. 26</b>	Gegenüberstellung Attribut 3: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	43
<b>Tab. 27</b>	Gegenüberstellung Attribut 7: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	43
<b>Tab. 28</b>	Gegenüberstellung Attribut 8: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	44
<b>Tab. 29</b>	Gegenüberstellung Attribut 9: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	44
<b>Tab. 30</b>	Gegenüberstellung Attribut 10: manuell erfasst – automatisiert extrahiert. Angaben in Anzahl Fließgewässeranbindungen.....	45
<b>Tab. 31</b>	Gegenüberstellung Attribut 11: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	46
<b>Tab. 32</b>	Gegenüberstellung Attribut 13: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	46
<b>Tab. 33</b>	Gegenüberstellung Attribut 14: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	47
<b>Tab. 34</b>	Gegenüberstellung Attribut 4: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	48
<b>Tab. 35</b>	Gegenüberstellung Attribut 15: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	48
<b>Tab. 36</b>	Gegenüberstellung Attribut 17: manuell erfasst – automatisiert extrahiert. Angaben in Metern.....	49
<b>Tab. 37</b>	Minimale Attributliste zur Seeuferbeurteilung (nach BAFU 2014a). Der Wirkungstyp beschreibt die Verwendbarkeit des Attributs für die Beurteilung: Zusatzinformation (z), Hauptattribut für Bewertung ( $b_1$ ), Bonus-Malus-Attribut ( $b_2$ ) oder übergeordneter Einfluss (e). .....	55

**Abkürzungsverzeichnis**

AU	Bundesinventar der Auengebiete von nationaler Bedeutung
BAFU	Bundesamt für Umwelt
BFS	Bundesamt für Statistik
BUWAL	Bundesamt für Umwelt, Wald und Landschaft (heute BAFU)
DHM25	Digitales Höhenmodell 1:25'000 (wird nicht mehr nachgeführt)
DOM	Digitales Oberflächenmodell (wird nicht mehr nachgeführt)
DSM	Digitales Seebodenmodell (wird nicht mehr nachgeführt)
Eawag	Eidgenössische Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz
FM	Bundesinventar der Flachmoore von nationaler Bedeutung
FWZ	Flachwasserzone
GeoIG	Geoinformationsgesetz
GeoIV	Geoinformationsverordnung
GSchG	Gewässerschutzgesetz
GSchV	Gewässerschutzverordnung
LABES	Landschaftsbeobachtung Schweiz
LV03	Landesvermessung 1903
LV95	Landesvermessung 1995
MESUBA	Methode der Seeuferbeurteilung und –aufwertung
MGDM	Minimales Geodatenmodell
NHG	Natur- und Heimatschutzgesetz
NIR	nahes Infrarot
PAN	panchromatisch
RGB	rot – grün – blau (additive Farbmischung)
SR	Systematische Rechtssammlung
swissALTI3D	Digitales Terrainmodell
SWISSIMAGE FCIR	Falschfarbeninfrarot-Orthofoto
swissTLM3D	Topografisches Landschaftsmodell mit Einbezug der dritten Dimension
swisstopo	Bundesamt für Landestopografie
TWW	Bundesinventar der Trockenwiesen und –weiden von nationaler Bedeutung
VECTOR25	Digitales Landschaftsmodell der Schweiz (wird nicht mehr nachgeführt)

# 1 Einleitung

## 1.1 Ausgangslage

In der dicht besiedelten Schweiz stehen Seen, Bäche und Flüsse unter grossem Druck. Sie sind von Siedlung und Landwirtschaft umgeben, werden zur Energiegewinnung genutzt und dienen der Erholung. Die verschiedenen menschlichen Nutzungen beeinflussen die Gewässer und ihre Funktion als natürliche Ökosysteme. Während früher die Wasserqualität im Mittelpunkt des Gewässerschutzes stand, wird seit einigen Jahren das Gewässer als Ganzes betrachtet. Ein Teil davon ist die Ökomorphologie. Mit der Kartierung des ökomorphologischen Zustands eines Gewässers wird die Gewässerstruktur (Sohle, Ufer, Umland) erfasst, die Beeinflussung durch den Menschen untersucht und die Lebensraumfunktion des Gewässers bewertet.

Im Jahr 2011 wurde das Schweizerische Gewässerschutzrecht revidiert. Aufgrund der Änderungen sind die Kantone unter anderem verpflichtet, bis Ende 2018 eine strategische Planung zur Revitalisierung von stehenden Gewässern zu erstellen. Um geeignete Massnahmen zur Revitalisierung festzulegen, sind gute Kenntnisse der Seen vorausgesetzt. Ein wichtiger Bestandteil ist der ökomorphologische Zustand des Gewässers. Während mit Ökomorphologie Stufe F (BUWAL 1998b) für Fliessgewässer schon seit gut 15 Jahren eine bewährte Methode vorliegt, existiert derzeit noch kein anerkanntes Vorgehen zur Kartierung von stehenden Gewässern.

Eine Richtlinie zur Uferkartierung von Seen wird zurzeit in einem gemeinsamen Projekt vom Bundesamt für Umwelt (BAFU) und der Eidgenössischen Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz (Eawag) erarbeitet. Sie soll ein einheitliches Verfahren zur Zustandserfassung der Schweizer Seen aufzeigen.

## 1.2 Motivation

Schweizweit vergleichbare Geodaten sind wünschenswert aber rar. Damit Monitoringaufgaben wahrgenommen werden können, sind aufwändige und fehlerbehaftete Integrationen kantonaler Datensätze in ein System nötig. Da bis heute erst vereinzelte Seeufer ökomorphologisch kartiert wurden, sind die Chancen für einen homogenen Datensatz „Ökomorphologie Seen“ intakt. Dazu beitragen soll ein praktikabler Ansatz, der ausschliesslich Geodaten des Bundes verwendet, um einige Parameter automatisiert zu bestimmen.

Die revidierte GSchV verpflichtet die Kantone zur Erhebung von Grundlagen für die Planung von Revitalisierungen. Die Kosten gehen zu einem grossen Teil zu Lasten der Kantone. Eine GIS-gestützte Voranalyse verringert den Kartieraufwand und die damit anfallenden Kosten.

### 1.3 Aufgabenstellung und Abgrenzung

Im Rahmen dieser Master Thesis soll eine GIS-gestützte Voranalyse erarbeitet werden, um die Kartierung der Ökomorphologie von Seen zu erleichtern. Eine Auswertekette auf bestehenden Geodaten soll die Effizienz der Kartierarbeiten steigern und zur Homogenität der Resultate bezüglich Inhalt, Qualität und Genauigkeit beitragen. Das Verfahren muss auf die gesamte Schweiz anwendbar sein. Daher sollen nur Daten verwendet werden, die flächendeckend verfügbar sind.

Im Speziellen soll folgende Forschungsfrage beantwortet werden:

Lassen sich Merkmale zur Beschreibung des ökomorphologischen Zustands der Schweizer Seen aus flächendeckend verfügbaren Geodaten automatisiert extrahieren?

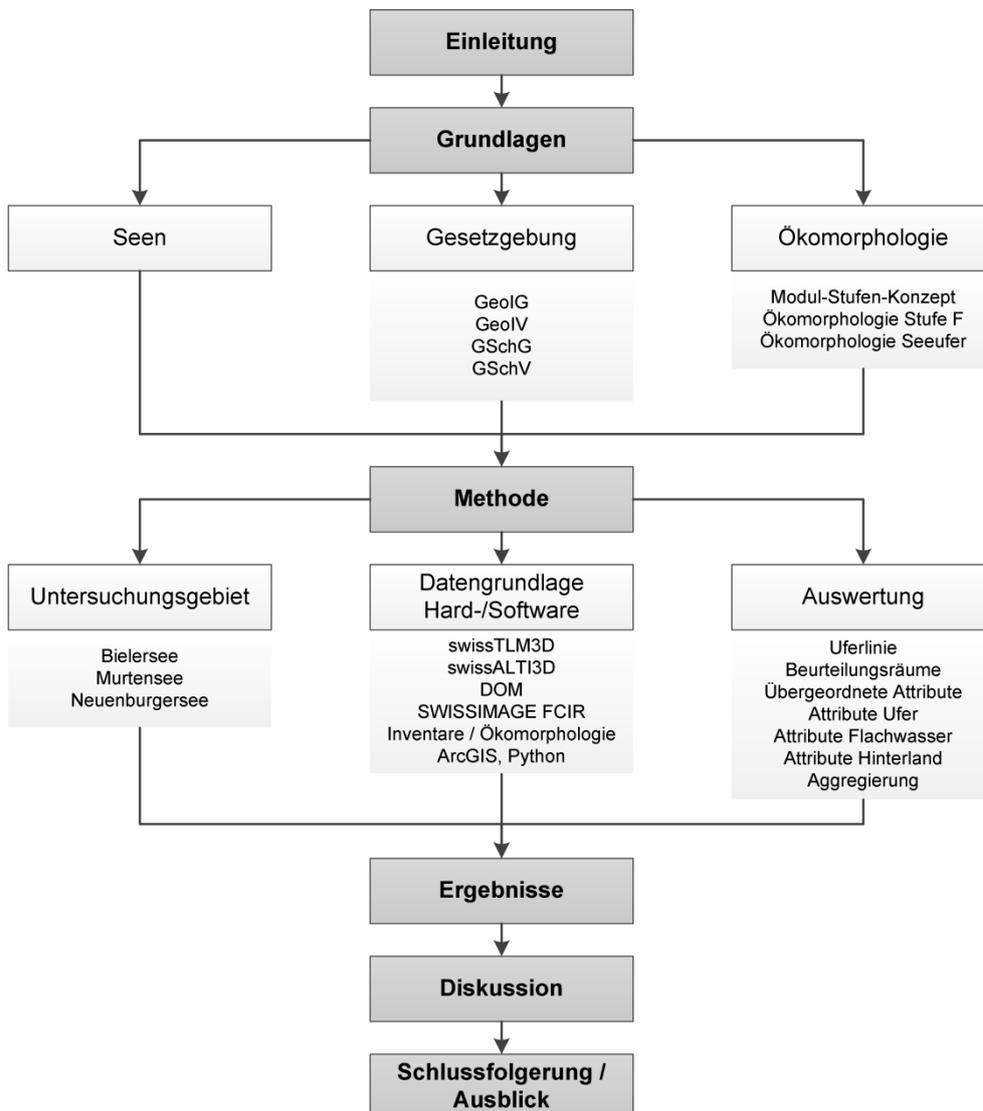
Vor diesem Hintergrund werden die folgenden untergeordneten Ziele bzw. Fragestellungen vertieft betrachtet:

- Wie gross müssen die Beurteilungsräume gewählt werden?
- Welche verfügbaren Geodaten eignen sich zur automatisierten Extraktion eines oder mehrerer Attribute zur Beschreibung des ökomorphologischen Zustands der Seen?
- Welche verfügbaren Geodaten eignen sich nicht zur Beschreibung des ökomorphologischen Zustands der Seen?
- Ist der vorgeschlagene Lösungsansatz praxistauglich?

Um die zu verarbeitende Datenmenge im Rahmen zu halten wird die GIS-gestützte Voranalyse exemplarisch anhand dreier Seen erarbeitet und geprüft. Als Resultat liegen keine abgeschlossenen Seeufer-Kartierungen vor.

## 1.4 Aufbau der Arbeit

Die vorliegende Master Thesis gliedert sich in sechs Bereiche (vgl. **Abb. 1**):



**Abb. 1** Aufbau der Master Thesis

Der einleitende Teil (Kap. 1) führt ins Thema ein. Dabei werden Motivation, Aufgabenstellung und Zielsetzung der Arbeit erläutert. Anschliessend folgt der Grundlagenteil (Kap. 2), der in die Bereiche „Seen“, „Gesetzgebung“ und „Ökomorphologie“ aufgeteilt ist.

In Kap. 3 wird die GIS-gestützte Voranalyse erläutert. Das Untersuchungsgebiet wird vorgestellt. Neben den verwendeten Geodaten und der eingesetzten Hard- und Software liegt das Hauptaugenmerk auf den Auswerteschritten. Die Resultate der Auswertung werden in Kap. 4 dargestellt. In Kap. 5 folgt die Diskussion und im 6. Kapitel werden Schlussfolgerungen gezogen sowie ein Ausblick gegeben.

## 2 Grundlagen

Die folgenden Seiten zeigen die Grundlagen, die für die vorliegende Arbeit von Wichtigkeit sind. Kap. 2.1 gibt einen Einblick in die Schweizer Seenlandschaft. In Kap. 2.2 wird die Notwendigkeit der Uferstrukturkartierung aufgezeigt während Kap. 2.3 einen Einblick ins Gebiet der Ökomorphologie gibt.

### 2.1 Seen

Seen sind stehende Gewässer, in welchen im Sommerhalbjahr über längere Zeit eine stabile Temperaturschichtung besteht (Spreafico & Weingartner 2005). Gemäss Arealstatistik 2004/2009 (BFS 2009a) sind knapp 3.5 % der Schweiz durch stehende Gewässer bedeckt. Selbst wenn nur die grösseren Gewässer mit einer Oberfläche von mindestens 10 ha gezählt werden, gibt es in der Schweiz gut 70 natürliche Seen und rund 100 Speicherseen (Schlosser et al. 2013).

Ein See lässt sich in verschiedene Bereiche einteilen (Spreafico & Weingartner 2005):

- Uferzone (Litoral):  
Die Uferzone ist der Übergangsbereich zwischen Land und Wasser. Es beinhaltet die durchlichteten Bereiche bis zur Tiefengrenze. Im Litoral ist die Artenvielfalt von Pflanzen und Tieren sehr gross – speziell bei flachen Ufern. Zugleich ist der Druck auf diese Zone durch Uferverbauungen und Badenutzung oft erheblich.
- Tiefer Seegrund (Profundal):  
Das Profundal liegt ab der Tiefengrenze unterhalb des Litorals. In diesem Bereich ist die Lichtintensität zu gering, als dass Photosynthese möglich wäre.
- Freiwasserzone (Pelagial):  
Das Pelagial weist bei entsprechend tiefen Seen die charakteristische Temperaturschichtung auf. In der lichtdurchfluteten obersten Schicht lebt eine Vielzahl von Kleinstlebewesen (Plankton); eine wichtige Nahrungsgrundlage für verschiedene Fischarten.

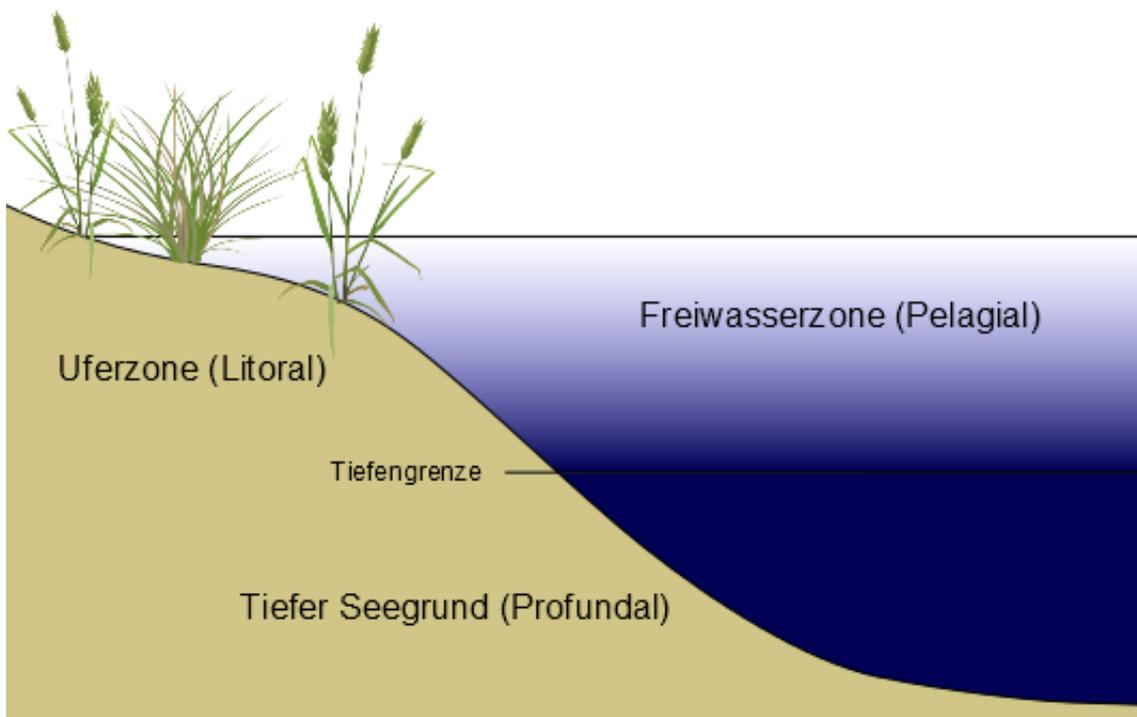


Abb. 2 Skizze der verschiedenen Bereiche im See (nach Spreafico & Weingartner 2005)

## 2.2 Gesetzgebung

An der Volksabstimmung vom 28. November 2004 wurde der Verfassungsartikel 75a „Vermessung“ (BV Art. 75a) angenommen. Dieser Volksentscheid stellt einen Meilenstein für die Geoinformation in der Schweiz dar. Die Schweizerische Bundesverfassung wurde mit dem Artikel ergänzt, der die verfassungsrechtlichen Grundlagen für die Landesvermessung, die amtliche Vermessung sowie die Harmonisierung der Geoinformation enthält. Der neue Artikel bildet den Grundstein für die weitere Gesetzgebung.

Auf den 1. Juli 2008 setzte der Schweizerische Bundesrat ein neues Gesetz in Kraft, das den Aufbau einer nationalen Geodateninfrastruktur ermöglicht. Das Bundesgesetz über Geoinformation (GeolG) hat zum Ziel, die breite Nutzung von Geoinformationen für Behörden, Wirtschaft, Gesellschaft und Wissenschaft zu ermöglichen. Dazu müssen Geodaten rasch, einfach, in der erforderlichen Qualität sowie zu angemessenen Kosten zur Verfügung stehen (GeolG Art. 1). Auf nationaler Ebene werden daher verbindliche Standards für die Erfassung und den Austausch von Geobasisdaten festgelegt. Geobasisdaten sind Geodaten, die auf einem rechtssetzenden Erlass des Bundes, eines Kantons oder einer Gemeinde beruhen (GeolG Art. 3).

Gemeinsam mit dem GeolG wurde die zugehörige Geoinformationsverordnung (GeolV) verabschiedet. Sie enthält fachliche und technische Ausführungen. Unter anderem wird zu jedem Geobasisdatensatz des Bundesrechts ein minimales Geodatenmodell gefordert (GeolV Art. 9). Anhang 1 der Verordnung zeigt den Katalog aller Geobasisdaten. Der Identifikator 191 umfasst die Geodaten zur Planung der Revitalisierung von Gewässern. Zu diesem Identifikator muss ein minimales Geodatenmodell (MGDM) erstellt werden, das die Kantone verpflichtet, Geodaten in dieser Form zur Verfügung zu stellen.

Während vielen Jahren war das Hauptziel des Gewässerschutzes, die Wasserqualität zu verbessern. Dank grossen Anstrengungen konnte die Wasserbelastung reduziert und gute Resultate verbucht werden. Das Schweizerische Gewässerschutzgesetz (GSchG) wurde im Jahr 1991 zum zweiten Mal revidiert. Seither steht der umfassende Schutzgedanke im Mittelpunkt, die Gewässer werden als Ganzes betrachtet. Sie haben viele Funktionen: Sie beeinflussen das Klima, haben eine wichtige Aufgabe als Lebensraum, prägen unser Landschaftsbild und dienen der Erholung.

2005 wurde die Volksinitiative „Lebendiges Wasser“ lanciert. Verbaute Gewässer sollten renaturiert, die Restwasser-Vorschriften durchgesetzt und Massnahmen gegen Schwall / Sunk bei Speicherkraftwerken ergriffen werden. Mit über 160'000 Unterschriften fand die Initiative grossen Zuspruch bei der Schweizer Bevölkerung. Der Bundesrat wollte die Initiative jedoch ablehnen, ohne einen Gegenvorschlag zu formulieren. Aufgrund einer Motion aus dem Parlament wurde trotzdem ein Gegenvorschlag erarbeitet, der vom Parlament gutgeheissen wurde und Gesetzesanpassungen mit sich zog. Die Initiantinnen und Initianten von „Lebendiges Wasser“ erachteten den Gegenvorschlag für ausreichend und zogen ihre Initiative zurück.

Im Jahr 2011 wurde das Schweizerische Gewässerschutzrecht revidiert. Die Renaturierung von Flüssen und Seen ist nun in der Gewässerschutzverordnung (GSchV) verankert. Die Kantone sind verpflichtet, für die Revitalisierung von Gewässern zu sorgen und dabei den Nutzen für Natur und Landschaft sowie die wirtschaftlichen Auswirkungen zu berücksichtigen (GSchG Art. 38a). Dazu muss eine Planung für die Revitalisierungen der nächsten 20 Jahre erstellt werden. Die Planungen zu den betroffenen Abschnitten und deren Revitalisierungsmassnahmen müssen von den Kantonen für Fliessgewässer bis zum 31. Dezember 2014, für stehende

Gewässer bis zum 31. Dezember 2018 verabschiedet werden. Eine wichtige Grundlage der Planungen ist der ökomorphologische Zustand der Gewässer (GSchV Art. 41d). Mit der Verankerung im Gewässerschutzrecht wird die Ökomorphologie zu einem Geobasisdatensatz (GeolG Art. 3) und die Kantone sind verpflichtet, ihre Gewässer nach dem vom BAFU vorgegebenen MGDM zu erfassen.

## 2.3 Ökomorphologie

In der Methode zur Untersuchung und Beurteilung der Fliessgewässer (BUWAL 1998b) wird der Begriff „Ökomorphologie“ sinngemäss folgendermassen definiert:

Die Ökomorphologie umfasst die Gesamtheit der strukturellen Gegebenheiten im und am Gewässer: die eigentliche Gewässermorphologie, wasserbauliche Massnahmen (z. B. Verbauungen des Ufers) sowie die Gegebenheiten im angrenzenden Umland (Bebauungen, Landnutzung, Vegetation).

### 2.3.1 Modul-Stufen-Konzept

Mit dem 1991 revidierten Gewässerschutzgesetz steht der umfassende Schutz der Gewässer im Mittelpunkt. Um den neuen Anforderungen zur Untersuchung und Überwachung der Gewässer gerecht zu werden, musste ein geeignetes Instrument entwickelt werden: Das Modul-Stufen-Konzept.

1998 wurde das Modul-Stufen-Konzept für Fliessgewässer publiziert (BUWAL 1998a). Es handelt sich dabei um ein gemeinsames Projekt des Bundesamts für Umwelt, Wald und Landschaft (BUWAL), der Eidgenössischen Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz (Eawag) und kantonaler Gewässerschutzfachstellen. Eine Gewässerthematik kann in einer bis drei Stufen angewandt werden (vgl. **Tab. 1**), nicht jede Methode lässt sich auf allen drei Stufen umsetzen.

**Tab. 1** Bearbeitungsstufen des Modul-Stufen-Konzepts (nach Eawag 2014)

Stufe	F	S	A
<b>Raum</b>	Flächendeckend (Region / Kanton)	Systembezogen (Gewässersystem)	Abschnittsbezogen (Gewässerabschnitt)
<b>Ziel</b>	Übersicht über den ökologischen Zustand und Defizite der Gewässer	Differenzierte Erfassung und Analyse ökologischer Defizite, Massnahmenkonzept	Problemanalyse für spezifische Fragestellungen
<b>Aufwand (pro Gewässerstrecke)</b>	Gering	Mittel	Gross
<b>Bewertung</b>	Punkteskala	Punkteskala / verbal	Verbal

Aufgrund der Ziele, die mit den Gewässeruntersuchungen verfolgt werden, wurden die wichtigsten Teilmethoden (Module) eruiert. Sie lassen sich in die Bereiche Hydrodynamik und Morphologie, Biologie sowie chemische und toxische Effekte einteilen. Mit Ökomorphologie Stufe F wurde 1998 das erste Modul publiziert. Seither wurden weitere veröffentlicht, die Arbeiten sind jedoch noch nicht abgeschlossen. Umgesetzt oder in Erarbeitung sind (vgl. Eawag 2014):

- Hydrodynamik und Morphologie:
  - Ökomorphologie F / S
  - Hydrologie F

- Äusserer Aspekt F
- Temperaturregime
- Biologie:
  - Makrozoobenthos F
  - Fische F / S
  - Kieselalgen F
  - Makrophyten F
- Chemische und toxische Effekte:
  - Chemie F
  - Ökotoxikologie F
- Synthese

Schlosser et al. (2013) zeigen ein Konzept für die Zustandsbeurteilung der Schweizer Seen auf. Dieser Leitfaden zum Vorgehen bei der Entwicklung von Modulen reiht sich in das Modul-Stufen-Konzept ein. Auch für die Untersuchung der stehenden Gewässer sollen verschiedene eigenständige Module entwickelt werden. Die Beurteilung erfolgt bezüglich des naturnahen Referenzzustands eines Gewässers. Als oberstes Ziel wird der naturnahe Zustand eines Gewässers festgelegt. Dieses Oberziel wird in Unterziele verschiedener Hierarchien unterteilt. Auf der untersten Stufe stehen die einzelnen Attribute. Ob ein Ziel erreicht wird, kann mit stetigen Werten zwischen 0 (= sehr schlechter Zustand) und 1 (= naturnaher Zustand) quantifiziert werden. Eine Abbildung in Zustandsklassen ist jederzeit möglich.

Das Prinzip der messbaren Grössen, die sich zur Überprüfung einzelner Ziele aggregieren lassen, soll sich durch alle zukünftig entwickelten Module zur Beurteilung der Seen durchziehen. Wichtig ist, dass die Attribute wertneutral erhoben werden. Eine Bewertung erfolgt nachgelagert an die Kartierung mit Hilfe von Wertfunktionen, die den einzelnen Attributausprägungen einen Wert zwischen 0 und 1 zuweisen. Dieser Aufbau lässt eine grosse Flexibilität zu. Bei neuen Erkenntnissen können die Wertfunktionen verändert oder die Zielhierarchien anders definiert und aktuellen Bedürfnissen angepasst werden. Eine komplette Neuerhebung der Attribute ist dabei nicht nötig.

Im Rahmen der Konzepterstellung stellten Schlosser et al. (2013) die zu erarbeitenden Module zusammen. Dabei richteten sie sich nach gesetzlichen Vorgaben. Mittels Expertenbefragung wurde die Dringlichkeit der Module festgelegt. „Ökomorphologie Seeufer“ hat neben dem Modul „Nährstoffe“ höchste Priorität.

### 2.3.2 Ökomorphologie Stufe F

Ökomorphologie Stufe F (BUWAL 1998b) ist ein Teil des Modul-Stufen-Konzepts. Ziel ist es, einen Überblick über den Zustand der Fließgewässer zu erhalten und die ökologischen Defizite zu erkennen. Die Methode befasst sich mit wichtigen ökologischen und morphologischen Eigenschaften von kleinen und mittleren Fließgewässern. Für Gewässer ab einer Breite von rund 15 m sollten die beiden Flussufer differenziert betrachtet werden, eine Anwendung von Stufe F ist dann nicht sinnvoll.

Bei einer flächendeckend anwendbaren Methode muss der Erhebungsaufwand pro Gewässerabschnitt verhältnismässig klein sein. Daher beschränkt sich die Zahl der Attribute auf die wichtigsten repräsentativen ökologischen und morphologischen Merkmale (vgl. **Tab. 2**).

**Tab. 2** Wichtigste Merkmale der Ökomorphologie Stufe F (nach BUWAL 1998b)

<b>Sohlenbreite</b>	Mittlere Breite der Gewässersohle innerhalb eines ausgewählten Gewässerabschnittes. Die Gewässersohle entspricht jenem Bereich, welcher in der Regel bei Hochwasser umgelagert wird und somit frei ist von höheren Wasserpflanzen und Landpflanzen. Normalerweise wird für die Bestimmung der Sohlenbreite der Abstand zwischen linkem und rechtem Böschungsfuss (Definition vgl. unten) verwendet.
<b>Wasserspiegelbreitenvariabilität</b>	Änderung der Wasserspiegelbreite innerhalb eines ausgewählten Gewässerabschnittes. Die Wasserspiegelbreite umfasst den bei mittlerem Wasserstand (Mittelwasser) überspülten Bereich des Gewässers.
<b>Verbauung der Sohle</b>	Ausmass der künstlichen (baulichen) Massnahmen zur Stabilisierung der Gewässersohle
<b>Verbauung des Böschungsfusses</b>	Ausmass der künstlichen (baulichen) Massnahmen zur Stabilisierung des Böschungsfusses. Der Böschungsfuss entspricht dem Übergang von Gewässersohle zu Böschung.
<b>Breite und Beschaffenheit des Uferbereiches</b>	Mittlere Breite des Uferbereiches innerhalb eines ausgewählten Gewässerabschnittes. Beschaffenheit=Bewuchs, Material und Struktur des Uferbereiches.  Der Uferbereich wird als der Bereich oberhalb des Böschungsfusses bis zum Gebiet mit „intensiver Landnutzung“ (Siedlungsgebiet, Gebäude, Strassen, Wege, intensiv genutztes Agrarland wie Acker, Weide u.a.) bezeichnet.

Die ökomorphologische Beurteilung von Fliessgewässern beinhaltet eine Bewertung vor Ort. Von Mündung bis Quelle wird das Gewässer von der kartierenden Person verfolgt. Sobald die Ausprägung eines der wichtigen Merkmale (vgl. **Tab. 2**) sich ändert, endet ein Gewässerabschnitt und alle Attribute werden für diesen Abschnitt erfasst. Neben den Hauptattributen, die zur Bestimmung der Natürlichkeitsklasse dienen, werden zusätzliche Informationen erfasst, die nicht in die Bewertung einfließen: Eindolung, Tiefenvariabilität, natürliche Abstürze, Makrophytenvorkommen, Algenbewuchs und Totholz. Zudem werden Durchgangshindernisse (Abstürze und Bauwerke) mit Typ und Höhe als Punktobjekte erfasst.

Für die einzelnen Bewertungsmerkmale werden Strafpunkte verteilt und pro Gewässerabschnitt aufsummiert. Die Punktesumme lässt eine Zuordnung in eine von fünf Zustandsklassen zu (vgl. **Tab. 3**).

**Tab. 3** Zustandsklassen Ökomorphologie Stufe F (nach BUWAL 1998b)

Zustandsklasse	Punktesumme	Darstellung	Bewertung ökomorphologischer Zustand
I natürlich / naturnah	0 – 1	<b>blaue Linie</b>	gut / genügend
II wenig beeinträchtigt	2 – 5	<b>grüne Linie</b>	
III stark beeinträchtigt	6 – 9	<b>gelbe Linie</b>	schlecht / ungenügend
IV naturfremd / künstlich	10 – 12	<b>rote Linie</b>	
eingedolt		<b>rot punktierte Linie</b>	

Durch die Abschnittsbildung entsteht oft eine stärkere Aggregation, die einzelnen Merkmale werden nicht nach der tatsächlichen Gegebenheit erfasst. Der Geodatensatz wird dadurch einfach interpretierbar, die Flexibilität für gezielte Auswertungen oder bei geänderten Anforderungen ist jedoch nicht gegeben.

Ökomorphologie Stufe F wurde auf einen grossen Teil der Schweizer Fliessgewässer angewandt. Die kantonalen Erhebungen erfolgten auf kantonalen Gewässernetzen in Massstäben von 1:2500 bis 1:25'000. Aufgrund der unterschiedlichen Detaillierungsgrade ist eine direkte Vergleichbarkeit nicht gegeben. Um eine nationale Auswertung zu ermöglichen, transferierte das Bundesamt für Umwelt im Jahr 2008 die verfügbaren Ökomorphologiedaten auf das digitale VECTOR25-Gewässernetz der swisstopo. Dabei gingen aus zwei Gründen gewisse Daten verloren (Zeh Weissmann et al. 2009):

- Grossmassstäbliche kantonale Gewässernetze enthalten (Kleinst-)Bäche, die im kleinmassstäblicheren VECTOR25-Gewässernetz nicht vorhanden sind.
- Weicht der Verlauf eines Bachabschnitts im nationalen Gewässernetz stark vom Verlauf des gleichen Bachabschnitts im kantonalen Gewässernetz ab, ist eine eindeutige Zuweisung nicht gewährleistet.

Bis im Jahr 2008 waren rund 29'000 km kartiert, wovon 23'887 km auf das VECTOR25-Gewässernetz transferiert werden konnten. Für rund 37 % der knapp 65'000-Fliesskilometer lag eine Klassierung vor. Für eine Hochrechnung des Gewässerzustands auf das Gebiet der ganzen Schweiz stratifizierten Zeh Weissmann et al. (2009) die Gewässerabschnitte nach vier Kriterien:

- biogeografische Regionen (Jura / Mittelland / Alpen)
- Höhenklassen (unter 600 m ü.M. / 600–1200 m ü.M. / 1200–2000 m ü.M. / über 2000 m ü.M.)
- Flussordnungszahlen nach Strahler (FLOZ 1 (Kleingewässer) / FLOZ 2 und 3 (mittlere Gewässer) / FLOZ 4 bis 9 (grosse Gewässer)
- Siedlung (in Siedlung / ausserhalb Siedlung)

Die Hochrechnung auf das ganze Gewässernetz zeigte, dass 78 % der Fliessgewässer einen guten ökomorphologischen Zustand aufweisen (natürlich / naturnah oder wenig beeinträchtigt). Die restlichen Gewässer (22 %) haben ein grosses Potenzial für Renaturierungen. Diese Zustandsbeobachtung floss als Indikator in die Landschaftsbeobachtung Schweiz (LABES) ein (Roth et al. 2010).

Seit der Hochrechnung der Daten von 2008 fand aufgrund des revidierten Gewässerschutzrechts eine grosse Entwicklung statt. Viele zusätzliche Gewässerstrecken wurden kartiert, einige renaturierte Abschnitte wurden nachgeführt. Eine erneute schweizweite Auswertung liesse Rückschlüsse auf die Veränderung zu.

### 2.3.3 Ökomorphologie Seeufer

Im Rahmen der Konzepterstellung für die Zustandsbeurteilung der Schweizer Seen stellten Schlosser et al. (2013) die zu erarbeitenden Module zusammen und legten die Dringlichkeit ihrer Entwicklung fest. „Ökomorphologie Seeufer“ hat höchste Priorität, da aufgrund eines gesetzlichen Auftrags bis Ende 2018 eine strategische Revitalisierungsplanung vorliegen muss. Aus diesem Grund wurde eine Vorstudie zur Untersuchung bestehender Seeuferbewertungen lanciert. Teiber et al. (2013) prüften bisherige Kartierungen auf ihre Methode und Übertragbarkeit auf die Schweizer Seen.

Für MESUBA (Methode der Seeuferbeurteilung und –aufwertung) wurden folgende Anforderungen definiert:

- Sie soll sich als eigenständige Methode in das Konzept für die Untersuchung und Beurteilung von Seen (Schlosser et al. 2013) einfügen.

- Sie soll den gesetzlichen Anforderungen genügen (Revitalisierungsplanung nach GSchV Art. 41d).
- Sie soll flächendeckend anwendbar sein. Auch mittlere und kleinere Gewässer (Richtwert stehende Gewässer mit einer Fläche von  $\geq 0.5$  ha) sollen sofern gewünscht mit der Methode kartiert werden können.
- Bestehende Kartierungen von Schweizer Seen sollen möglichst einfach in die neue Methode überführt werden können.

Auf diese Richtlinien sollten bisherige Kartierungen geprüft werden. Weltweit wurden schon einige Seen ökomorphologisch erfasst. Dabei wurden verschiedenste Methoden eingesetzt und unzählige unterschiedliche Merkmale erhoben. Das Vorgehen der bisherigen Seeuferkartierungen lässt sich grob in zwei Typen einteilen: Felderhebung oder Auswertung von Luftbildern.

Teiber et al. (2013) betrachteten 19 von rund 30 identifizierten Ansätzen als relevant für die Methodenentwicklung. Alle erfassten Attribute dieser 19 Ansätze wurden übergeordneten Kriterienklassen zugewiesen. So konnten redundante Merkmale und identische Attribute, die unterschiedlich bezeichnet wurden, identifiziert werden. In weiteren iterativen Schritten wurde der Kriterienkatalog auf die wesentlichen Merkmale reduziert. Schliesslich lieferte die Vorstudie einen ersten Vorschlag für eine minimale Attributliste, die als Vorschlag für eine Schweizer Methode „Ökomorphologie Seeufer“ dient.

In einer Begleitgruppe, die sich aus Fachpersonen aus BAFU, verschiedenen Kantonen und Privatwirtschaft zusammensetzt, wird die endgültige Methode zur Seeuferkartierung erarbeitet. Erste Diskussionen zeigten, dass die von Teiber et al. (2013) vorgeschlagene minimale Attributliste eine sehr gute Ausgangslage bildet. Trotzdem wurden seit der Vorstudie schon mehrere Anpassungen an der Auswahl der Merkmale sowie an deren Ausprägungen vorgenommen.

Der derzeit aktuelle Stand der Methodenentwicklung (BAFU 2014a) umfasst 18 Merkmale, die den Zustand der Seen beschreiben sollen. Neben zwei allgemein gültigen Attributen sind die restlichen 16 einem bestimmten Beurteilungstreifen zugewiesen (Ufer, Flachwasser, Hinterland). Da der Zeit- und Materialaufwand bei Feldkartierungen enorm ist, sind keine Vorortaufnahmen vorgesehen. Stattdessen soll die Erfassung der Merkmale am Computer mit Hilfe von Ortho- und Schrägbildern erfolgen.

Neben der Festlegung des Attributkatalogs und der Erfassungsmethode ist die Klärung der Abschnittsbildung ein weiterer wichtiger Punkt. Bei der Erfassung der Ökomorphologie von Fliessgewässern wird eine Abschnittsgrenze im Feld vom Kartierer festgelegt, sobald sich ein wichtiger Parameter ändert. Anschliessend werden für den definierten Abschnitt alle Merkmale erfasst, was die Resultate weniger scharf abgrenzt. In vielen bisherigen Seeufer-Kartierungen wurden feste Abschnittslängen verwendet. So wurde das Ufer des Wohlensees mit einer definierten Abschnittslänge von 50 m erfasst (SigmaPlan 2011). Im Rahmen der Erarbeitung von MESUBA wurde die Umsetzung von Abschnitten diskutiert. Dabei wurde festgelegt, dass jedes Attribut unabhängig von den anderen erfasst wird. Ein grosser Vorteil besteht darin, dass die Originaldaten weiterhin verfügbar sind und zu einem späteren Zeitpunkt auf verschiedene Arten aggregiert werden können. Dies ermöglicht einen flexibleren Umgang bei allfälligen Anpassungen der Bewertung und Definitionen verschiedener Zielhierarchien. Ausserdem können einzelne Merkmale gezielt ausgewertet werden, ohne durch die Abschnittsbildung verwässert zu sein.

Für die vorliegende Arbeit ausschlaggebend sind die in **Tab. 4** aufgelisteten 18 Merkmale, die dem Entwicklungsstand vom April 2014 entsprechen (BAFU 2014a).

**Tab. 4** Minimale Attributliste zur Seeuferbeurteilung (nach BAFU 2014a). Der Wirkungstyp beschreibt die Verwendbarkeit des Attributs für die Beurteilung: Zusatzinformation (z), Hauptattribut für Bewertung (b<sub>1</sub>), Bonus-Malus-Attribut (b<sub>2</sub>) oder übergeordneter Einfluss (e). Eine detaillierte Auflistung mit den Attributausprägungen ist in den Kapiteln 3.5 bis 3.8 sowie im Anhang 1 zu finden.

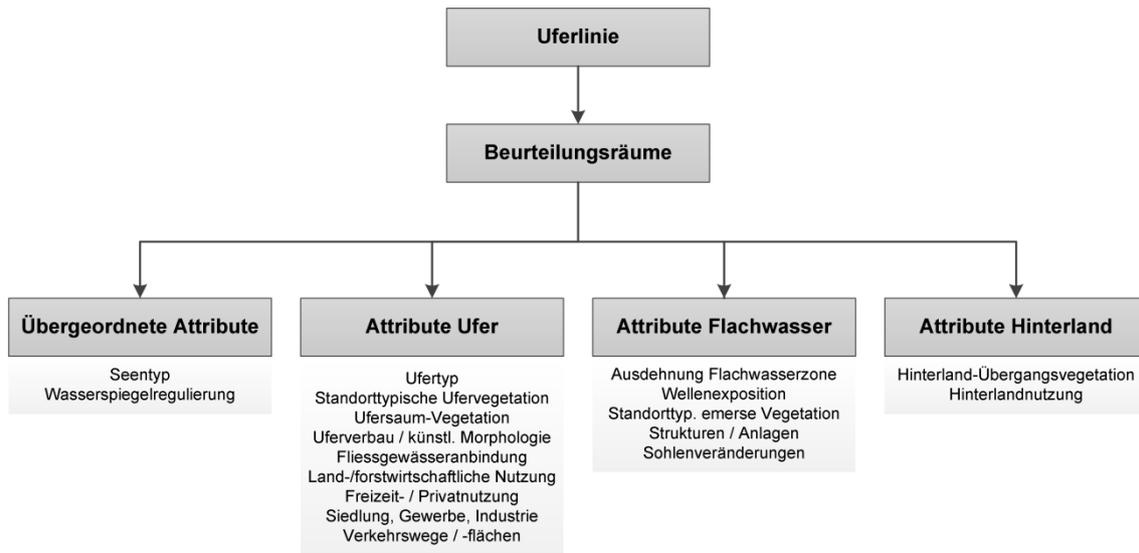
Nr.	Attribut	Wirkungstyp	Beurteilungstreifen
1	Typ stehende Gewässer	e	-
2	Wasserspiegelregulierung	e	
3	Ufertyp	e	Ufer
4	Ausdehnung der Flachwasserzone	e	Flachwasser
5	Wellenexposition	e	
6	Standorttypische emerse Vegetation der Flachwasserzone	z	Flachwasser
7	Standorttypische Ufervegetation	z	Ufer
8	Ufersaum-Vegetation	b <sub>1</sub>	Ufer
9	Uferverbau und künstliche Morphologie	b <sub>1</sub>	
10	Fliessgewässeranbindung (Mündung / Abfluss)	b <sub>1</sub>	
11	Land- und forstwirtschaftliche Nutzung im Uferstreifen	b <sub>1</sub>	
12	Freizeit- und Privatnutzungen im Uferstreifen	b <sub>1</sub>	
13	Siedlung, Gewerbe, Industrie im Uferstreifen	b <sub>1</sub>	
14	Verkehrswege und -flächen im Uferstreifen	b <sub>1</sub>	
15	Strukturen / Anlagen in der Flachwasserzone	b <sub>1</sub>	Flachwasser
16	Sohlenveränderungen in der Flachwasserzone	b <sub>2</sub>	Flachwasser
17	Hinterland-Übergangsvegetation	b <sub>2</sub>	Hinterland
18	Hinterlandnutzung	b <sub>2</sub>	

Mit den erfassten Merkmalen zum ökomorphologischen Zustand soll ein Gesamtüberblick über den Zustand der Uferbereiche gegeben werden. Die qualitative und quantitative Bewertung des Uferzustands lässt vorhandene Defizite erkennen und damit den Handlungsbedarf abschätzen. Auf dieser Basis soll die strategische Revitalisierungsplanung erarbeitet und Massnahmen definiert werden (GSchV Art. 41d).

Bei periodischen Erhebungen ist ein Monitoring, also ein Vergleich der Seeuferzustände zu verschiedenen Zeitpunkten, möglich. Die differenzierte Erfassung der Attribute lässt bei Bedarf auch Überprüfungen und Neuerhebungen einzelner Merkmale zu, es muss nicht zwingend eine Vollkartierung durchgeführt werden. Dies ist ein grosser Vorteil gegenüber der für Fliessgewässer verwendeten Methode „Ökomorphologie Stufe F“.

### 3 Methode

Anhand eines Testgebiets wird ein Ablauf für die GIS-gestützte Voranalyse entwickelt. Dazu dient das Seeland. Wie es der Name sagt, umfasst diese Schweizer Region verschiedenen Seen: Bieler-, Murten- und Neuenburgersee. In Kap. 3.1 werden sie vorgestellt. Die für die Analyse verwendeten Geodaten und die eingesetzte Hard- und Software sind in Kap. 3.2 beschrieben. Die einzelnen Arbeitsschritte der Voranalyse sind in **Abb. 3** zu sehen und werden in Kap. 3.3 bis 3.8 erläutert.



**Abb. 3** Schritte der GIS-gestützten Analyse

Bis auf die verwendeten Orthofotos, die in LV95 (Landesvermessung 1995) ausgeliefert werden, liegen alle Geodaten im herkömmlichen Bezugssystem LV03 (Landesvermessung 1903) vor. Aus diesem Grund werden die Analysen im zurzeit noch weiter verbreiteten System LV03 durchgeführt.

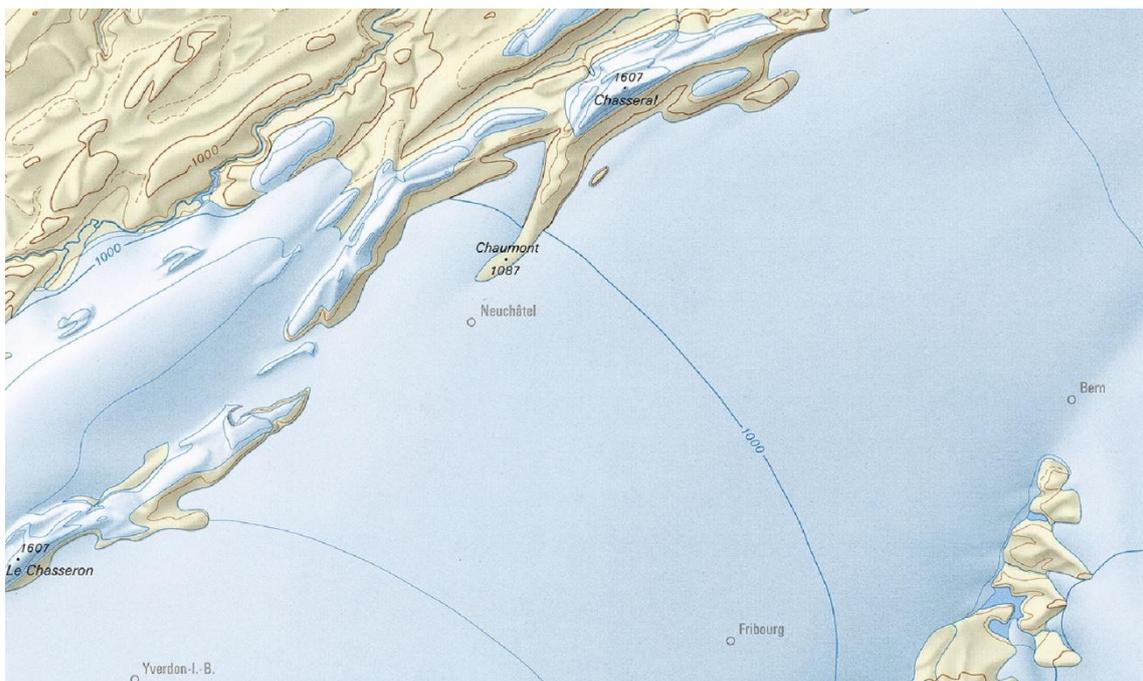
### 3.1 Untersuchungsgebiet

Als Untersuchungsgebiet dient das Drei-Seen-Land mit Bieler-, Murten- und Neuenburgersee. In **Abb. 4** ist die geografische Nähe der drei Seen sichtbar.



**Abb. 4** Die drei Jurarandseen: Bielersee (frz. Lac de Bielle), Murtensee (frz. Lac de Morat) und Neuenburgersee (frz. Lac de Neuchâtel)  
PK1000 im Massstab 1:500'000 (Quelle: Bundesamt für Landestopografie)

Bini et al. (2009) zeigen die Vergletscherung während der letzten Eiszeit auf. Der Rhonegletscher erstreckte sich bis Solothurn und bedeckte das gesamte Untersuchungsgebiet mit einer mehreren hundert Meter dicken Eisschicht (vgl. **Abb. 5**).



**Abb. 5** Die Vergletscherung der Schweiz zur letzten Eiszeit im Massstab 1:500'000 (Quelle: Bini et al. 2009)

Die Region Drei-Seen-Land gehört zum Schweizer Mittelland. Der Untergrund besteht mehrheitlich aus Molasse. Während der letzten Eiszeit flossen die Eismassen grundsätzlich nach Nordwesten. In der Westschweiz wurden sie am Jurarand nach Nordosten abgelenkt (Labhart 2009). Nach dem Rückzug des Rhonegletschers blieb eine tiefgelegene Senke am Jurasüdfuss zurück. Die Endmoräne in Solothurn staute mit der Zeit einen durch die Aare gespiesenen, riesigen See auf, den so genannten „Solothurnersee“. Das ganze Drei-Seen-Land stand bis auf die höchsten Molassehügel unter Wasser. Mit einem langwierigen Erosionsprozess wurde die Endmoräne schliesslich bezwungen und das gestaute Wasser befreit. Zurück blieben kleinere Seen (Nast 2006).

Im sehr flachen Seeland kam es aufgrund des geringen Gefälls der Aare immer wieder zu Überschwemmungen. Durch Geschiebeablagerungen war die Entwässerung unberechenbar und es entstanden grosse versumpfte Gebiete, die nicht landwirtschaftlich genutzt werden konnten. Nach vereinzelt Schutzmassnahmen erfolgte zwischen 1868 und 1891 die Erste Juragewässerkorrektur. Der Seespiegel der drei Seen wurde um 2.5 m gesenkt. Damit es nicht zu Überschwemmungen kommt musste der Abfluss des Bielersees vergrössert werden, indem das alte Zihlbett im Nordosten des Sees zum Nidau-Büren-Kanal erweitert und kanalisiert wurde. Zusätzlich wurde mit dem Hagneckkanal ein komplett neuer, künstlicher Flusslauf erstellt und ein Grossteil des Aarewassers in den Bielersee umgeleitet. Die Verbindungen zwischen Murten- und Neuenburgersee (Broye) und zwischen Neuenburger- und Bielersee (Zihl) wurden ausgebaut und begradigt. Mit der Seespiegelabsenkung ging ein Landgewinn von rund 30 km<sup>2</sup> einher. Ein Netz mit Drainagen und kleineren Kanälen entsumpfte das Gebiet (Nast 2006).

Trotz enormen Eingriffen kam es auch nach der Ersten Juragewässerkorrektur noch zu katastrophalen Überschwemmungen. Einerseits hatten sich die Torfböden gesetzt, andererseits stimmte bei Hochwasser das Verhältnis zwischen Zu- und Abfluss nicht. Eine weitere Korrektur wurde geplant. Die Bauarbeiten zur Zweiten Juragewässerkorrektur fanden zwischen 1962 und 1973 statt. Die Seespiegel wurden um einen weiteren Meter abgesenkt. Dadurch wurde die Petersinsel im Bielersee zu einer Halbinsel. Um den Ausgleich zwischen den drei Seen zu verbessern, wurden Broye- und Zihlkanal auf den zwei- bis dreifachen Querschnitt ausgebaut und der Nidau-Büren-Kanal stark abgesenkt. Der Bau eines Regulierwehrs beim Ausfluss des Bielersees schafft einer hydraulische Einheit bestehend aus Bieler-, Neuenburger- und Murtensee. Nach der Zweiten Korrektur wurde das Seeland zu einem der wichtigsten Gemüseanbaugebiete der Schweiz (Nast 2006).

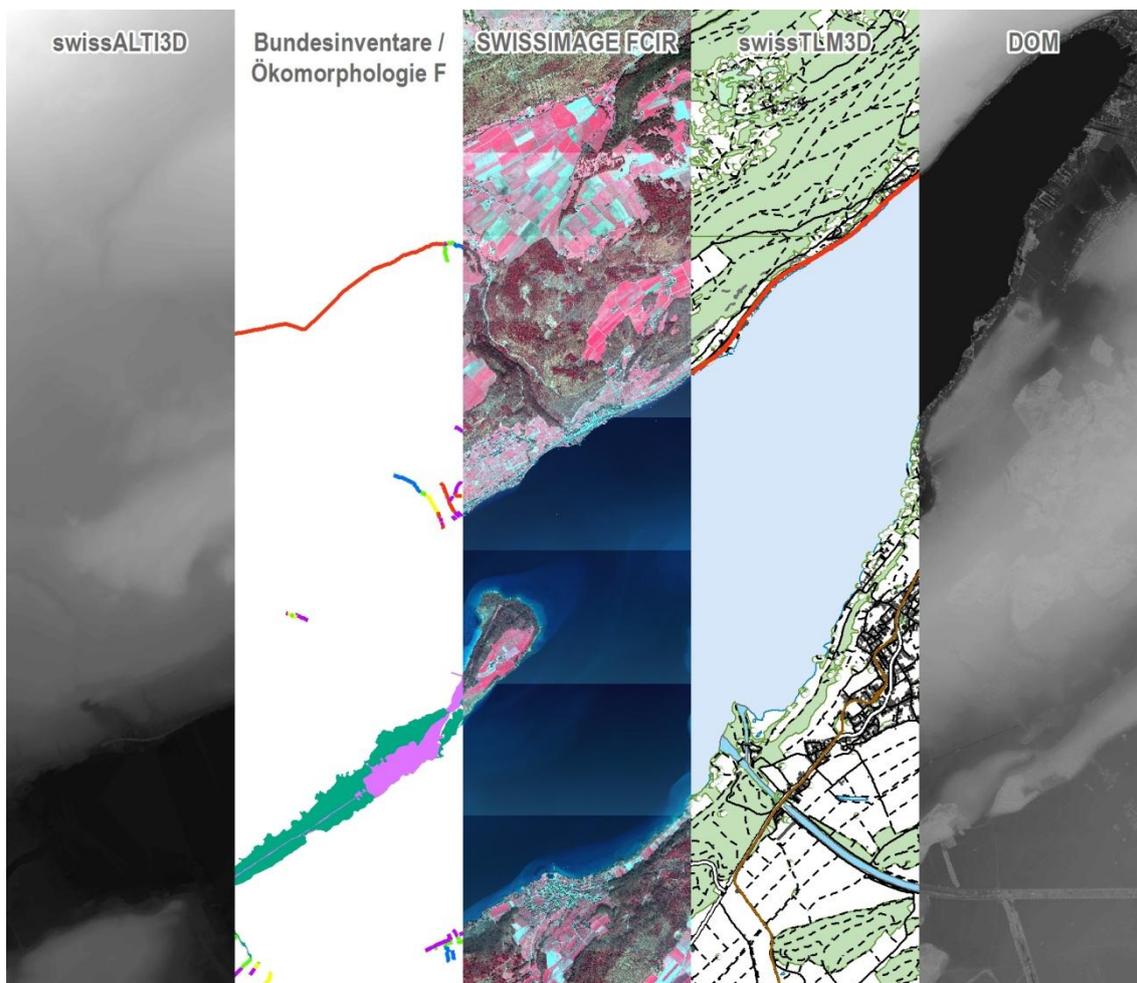
Die drei Seen unterscheiden sich bezüglich Grösse, Tiefe und Uferstruktur. Beim Neuenburgersee handelt es sich um den grössten, komplett in der Schweiz liegenden See. Die wichtigsten Angaben zu den drei Seen sind in **Tab. 5** aufgelistet. Die identische Höhe über Meer ist auf die beiden Juragewässerkorrekturen zurückzuführen.

**Tab. 5** Kennzahlen der drei Seen, die als Beispiel für die GIS-gestützte Analyse verwendet werden (nach BFS 2009b).

Kennzahl	Bielersee	Murtensee	Neuenburgersee
Fläche [km <sup>2</sup> ]	39.4	22.7	215.0
Höhe über Meer [m]	429	429	429
Gesamter Inhalt [m <sup>3</sup> ]	1240	600	14'170
Maximale Tiefe [m]	74	46	153

### 3.2 Datengrundlage, Hard- und Software

Gemäss Zielsetzung dieser Arbeit soll das Verfahren auf die gesamte Schweiz anwendbar sein. Es werden nur Daten verwendet, die flächendeckend in homogener Qualität und Form verfügbar sind. Insofern beschränkt sich die Auswahl der Grundlagedaten auf Produkte der Bundesämter swisstopo und BAFU, sowie auf die kantonalen Ökomorphologiedaten Stufe F. Die einzelnen Produkte werden in Kap. 3.2.1 bis 3.2.6 kurz beschrieben, **Abb. 6** visualisiert die Daten am Beispiel des Bielersees.



**Abb. 6** Verwendete Grundlagedaten der Region Bielersee im Massstab 1:85'000  
(Quelle Geodaten: Bundesamt für Landestopografie)

#### 3.2.1 Topografisches Landschaftsmodell swissTLM3D

Bis im Jahr 2008 war VECTOR25 das detaillierteste Landschaftsmodell der Schweiz. Es handelte sich dabei um ein Modell von swisstopo, das inhaltlich und geometrisch auf der Landeskarte 1:25'000 basierte und im Nachführungsrhythmus von sechs Jahren aktualisiert wurde. Ab Frühling 2008 wurde das Topografische Landschaftsmodell TLM aufgebaut. Für die Version 1.0 wurden die generalisierten Geometrien von VECTOR25 übernommen. Die dritte Dimension wurde einbezogen, so dass aus dem TLM das Produkt swissTLM3D<sup>1</sup> wurde. Bei der Überarbeitung werden nun die generalisierten und kartografisch verdrängten Geometrien ver-

<sup>1</sup> Vgl. swisstopo (2014a)

bessert, so dass nach einer ersten gesamthaften Bereinigung ein Modell in hoher Genauigkeit vorliegen wird. Diese Überarbeitung ist noch nicht abgeschlossen. Das Modell liegt zurzeit in der Version 1.2 vor, in der vorliegenden Arbeit wird die Version 1.1 verwendet.

swissTLM3D beschreibt rund 6 Millionen Objekte mit Lage und Form, dem dazugehörigen Objekttyp und weiteren Attributen. Die flächendeckende Verfügbarkeit, der umfangreichen thematische Inhalt (vgl. **Tab. 6**), der Einbezug der dritten Dimension und die hohe geometrische Genauigkeit sind die grössten Vorteile des swissTLM3D.

Die Lage- und Höhengenaugkeit beträgt

- bei VECTOR25: 3 – 8 m
- bei überarbeitetem swissTLM3D:
  - 0.2 – 1.5 m für gut definierte Objekte (z. B. Strassen und Gebäude)
  - 1 – 3 m für nicht klar abgrenzbare Objekte (z. B. Wald)

**Tab. 6** Topics in swissTLM3D (nach swisstopo 2014a)

Topic	Beschreibung
Strassen und Wege	Strassen- und Wegnetz, Fuss- und Fahrradwege
Öffentlicher Verkehr	Eisenbahnlinien und andere Bahnen, Schifffahrtswege und Haltestellen des öffentlichen Verkehrs
Bauten	Gebäude, Mauern, Dämme, Konstruktionen im Zusammenhang mit Transport, Kommunikation, Energie, Sport und Freizeit
Areale	Bereiche mit einer speziellen Landnutzung (Transport, Lebensraum, Industrie etc.)
Bodenbedeckung	Bodenbedeckung unabhängig der Landnutzung
Gewässernetz	Wasserläufe und Seeumrisse
Einzelobjekte	Selektion markanter Objekte, welche zur Orientierung beitragen (Kreuze, Quellen, Höhlen)
Namen	Unterschiedliche Namen (Berge, Gebiete, Ortschaften)

Für die gezielte Auswahl der benötigten Topics und Feature Classes bietet swisstopo (2012) die nötige Hilfe.

### 3.2.2 Höhenmodell swissALTI3D

Im Jahr 2000 nahm swisstopo in Zusammenarbeit mit privaten Unternehmen mittels Airborne-Laser-Scanning ein digitales Höhenmodell auf. Bis auf eine Höhenlage von 2000 Metern wurde mit einer hohen Punktdichte von rund 30 cm das Gelände präzise erfasst. Gebiete über 2000 Meter wurden ergänzt mit dem Digitalen Höhenmodell DHM25, das auf der Landeskarte 1:25'000 basiert. Daraus resultierte das Geländemodell DTM-AV. Mit der Erstellung des Topografischen Landschaftsmodells wurde auf ein neues Produkt swissALTI3D<sup>2</sup> umgestellt. Es handelt sich dabei um ein präzises digitales Höhenmodell, das die Oberfläche der Schweiz ohne Bewuchs und Bebauung beschreibt. Es ist als 2m-Raster erhältlich und wird in einem Nachführungszyklus von 6 Jahren aktualisiert. Zurzeit liegt das Modell in der Version 1.1 vor.

<sup>2</sup> Vgl. swisstopo (2014b)

Die Lage- und Höhengenaugigkeit beträgt

- Laserpunkte unterhalb 2000 m ü. M.  $\pm 0.5$  m
- Stereokorrelation oberhalb 2000 m ü. M. 1-3 m
- manuelle Nachführung (Einzelpunkte, Bruchkanten und Flächen) 25-100 cm

### 3.2.3 Digitales Oberflächenmodell DOM

Im Jahr 2000 nahm swisstopo in Zusammenarbeit mit privaten Unternehmen mittels Airborne-Laser-Scanning ein digitales Oberflächenmodell auf. Bis auf eine Höhenlage von 2000 Metern wurde mit einer hohen Punktdichte von rund 30 cm die Oberfläche präzise erfasst. Das entstandene Produkt DOM<sup>3</sup> zeigt die beständig sichtbare Oberflächen und beinhaltet damit Wälder, Gebäude und weitere Kunstbauten. Das Produkt kann als 2m-Grid bezogen werden, wobei Gebiete über 2000 Metern mit dem Digitalen Höhenmodell DHM25, das auf der Landeskarte 1:25'000 basiert, ergänzt wurden. Das DOM wird nicht nachgeführt und der Datensatz nicht angepasst.

Die Höhengenaugigkeit beträgt

- im offenen Gebiet  $\pm 0.5$  m
- in Vegetationsgebieten  $\pm 1.5$  m
- über 2000 m ü. M. 3-8 m

### 3.2.4 Orthofoto SWISSIMAGE FCIR

Seit 2005 werden die Luftbildaufnahmen, die unter anderem Grundlage für die Nachführung der Landeskarten sind, mit einer Digitalkamera des Typs Leica-ADS (airborne digital sensor) aufgenommen. Der ADS40- bzw. ADS80-Sensor erfasst fünf spektrale Kanäle: panchromatisch (PAN), nahes Infrarot (NIR), rot, grün und blau. Die Schweiz wird sektorweise im Dreijahresrhythmus befliegen. Während die Bilddaten im Flachland mit einer Bodenauflösung von 25 cm erfasst werden sind es im Berggebiet 50 cm. Das bekannteste Produkt dieser Aufnahmen ist SWISSIMAGE, ein radiometrisch korrigiertes Echtfarben-Orthofotomosaik. Erst seit der Befliegung von 2010 wird mit der Kombination der Kanäle NIR, rot und grün zusätzlich SWISSIMAGE FCIR<sup>4</sup>, ein Falschfarbeninfrarot-Orthofotomosaik, zur Verfügung gestellt. Die FCIR-Luftbilder werden in Kacheln von 4.375 \* 3 km verwaltet. Jeweils 16 Kacheln decken ein Landeskartenblatt 1:25'000 ab. Im Gegensatz zu SWISSIMAGE werden herstellerseitig keine radiometrischen und manuellen geometrischen Anpassungen durchgeführt.

Die Standardabweichung für die Lagegenauigkeit beträgt

- in ebenem Gelände  $\pm 0.25$  m
- in unebenem Gelände  $\pm 3-5$  m

Das Untersuchungsgebiet wurde in den Jahren 2010 und 2011 befliegen.

### 3.2.5 Bundesinventare

Das BAFU stellt verschiedenste Geodatensätze zur Verfügung<sup>5</sup>. Gemäss Art. 18 des Natur- und Heimatschutzgesetzes (NHG) scheidet der Bundesrat Biotop von nationaler Bedeutung aus

---

<sup>3</sup> Vgl. swisstopo (2014c)

<sup>4</sup> Vgl. swisstopo (2014d)

<sup>5</sup> Vgl. BAFU (2014b)

und legt die Schutzziele fest. Die Aufnahme eines Objektes in ein Bundesinventar bedeutet, dass es ungeschmälert erhalten werden soll. Für die Beurteilung der uferseitigen Gebiete interessant sind die folgenden zwei Themen:

#### **Bundesinventar der Auengebiete von nationaler Bedeutung (AU)**

Auen sind wichtige natürliche Lebensräume im Überschwemmungsbereich von Gewässern. Für das Bundesinventar wurden an natürlichen oder naturnahen Gewässern Auengebiete von mindestens 2 ha aufgenommen, an korrigierten Gewässern mussten sie auf einer Fläche von mindestens 5 ha typische Auenvegetation aufweisen. Die Objekte wurden auf Basis der Landeskarte 1:25'000 kartiert. Der aktuelle Stand des Inventars stammt aus dem Jahr 2007.

#### **Bundesinventar der Flachmoore von nationaler Bedeutung (FM)**

Flachmoore sind Überreste der ursprünglichen Natur- und Kulturlandschaft und bieten einer Vielzahl gefährdeter Tier- und Pflanzenarten einen natürlichen Lebensraum. Biotopflächen mit einer Mindestfläche von 1 ha wurden ins Inventar aufgenommen. Dabei wurden Flachmoorflächen, die mehr als 100 m auseinander liegen, als separate Objekte betrachtet. Die Kartierung basiert auf der Landeskarte 1:25'000. Der aktuelle Stand des Inventars stammt aus dem Jahr 2007.

### **3.2.6 Ökomorphologie Stufe F**

Im Jahr 1998 wurde eine Methode zur Erfassung des Natürlichkeitsgrads von Fliessgewässern veröffentlicht (BUWAL 1998b). Die bis 2008 kantonal erfassten Daten wurden zusammengezogen und analysiert. Rund 37 % der Schweizer Fliessgewässer waren zu diesem Zeitpunkt ökomorphologisch kartiert (Zeh Weissmann et al. 2009). Im Hinblick auf die strategischen Revitalisierungsplanungen der Fliessgewässer, die Ende 2014 verabschiedet werden, wurden in der Zwischenzeit viele weitere Gewässer ökomorphologisch erhoben.

Obwohl es sich bei den Ökomorphologiedaten Stufe F nicht um einen Bundesdatensatz handelt, eignet er sich für die Auswertungen, da die Daten nach einheitlicher Methode mit vorgegebenem Datenmodell erfasst werden. Im Rahmen der strategischen Revitalisierungsplanung von Fliessgewässern müssen die Geodaten bis Ende 2014 nach dem MGDM an den Bund geliefert werden.

### **3.2.7 Eingesetzte Hard- und Software**

Für die Entwicklung, Operationalisierung und Auswertung des vorgestellten Lösungsansatzes kommen folgende Software(-pakete) zum Einsatz:

- ArcGIS 10.0 (mit Python 2.6)
- Access 2007
- ET Geowizards 10.0

Die Auswertung soll so konzipiert werden, dass sie mit einem stärkeren Desktopcomputer ausgeführt werden kann. Im Rahmen der Master Thesis wird ein HP Notebook mit Intel® Core™ i7 CPU 2.90 GHz und 8 GB RAM eingesetzt.

### 3.3 Uferlinie

Eine zentrale Grundlage zur Beurteilung des Zustands eines Seeufers ist die Uferlinie selbst, die definiert werden muss. Teiber et al. (2013) legen die Mittelwasserlinie als Uferlinie fest. Für Analysen und auch Beurteilungen am Bildschirm ist eine räumliche Festlegung nötig, um auf dieser Basis die Beurteilungsräume definieren zu können.

Zudem werden zur Abspeicherung der Daten die einzelnen Attribute zur Beschreibung eines Seeufers auf die Uferlinie projiziert. Damit die Vergleichbarkeit zwischen den Schweizer Seen gewährleistet ist, muss ein einheitlicher Datensatz verwendet werden. Es bieten sich die Uferlinien aus swissTLM3D an. Eine genauere Prüfung der Uferlinien der drei Seeland-Seen zeigt folgende Schwierigkeiten auf:

- Molen oder Buhnen werden meist nicht als Bauwerke, die in den See ragen, erfasst. Üblicherweise folgt die Uferlinie den Bauwerken, was eine massive Verlängerung bewirkt (vgl. **Abb. 7**).
- Während die Uferlinien im älteren Landschaftsmodell VECTOR25 noch ähnlich detailliert waren, variiert der Detaillierungsgrad der swissTLM3D-Linien stark (vgl. **Tab. 7**).

**Tab. 7** Der Neuenburgersee ist in swissTLM3D mit gut doppelt so vielen Stützpunkten pro km erfasst als der Bieler- und Murtensee. Bei VECTOR25 waren es deutlich weniger.

Kennzahl	Bielersee	Murtensee	Neuenburgersee
Uferlinie [km]	55.9	25.0	173.2
Stützpunkte	4593	1776	30'388
Stützpunkte / km (swissTLM3D)	82	71	175
Stützpunkte / km (VECTOR25)	87	90	56

Im Rahmen der vorliegenden Arbeit wird die Uferlinie von Bieler-, Murten- und Neuenburgersee aus dem swissTLM3D-Topic Gewässernetz extrahiert. Es werden keine Inseln einbezogen. Eine Ausnahme bildet die für den Bielersee markante Petersinsel, die mit der Zweiten Jura-gewässerkorrektur zur Halbinsel wurde.

In **Abb. 7** ist oben ein Auszug aus dem swissTLM3D zu sehen. Auch sehr kleine Formen sind in diesem Datensatz abgebildet. Unten links ist die Uferlinie gemäss swissTLM3D rot hervorgehoben. Meist folgt sie Wellenbrecher oder Uferverbauungen, obwohl es sich dabei um Bauten in der Flachwasserzone handelt. Damit die Beurteilungsräume nicht zu stark verfälscht werden, wird die extrahierte Uferlinie bei diesen Spezialfällen manuell angepasst. Die geometrische Änderung ist in **Abb. 7** unten rechts zu sehen. Die generalisierte Uferlinie floss in die weiteren Analyseschritte ein.

Die Vereinfachung der Uferlinie hat einen grossen Einfluss auf deren Länge. Beim Bielersee verringert sich die Länge von 55.9 auf 52.1 km, beim Murtensee von 25.0 auf 23.4 km und beim Neuenburgersee gar von 173.2 auf 130.3 km.



**Abb. 7** Die aus dem swissTLM3D (oben) automatisch extrahierte Uferlinie folgt Bauten im Flachwasser (unten links: rote Linie). Sie wird manuell vereinfacht (unten rechts: rote Linie).  
(Quelle Geodaten: Bundesamt für Landestopografie)

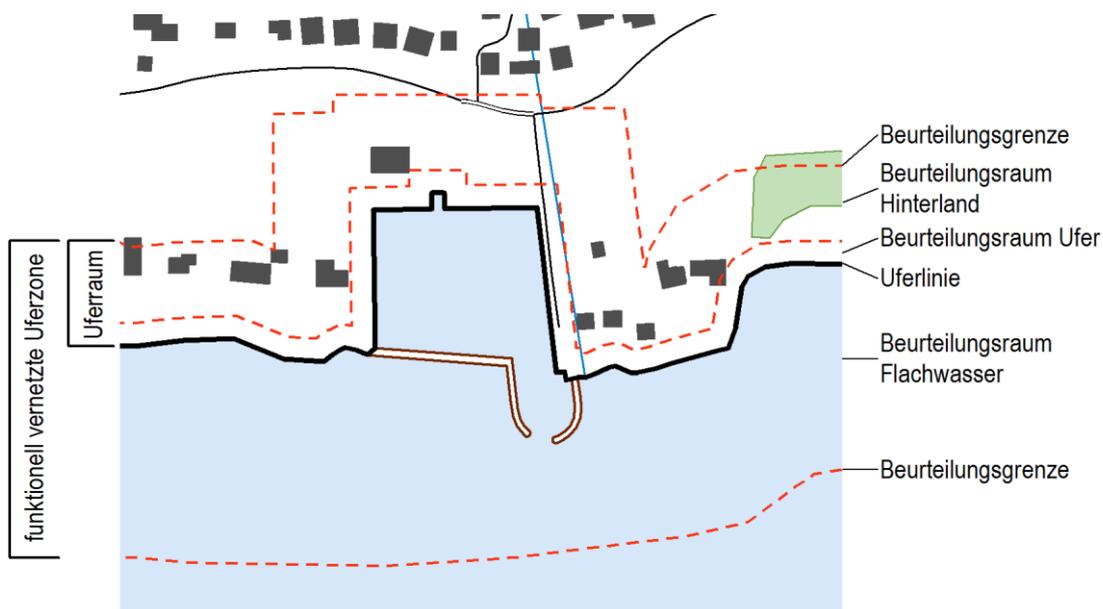
### 3.4 Beurteilungsräume zur Bestimmung der Attribute

Für die Seeufererhebung und –bewertung schlagen Teiber et al. (2013) drei Beurteilungstreifen vor:

- Beurteilungsraum Ufer
- Beurteilungsraum Flachwasser
- Beurteilungsraum Hinterland

Die definitive Abgrenzung der Beurteilungsräume soll im Verlauf der Methodenentwicklung klar festgelegt werden. Die Tiefen der Beurteilungsräume werden im Rahmen der vorliegenden Arbeit wie folgt festgelegt: Der zentrale Beurteilungstreifen Ufer umfasst die Zone zwischen Uferlinie und der 15 m landwärts verschobenen Linie. Die Breite von 15 m entspricht dem minimalen Gewässerraum für stehende Gewässer (GSchV Art. 41b). Seeseitig schliesst der Beurteilungsraum Flachwasser bis zur Haldenkante an (d.h. bis zu einer Wassertiefe von 4 bis 5 m). Der Beurteilungsraum Hinterland ist grundsätzlich ab dem Uferbereich landwärts 50 m breit. Bei sehr steilem Ufer ist der Einfluss des weiter entfernten Bereichs auf den See gering. Aus diesem Grund wird die Breite des Hinterlands verkleinert, sofern der Höhenunterschied zum Mittelwasserstand mehr als 20 m beträgt. Eine Mindestbreite von 20 m bleibt jedoch bestehen.

Im Rahmen der vorliegenden Arbeit werden die Beurteilungsräume wie in **Abb. 8** bezeichnet festgelegt.



**Abb. 8** Beurteilungsräume zur Bestimmung der Attribute (nach BAFU 2014a)  
(Quelle Geodaten: Bundesamt für Landestopografie)

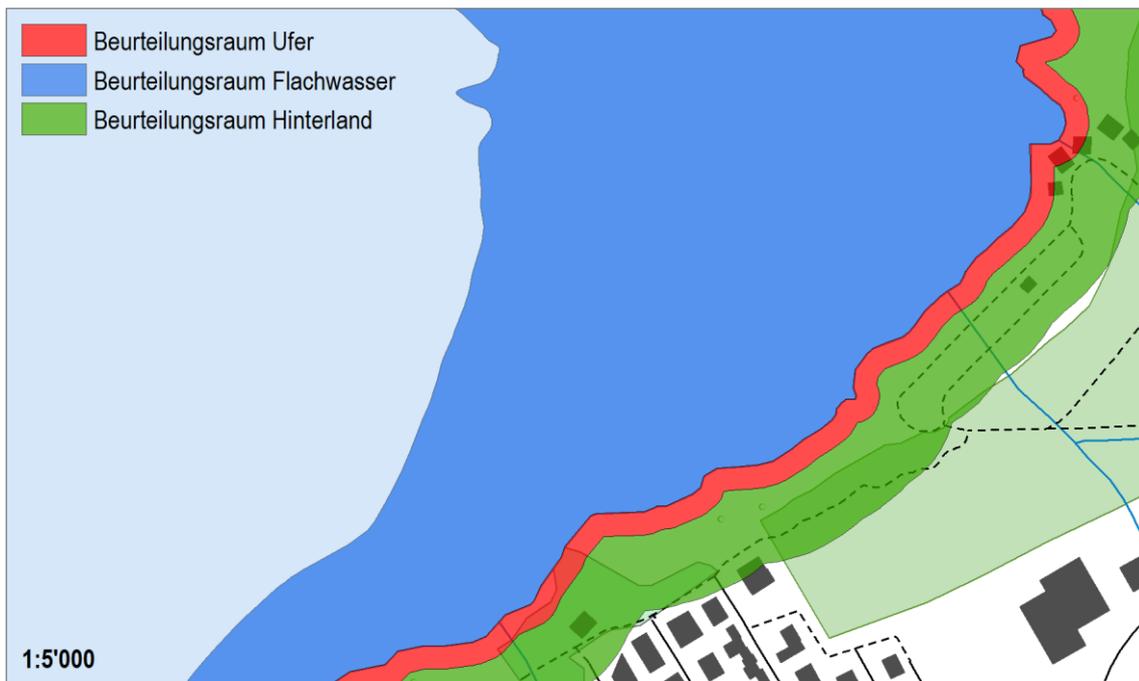
Die drei Beurteilungsräume werden automatisiert mit den zur Verfügung stehenden Geodaten berechnet. Zur Bestimmung des Uferbereichs kommen die geometrisch leicht vereinfachten Uferlinien aus dem swissTLM3D zum Einsatz. Der Beurteilungsraum Ufer umfasst die Zone zwischen Uferlinie und der 15 m landwärts verschobenen Linie. Damit bei weiteren Analysen auch Inhalte einbezogen werden, die genau auf der Uferlinie liegen, wird der Uferbereich um 0.5 m seewärts vergrößert.

Der Beurteilungsraum Flachwasser schliesst seeseitig an den Beurteilungsraum Ufer an und erstreckt sich bis zu einer Wassertiefe von 4 bis 5 m. Diese Tiefe ist auch bei kleineren

Gewässern anwendbar. Die Flachwasserzone ist lichtdurchflutet und weist eine andere Reflexion auf als die Freiwasserzone. Für die Bestimmung der seeseitigen Begrenzung wird daher das Falschfarbenorthofoto SWISSIMAGE FCIR eingesetzt. Da herstellerseitig keine radiometrischen Korrekturen vorgenommen wurden, sind die Fehler in den Daten sichtbar. Ein empirisch ermittelter Schwellwert im Grünkanal liefert eine Fläche im See, die aufgrund der geringen Reflexion der Freiwasserzone entspricht. Mit einem Generalisierungsalgorithmus wird das Polygon geometrisch vereinfacht. Für den Beurteilungsraum Flachwasser wird eine Mindestbreite von 5 m festgelegt, eine maximale Breite wird nicht definiert.

Der Beurteilungsraum Hinterland schliesst uferseitig an den Beurteilungsraum Ufer an. Bei sehr steilen Ufern verringert sich die Breite von den üblichen 50 m auf minimal 20 m. In dieser Bandbreite hängt die tatsächliche Breite vom Höhenunterschied zum Mittelwasserstand ab (maximal 20 m). Zur Berechnung des Hinterlands wird neben den geometrisch leicht vereinfachten Uferlinien aus dem swissTLM3D das Terrainmodell swissALTI3D zur Bestimmung des Höhenunterschieds verwendet.

In **Abb. 9** sind am Beispiel des flachen nordöstlichen Murtenseeufer die drei Beurteilungsräume dargestellt.



**Abb. 9** Umgesetzte Beurteilungsräume zur Bestimmung der Attribute  
(Quelle Geodaten: Bundesamt für Landestopografie)

Der Python Code zur Erstellung der drei Beurteilungsräume ist in Anhang 2 zu finden.

### 3.5 Übergeordnete Attribute

Zwei Attribute zur Beschreibung der Seen sind übergeordneter Natur und haben keinen direkten Einfluss auf die Seeuferbeurteilung.

#### 3.5.1 Seentyp

Für jeden ökomorphologisch zu erfassenden See wird der Seentyp festgelegt. Es handelt sich dabei um ein Merkmal, das Anhand von Grösse, Lage und Entstehung zugewiesen wird. Aufgrund des fehlenden bzw. einfachen räumlichen Bezugs wird der Seentyp im Rahmen der vorliegenden Arbeit nicht weiter untersucht. Die verschiedenen Ausprägungen sind in **Tab. 8** aufgelistet.

**Tab. 8** Ausprägungen des Merkmals „Seentyp“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
1	Seentyp	1.01 Bergsee oberhalb der Baumgrenze 1.02 Bergsee unterhalb der Baumgrenze 1.03 Voralpensee gross 1.04 Voralpensee klein 1.05 Mittellandsee gross 1.06 Mittellandsee klein 1.07 Kleingewässer (Weiher) 1.08 Teich 1.09 Stausee 1.10 Flusstau (aufgestautes Fließgewässer)	nicht Bestandteil der Klassifizierung

#### 3.5.2 Wasserspiegelregulierung

Die Wasserspiegelregulierung eines Sees wird mit Hilfe von historischem Wissen für jeden See erfasst. Aufgrund des fehlenden bzw. einfachen räumlichen Bezugs wird dieses Merkmal im Rahmen der vorliegenden Arbeit nicht weiter untersucht. Die verschiedenen Ausprägungen sind in **Tab. 9** aufgelistet.

**Tab. 9** Ausprägungen des Merkmals „Wasserspiegelregulierung“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
2	Wasserspiegelregulierung	2.01 Wasserspiegel nicht reguliert 2.02 Wasserspiegel statisch reguliert (Pegel wenig schwankend) 2.03 Wasserspiegel dynamisch reguliert (Pegel stark schwankend) 2.04 Wasserspiegel historisch verändert - nicht reguliert 2.05 Wasserspiegel historisch verändert - reguliert	nicht Bestandteil der Klassifizierung

### 3.6 Beurteilungsraum Ufer

Insgesamt neun Attribute sind im Bereich des zentralen Beurteilungsraums anzusiedeln. Bis auf den Ufertyp und die standorttypische Ufervegetation fließen alle Merkmale direkt in die Bewertung des Seeufers ein.

Der Python Code zur Auswertung der Attribute ist in Anhang 3 zu finden.

#### 3.6.1 Ufertyp

Rey et al. (2009) haben Steilheit der landseitigen Verhältnisse im Renaturierungsleitfaden Bodensee in drei Klassen eingeteilt: Steilufer ( $\geq 1:5$ ), mittelsteiles Ufer ( $1:5 - 1:20$ ) und Flachufer ( $\leq 1:20$ ). Aufgrund der Schweizer Topografie werden für die Methode Ökomorphologie Seeufer zwei weitere Klassen zugefügt und die Neigungsklassen leicht angepasst. Der Ufertyp hat informativen Charakter und keinen Einfluss auf die Bewertung eines Gewässerabschnitts. Die verschiedenen Ausprägungen sind in **Tab. 10** aufgelistet.

**Tab. 10** Ausprägungen des Merkmals „Ufertyp“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
3	Ufertyp	3.01 Sehr steiles Ufer (Neigung $\geq 1:2.5$ ) 3.02 Steilufer (Neigung $1:2.5 - 1:5$ ) 3.03 Mittelsteiles Ufer (Neigung $1:5 - 1:12.5$ ) 3.04 Flachufer (Neigung $1:12.5 - 1:25$ ) 3.05 Sehr flaches Ufer (Neigung $< 1:25$ )	nicht Bestandteil der Klassifizierung

Um die unterschiedlichen Gefällsklassen automatisiert zu extrahieren, wird das Geländemodell swissALTI3D verwendet.

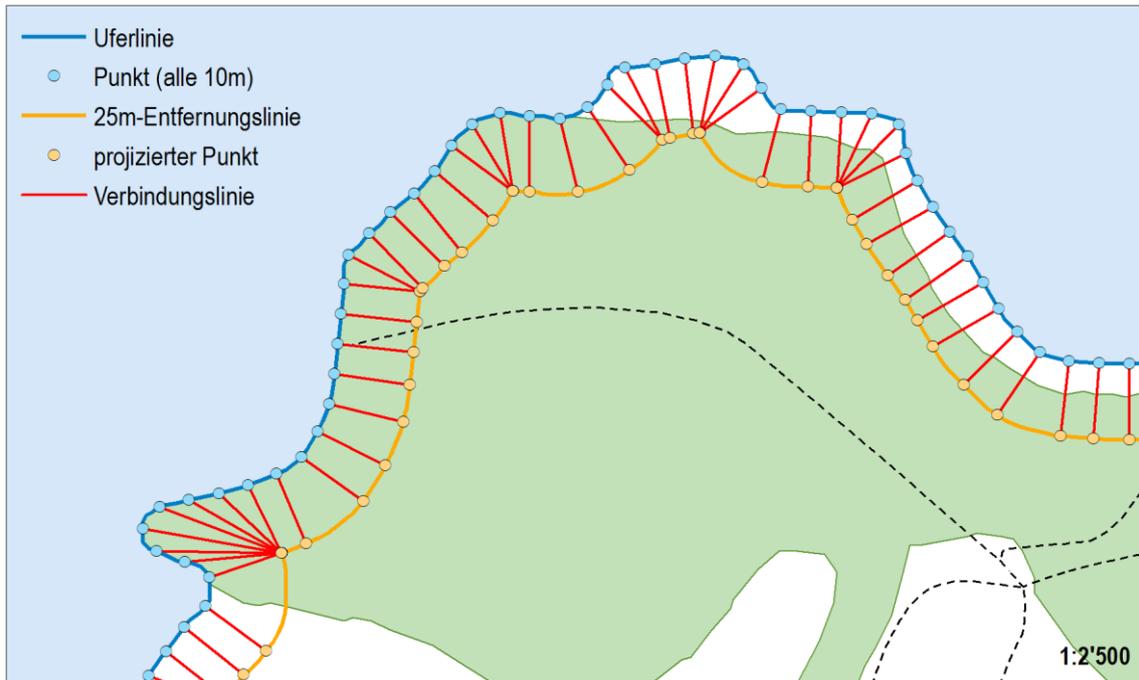
#### Vorgehen der Auswertung

Die Neigung wird in regelmässigen Abständen bestimmt. Dazu wird alle 10 m auf der Uferlinie ein Punkt erstellt. Diese Punkte werden auf eine vom Ufer entfernte Linie projiziert. Die Abstände der projizierten Punkte sind nicht mehr zwingend regelmässig. Für die Verbindungslinie eines Punktepaares wird schliesslich das Gefälle berechnet (vgl. **Abb. 10**).

Tests zeigen, dass die 15m-Tiefe des Uferbereichs zu schmal ist für die Bestimmung des Gefälles. Schon kleine Abweichungen in der Uferlinie führen zu rasch wechselnden Neigungen, die nicht der Realität entsprechen. Eine Breite von 25 m liefert stabilere und zuverlässigere Resultate.

Der Höhenunterschied zwischen dem äusseren Ende und dem Seespiegel wird mit Hilfe des Geländemodells swissTLM3D bestimmt. Aus Höhenunterschied und Distanz ergibt sich das Gefälle, das sich in eine der vorgegebenen Klassen einordnen lässt. Die Neigungsklasse gilt für jeweils 10 m der Uferlinie (vom auf der Uferlinie liegenden Punkt je 5 m auf beide Seiten). Damit wird sichergestellt, dass für die ganze Uferlinie eine lückenlose Bestimmung des Ufertyps vorliegt.

In einem letzten Schritt werden nebeneinander liegende Teilstücke mit gleicher Ausprägung zusammengefasst.



**Abb. 10** Vorgehen bei der Bestimmung des Ufertyps am Beispiel des Bielersees (Quelle Geodaten: Bundesamt für Landestopografie)

Aufgrund der geometrischen Gegebenheiten betragen die Horizontalabstände der Linien mindestens 25 m. Schon geringe Abweichungen der Uferlinie können grössere Höhenunterschiede zur Folge haben. Um Auswirkungen dieser Unsicherheiten in der Uferlinie gering zu halten, wird bei den inneren Punkten von der mittleren Seespiegellhöhe ausgegangen.

Der Python Code zur Auswertung des Ufertyps ist in Anhang 3.01 zu finden.

### 3.6.2 Standorttypische Ufervegetation

Mit vier Ausprägungen wird erfasst, ob die Ufervegetation standorttypisch ist oder nicht. Dabei muss die Vegetation mindestens so breit wie der Uferbereich sein ( $\geq 15$  m). Das Attribut hat keinen Einfluss auf die Bewertung eines Gewässerabschnitts. Die verschiedenen Ausprägungen sind in **Tab. 11** aufgelistet.

**Tab. 11** Ausprägungen des Merkmals „Standorttypische Ufervegetation“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
7	Standorttypische Ufervegetation	7.01 Standorttypische Ufervegetation fehlend 7.02 Typische Sukzession (Ried, Riedwiese, Flachmoor dominierend) 7.03 Typische Sukzession (Uferwald dominierend) 7.04 Standorttypische Vegetation oberhalb der Baumgrenze	nicht Bestandteil der Klassifizierung

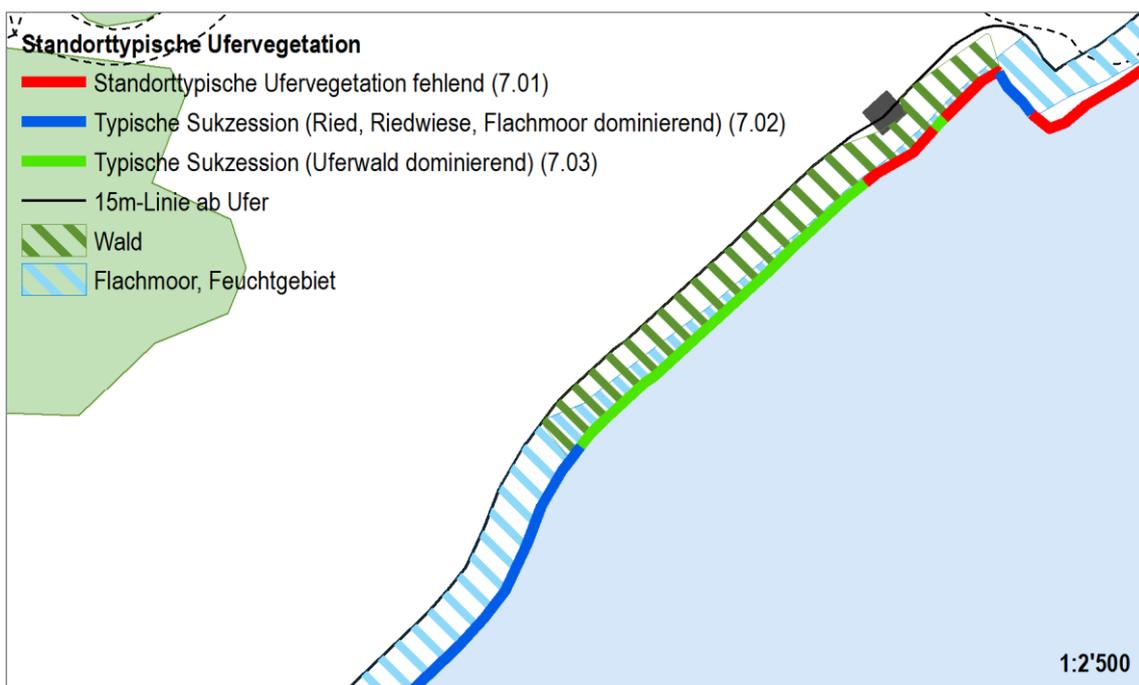
Flächendeckend verfügbare Geodaten zur Vegetation sind in der swissTLM3D-Topic „Bodenbedeckung“ sowie in den Bundesinventaren der Flachmoore und Auen enthalten. Die Kategorien werden aus folgenden Inhalten extrahiert:

- 7.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 7.02: Bundesinventar der Flachmoore von nationaler Bedeutung, Bundesinventar der Auen von nationaler Bedeutung, Feuchtgebiete aus der Bodenbedeckung
- 7.03: Wald, Wald offen und Gebüschwald aus der Bodenbedeckung
- 7.04: keine flächendeckend verfügbaren Informationen

### Vorgehen der Auswertung

Die für die Auswertung relevanten Polygone werden aus den Grundlagendaten extrahiert und thematisch zusammengefügt. Um sicherzustellen, dass geringfügige Abweichungen im Bereich der Uferlinie nicht zum Ausschluss der standorttypischen Ufervegetation führen, werden die Flächen mit einem Meter gepuffert. Anschliessend folgt das Ausstanzen auf 15 m ab Uferlinie.

Die Vegetation ist dann mindestens 15 m breit, wenn für einen bestimmten Bereich sowohl die Uferlinie als auch die Aussenbegrenzung des Uferraums identisch sind mit der Umrandung der ausgestanzten Vegetation (vgl. **Abb. 11**). Bei Überlappungen hat die Kategorie 7.02 aufgrund des selteneren Vorkommens Vorrang gegenüber 7.03.



**Abb. 11** Damit einem Uferabschnitt der Typ 7.02 bzw. 7.03 zugewiesen wird, müssen die Begrenzungslinie des ausgestanzten Polygons (Wald bzw. Flachmoor / Feuchtgebiet) sowohl mit der Uferlinie als auch mit der 15 m-Linie ab Ufer identisch sein. Andernfalls weist die Ufervegetation nicht die geforderte Breite auf.  
(Quelle Geodaten: Bundesamt für Landestopografie)

Die Lokalisierung der identischen Uferstücke auf der kilometrierten Uferlinie ist mit der üblichen ArcGIS-Funktion „Locate Features along Routes“ möglich. Die Lokalisierung der identischen Linienstücke der äusseren Begrenzung mit der herkömmlichen Funktion ist fehleranfällig. Oft werden die Abschnitte zu kurz. Daher werden stattdessen Anfangs- und Endpunkt jeder äusseren Linie auf das Ufer projiziert.

Der Python Code zur Auswertung der Ufervegetation ist in Anhang 3.02 zu finden.

### 3.6.3 Ufersaum-Vegetation

Das Vorkommen standorttypischer Ufersaum-Vegetation soll nach aktuellem Stand der Methodenentwicklung (BAFU 2014a) mit vier Ausprägungen (8.01 bis 8.04) erfasst werden. Um Aussagen zur gesamten Uferlinie machen zu können, wird im Rahmen der vorliegenden Arbeit eine zusätzliche Ausprägung ergänzt (8.05). Das Attribut fließt direkt in die Uferbewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 12** aufgelistet. Das Vorgehen bei der Auswertung und auch die Resultate sind ähnlich wie bei Attribut 7 (Standorttypische Ufervegetation). Aufgrund der geringen Ausdehnung des Ufersaums werden Einzelbäume, Gebüsch und Baum- bzw. Gebüschreihen auch einbezogen.

**Tab. 12** Ausprägungen des Merkmals „Ufersaum-Vegetation“ und dessen Klassifizierung (nach BAFU 2014a, abgeändert).

Nr.	Attribut	Ausprägung	Klassifizierung
8	Ufersaum-Vegetation	8.01 Ufersaum-Vegetation standorttypisch Gehölz $\geq$ 3 m Breite	vorhanden
		8.02 Ufersaum-Vegetation standorttypisch Röhricht / Ried $\geq$ 3 m Breite	vorhanden
		8.03 Ufersaum-Vegetation standorttypisch Gehölz $<$ 3 m Breite	fehlend
		8.04 Ufersaum-Vegetation standorttypisch Röhricht / Ried $<$ 3 m Breite	fehlend
		8.05 Keine standorttypische Ufersaum-Vegetation	fehlend

Flächendeckend verfügbare Geodaten zur Vegetation des Ufersaums sind im swissTLM3D-Topic „Bodenbedeckung“ (Feature Classes Bodenbedeckung, Einzelbaum / Gebüsch, Baum- / Gebüschreihe) sowie in den Bundesinventaren der Flachmoore und Auen enthalten. Die Kategorien werden aus folgenden Inhalten extrahiert:

- 8.01: Wald, Wald offen und Gebüschwald aus der Bodenbedeckung, Einzelbäume (5m-Krone), Baum- und Gebüschreihen (10m-Breite)
- 8.02: Bundesinventar der Flachmoore von nationaler Bedeutung, Bundesinventar der Auen von nationaler Bedeutung, Feuchtgebiete
- 8.03: Wald, Wald offen und Gebüschwald aus der Bodenbedeckung, Einzelbäume (5m-Krone), Baum- und Gebüschreihen (10m-Breite)
- 8.04: Bundesinventar der Flachmoore von nationaler Bedeutung, Bundesinventar der Auen von nationaler Bedeutung, Feuchtgebiete
- 8.05: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind

#### Vorgehen der Auswertung

Die für die Auswertung relevanten Kategorien werden aus den Grundlegendaten extrahiert. Einzelbäume und Baum- / Gebüschreihen werden gepuffert. Alle resultierenden Polygon-Datensätze werden zusammengefügt. Um sicherzustellen, dass geringfügige Abweichungen im Bereich der Uferlinie nicht zum Ausschluss der standorttypischen Ufersaum-Vegetation führen, werden die Flächen mit einem Meter gepuffert. Anschliessend folgt das Ausstanzen auf 3 m ab Uferlinie.

Die Vegetation ist dann mindestens 3 m breit, wenn für einen bestimmten Bereich sowohl die Uferlinie als auch die 3 m entfernte Aussenbegrenzung identisch sind mit der Umrandung der ausgestanzten Vegetation. Ist nur eine Begrenzungslinie identisch, so ist die Ufersaum-

vegetation schmäler als 3 m. Bei Überlappungen haben die Kategorien 8.02 und 8.04 Vorrang gegenüber 8.01 und 8.03.

Die Lokalisierung der identischen Uferstücke auf der kilometrierten Uferlinie ist mit der üblichen ArcGIS-Funktion „Locate Features along Routes“ möglich. Die Lokalisierung der identischen Linienstücke der äusseren Begrenzung mit der herkömmlichen Funktion ist fehleranfällig. Oft werden die Abschnitte zu kurz. Daher werden stattdessen Anfangs- und Endpunkt jeder äusseren Linie auf das Ufer projiziert.

Der Python Code zur Auswertung der Ufersaum-Vegetation ist in Anhang 3.03 zu finden.

### 3.6.4 Uferverbau und künstliche Morphologie

Die Verbauungsart ist ein wichtiges Merkmal zur Beurteilung eines Seeufers. Uferverbau und künstliche Morphologie werden mit sechs verschiedenen Ausprägungen erfasst, die direkt in die Seeuferbewertung einfließen. Die Uferlinie wird lückenlos einer Kategorie zugewiesen. Die verschiedenen Ausprägungen sind in **Tab. 13** aufgelistet.

**Tab. 13** Ausprägungen des Merkmals „Uferverbau und künstliche Morphologie“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
9	Uferverbau und künstliche Morphologie	9.01 Keine Uferverbauungen	nicht beeintr.
		9.02 Geringfügige Uferverbauungen (geringe Spuren)	gering beeintr.
		9.03 Kleinere Uferverbauungen (deutliche Spuren)	beeinträchtigt
		9.04 Hartverbau durchlässig	naturfern
		9.05 Hartverbau undurchlässig	naturfremd
		9.06 Regenwasserentlastung	naturfremd

Flächendeckend verfügbare Geodaten zur Verbauung des Ufers sind im swissTLM3D-Topic „Bauten“ (Feature Class Verbauung Mauer) enthalten. Die Inhalte lassen allerdings keine Unterscheidungen der Kategorien zu. Folgende Informationen werden extrahiert:

- 9.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 9.02 – 9.04: keine flächendeckend verfügbaren Informationen
- 9.05: Gewässerverbauung aus Verbauung Mauer
- 9.06: keine flächendeckend verfügbaren Informationen

#### Vorgehen der Auswertung

Aus den zur Verfügung stehenden Daten werden die Gewässerverbauungen extrahiert. Anfangs- und Endpunkte dieser Linien werden auf die Uferlinie projiziert. Da im swissTLM3D nur Mauern erfasst sind, werden die betroffenen Abschnitte der Kategorie 9.05 zugewiesen. Mangels weiterer Informationen gelten die restlichen Uferabschnitte als unverbaut (9.01).

Der Python Code zur Auswertung Uferverbau und künstliche Morphologie ist in Anhang 3.04 zu finden.

### 3.6.5 Fliessgewässeranbindung (Mündung / Abfluss)

Ein 50 cm hohes Hindernis ist für die meisten Fische unüberwindbar, gewisse Arten scheitern schon an 20 cm hohen Abstürzen. Eine solche Durchgangsstörung trennt die oberhalb liegenden Gewässerabschnitte vom Lebensraum See ab. Aus diesem Grund werden alle Fliessgewässer, die in den zu beurteilenden See münden oder von ihm abfließen, bezüglich ihrer Struktur und der Aufstiegshindernisse untersucht. Die Uferlinie wird dabei nur punktuell beurteilt. Die verschiedenen Ausprägungen dieses Merkmals sind in **Tab. 14** aufgelistet.

**Tab. 14** Ausprägungen des Merkmals „Fließgewässeranbindung“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
10	Fließgewässeranbindung (Mündung / Abfluss)	10.01 Anbindung naturnah ohne Aufstiegshindernis	nicht beeintr.
		10.02 Anbindung naturnah mit Aufstiegshindernis	beeinträchtigt
		10.03 Anbindung verbaut ohne Aufstiegshindernis	naturfern
		10.04 Anbindung verbaut mit Aufstiegshindernis	naturfremd
		10.05 Anbindung verdolt	naturfremd

Detaillierte Informationen zum Natürlichkeitsgrad von Fliessgewässern und zu Aufstiegshindernissen stehen in den kantonalen Ökomorphologiedaten Stufe F zur Verfügung. Sie wurden nach einer einheitlichen Methode erhoben. Obwohl im Zusammenhang mit der Strategischen Revitalisierungsplanung der Fliessgewässer (GSchV, Art. 41d) viele zusätzliche Gewässer ökomorphologisch kartiert wurden, bestehen noch Lücken in den einzelnen Datensätzen. Um alle bekannten Mündungen zu lokalisieren, werden daher zusätzlich die Fliessgewässer aus dem swissTLM3D überprüft.

Folgende Objekte aus den Ökomorphologiedaten Stufe F gelten als Hindernisse und sind für einige Fische nicht überwindbar:

- Natürliche und künstliche Abstürze mit einer Höhe von mindestens 20 cm
- Bauwerke mit Ausnahme von Fischpässen, Durchlässen und Brücken

Aus den Abschnittsdaten der Ökomorphologie sind die Attribute Gewässersohlenbreite und die Klassierung entscheidend.

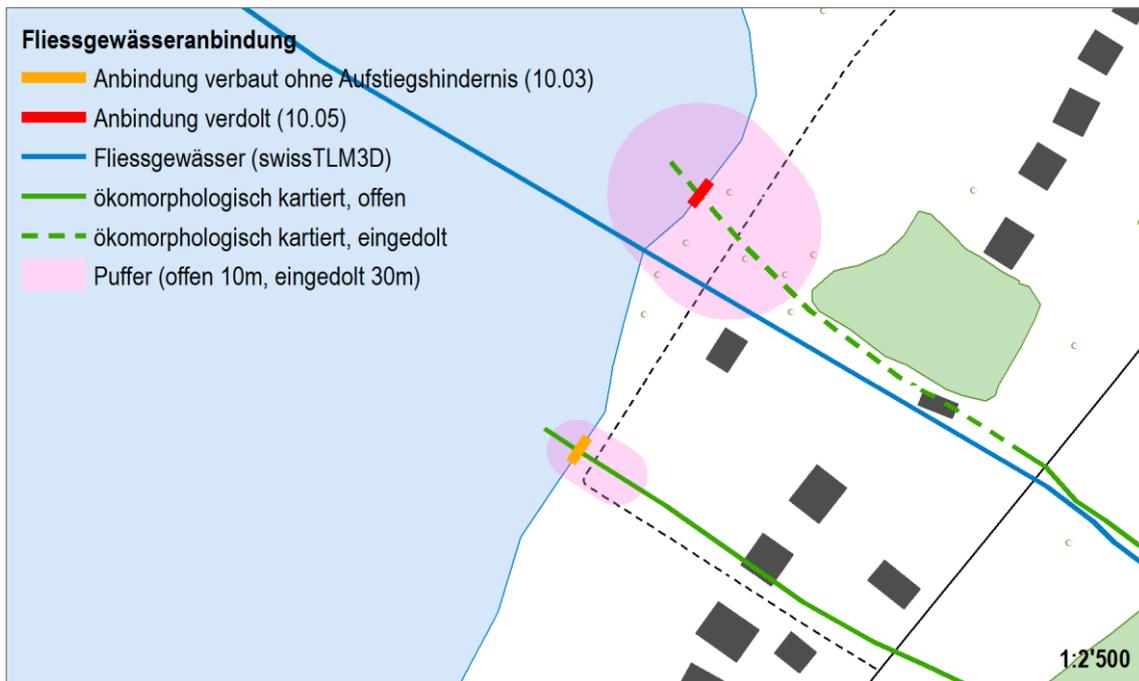
#### Vorgehen der Auswertung

Bei der Aufbereitung der verwendeten Geodaten werden die verschiedenen kantonalen Datensätze in ein File integriert. Grundlage für die Kartierungen sind meist kantonale Gewässernetze, die detaillierter sind als das Gewässernetz swissTLM3D.

Abschnitte, Abstürze und Bauwerke, die innerhalb des Beurteilungsraums liegen, werden extrahiert. Informationen zu den Durchgangshindernissen werden an die Abschnitte gefügt. Damit kann die geltende Ausprägung bestimmt werden (10.01 bis 10.05). Diese Informationen zur Anbindung werden auf die Uferlinie projiziert. Die Ausdehnung auf der Linie wird mit mindestens 10 m festgelegt. Ist die Gewässersohle breiter, wird die Ausdehnung dementsprechend vergrößert.

SwissTLM3D und die kantonalen Gewässernetze weisen geometrische Abweichungen auf, die vor allem bei eingedolten Gewässern sehr gross sein können. Es muss sichergestellt werden, dass keine Mündungen aufgenommen werden, die schon kartiert sind. Dazu werden alle swissTLM3D-Gewässer gelöscht, die ganz oder teilweise innerhalb eines Puffers eines

kartierten Abschnitts liegen. Die Grösse der Puffer wurde empirisch ermittelt und bei offenen Gewässern mit 10 m, bei eingedolten Gewässern mit 30 m festgelegt (vgl. **Abb. 12**).



**Abb. 12** Das swissTLM3D weicht vor allem bei eingedolten Abschnitten stark vom kantonalen Gewässernetz ab (Beispiel Bielersee). Mit Hilfe von unterschiedlich grossen Puffern werden bei eingedolten Abschnitten auch weiter vom kantonalen Datensatz entfernte Gewässer gelöscht und nicht als zusätzliche Mündung erkannt. (Quelle Geodaten: Bundesamt für Landestopografie)

Die zusätzlichen Anbindungen aus swissTLM3D werden auf die Uferlinie projiziert. Da in den Daten keine Informationen zur Gewässerbreite vorliegen wird von einer Breite von 10 m ausgegangen. Keine der vordefinierten Klassierungen kann zugewiesen werden. Die Mündungen oder Abflüsse werden daher mit der Ausprägung 10.00 erfasst.

Der Python Code zur Auswertung der Anbindungen von Fließgewässern ist in Anhang 3.05 zu finden.

### 3.6.6 Land- und forstwirtschaftliche Nutzung im Uferstreifen

Die Klassierung der Uferlinie bezüglich der land- und forstwirtschaftlichen Nutzung im Uferstreifen erfolgt lückenlos und fließt direkt in die Bewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 15** aufgelistet.

**Tab. 15** Ausprägungen des Merkmals „Land- und forstwirtschaftliche Nutzung im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
11	Land- und forstwirtschaftliche Nutzung im Uferstreifen	11.01 Keine Nutzungen	nicht beeintr.
		11.02 Extensiv genutzte Land- und Forstwirtschaftsflächen	gering beeintr.
		11.03 Intensiv genutzte Land- und Forstwirtschaftsflächen	naturfern

Zur Bestimmung der land- und forstwirtschaftlichen Nutzung im Uferbereich werden die swissTLM3D-Topics „Bodenbedeckung“ sowie „Nutzungsareal“ verwendet. Die Kategorien werden aus folgenden Inhalten extrahiert:

- 11.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 11.02: Wald aus der Bodenbedeckung; keine flächendeckend verfügbaren Informationen zur Landwirtschaft
- 11.03: Baumschulen, Obstanlagen und Reben aus den Nutzungsarealen; keine flächendeckend verfügbaren Informationen zur Landwirtschaft

### Vorgehen der Auswertung

Die Ausprägungen 11.02 und 11.03 werden einzeln geprüft. Dazu werden die relevanten Daten aus dem swissTLM3D extrahiert und auf die Uferlinie projiziert. Es ist theoretisch möglich, dass für einen Gewässerabschnitt beide Ausprägungen zutreffen. In diesem Fall hat die Kategorie 11.03 Vorrang gegenüber 11.02. Alle weiteren Abschnitte fallen in die Kategorie 11.01.

Der Python Code zur Auswertung der land- und forstwirtschaftlichen Nutzung im Uferbereich ist in Anhang 3.06 zu finden.

### 3.6.7 Freizeit- und Privatnutzung im Uferstreifen

Die Klassierung der Uferlinie bezüglich der Freizeit- und Privatnutzung im Uferstreifen erfolgt lückenlos und fließt direkt in die Bewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 16** aufgelistet.

**Tab. 16** Ausprägungen des Merkmals „Freizeit- und Privatnutzung im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
12	Freizeit- und Privatnutzungen im Uferstreifen	12.01 Keine Nutzungen	nicht beeintr.
		12.02 Leichte Nutzungsspuren durch Freizeitbetrieb	gering beeintr.
		12.03 Extensive Privat- und Freizeitnutzungen	beeinträchtigt
		12.04 Intensive Privat- und Freizeitnutzungen	naturfern
		12.05 Sehr starke Privat- und Freizeitnutzungen	naturfremd

Das swissTLM3D enthält keine Informationen zu Freizeit- und Privatnutzungen. Auch aus SWISSIMAGE FCIR lassen sich Nutzungen nur mittels Interpretation erfassen. Mit der Arealstatistik (BFS 2009a) liegt zwar ein flächendeckend verfügbarer Geodatensatz mit Nutzungskategorien vor, die Auflösung von 1ha genügt jedoch nicht den Anforderungen von Ökomorphologie Seeufer. In der vorliegenden Arbeit wird das Attribut zur Freizeit- und Privatnutzung nicht weiter analysiert.

### 3.6.8 Siedlung, Gewerbe, Industrie im Uferstreifen

Die Klassierung der Uferlinie bezüglich Siedlung, Gewerbe und Industrie im Uferstreifen erfolgt lückenlos und fließt direkt in die Bewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 17** aufgelistet.

**Tab. 17** Ausprägungen des Merkmals „Siedlung, Gewerbe, Industrie im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
13	Siedlung, Gewerbe, Industrie im Uferstreifen	13.01 Keine Uferbauwerke	nicht beeintr.
		13.02 Vereinzelte Kleingebäude (Hütte / Pavillon / kleines Nebengebäude)	gering beeintr.
		13.03 Freistehende Einzelbauwerk	beeinträchtigt
		13.04 Lückige Bebauung (versiegelte Fläche $\leq$ Freifläche)	naturfern
		13.05 Dichte Bebauung (versiegelte Fläche $>$ Freifläche)	naturfremd
		13.06 Standortgebundene Uferbauten / Uferanlagen landseitig	naturfremd

Zur Bestimmung von Siedlung, Gewerbe und Industrie im Uferstreifen wird das swissTLM3D-Topic „Gebäude“ verwendet:

- 13.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 13.02: einzelnes Gebäude  $\leq 50 \text{ m}^2$
- 13.03: einzelnes Gebäude  $> 50 \text{ m}^2$
- 13.04: mehrere Gebäude, versiegelte Fläche  $\leq$  Freifläche
- 13.05: mehrere Gebäude, versiegelte Fläche  $>$  Freifläche
- 13.06: keine flächendeckend verfügbaren Informationen

#### Vorgehen der Auswertung

Um nahe beieinander liegende Gebäude als zusammenhängende Siedlungen zu erkennen werden die Einzelbauten gepuffert. Nach einem Verschnitt mit dem Uferstreifen wird jedem Siedlungspolygon die Anzahl Gebäude und das Verhältnis zwischen bebaut und unbebaut zugewiesen. Mit dieser Information ist eine Klassierung in 13.02 bis 13.05 möglich. Die gültigen Bereiche werden anschliessend auf die Uferlinie projiziert.

Der Python Code zur Auswertung von Siedlung, Gewerbe und Industrie im Uferbereich ist in Anhang 3.07 zu finden.

### 3.6.9 Verkehrswege und –flächen im Uferstreifen

Die Klassierung der Uferlinie bezüglich Verkehrsanlagen erfolgt lückenlos und fliesst direkt in die Bewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 18** aufgelistet.

**Tab. 18** Ausprägungen des Merkmals „Verkehrswege und –flächen im Uferstreifen“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
14	Verkehrswege und -flächen im Uferstreifen	14.01 Keine Verkehrsinfrastruktur	nicht beeintr.
		14.02 Unbefestigter, ungeteeter schmaler Pfad / Fussweg / Veloweg	gering beeintr.
		14.03 Befestigter, geteeter schmaler Pfad / Fussweg / Veloweg	beeinträchtigt
		14.04 Unbefestigter, ungeteeter Fahrweg	beeinträchtigt
		14.05 Unversiegelte Parkplatzfläche	naturfern
		14.06 Unbefestigte, ungeteerte Uferpromenade / Quaianlage	naturfern
		14.07 Versiegelte Parkplatzfläche	naturfremd
		14.08 Befestigte, geteerte Uferpromenade / Quaianlage	naturfremd
		14.09 Befestigte, geteerte Strasse (Fahrweg bis Kantonsstrasse)	naturfremd
		14.10 Befestigte, geteerte Strasse (Autobahn)	naturfremd
		14.11 Bahntrasse	naturfremd

Zur Bestimmung von Verkehrsanlagen im Uferbereich werden die swissTLM3D-Topics „Strassen und Wege“ sowie „Öffentlicher Verkehr“ verwendet. Die für die Auswertung relevanten Elemente sind als Linien erfasst. Ihnen werden folgende Breiten zugewiesen, die aufgrund des TLM-Objektkatalogs (swisstopo 2012) festgelegt wurden:

- 14.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 14.02/14.03: 1m-Weg (1m)
- 14.04: 2m-Weg (2m)
- 14.05/14.07: Parkplatz (Umriss, daher keine Breite)
- 14.06: keine flächendeckend verfügbaren Informationen
- 14.08: keine flächendeckend verfügbaren Informationen
- 14.09: Autostrasse (15m); 10m-Strasse / Einfahrt / Ausfahrt / Verbindung (10m); 6m-Strasse (6m); 4m-Strasse / Zufahrt / Dienstzufahrt (4m); 3m-Strasse (3m)
- 14.10: Autobahn (40m)
- 14.11: Normalspurbahn (15m); Schmalspurbahn (10m)

#### Vorgehen der Auswertung

Die Ausprägungen werden soweit möglich einzeln geprüft. Unklarheiten bestehen bei der Belagsart (befestigt / unbefestigt). In diesen Fällen wird die schlechter bewertete Ausprägung zugewiesen.

Die relevanten Daten werden aus dem swissTLM3D extrahiert und gepuffert. Erst anschliessend erfolgt ein Ausstanzen mit dem Beurteilungsraum. So wird sichergestellt, dass auch Verkehrsanlagen, deren Mittellinie unmittelbar neben dem Uferbereich liegen, in die Auswertung einbezogen werden.

Die Polygone bzw. im Fall von Parkplätzen Polylinien werden auf die Uferlinie projiziert und mit den restlichen Ausprägungen verrechnet. Es ist möglich, dass für einen Gewässerabschnitt verschiedene Ausprägungen zutreffen. So kann beispielsweise sowohl ein schmaler Fussweg als auch ein Bahntrasse innerhalb des Beurteilungsraums liegen. In diesem Fall wird immer die Kategorie übernommen, welche zu einer schlechteren Beurteilung führt (vgl. **Abb. 13**).



**Abb. 13** Für einen Uferabschnitt können mehrere Kategorien von Verkehrsanlagen zutreffen (Beispiel Bielersee). Obwohl ein Fussweg näher beim Seeufer liegt als die Bahnlinie, wird die Kategorie übernommen, welche zu einer schlechteren Bewertung führt, d.h. die Ausprägung ist 14.11 und nicht 14.03. (Quelle Geodaten: Bundesamt für Landestopografie)

Der Python Code zur Auswertung der Verkehrswege und –flächen im Uferbereich ist in Anhang 3.08 zu finden.

### 3.7 Beurteilungsraum Flachwasser

Insgesamt fünf Attribute sind im Bereich des Flachwassers anzusiedeln. Nur ein Merkmal (Strukturen / Anlagen in der Flachwasserzone) fließt direkt in die Bewertung des Seeufers ein.

Der Python Code zur Auswertung der Attribute ist in Anhang 3 zu finden.

#### 3.7.1 Ausdehnung der Flachwasserzone

Die Breite der Flachwasserzone wird in fünf Klassen erfasst. Eine lückenlose Auswertung ist möglich. Das Attribut hat informativen Charakter und keinen Einfluss auf die Bewertung eines Gewässerabschnitts. Die verschiedenen Ausprägungen sind in **Tab. 19** aufgelistet. Teiber (2010) legte die Klassengrenzen der Breite im Rahmen der Seeuferbewertung Vierwaldstättersee fest.

**Tab. 19** Ausprägungen des Merkmals „Ausdehnung der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
4	Ausdehnung der Flachwasserzone	4.01 Sehr schmale Flachwasserzone (Breite $\leq 10$ m) 4.02 Schmale Flachwasserzone (Breite 10 – 50 m) 4.03 Mittelbreite Flachwasserzone (Breite 50 – 100 m) 4.04 Breite Flachwasserzone (Breite 100 – 250 m) 4.05 Sehr breite, ausgedehnte Flachwasserzone (Breite $> 250$ m)	nicht Bestandteil der Klassifizierung

#### Vorgehen der Auswertung

Die Ausdehnung der Flachwasserzone wird in regelmässigen Abständen bestimmt. Dazu wird alle 10 m auf der Uferlinie ein Punkt erstellt. Diese Punkte werden auf die innere Begrenzung des Beurteilungsraums Flachwasser projiziert. Die Abstände der projizierten Punkte sind nicht mehr zwingend regelmässig. Die Länge der Verbindungslinie eines Punktepaares stellt die Breite der Flachwasserzone dar und kann in die vorgegebenen Klassen eingeteilt werden. Anschließend werden nebeneinanderliegende Teilstücke mit gleicher Ausprägung zusammengefasst.

Der Python Code zur Auswertung der Ausdehnung der Flachwasserzone ist in Anhang 3.09 zu finden.

### 3.7.2 Wellenexposition

Die Wellenexposition soll als zusätzliches Merkmal erfasst werden, dient jedoch nicht der Klassifizierung eines Uferabschnitts. Der Uferlinie kann lückenlos eine Ausprägung zugewiesen werden. Die verschiedenen Ausprägungen sind in **Tab. 20** aufgelistet.

**Tab. 20** Ausprägungen des Merkmals „Wellenexposition“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
5	Wellenexposition	5.01 Bereich starkem Wellenschlag ausgesetzt 5.02 Bereich mittlerem Wellenschlag ausgesetzt 5.03 Bereich geringem Wellenschlag ausgesetzt 5.04 Keine Information vorhanden	nicht Bestandteil der Klassifizierung

Zur Bestimmung der Wellenexposition sind Modellierungen und detaillierte Kenntnisse von Wind und Untergrund nötig. Aus den zur Verfügung stehenden Geodaten lassen sich keine Rückschlüsse ziehen. Daher wird dieses Merkmal in der vorliegenden Arbeit nicht weiter analysiert. Die Anwendung der Methode Seeufer aufgrund von Ortho- und Schrägbildern lässt keine Aussagen zur Wellenexposition zu. Für einzelne Schweizer Seen steht jedoch ein Wellenatlas zur Verfügung. Darin wird aufgezeigt, wie häufig welche Wellen zu erwarten sind.

### 3.7.3 Standorttypische emerse Vegetation der Flachwasserzone

Als ergänzendes Merkmal ohne Einfluss auf die Klassifizierung soll das Röhrichtvorkommen in der Flachwasserzone erfasst werden. Der Uferlinie kann lückenlos eine Ausprägung zugewiesen werden. Die verschiedenen Ausprägungen sind in **Tab. 21** aufgelistet.

**Tab. 21** Ausprägungen des Merkmals „Standorttypische emerse Vegetation der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
6	Standorttypische emerse Vegetation der Flachwasserzone	6.01 Kein emerser Bewuchs 6.02 Röhrichtsaum < 5 m Breite 6.03 Röhrichtband 5-20 m Breite 6.04 Röhrichtfläche > 20 m Breite	nicht Bestandteil der Klassifizierung

Das swissTLM3D enthält keine Informationen zum Röhrichtvorkommen. Auf dem SWISSIMAGE FCIR liessen sich mittels Vegetationsindizes vegetationsreichere Gebiete im See lokalisieren. Nach dem aktuellen Stand der Methode Seeufer soll bewusst kein Schwimmblatt erfasst werden. Da die Differenzierung zwischen verschiedenen emersen und submersen Pflanzen kaum möglich ist, fällt dieser Ansatz weg. Die standorttypische emerse Vegetation der Flachwasserzone wird in der vorliegenden Arbeit nicht weiter analysiert.

### 3.7.4 Strukturen / Anlagen in der Flachwasserzone

Strukturen und Anlagen in der Flachwasserzone stören submerse Makrophyten. Der Uferlinie kann lückenlos eine Ausprägung zugewiesen werden. Das Merkmal fliesst als Bonus-Malus-Attribut in die gesamthafte Bewertung des Ufers ein. Die verschiedenen Ausprägungen sind in **Tab. 22** aufgelistet.

**Tab. 22** Ausprägungen des Merkmals „Strukturen / Anlagen in der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
15	Strukturen / Anlagen in der Flachwasserzone	15.01 Flachwasserzone ohne Nutzung durch Strukturen / Anlagen	nicht beeintr.
		15.02 Flachwasserzone mit geringer Nutzung durch Strukturen / Anlagen	gering beeintr.
		15.03 Flachwasserzone mit mässiger Nutzung durch Strukturen / Anlagen	beeinträchtigt
		15.04 Flachwasserzone mit ausgeprägter Nutzung durch Strukturen / Anlagen	naturfern
		15.05 Flachwasserzone mit starker Nutzung durch Strukturen / Anlagen	naturfremd

- 15.01: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind
- 15.02 – 15.05: keine flächendeckend verfügbaren Informationen; Verbauung Mauer aus swissTLM3D als Hinweise

#### Vorgehen der Auswertung

Um die Stärke der Nutzung der Flachwasserzone beurteilen zu können müssten viele Informationen verfügbar sein, wie z.B. Badeanstalten, Bojen, Hafenanlagen oder Stege. Flächendeckend liegen aber kaum Daten vor. Zur automatisierten Bestimmung der Nutzung werden aus dem Landschaftsmodell swissTLM3D Mauern einbezogen. Da der Perimeter der Flachwasserzone 0.5m ab der Uferlinie beginnt, besteht keine Gefahr, dass die Ufermauer fälschlicherweise als Anlage im Flachwasser interpretiert wird. Stattdessen fallen Hafenbauten in den Beurteilungsbereich.

Die Mauern werden nach dem Verschnitt mit der Flachwasserzone auf die Uferlinie projiziert. Anlagen, die im swissTLM3D erfasst sind, sind mit grosser Wahrscheinlichkeit ein Zeichen für starke Nutzungen. Aus diesem Grund werden die betroffenen Gewässerabschnitte dem Typ 11.05 zugewiesen.

Der Python Code zur Auswertung der Strukturen und Anlagen in der Flachwasserzone ist in Anhang 3.10 zu finden.

### 3.7.5 Sohlenveränderung in der Flachwasserzone

Veränderungen in der Sohle lassen auf Störungen im Flachwasserbereich schliessen. Der Uferlinie kann lückenlos eine Ausprägung zugewiesen werden. Die Sohlenveränderung fliesst als Bonus-Malus-Attribut in die gesamthafte Bewertung des Ufers ein. Die verschiedenen Ausprägungen sind in **Tab. 23** aufgelistet.

**Tab. 23** Ausprägungen des Merkmals „Sohlenveränderung in der Flachwasserzone“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
16	Sohlenveränderungen in der Flachwasserzone	16.01 Keine Nutzung	nicht beeintr.
		16.02 Kiesvorschüttung standortgerecht	gering beeintr.
		16.03 Kiesvorschüttung als Aufwertungs- oder Schutzmassnahme	gering beeintr.
		16.04 Leichte Störung erkennbar	beeinträchtigt
		16.05 Ausbaggerungen geringfügig	beeinträchtigt
		16.06 Ausbaggerungen stark	naturfern
		16.07 Kiesvorschüttung standortfremd	naturfremd
		16.08 Künstliches Sohlsubstrat	naturfremd

Um allfällige Sohlenveränderungen in der Flachwasserzone mittels Analyse zu erkennen, müssten zwei Zeitstände eines Seebodenmodells vorhanden sein. Das Terrainmodell swissALTI3D wird periodisch nachgeführt, die Seeflächen sind jedoch mit dem mittleren Wasserstand modelliert. Flächendeckende Informationen zum Seeboden der Schweizer Seen sind im digitalen Seebodenmodell (DSM) der swisstopo vorhanden. Das DSM wurde aus den generalisierten Tiefenlinien der Schweizer Landeskarten 1:25'000 erstellt und wird nicht nachgeführt. Mangelhafte Genauigkeit und ungenügende Aktualität lassen keine Rückschlüsse auf die Sohlenveränderung in der Flachwasserzone zu. Das Merkmal wird in der vorliegenden Arbeit nicht weiter analysiert.

### 3.8 Beurteilungsraum Hinterland

Zwei Bonus-Malus-Attribute beschreiben den Zustand des Hinterlands.

Der Python Code zur Auswertung der Attribute ist in Anhang 3 zu finden.

#### 3.8.1 Hinterland-Übergangsvegetation

Das Hinterland wirkt positiv auf den ökomorphologischen Zustand, wenn unmittelbar an den Uferstreifen eine standorttypische Vegetation von mindestens 15 m Breite anschliesst. Das Vorkommen standorttypischer Hinterland-Übergangsvegetation soll nach aktuellem Stand der Methodenentwicklung (BAFU 2014a) mit vier Ausprägungen (17.01 bis 17.04) erfasst werden. Um Aussagen zur gesamten Uferlinie machen zu können, wird im Rahmen der vorliegenden Arbeit eine zusätzlich Ausprägung ergänzt (17.05). Das Attribut fließt als Zustand aufwertendes Attribut in die Uferbewertung ein. Die verschiedenen Ausprägungen sind in **Tab. 24** aufgelistet. Das Vorgehen bei der Auswertung ist vergleichbar mit dem der Attribute 7 und 8 (Standorttypische Ufervegetation und Ufersaum-Vegetation).

**Tab. 24** Ausprägungen des Merkmals „Hinterland-Übergangsvegetation“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
17	Hinterland-Übergangsvegetation	17.01 Übergangsvegetation standorttypisch Gehölz $\geq$ 15 m Breite	vorhanden
		17.02 Übergangsvegetation standorttypisch Röhricht / Ried $\geq$ 15 m	vorhanden
		17.03 Übergangsvegetation standorttypisch Gehölz $<$ 15 m Breite	fehlend
		17.04 Übergangsvegetation standorttypisch Röhricht / Ried $<$ 15 m Breite	fehlend
		17.05 Keine standorttypische Übergangsvegetation	fehlend

Flächendeckend verfügbare Geodaten zur Hinterland-Übergangsvegetation sind im swissTLM3D-Topic „Bodenbedeckung“ sowie in den Bundesinventaren der Flachmoore und Auen enthalten. Die Kategorien werden aus folgenden Inhalten extrahiert:

- 17.01: Wald, Wald offen und Gebüschwald aus der Bodenbedeckung
- 17.02: Bundesinventar der Flachmoore von nationaler Bedeutung, Bundesinventar der Auen von nationaler Bedeutung, Feuchtgebiete
- 17.03: Wald, Wald offen und Gebüschwald aus der Bodenbedeckung
- 17.04: Bundesinventar der Flachmoore von nationaler Bedeutung, Bundesinventar der Auen von nationaler Bedeutung, Feuchtgebiete
- 17.05: Uferabschnitte, die nicht den anderen Kategorien zugewiesen sind

#### Vorgehen der Auswertung

Die für die Auswertung relevanten Kategorien werden aus den Grundlagendaten extrahiert, zusammengefügt und mit einem 15 m-breiten Band (ab Grenze Beurteilungsraum Ufer) gestanzt.

Die Vegetation ist dann mindestens 15 m breit, wenn für einen bestimmten Bereich sowohl die Begrenzungslinie zum Uferstreifen als auch die 15 m entfernte Aussenbegrenzung identisch sind mit der Umrandung der ausgestanzten Vegetation. Ist nur die Begrenzungslinie zum

Uferstreifen identisch, so ist die Ufersaumvegetation schmäler als 15 m. Bei Überlappungen haben die Kategorien 17.02 und 17.04 Vorrang gegenüber 17.01 und 17.03.

Die Lokalisierung der identischen Begrenzungslinien auf der kilometrierten Uferlinie mit der herkömmlichen ArcGIS-Funktion „Locate Features along Routes“ ist fehleranfällig. Oft werden die Abschnitte zu kurz. Daher werden stattdessen Anfangs- und Endpunkt jeder Linie auf das Ufer projiziert.

Der Python Code zur Auswertung der Hinterland-Übergangsvegetation ist in Anhang 3.11 zu finden.

### 3.8.2 Hinterlandnutzung

Der Uferlinie kann lückenlos ein Nutzungsart bzw. eine Nutzungsstärke des Hinterlands zugewiesen werden. Die Hinterlandnutzung fließt als Bonus-Malus-Attribut in die gesamthafte Bewertung des Ufers ein. Die verschiedenen Ausprägungen sind in **Tab. 25** aufgelistet.

**Tab. 25** Ausprägungen des Merkmals „Hinterlandnutzung“ und dessen Klassifizierung (nach BAFU 2014a).

Nr.	Attribut	Ausprägung	Klassifizierung
18	Hinterlandnutzung	18.01 Keine Nutzung	nicht beeintr.
		18.02 Geringfügige Nutzung	gering beeintr.
		18.03 Mässige Nutzung (zwischen gering und ausgeprägt)	beeinträchtigt
		18.04 Ausgeprägte Nutzung	naturfern
		18.05 Sehr starke, ± flächendeckende Nutzung	naturfremd

Die Stärke der Hinterlandnutzung kann sich aus verschiedenen Faktoren zusammensetzen. Bei der Methodenentwicklung ist angedacht, dass neben Siedlung und Verkehrsanlagen auch die Freizeitnutzung sowie Land- und Forstwirtschaft in eine gemeinsame Nutzungsstärke integriert werden sollen. Sehr viele Informationen müssen gegeneinander abgewägt werden. Einige Inhalte wie Freizeitnutzung und Landwirtschaft, die mit einbezogen werden müssen, sind nicht in den verfügbaren Geodatenätzen vorhanden. Eine Automatisierung macht in diesem Fall keinen Sinn. In der vorliegenden Arbeit wird das Attribut Hinterlandnutzung nicht weiter analysiert.

### 3.9 Aggregierung

Bei der Erfassung der Ökomorphologie von Fließgewässern wird eine Abschnittsgrenze im Feld vom Kartierer festgelegt, sobald sich ein wichtiger Parameter ändert. Anschliessend werden für den definierten Abschnitt alle Merkmale erfasst, was die einzelnen Merkmale weniger scharf abgrenzt.

Im Rahmen der Erarbeitung von MESUBA wurde die Umsetzung von Abschnitten diskutiert. Dabei wurde festgelegt, dass jedes Attribut unabhängig von den anderen erfasst wird. Damit sind die Originaldaten weiterhin verfügbar und können zu einem späteren Zeitpunkt auf verschiedene Arten aggregiert werden. Dies ermöglicht einen flexibleren Umgang bei allfälligen Anpassungen der Bewertung und lässt gezielte Auswertungen einzelner Merkmale zu.

Um eine Gesamtaussage zu Uferabschnitten machen zu können, ist eine an die Kartierung anschliessende Aggregierung der Bewertungs- und Bonus-Malus-Attribute nötig. Denkbar sind zwei verschiedene Vorgehensarten:

- Feste Abschnittslängen  
Eine fixe Abschnittslänge wird passend zur Betrachtungstiefe festgelegt. Für jeden Abschnitt werden die einzelnen Zielhierarchien berechnet.
- Variable Abschnittslängen  
Ein oder mehrere Attribute werden als „Taktgeber“ ausgewiesen. Sobald die Ausprägung eines taktgebenden Attributs ändert, wird ein neuer Abschnitt gebildet.

Da verschiedene Bewertungskriterien nicht automatisiert aus verfügbaren Geodaten extrahierbar sind, ist eine Aggregierung der Attribute auf eine höhere Zielhierarchie nicht möglich.

## 4 Ergebnisse

Bisher wurde kein Schweizer See nach der neu entwickelten Methode kartiert. Für einen Bereich des Bielersees liegen Testkartierungen auf Basis der von Teiber et al. (2013) vorgeschlagenen ersten Version des minimalen Attributkatalogs vor. Das Ziel dieser Tests war die Abschätzung des Zeitaufwands bei der Erfassung am Bildschirm.

Da die Testkartierung auf einem älteren Stand der Attributliste basiert und ohne direkten räumlichen Bezug mit Hilfe eines Grafikprogramms erfolgte, eignen sich diese Daten nicht zur Überprüfung der automatisch extrahierten Information. Im Rahmen der vorliegenden Arbeit wurde für jeden der drei Seen im Untersuchungsgebiet je einen Bereich von 2 km manuell erfasst. Die manuelle Kartierung erfolgte ohne Einbezug der automatisch erhobenen Daten. Zur Klassierung wurden verschiedene Grundlagedaten konsultiert: Orthofotos, das Landschaftsmodell swissTLM3D, Schweizer Landeskarten, sowie das Terrainmodell swissALTI3D.

Für die Überprüfung der Qualität der Resultate wurden drei möglichst unterschiedliche Uferbereiche ausgewählt. In Kap. 4.1 bis Kap. 4.11 werden die automatisch gewonnenen Informationen mit den manuell erfassten Abschnitten verglichen. Es werden nur die Attribute überprüft, für die zumindest ein Teil der Ausprägungen extrahiert werden konnte.



Abb. 14 Die für die Überprüfung der automatisch erfassten Attribute gewählten Uferbereiche sind rot dargestellt. PK1000 im Massstab 1:300'000 (Quelle: Bundesamt für Landestopografie)

### 4.1 Ufertyp

Die Steilheit des Ufers kann mit Hilfe des Terrainmodells automatisch erfasst werden. Alle Ausprägungen sind mit geringem Rechenaufwand bestimmbar. Manuell wird das Ufer mit generierten Höhenkurven klassiert. In **Tab. 26** sind die manuell erfassten Kategorien den automatisch erhobenen gegenübergestellt.

**Tab. 26** Gegenüberstellung Attribut 3: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert				
		3.01	3.02	3.03	3.04	3.05
Manuell	3.01	-	-	-	-	-
	3.02	-	261	44	-	-
	3.03	-	24	1676	59	-
	3.04	-	-	361	2255	399
	3.05	-	-	-	220	701

Mit gut 80 % der Uferkilometer, die mit beiden Methoden gleich kategorisiert wurden, ist eine deutliche Übereinstimmung zu sehen. Bei den restlichen 20 % beträgt der Unterschied eine Klasse.

Die automatisiert erhobenen Daten sind zur Verwendung für die Bestimmung der Ökomorphologie von Seeufern geeignet.

## 4.2 Standorttypische Ufervegetation

Die standorttypische Ufervegetation wird aus dem Landschaftsmodell swissTLM3D und den Bundesinventaren extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos. In **Tab. 26** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 27** Gegenüberstellung Attribut 7: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert			
		7.01	7.02	7.03	7.04
Manuell	7.01	3510	21	-	-
	7.02	37	1342	-	-
	7.03	31	1023	35	-
	7.04	-	-	-	-

81 % der Uferkilometer sind manuell und automatisiert den gleichen Klassen zugewiesen. Trotz dieser sehr hohen Übereinstimmung fällt eine Zahl negativ auf. Gut ein Uferkilometer wurde automatisiert dem Typ 7.02 (Ried, Riedwiese, Flachmoor dominierend) zugewiesen, obwohl Typ 7.03 (Wald) dominiert. Dies hängt mit Überlagerungen mehrerer Bodenbedeckungen im Landschaftsmodell, wie z.B. Wald und Feuchtgebiet, sowie den zusätzlich integrierten Bundesinventaren zusammen. Die fehlerhafte Zuordnung zum Typ 7.02 ist mit der Festlegung, dass dieser Ausprägung Vorrang gegenüber 7.03 hat, zu erklären.

Die automatisiert erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

### 4.3 Ufersaum-Vegetation

Die standorttypische Ufersaum-Vegetation wird aus dem Landschaftsmodell swissTLM3D und den Bundesinventaren extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos. In **Tab. 28** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 28** Gegenüberstellung Attribut 8: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert				
		8.01	8.02	8.03	8.04	8.05
Manuell	8.01	31	1117	-	-	-
	8.02	-	1404	-	2	49
	8.03	-	-	-	-	-
	8.04	-	-	-	-	-
	8.05	-	-	-	-	3397

Bei der Ufersaum-Vegetation sind 81 % der Uferkilometer manuell und automatisiert der gleichen Klasse zugewiesen. Trotz dieser sehr hohen Übereinstimmung fällt eine Zahl negativ auf. Gut ein Uferkilometer wurde automatisiert dem Typ 8.02 (Röhricht / Ried) zugewiesen, obwohl Typ 8.01 (Gehölz) dominiert. Grund dafür ist wie bei Attribut 7, dass die verschiedenen Bodenbedeckungsarten in swissTLM3D und die Bundesinventare sich überschneiden. Im Fall einer Mehrfachklassierung wird Zugunsten von Röhricht / Ried klassiert. Auffällig ist auch, dass die sehr schmale Ufersaumvegetationen <3 m (8.03 und 8.04) nicht bzw. kaum vorkommen.

Die automatisiert erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

### 4.4 Uferverbau und künstliche Morphologie

Die Uferverbauung wird aus dem Landschaftsmodell swissTLM3D extrahiert. Es sind nur Mauern erfasst, die dem Typ 9.05 zugewiesen werden. Die restlichen Ausprägungen können nicht bestimmt werden. Die manuelle Erfassung erfolgt auf Basis von Orthofotos. Das Erkennen des Verbauungsmaterials und der Stärke ist schwierig und aufgrund von der Verdeckung durch Baumkronen nur bedingt möglich. Um eine zuverlässige Aussage zum Attribut 9 machen zu können, sind Schrägbilder zwingend nötig. In **Tab. 29** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 29** Gegenüberstellung Attribut 9: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert					
		9.01	9.02	9.03	9.04	9.05	9.06
Manuell	9.01	1965	-	-	-	-	-
	9.02	775	-	-	-	24	-
	9.03	-	-	-	-	71	-
	9.04	634	-	-	-	958	-
	9.05	220	-	-	-	1353	-
	9.06	-	-	-	-	-	-

Etwas mehr als 50 % der Uferkilometer sind manuell und automatisiert der gleichen Klasse zugewiesen. Dass das Resultat nicht schlechter ausfiel hängt mit dem sehr natürlichen Ufer des gewählten Ausschnitts am Neuenburgersee sowie dem teilweise stark verbauten Ufer der Ausschnitte von Murten- und Bielersee zusammen.

Die automatisiert erhobenen Daten sind nicht geeignet zur Verwendung bei der Bestimmung des ökomorphologischen Zustands.

#### 4.5 Fließgewässeranbindung (Mündung / Abfluss)

Die Fließgewässeranbindung wird aus den kantonalen Ökomorphologiedaten Stufe F sowie dem Landschaftsmodell swissTLM3D extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos und mit Hilfe der Schweizer Landeskarte. In **Tab. 30** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 30** Gegenüberstellung Attribut 10: manuell erfasst – automatisiert extrahiert. Angaben in Anzahl Fließgewässeranbindungen.

		Automatisiert					
		10.00	10.01	10.02	10.03	10.04	10.05
Manuell	10.01	-	1	-	-	-	-
	10.02	-	-	-	-	-	-
	10.03	1	-	-	5	-	-
	10.04	-	-	-	-	-	-
	10.05	-	-	-	-	-	7
	Keine Anb.	1	-	-	-	-	-

Mit rund 87 % Übereinstimmung zwischen manueller Kartierung und automatischer Bestimmung der Fließgewässeranbindung liegt ein gutes Resultat vor. Allerdings stützt sich dieser Prozentsatz auf nur 14 reale Mündungen. Trotzdem kann gesagt werden, dass mit den kantonalen Ökomorphologiedaten Stufe F ein Datensatz vorliegt, der für weitere Auswertungen geeignet ist. Sämtliche Ausprägungen lassen sich automatisiert extrahieren. Bei diesem Vorgehen werden Durchgangshindernisse einbezogen, sofern sie im Uferstreifen liegen. Oft liegt ein Absturz oder Bauwerk aber leicht zurückversetzt in Hinterland. Die Störung wirkt sich jedoch auf den See aus. Eine Erweiterung des Betrachtungsraums zur Unterscheidung zwischen Typ 10.01 und 10.02 sowie zwischen 10.03 und 10.04 muss im Rahmen der weiteren Methodentwicklung geprüft werden.

Aus dem swissTLM3D wurden zusätzlich zwei Anbindungen mit der Ausprägung 10.00 erfasst. Eine davon ist in den Grundlagedaten nicht ökomorphologisch kartiert, die andere wurde aufgrund starker geometrischer Abweichungen zwischen swissTLM3D und dem kantonalen Gewässernetz fälschlicherweise erfasst.

Die automatisiert erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden. Die aufgrund fehlender Ökomorphologiedaten nicht einer der fünf Klassen zugewiesene Anbindungen müssen manuell klassiert oder allenfalls gelöscht werden.

#### 4.6 Land- und forstwirtschaftliche Nutzung im Uferstreifen

Die land- und forstwirtschaftliche Nutzung wird aus dem Landschaftsmodell swissTLM3D extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos. In **Tab. 31** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 31** Gegenüberstellung Attribut 11: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert		
		11.01	11.02	11.03
Manuell	11.01	4634	21	11
	11.02	68	1037	-
	11.03	-	-	228

98 % der Uferkilometer sind sowohl manuell als auch automatisiert der gleichen Klasse zugewiesen. Dieses Resultat fällt viel zu positiv aus und hängt mit der Wahl der Testbereiche zusammen. Neben sumpfigen, nicht genutzten Gebieten sind Siedlung, Wald und Rebflächen vorherrschend. Diese Kategorien können aus den vorhandenen Geodaten einfach extrahiert werden. Sobald Landwirtschaftsflächen in den Bereich des Uferstreifens stossen, werden die Resultate aufgrund fehlender Geodaten schlechter.

Die automatisiert erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

#### 4.7 Siedlung, Gewerbe, Industrie im Uferstreifen

Siedlung, Gewerbe und Industrie wird aus dem Landschaftsmodell swissTLM3D extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos und mit Hilfe der Schweizer Landeskarte. In **Tab. 30** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 32** Gegenüberstellung Attribut 13: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert					
		13.01	13.02	13.03	13.04	13.05	13.06
Manuell	13.01	4985	33	32	111	-	-
	13.02	-	54	48	-	-	-
	13.03	7	-	78	-	-	-
	13.04	70	-	-	500	-	-
	13.05	-	-	-	-	-	-
	13.06	-	-	52	29	-	--

Die Übereinstimmung von 93 % hängt grösstenteils mit dem stark vertretenen Typ 13.01 zusammen. Trotzdem zeigt die visuelle Prüfung, dass Informationen zur Bebauungsdichte aus den Gebäudedaten gewonnen werden können. Einzig Typ 13.06 kann nicht ermittelt werden. Aus den Gebäudedaten ist nicht erkennbar, ob es sich um standortgebundene Uferbauten handelt.

Die automatisch erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

#### 4.8 Verkehrswege und –flächen im Uferstreifen

Aufgrund der grösseren Datenmenge der Verkehrswege und der verschiedenen Pufferschritte während der Analyse ist die Rechenzeit bei Attribut 14 grösser als bei den meisten anderen. Trotzdem ist eine Verarbeitung der drei Seen innerhalb von rund einer Stunde möglich. Die Verkehrsanlagen werden aus dem Landschaftsmodell swissTLM3D extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos, der Schweizer Landeskarten und swissTLM3D. In **Tab. 31** sind die manuell erfassten Kategorien den automatisch erhobenen gegenübergestellt.

**Tab. 33** Gegenüberstellung Attribut 14: manuell erfasst – automatisch extrahiert. Angaben in Metern.

		Automatisiert											
		14.01	14.02	14.03	14.04	14.05	14.06	14.07	14.08	14.09	14.10	14.11	
Manuell	14.01	3885	-	-	-	-	-	-	-	-	-	-	-
	14.02	-		-	-	-	-	-	-	-	-	-	-
	14.03	-	-	284	505	-	-	-	-	4	-	-	-
	14.04	-	-	-		-	-	-	-	-	-	-	-
	14.05	-	-	-	-		-	-	-	-	-	-	-
	14.06	-	-	-	-	-		-	-	-	-	-	-
	14.07	-	-	-	-	-	-		-	-	-	-	-
	14.08	21	-	65	364	-	-	-		12	-	-	-
	14.09	12	-	-	44	-	-	-	-		22	-	14
	14.10	-	-	-	-	-	-	-	-	-			-
	14.11	30	-	-	-	-	-	-	-	-	-	-	

Die Übereinstimmung von 82 % zeigt, dass die Differenzierung und der Detaillierungsgrad von swissTLM3D sehr gut auf die Anforderungen der Methode Ökomorphologie Seeufer passt. Die meisten Ausprägungen werden gut erfasst, eine Ausnahme bildet die Versiegelung bei Parkplätzen und schmalen Wegen. Eine Unterscheidung zwischen 14.02 und 14.03 sowie zwischen 14.05 und 14.07 ist mit dem swissTLM3D nicht möglich. Zudem bilden Uferpromenaden keine eigene Kategorie sondern werden entsprechend ihrer Breite den Weg- und Strassenkategorien zugewiesen.

Die automatisch erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

#### 4.9 Ausdehnung der Flachwasserzone

Die Ausdehnung der Flachwasserzone kann mit geringem Aufwand automatisiert berechnet werden. Die manuelle Klassierung erfolgt mit Hilfe von Orthofotos. In **Tab. 34** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 34** Gegenüberstellung Attribut 4: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert				
		4.01	4.02	4.03	4.04	4.05
Manuell	4.01	687	10	-	-	-
	4.02	103	523	317	-	-
	4.03	-	27	1027	186	-
	4.04	-	-	46	836	-
	4.05	-	-	-	13	2225

Mit rund 88 % der Uferkilometer, die mit beiden Methoden gleich kategorisiert wurden, ist eine deutliche Übereinstimmung zu sehen. Beschönigt wird das Bild dadurch, dass ein grosser Bereich der untersuchten sechs Kilometer sehr flach ist und in Kategorie 4.05 fällt. Bei den nicht gleich klassierten Abschnitten beträgt der Unterschied eine Klasse.

Die automatisiert erhobenen Daten sind zur Verwendung für die Bestimmung der Ökomorphologie von Seeufnern geeignet.

#### 4.10 Strukturen / Anlagen der Flachwasserzone

Die Nutzung in der Flachwasserzone wird soweit möglich automatisiert berechnet. Die manuelle Klassierung erfolgt mit Hilfe von Orthofotos und der Schweizer Landeskarte. In **Tab. 34** sind die manuell erfassten Kategorien den automatisiert erhobenen gegenübergestellt.

**Tab. 35** Gegenüberstellung Attribut 15: manuell erfasst – automatisiert extrahiert. Angaben in Metern.

		Automatisiert				
		15.01	15.02	15.03	15.04	15.05
Manuell	15.01	3658	-	-	-	-
	15.02	1281	-	-	-	-
	15.03	355	-	-	-	-
	15.04	290	-	-	-	-
	15.05	244	-	-	-	172

Nur punktuell fliessen Hafenmauern bei der GIS-Analyse ein und werden dem Typ 15.05 zugewiesen. Leichtere Nutzungen wie beispielsweise Badeflosse oder Pfahlbauten sind in den Geodatensätzen nicht vorhanden. Die Auswertung zeigt, dass eine Automatisierung aufgrund fehlender Grundlagedaten keinen Sinn macht. Von sechs Kilometern Uferlinie wurde auf nur 172 m eine Störung in der Flachwasserzone erkannt.

Die automatisch erhobenen Daten sind nicht geeignet zur Verwendung bei der Bestimmung des ökomorphologischen Zustands.

#### 4.11 Hinterland-Übergangsvegetation

Die standorttypische Hinterland-Übergangsvegetation wird aus dem Landschaftsmodell swissTLM3D und den Bundesinventaren extrahiert. Die manuelle Erfassung erfolgt auf Basis von Orthofotos. In **Tab. 36** sind die manuell erfassten Kategorien den automatisch erhobenen gegenübergestellt.

**Tab. 36** Gegenüberstellung Attribut 17: manuell erfasst – automatisch extrahiert. Angaben in Metern.

		Automatisiert				
		17.01	17.02	17.03	17.04	17.05
Manuell	17.01	40	1058	-	-	-
	17.02	-	1229	-	30	194
	17.03	47	-	-	2	-
	17.04	-	84	-	2	-
	17.05	-	-	-	-	3314

Auch bei der Ufersaumvegetation wurde mit 76 % ein grosser Anteil manuell und automatisch der gleichen Klasse zugewiesen. Trotz dieser sehr hohen Übereinstimmung fällt eine Zahl negativ auf. Gut ein Uferkilometer wurde automatisch dem Typ 17.02 (Röhricht / Ried) zugewiesen, obwohl Typ 17.01 (Gehölz) dominiert. Grund dafür ist wie bei den Attributen 7 und 8, dass die verschiedenen Bodenbedeckungsarten in swissTLM3D und die Bundesinventare sich überschneiden. Im Fall einer Mehrfachklassierung wird Zugunsten von Röhricht / Ried klassiert.

Die automatisch erhobenen Daten sind als unterstützendes Hilfsmittel für die manuelle Kartierung geeignet, sollten jedoch nicht ohne Prüfung übernommen werden.

## 5 Diskussion

Aus den Ergebnissen lassen sich folgende Antworten auf die in Kap. 1 gestellten Forschungsfragen geben:

*Lassen sich Merkmale zur Beschreibung des ökomorphologischen Zustands der Schweizer Seen aus flächendeckend verfügbaren Geodaten automatisiert extrahieren?*

Diese Frage muss differenziert beantwortet werden. Die Resultate der automatischen Extraktion sind durchaus positiv beim Ufertyp und bei der Ausdehnung der Flachwasserzone. Die gewünschten Ausprägungen dieser Kennwerte lassen sich aus dem Geländemodell swissALTI3D und aus dem Falschfarbenorthofoto SWISSIMAGE FCIR gewinnen. Bei sieben weiteren Merkmalen lassen sich zumindest einige Ausprägungen gewinnen. Es handelt sich dabei um folgende Attribute: Standorttypische Ufervegetation; Ufersaum-Vegetation; Fliessgewässeranbindung; Land- und forstwirtschaftliche Nutzung im Uferstreifen; Siedlung, Gewerbe, Industrie im Uferstreifen; Verkehrswege und -flächen im Uferstreifen; Hinterland-Übergangsvegetation. Bei den restlichen neun Attributen ist eine automatisierte Voranalyse nicht möglich oder lohnt sich aufgrund der schlechten Grundlagesituation nicht.

*Wie gross müssen die Beurteilungsräume gewählt werden?*

Die meisten Kriterien betreffen den unmittelbar an die Mittelwasserlinie grenzenden Raum. Der zentrale Beurteilungstreifen wurde im Rahmen der Methodenerarbeitung auf 15 m festgesetzt. Damit ist er gleich gross wie der gesetzlich festgelegte minimale Gewässerraum (GSchV Art. 41b). Im Laufe der Methodenerarbeitung wurde der seeseitige Raum mit der Haldenkante begrenzt.

Zurzeit liegt noch keine methodenbedingte Festlegung zur Grösse des Hinterlands vor. Der Ansatz von einer festen Breite ab dem Beurteilungstreifen Ufer, sofern die Höhendifferenz zum mittleren Seespiegel nicht zu gross ist, hat sich bewährt.

*Welche verfügbaren Geodaten eignen sich zur automatisierten Extraktion eines oder mehrerer Attribute zur Beschreibung des ökomorphologischen Zustands der Seen?*

Es stellte sich heraus, dass vor allem das Landschaftsmodell swissTLM3D einen grossen Nutzen bringt. Neben Informationen zu Verkehrswegen lassen sich kombiniert mit den Bundesinventaren Aussagen zur Vegetation machen.

Zur Bestimmung des Ufertyps dient das Geländemodell swissALTI3D.

Die Ökomorphologie Stufe F liefert gute Hinweise zur Fliessgewässeranbindung. In lückenhaften Gebieten kann das Landschaftsmodell swissTLM3D als Ergänzung beigezogen werden.

Indirekt verwendbar zur Merkmalsgewinnung ist das Falschfarbenbild SWISSIMAGE FCIR. Mit der Bestimmung des Beurteilungsraums Flachwasser ist die Ausdehnung der Flachwasserzone berechenbar.

*Welche verfügbaren Geodaten eignen sich nicht zur Beschreibung des ökomorphologischen Zustands der Seen?*

Es hat sich gezeigt, dass das Oberflächenmodell DOM keinen Mehrwert bietet. Für die automatisierte Bestimmung der Ökomorphologie-Merkmale ist auch das Falschfarbenbild SWISSIMAGE FCIR nicht geeignet, da die Ausprägungen der Merkmale oft nur geringe Unter-

schiede aufweisen, die mittels Multispektralanalyse nicht erkannt werden können. Die Arealstatistikdaten des Bundesamts für Statistik weisen eine zu grobe Auflösung auf (ha-Raster) und wurden von vornherein nicht in die Analysen einbezogen.

*Ist der vorgeschlagene Lösungsansatz praxistauglich?*

Aufgrund der benötigten zeitlichen und materiellen Ressourcen ist der gewählte Ansatz praxistauglich. Es müssen keine zusätzlichen Daten erhoben werden. Mit der Verwendung von Datensätzen, die flächendeckend verfügbar sind, beschränkt sich die Anzahl der Datenherren auf Bundesämter. Der sonst oft grosse zeitliche Aufwand der Datenbeschaffung ist in diesem Fall gering. Auch der Aufwand für die Aufbereitung der Daten hält sich in Grenzen. Neben der Transformation in einen einheitlichen Bezugsrahmen fallen Arbeiten beim Zusammenführen der Daten an. Das Landschaftsmodell swissTLM3D ist blattschnittfrei und kann wie die Bundesinventare und Ökomorphologiedaten der Fließgewässer direkt verwendet werden. Beim Geländemodell swissALTI3D und dem Falschfarbenbild SWISSIMAGE FCIR müssen erst viele Kacheln zusammengeführt werden. Vor allem beim Falschfarbenbild ist der Rechenaufwand gross und die Bearbeitung muss aufgrund der umfangreichen Datenmenge eingeschränkt werden. Trotzdem können alle Analysen mit einem gängigen Notebook durchgeführt werden.

Das gewählte Vorgehen mit der Berechnung der Beurteilungsräume sowie der automatisierten Erstellung einiger Attribute-Eventlayer hat sich bewährt.

## 6 Schlussfolgerung und Ausblick

Bei der Bearbeitung der vorliegenden Arbeit war die Verwendung der stets aktuellsten Attributliste hinderlich. Obwohl die Liste mit Ausprägungen und Definitionen immer verbessert wurde, hatte es für die Analyse einen grossen Mehraufwand bei den Anpassungen der Skripts zur Folge. Besser wäre die Festsetzung des ersten Vorschlags von Teiber et al. (2013) gewesen. Erst nach Veröffentlichung der Methode „Ökomorphologie Seeufer“ macht eine definitive Umsetzung Sinn.

Aus den Erfahrungen im Rahmen der vorliegenden Arbeit wird klar, dass die Festlegung der Uferlinie eine zentrale Rolle spielt. Sie definiert die Beurteilungstreifen und bildet die Bezugslinie für die Attribute. Ein homogener aber trotzdem aktueller Datensatz für die ganze Schweiz wäre wünschenswert. Die bestehenden Landschaftsmodelle swissTLM3D und VECTOR25 erfüllen diese Forderungen beide nicht.

Für die Bestimmung einzelner Attribute eignen sich die vorhandenen Datensätze durchaus. Auch wenn bei vielen Attributen nur ein Teil der Ausprägungen extrahiert werden kann, wird der Kartierer davon profitieren können, was die manuelle Bearbeitung effizienter gestalten wird. Ein wesentlicher Punkt als nächster Schritt ist die Erstellung eines Erfassungstools. Idealerweise werden die für die automatisierte Auswertung empfohlenen Attribute vorgerechnet und in das Erfassungstool integriert. Damit stünden die Daten für Korrekturen zur Verfügung. Neben der effizienteren Bearbeitung wäre die Chance, dass schweizweit auf einer einheitlichen geometrischen Basis mit möglichst gleichem Detaillierungsgrad erfasst würde, intakt.

Aus bisherigen Kartierungen wird klar, dass feste Abschnittslängen ein ruhigeres, einfacher interpretierbares Bild geben. Für Aufgaben mit einer klaren Fokussierung auf einen Teilaspekt der Ökomorphologie Seeufer sind variable Abschnittslängen mit einem taktgebenden Attribut besser geeignet. Es ist denkbar, dass die unterschiedlichen Ansätze in das Erfassungstool integriert und den verschiedenen Interessensgruppen zur Verfügung gestellt werden.

Die nächsten Schritte der Methodenentwicklung und Umsetzung werden einem sehr straffen Zeitplan folgen. Im Hinterkopf ist die Revitalisierungsplanung der Seeufer, die bis Ende 2018 abgeschlossen sein soll.

## Literaturverzeichnis

- Bini A.; Buoncristiani J.-F.; Couterrand S.; Ellwanger D.; Felber M.; Florineth D.; Graf H. R.; Keller O.; Kelly M.; Schlüchter C.; Schoeneich P. (2009): Die Schweiz während ihres letzteiszeitlichen Maximums (LGM) 1:500'000. Bundesamt für Landestopografie swisstopo, Bern.
- BAFU (2014a): Methoden der Seeuferbeurteilung und –aufwertung (MESUBA). Unterlagen Attributliste vom 27.04.2014, unveröffentlicht.
- BAFU (2014b): Verfügbare Geodaten im BAFU. URL: <http://www.bafu.admin.ch/gis/02911/index.html?lang=de> [Abruf: 2014-05-30].
- BFS (2009a): Arealstatistik 2004/2009. Bundesamt für Statistik, Neuchâtel.
- BFS (2009b): Statistisches Lexikon der Schweiz: Die grössten Seen und höchsten Berge (je-d-02.02.01.01). Bundesamt für Statistik, Neuchâtel.
- BUWAL (1998a): Modul-Stufen-Konzept. Methoden zur Untersuchung und Beurteilung der Fliessgewässer in der Schweiz. Bundesamt für Umwelt, Wald und Landschaft, Bern.
- BUWAL (1998b): Ökomorphologie Stufe F (flächendeckend). Methoden zur Untersuchung und Beurteilung der Fliessgewässer in der Schweiz. Bundesamt für Umwelt, Wald und Landschaft, Bern.
- Eawag (2014): Modul-Stufen-Konzept. URL: <http://www.modul-stufen-konzept.ch/fg/> [Abruf: 2014-05-30].
- Labhart T. (2009): Geologie der Schweiz. hep Verlag, Bern.
- Nast M. (2006): Überflutet – überlebt – überlistet. Die Geschichte der Juragewässerkorrektion. Verein Schlossmuseum Nidau, Nidau.
- Rey P.; Teiber P.; Huber M. (2009): Renaturierungsleitfaden Bodenseeufers, IGKB, Bregenz.
- Roth U.; Schwick Ch.; Spichtig F. (2010): Zustand der Landschaft in der Schweiz. Zwischenbericht Landschaftsbeobachtung Schweiz (LABES). Umwelt-Zustand Nr. 1010. Bundesamt für Umwelt, Bern.
- Schlosser J. A.; Haertel-Borer S.; Liechti P.; Reichert P. (2013): Konzept für die Untersuchung und Beurteilung der Seen in der Schweiz. Anleitung zur Entwicklung und Anwendung von Beurteilungsmethoden. Umwelt-Wissen Nr. 1326. Bundesamt für Umwelt, Bern.
- Sigmaplan (2011): Ökomorphologie Wohlensee: Kartier- und Methodenbericht. Amt für Wasser und Abfall des Kantons Bern.
- Spreafico M.; Weingartner R. (2005): Hydrologie der Schweiz – Ausgewählte Aspekte und Resultate. Berichte des BWG, Serie Wasser Nr. 7, Bern.
- swisstopo (2012): Objektkatalog swissTLM3D 1.1. Bundesamt für Landestopografie swisstopo, Bern.
- swisstopo (2014a): swissTLM3D. URL: <http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/landscape/swissTLM3D.html> [Abruf: 2014-05-30].

swisstopo (2014b): swissALTI3D. URL: <http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/height/swissALTI3D.html> [Abruf: 2014-05-30].

swisstopo (2014c): DOM. URL: [http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/height/dom\\_dtm-av.html](http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/height/dom_dtm-av.html) [Abruf: 2014-05-30].

swisstopo (2014d): SWISSIMAGE FCIR. URL: [http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/images/ortho/swissimage/SWISSIMAGE\\_FCIR.html](http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/images/ortho/swissimage/SWISSIMAGE_FCIR.html) [Abruf: 2014-05-30].

Teiber P. (2010): Seeuferbewertung Vierwaldstättersee 2008. Aufsichtskommission Vierwaldstättersee.

Teiber P.; Niederberger K.; Rey P. (2013): Methoden der Seeuferbeurteilung und –aufwertung, Vorstudie. Bericht im Auftrag des Bundesamts für Umwelt, Bern.

Zeh Weissmann H.; Könitzer C.; Bertiller A. (2009): Strukturen der Fließgewässer in der Schweiz. Zustand von Sohle, Ufer und Umland (Ökomorphologie). Umwelt-Zustand Nr. 0926. Bundesamt für Umwelt, Bern.

### **Rechtliche Grundlagen**

BV: Bundesverfassung der Schweizerischen Eidgenossenschaft vom 18. April 1999. SR 101.

GeolG: Bundesgesetz vom 5. Oktober 2007 über Geoinformation (Geoinformationsgesetz, GeolG). SR 510.62.

GeolV: Verordnung vom 21. Mai 2008 über Geoinformation (Geoinformationsverordnung, GeolV). SR 510.620.

GSchG: Bundesgesetz vom 24. Januar 1991 über den Schutz der Gewässer (Gewässerschutzgesetz, GSchG). SR 814.20.

GSchV: Gewässerschutzverordnung vom 24. Januar 1991. SR 814.201.

NHG: Bundesgesetz vom 1. Juli 1966 über den Natur- und Heimatschutz (NHG). SR 451.

## Anhang 1: Minimale Attributliste Ökomorphologie Seeufer

**Tab. 37** Minimale Attributliste zur Seeuferbeurteilung (nach BAFU 2014a). Der Wirkungstyp beschreibt die Verwendbarkeit des Attributs für die Beurteilung: Zusatzinformation (z), Hauptattribut für Bewertung (b<sub>1</sub>), Bonus-Malus-Attribut (b<sub>2</sub>) oder übergeordneter Einfluss (e).

Nr.	Attribut	Ausprägung	Wirkungstyp
1	Typ stehende Gewässer	1.01 Bergsee oberhalb der Baumgrenze 1.02 Bergsee unterhalb der Baumgrenze 1.03 Voralpensee gross 1.04 Voralpensee klein 1.05 Mittellandsee gross 1.06 Mittellandsee klein 1.07 Kleingewässer (Weiher) 1.08 Teich 1.09 Stausee 1.10 Flusstau (aufgestautes Fließgewässer)	e
2	Wasserspiegelregulierung	2.01 Wasserspiegel nicht reguliert 2.02 Wasserspiegel statisch reguliert (Pegel wenig schwankend) 2.03 Wasserspiegel dynamisch reguliert (Pegel stark schwankend) 2.04 Wasserspiegel historisch verändert - nicht reguliert 2.05 Wasserspiegel historisch verändert - reguliert	e
3	Ufertyp	3.01 Sehr steiles Ufer (Neigung $\geq 1:2.5$ ) 3.02 Steilufer (Neigung $< 1:2.5 - 1:5$ ) 3.03 Mittelsteiles Ufer (Neigung $< 1:5 - 1:12.5$ ) 3.04 Flachufer (Neigung $< 1:12.5 - 1:25$ ) 3.05 Sehr flaches Ufer (Neigung $< 1:25$ )	e
4	Ausdehnung der Flachwasserzone (bis 4 m Wassertiefe)	4.01 Sehr schmale Flachwasserzone (Breite $\leq 10$ m) 4.02 Schmale Flachwasserzone (Breite $> 10$ m $\leq 50$ m) 4.03 Mittelbreite Flachwasserzone (Breite $> 50$ m $\leq 100$ m) 4.04 Breite Flachwasserzone (Breite $> 100$ m $\leq 250$ m) 4.05 Sehr breite, ausgedehnte Flachwasserzone (Breite $> 250$ m)	e
5	Wellenexposition	5.01 Bereich starkem Wellenschlag ausgesetzt 5.02 Bereich mittlerem Wellenschlag ausgesetzt 5.03 Bereich geringem Wellenschlag ausgesetzt 5.04 Keine Informationen vorhanden	e
6	Standorttypische emerse Vegetation der Flachwasserzone	6.01 Kein emerser Bewuchs 6.02 Röhrichsaum $< 5$ m Breite 6.03 Röhrichtband $5 - 20$ m Breite 6.04 Röhrichtfläche $> 20$ m Breite	z
7	Standorttypische Ufervegetation	7.01 Standorttypische Ufervegetation fehlend 7.02 Typische Sukzession (Ried, Riedwiese, Flachmoor dominierend) 7.03 Typische Sukzession (Uferwald dominierend) 7.04 Standorttypische Vegetation oberhalb der Baumgrenze	z
8	Ufersaum-Vegetation	8.01 Ufersaum-Vegetation standorttypisch Gehölz $\geq 3$ m Breite 8.02 Ufersaum-Vegetation standorttypisch Röhricht / Ried $\geq 3$ m Breite 8.03 Ufersaum-Vegetation standorttypisch Gehölz $< 3$ m Breite 8.04 Ufersaum-Vegetation standorttypisch Röhricht / Ried $< 3$ m Breite	b <sub>1</sub>
9	Uferverbau und künstliche Morphologie	9.01 Keine Uferverbauungen 9.02 Geringfügige Uferverbauungen (geringe Spuren) 9.03 Kleinere Uferverbauungen (deutliche Spuren) 9.04 Hartverbau durchlässig 9.05 Hartverbau undurchlässig 9.06 Regenwasserentlastung	b <sub>1</sub>

Nr.	Attribut	Ausprägung	Wirkungstyp
10	Fliessgewässeranbindung (Mündung / Abfluss)	10.01 Anbindung naturnah ohne Aufstiegshindernis 10.02 Anbindung naturnah mit Aufstiegshindernis 10.03 Anbindung verbaut ohne Aufstiegshindernis 10.04 Anbindung verbaut mit Aufstiegshindernis 10.05 Anbindung verdolt	b <sub>1</sub>
11	Land- und forstwirtschaftliche Nutzung im Uferstreifen	11.01 Keine Nutzungen 11.02 Extensiv genutzte Land- und Forstwirtschaftsflächen 11.03 Intensiv genutzte Land- und Forstwirtschaftsflächen	b <sub>1</sub>
12	Freizeit- und Privatnutzungen im Uferstreifen	12.01 Keine Nutzungen 12.02 Leichte Nutzungsspuren durch Freizeitbetrieb 12.03 Extensive Privat- und Freizeitnutzungen 12.04 Intensive Privat- und Freizeitnutzungen 12.05 Sehr starke Privat- und Freizeitnutzungen	b <sub>1</sub>
13	Siedlung, Gewerbe, Industrie im Uferstreifen	13.01 Keine Uferbauwerke 13.02 Vereinzelte Kleingebäude (Hütte / Pavillon / kleines Nebengebäude) 13.03 Freistehende Einzelbauwerk 13.04 Lückige Bebauung (versiegelte Fläche ≤ Freifläche) 13.05 Dichte Bebauung (versiegelte Fläche >> Freifläche) 13.06 Standortgebundene Uferbauten / Uferanlagen landseitig	b <sub>1</sub>
14	Verkehrswege und -flächen im Uferstreifen	14.01 Keine Verkehrsinfrastruktur 14.02 Unbefestigter, ungeteeter schmaler Pfad / Fussweg / Veloweg 14.03 Befestigter, geteeter schmaler Pfad / Fussweg / Veloweg 14.04 Unbefestigter, ungeteeter Fahrweg 14.05 Unversiegelte Parkplatzfläche 14.06 Unbefestigte, ungeteerte Uferpromenade / Quaianlage 14.07 Versiegelte Parkplatzfläche 14.08 Befestigte, geteerte Uferpromenade / Quaianlage 14.09 Befestigte, geteerte Strasse (Fahrweg bis Kantonsstrasse) 14.10 Befestigte, geteerte Strasse (Autobahn) 14.11 Bahntrasse	b <sub>1</sub>
15	Strukturen / Anlagen in der Flachwasserzone	15.01 Flachwasserzone ohne Nutzung durch Strukturen / Anlagen 15.02 Flachwasserzone mit geringer Nutzung durch Strukturen / Anlagen 15.03 Flachwasserzone mit mässiger Nutzung durch Strukturen / Anlagen 15.04 Flachwasserzone mit ausgeprägter Nutzung durch Strukturen / Anlagen 15.05 Flachwasserzone mit starker Nutzung durch Strukturen / Anlagen	b <sub>1</sub>
16	Sohlenveränderungen in der Flachwasserzone	16.01 Keine Nutzung 16.02 Kiesvorschüttung standortgerecht 16.03 Kiesvorschüttung als Aufwertungs- oder Schutzmassnahme 16.04 Leichte Störungen erkennbar 16.05 Ausbaggerungen geringfügig 16.06 Ausbaggerungen stark 16.07 Kiesvorschüttungen standortfremd 16.08 Künstliches Sohlsubstrat	b <sub>2</sub>
17	Hinterland-Übergangsvegetation	17.01 Übergangsvegetation standorttypisch Gehölz ≥ 15 m Breite 17.02 Übergangsvegetation standorttypisch Röhricht / Ried ≥ 15 m Breite 17.03 Übergangsvegetation standorttypisch Gehölz < 15 m Breite 17.04 Übergangsvegetation standorttypisch Röhricht / Ried < 15 m Breite 17.05 Keine standorttypische Übergangsvegetation	b <sub>2</sub>

Nr.	Attribut	Ausprägung	Wirkungs- typ
18	Hinterlandnutzung	18.01 Keine Nutzung 18.02 Geringfügige Nutzung 18.03 Mässige Nutzung (zwischen gering und ausgeprägt) 18.04 Ausgeprägte Nutzung 18.05 Sehr starke, ± flächendeckende Nutzung	b <sub>2</sub>

## Anhang 2: Erstellung Beurteilungsräume (Python-Skript)

```

#####
# 1. Beurteilungsraum Ufer: 15m-Pufferband uferseitig, 0.5m-Pufferband seeseitig
# 2. Beurteilungsraum Flachwasserzone: 0.5m bis 5m seeseitig als Minimum
# 3. Beurteilungsraum Hinterland: 15m bis 35m uferseitig als Minimum, bis 65m als
    Maximum
#####

# Libraries/allgemeine Einstellungen
import arcpy, os
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
env.overwriteOutput = True
arcpy.env.outputMFlag = "Disabled"
arcpy.env.outputZFlag = "Disabled"
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
path = os.path.abspath("../")
path = path.replace("\\", r"\\")
inputPath = path + "\\20_bearbeitete_Daten\\"
outputPath = path + "\\20_bearbeitete_Daten\\30_Raeume\\"

# Listen/Variablen
Seespiegel = 429
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (temp.gdb wird am Ende geloescht)
arcpy.CreateFileGDB_management(outputPath, "raeume.gdb", "10.0")
arcpy.CreateFileGDB_management(outputPath, "temp.gdb", "10.0")

#####
# 1. Beurteilungsraum Ufer
#   Resultate: Pro See drei Feature Classes
#   - ufer_see (Beurteilungsraum Ufer)
#   - ufer_see_aussen_1 (Aussenlinien Beurteilungsraum Ufer)
#   - ufer_see_innen_1 (Innenlinien Beurteilungsraum Ufer)

try:
    for See in listeSeen:
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\ufer_" + See + "_aussen", "15")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\ufer_" + See + "_aussen",
            outputPath + "raeume.gdb\\ufer_" + See + "_aussen_1")
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\ufer_" + See + "_innen", "-0.5")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\ufer_" + See + "_innen",
            outputPath + "raeume.gdb\\ufer_" + See + "_innen_1")
        arcpy.Erase_analysis(outputPath + "temp.gdb\\ufer_" + See + "_aussen",
            outputPath + "temp.gdb\\ufer_" + See + "_innen", outputPath +
            "raeume.gdb\\ufer_" + See)
except:
    print "Ein Fehler beim Erstellen der Beurteilungsraeume Ufer aufgetreten."
    print arcpy.GetMessages()

print "Beurteilungsraeume Ufer erstellt."

#####
# 2. Beurteilungsraum Flachwasser
#   Resultate: Pro See drei Feature Classes
#   - flachwasser_see (Beurteilungsraum Flachwasser)
#   - flachwasser_see_aussen_1 (Aussenlinien Beurteilungsraum Flachwasser)
#   - flachwasser_see_innen_1 (Innenlinien Beurteilungsraum Flachwasser)

try:
    for See in listeSeen:

```

```

arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
    "_f", outputPath + "temp.gdb\\flachwasser_" + See + "_aussen", "-0.5")
arcpy.FeatureToLine_management(outputPath + "temp.gdb\\flachwasser_" + See +
    "_aussen", outputPath + "raeume.gdb\\flachwasser_" + See + "_aussen_1")
arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
    "_f", outputPath + "temp.gdb\\flachwasser_" + See + "_innen_min", "-5")
outReclass = Reclassify(inputPath + "20_Geodaten\\FCIR.gdb\\" + See + "_gruen",
    "Value", RemapRange([[0,3300,1],[3300,100000, "NODATA"]]), "NODATA")
outReclass.save(outputPath + "temp.gdb\\flachwasser_R_" + See + "_innen_max")
arcpy.RasterToPolygon_conversion(outputPath + "temp.gdb\\flachwasser_R_" + See +
    "_innen_max", outputPath + "temp.gdb\\flachwasser_" + See + "_innen_max_holes",
    "SIMPLIFY", "VALUE")
arcpy.EliminatePolygonPart_management(outputPath + "temp.gdb\\flachwasser_" +
    See + "_innen_max_holes", outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen_max_many", "AREA", 100000, "", "ANY")
arcpy.Select_analysis(outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen_max_many", outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen_max_detail", '"Shape_Area" > 100000')
arcpy.SmoothPolygon_cartography(outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen_max_detail", outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen_max", "PAEK", 50)
arcpy.Clip_analysis(outputPath + "temp.gdb\\flachwasser_" + See + "_innen_max",
    outputPath + "temp.gdb\\flachwasser_" + See + "_innen_min", outputPath +
    "temp.gdb\\flachwasser_" + See + "_innen")
arcpy.FeatureToLine_management(outputPath + "temp.gdb\\flachwasser_" + See +
    "_innen", outputPath + "raeume.gdb\\flachwasser_" + See + "_innen_1")
arcpy.Erase_analysis(outputPath + "temp.gdb\\flachwasser_" + See + "_aussen",
    outputPath + "temp.gdb\\flachwasser_" + See + "_innen", outputPath +
    "raeume.gdb\\flachwasser_" + See)
except:
    print "Ein Fehler beim Erstellen der Beurteilungsraeume Flachwasser aufgetreten."
    print arcpy.GetMessages()

print "Beurteilungsraeume Flachwasser erstellt."

#####
# 3. Beurteilungsraum Hinterland
# Resultate: Pro See drei Feature Classes
# - hinterland_see (Beurteilungsraum Hinterland)
# - hinterland_see_aussen_1 (Aussenlinien Beurteilungsraum Hinterland)
# - hinterland_see_innen_1 (Innenlinien Beurteilungsraum Hinterland)

try:
    for See in listeSeen:
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\hinterland_" + See + "_innen", "15")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\hinterland_" + See +
            "_innen", outputPath + "raeume.gdb\\hinterland_" + See + "_innen_1")
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\hinterland_" + See + "_aussen_50", "65")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\hinterland_" + See +
            "_aussen_50", outputPath + "temp.gdb\\hinterland_" + See + "_aussen_50_1")
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\hinterland_" + See + "_aussen_20", "35")
        arcpy.sa.Contour (inputPath + "20_Geodaten\\DTM.gdb\\" + See + "_DTM",
            outputPath + "temp.gdb\\hinterland_" + See + "_hoehe_20_1", 1000, Seespiegel +
            20)
        arcpy.Merge_management([outputPath + "temp.gdb\\hinterland_" + See +
            "_aussen_50_1", outputPath + "temp.gdb\\hinterland_" + See + "_hoehe_20_1"],
            outputPath + "temp.gdb\\hinterland_" + See + "_hoehe_merge")
        arcpy.FeatureToPolygon_management(outputPath + "temp.gdb\\hinterland_" + See +
            "_hoehe_merge", outputPath + "temp.gdb\\hinterland_" + See + "_hoehe_20_merge")
        arcpy.Union_analysis ([outputPath + "temp.gdb\\hinterland_" + See +
            "_aussen_50", outputPath + "temp.gdb\\hinterland_" + See + "_aussen_20",
            outputPath + "temp.gdb\\hinterland_" + See + "_hoehe_20_merge"], outputPath +
            "temp.gdb\\hinterland_" + See + "_union", "ONLY FID")
        arcpy.Select_analysis(outputPath + "temp.gdb\\hinterland_" + See + "_union",
            outputPath + "temp.gdb\\hinterland_" + See + "_aussen_multi",
            '"FID_Hinterland_' + See + '_aussen_20" = 1 OR ("FID_Hinterland_' + See +
            '_aussen_50" = 1 AND "FID_Hinterland_' + See + '_hoehe_20_merge" > -1)')
        arcpy.Dissolve_management(outputPath + "temp.gdb\\hinterland_" + See +
            "_aussen_multi", outputPath + "temp.gdb\\hinterland_" + See + "_aussen")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\hinterland_" + See +
            "_aussen", outputPath + "raeume.gdb\\hinterland_" + See + "_aussen_1")

```

```
        arcpy.Erase_analysis(outputPath + "temp.gdb\\hinterland_" + See + "_aussen",
        outputPath + "temp.gdb\\hinterland_" + See + "_innen", outputPath +
        "raeume.gdb\\hinterland_" + See)
except:
    print "Ein Fehler beim Erstellen der Beurteilungsraeume Hinterland aufgetreten."
    print arcpy.GetMessages()

print "Beurteilungsraeume Hinterland erstellt."

arcpy.Delete_management(outputPath + "temp.gdb")
```

## Anhang 3: Auswertung der einzelnen Attribute (Python-Skript)

### Anhang 3.01: Ufertyp

```
# Libraries/allgemeine Einstellungen
import arcpy, os
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
path = os.path.abspath("../")
path = path.replace("\\", r"\")
inputPath = path + "\\20_bearbeitete_Daten\\"
outputPath = path + "\\20_bearbeitete_Daten\\40_Attribute\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']
Seespiegel = 429

# file geodatabase erstellen (temp.gdb wird am Ende geloescht)
arcpy.CreateFileGDB_management(outputPath, "attribute.gdb", "10.0")
arcpy.CreateFileGDB_management(outputPath, "temp.gdb", "10.0")

#####
# Ufertyp
# Resultate: Ein Event Table pro See
# - Attr03 (Attribut Ufertyp Nr. 3)

try:
    for See in listeSeen:
        arcpy.gp.ET_GPStationPoints(inputPath + "10_Perimeter\\ufer_v.gdb\\general_" +
            See, outputPath + "temp.gdb\\innen_10m_" + See, 10)
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", outputPath + "temp.gdb\\ufer_" + See + "_25m", "25")
        arcpy.FeatureToLine_management(outputPath + "temp.gdb\\ufer_" + See + "_25m",
            outputPath + "temp.gdb\\ufer_" + See + "_25m_1")
        arcpy.gp.ET_GPSnapPoints(outputPath + "temp.gdb\\innen_10m_" + See, outputPath +
            "temp.gdb\\ufer_" + See + "_25m_1", outputPath + "temp.gdb\\aussen_10m_noZ_" +
            See, 50, False, True)
        arcpy.sa.ExtractValuesToPoints(outputPath + "temp.gdb\\aussen_10m_noZ_" + See,
            inputPath + "20_Geodaten\\DTM.gdb\\" + See + "_DTM", outputPath +
            "temp.gdb\\aussen_10m_" + See)
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See, "RID",
            "SHORT")
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See, "VON",
            "LONG")
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See, "BIS",
            "LONG")
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See,
            "Gefaelle", "DOUBLE")
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See, "Typ",
            "DOUBLE")
        arcpy.AddField_management(outputPath + "temp.gdb\\aussen_10m_" + See,
            "Schluessel", "TEXT", "", "", 10)
        rows = arcpy.UpdateCursor(outputPath + "temp.gdb\\aussen_10m_" + See)
        for row in rows:
            row.RID = 1
            row.VON = row.ET_STATION - 5
            row.BIS = row.ET_STATION + 5
            row.Schluessel = str(row.ET_STATION)[-2]
            rows.updateRow(row)
        del row
```

```
del rows
arcpy.Merge_management([outputPath + "temp.gdb\\aussen_10m_" + See, outputPath +
"temp.gdb\\innen_10m_" + See], outputPath + "temp.gdb\\punkte_10m_" + See)
arcpy.gp.ET_GPPointToPolyline(outputPath + "temp.gdb\\punkte_10m_" + See,
outputPath + "temp.gdb\\linien_10m_" + See, "ET_STATION")
arcpy.JoinField_management(outputPath + "temp.gdb\\aussen_10m_" + See,
"Schluessel", outputPath + "temp.gdb\\linien_10m_" + See, "ET_ID")
rows = arcpy.UpdateCursor(outputPath + "temp.gdb\\aussen_10m_" + See)
for row in rows:
    row.Gefaelle = round(((row.RASTERVALU - Seespiegel)/row.Shape_Length*100),
1)
    if row.Gefaelle >= 40:
        row.Typ = 3.01
    elif row.Gefaelle >= 20:
        row.Typ = 3.02
    elif row.Gefaelle >= 8:
        row.Typ = 3.03
    elif row.Gefaelle >= 4:
        row.Typ = 3.04
    elif row.Gefaelle < 4:
        row.Typ = 3.05
    else:
        row.Klasse = 0
    rows.updateRow(row)
del row
del rows
arcpy.TableToTable_conversion(outputPath + "temp.gdb\\aussen_10m_" + See,
outputPath + "temp.gdb", "Typ_" + See)
arcpy.DissolveRouteEvents_lr(outputPath + "temp.gdb\\Typ_" + See, "RID LINE VON
BIS", "Typ", outputPath + "attribute.gdb\\Attr03_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen des Ufertyps aufgetreten."
    print arcpy.GetMessages()

print "Ufertyp erstellt."

arcpy.Delete_management(outputPath + "temp.gdb")
```

## Anhang 3.02: Standorttypische Ufervegetation

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
# Standorttypische Ufervegetation
# Resultate: Ein Event Table pro See
# - Attr07 (Attribut Standorttypische Ufervegetation Nr. 7)

try:
    for See in listeSeen:
        arcpy.Erase_analysis(inputPath + "30_Raeume\\raeume.gdb\\ufer_" + See, inputPath
            + "10_Perimeter\\ufer_v.gdb\\general_" + See + "f", tempGDBPfad + "Ufer")
        if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_BODENBEDECKUNG"):
            arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_BODENBEDECKUNG", tempGDBPfad + "Wald_alle", '"OBJEKTART" in (\Wald\, \Wald
                offen\, \Gebueschwald\')')
            arcpy.Buffer_analysis(tempGDBPfad + "Wald_alle", tempGDBPfad +
                "Wald_buffer", "1", "FULL", "ROUND", "ALL")
            arcpy.Clip_analysis(tempGDBPfad + "Wald_buffer", tempGDBPfad + "Ufer",
                tempGDBPfad + "Wald_multi")
            arcpy.MultipartToSinglepart_management(tempGDBPfad + "Wald_multi",
                tempGDBPfad + "Wald")
            arcpy.PolygonToLine_management(tempGDBPfad + "Wald", tempGDBPfad + "Wald_1",
                "IGNORE_NEIGHBORS")
            arcpy.Intersect_analysis([tempGDBPfad + "Wald_1", inputPath +
                "10_Perimeter\\ufer_v.gdb\\general_" + See], tempGDBPfad + "Wald_innen_1")
            arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Wald_innen_1", inputPath +
                "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "1", tempMDBPfad +
                "Attr07_3_i", "RID LINE VON BIS", "ALL", "NO_DISTANCE", "NO_ZERO", "NO_FIELDS")
            arcpy.Intersect_analysis([tempGDBPfad + "Wald_1", inputPath +
                "30_Raeume\\raeume.gdb\\ufer_" + See + "_ausсен_1"], tempGDBPfad +
                "Wald_aussen_multi_1")
            arcpy.MultipartToSinglepart_management(tempGDBPfad + "Wald_aussen_multi_1",
                tempGDBPfad + "Wald_aussen_1")
            arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Wald_aussen_1", tempGDBPfad +
                "Wald_aussen_p", "Node")
            arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Wald_aussen_p", inputPath
                + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad +
                "Attr07_3_a_temp", "RID POINT POSITION")

            arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_BODENBEDECKUNG", tempGDBPfad + "Feuchtgebiet_TLM", '"OBJEKTART" =
                \Feuchtgebiet\')
            arcpy.Select_analysis(inputPath + "20_Geodaten\\INV.gdb\\" + See + "_INV",
                tempGDBPfad + "Inventare", '"AU" = 1 OR "FM" = 1 OR "HM" = 1')

```

```

arcpy.Merge_management([tempGDBPfad + "Feuchtgebiet_TLM", tempGDBPfad +
    "Inventare"], tempGDBPfad + "Feuchtgebiet_alle")
arcpy.Buffer_analysis(tempGDBPfad + "Feuchtgebiet_alle", tempGDBPfad +
    "Feuchtgebiet_buffer", "1", "FULL", "ROUND", "ALL")
arcpy.Clip_analysis(tempGDBPfad + "Feuchtgebiet_buffer", tempGDBPfad + "Ufer",
    tempGDBPfad + "Feuchtgebiet_multi")
arcpy.Dissolve_management(tempGDBPfad + "Feuchtgebiet_multi", tempGDBPfad +
    "Feuchtgebiet", "", "", "SINGLE_PART")
arcpy.PolygonToLine_management(tempGDBPfad + "Feuchtgebiet", tempGDBPfad +
    "Feuchtgebiet_1", "IGNORE_NEIGHBORS")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_1", inputPath +
    "10_Perimeter\\ufer_v.gdb\\general_" + See], tempGDBPfad +
    "Feuchtgebiet_innen_1")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_innen_1",
    inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "1", tempMDBPfad +
    "Attr07_2_i", "RID LINE VON BIS", "ALL", "NO_DISTANCE", "NO_ZERO", "NO_FIELDS")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_1", inputPath +
    "30_Raeume\\raeume.gdb\\ufer_" + See + "_ausсен_1"], tempGDBPfad +
    "Feuchtgebiet_aussen_multi_1")
arcpy.MultipartToSinglepart_management(tempGDBPfad +
    "Feuchtgebiet_aussen_multi_1", tempGDBPfad + "Feuchtgebiet_aussen_1")
arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Feuchtgebiet_aussen_1",
    tempGDBPfad + "Feuchtgebiet_aussen_p", "Node")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_aussen_p",
    inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
    + "Attr07_2_a_temp", "RID POINT POSITION")

arcpy.CreateTable_management(tempMDBPfad, "Attr07_2_a", outputPath + "template")
arcpy.CreateTable_management(tempMDBPfad, "Attr07_3_a", outputPath + "template")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr07_2_a_temp", tempMDBPfad,
    "Attr07_2_a_temp")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr07_3_a_temp", tempMDBPfad,
    "Attr07_3_a_temp")
arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
    See, tempMDBPfad, "Attr07_1")
arcpy.AddField_management(tempMDBPfad + "Attr07_1", "VON", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr07_1", "BIS", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr07_1", "TYP1", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr07_2_i", "TYP2i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr07_2_a", "TYP2a", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr07_3_i", "TYP3i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr07_3_a", "TYP3a", "DOUBLE")

constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
    Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
    oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
conn = adodbapi.connect(constr)
cur = conn.cursor()
cur.execute("UPDATE Attr07_1 SET VON = 0, BIS = round(Shape_Length, 0), TYP1 =
    7.01")
cur.execute("UPDATE Attr07_2_i SET TYP2i = 7.02")
cur.execute("UPDATE Attr07_3_i SET TYP3i = 7.03")
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
    Attr07_2_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 7.02
    cur.execute("INSERT INTO Attr07_2_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
    l))
    h = h+1
    conn.commit()
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
    Attr07_3_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 7.03

```

```

        cur.execute("INSERT INTO Attr07_3_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
        h = h+1
        conn.commit()
cur.close()
conn.close()

arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr07_1", "RID LINE VON BIS",
tempMDBPfad + "Attr07_2_i", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr07_12_i", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr07_12_i", "RID LINE VON BIS",
tempMDBPfad + "Attr07_2_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr07_12", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr07_12", "RID LINE VON BIS",
tempMDBPfad + "Attr07_3_i", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr07_13_i", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr07_13_i", "RID LINE VON BIS",
tempMDBPfad + "Attr07_3_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr07_13", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")

arcpy.AddField_management(tempMDBPfad + "Attr07_13", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr07_13")
for row in rows:
    if row.TYP2i == 7.02:
        if row.TYP2a == 7.02:
            row.TYP = 7.02
        elif row.TYP3i == 7.03:
            if row.TYP3a == 7.03:
                row.TYP = 7.03
            else:
                row.TYP = 7.01
        else:
            row.TYP = 7.01
    elif row.TYP3i == 7.03:
        if row.TYP3a == 7.03:
            row.TYP = 7.03
        else:
            row.TYP = 7.01
    else:
        row.TYP = 7.01
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr07_13", "RID LINE VON BIS",
"TYP", outputPath + "Attr07_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der standorttypischen Ufervegetation aufgetreten."
    print arcpy.GetMessages()

print "Standorttypische Ufervegetation erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

## Anhang 3.03: Ufersaum-Vegetation

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
#    Ufersaum-Vegetation
#    Resultate: Ein Event Table pro See
#    - Attr08 (Attribut Ufersaum-Vegetation Nr. 8)

try:
    for See in listeSeen:
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", tempGDBPfad + "See_3m", "3")
        arcpy.Erase_analysis(tempGDBPfad + "See_3m", inputPath +
            "10_Perimeter\\see_v.gdb\\general_" + See + "_f", tempGDBPfad + "Ufer_3m")
        arcpy.PolygonToLine_management(tempGDBPfad + "Ufer_3m", tempGDBPfad +
            "Ufer_3m_1", "IGNORE_NEIGHBORS")

        arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_BODENBEDECKUNG", tempGDBPfad + "Wald_f", "'OBJEKTART' in ('Wald\\', 'Wald
            offen\\', 'Gebueschwald\\')')
        arcpy.Buffer_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_EINZELBAUM_GEBUESCH", tempGDBPfad + "Baum_f", "2.5")
        arcpy.Buffer_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_BAUM_GEBUESCHREIHE", tempGDBPfad + "Baumreihe_f", "5")
        arcpy.Merge_management([tempGDBPfad + "Wald_f", tempGDBPfad + "Baum_f",
            tempGDBPfad + "Baumreihe_f"], tempGDBPfad + "Gehoelz_alle")
        arcpy.Buffer_analysis(tempGDBPfad + "Gehoelz_alle", tempGDBPfad + "Gehoelz",
            "1", "FULL", "ROUND", "ALL")
        arcpy.Clip_analysis(tempGDBPfad + "Gehoelz", tempGDBPfad + "Ufer_3m",
            tempGDBPfad + "Gehoelz_3m_multi")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Gehoelz_3m_multi",
            tempGDBPfad + "Gehoelz_3m")
        arcpy.PolygonToLine_management(tempGDBPfad + "Gehoelz_3m", tempGDBPfad +
            "Gehoelz_3m_1", "IGNORE_NEIGHBORS")
        arcpy.Intersect_analysis([tempGDBPfad + "Gehoelz_3m_1", inputPath +
            "10_Perimeter\\ufer_v.gdb\\general_" + See], tempGDBPfad + "Gehoelz_innen_1")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Gehoelz_innen_1", inputPath +
            "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "1", tempMDBPfad +
            "Attr08_1_i", "RID LINE VON BIS", "ALL", "NO_DISTANCE", "NO_ZERO", "NO_FIELDS")
        arcpy.Intersect_analysis([tempGDBPfad + "Gehoelz_3m", tempGDBPfad +
            "Ufer_3m_1"], tempGDBPfad + "Gehoelz_aussen_3m_multi_1")
        arcpy.MultipartToSinglepart_management(tempGDBPfad +
            "Gehoelz_aussen_3m_multi_1", tempGDBPfad + "Gehoelz_aussen_3m_1")
        arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Gehoelz_aussen_3m_1", tempGDBPfad
            + "Gehoelz_aussen_3m_p", "Node")

```

```

arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Gehoelz_aussen_3m_p",
inputPath + "10_Perimeter\uferv.gdb\route_" + See, "RID", "20 Meters",
tempGDBPfad + "Attr08_1_a_temp", "RID POINT POSITION")

arcpy.Select_analysis(inputPath + "20_Geodaten\TLM.gdb\" + See +
"_BODENBEDECKUNG", tempGDBPfad + "Feuchtgebiet_f", "'OBJEKTART" in
(\\'Feuchtgebiet\\'))
arcpy.Select_analysis(inputPath + "20_Geodaten\INV.gdb\" + See + "_INV",
tempGDBPfad + "Inventare_f", "'AU" = 1 OR "FM" = 1 OR "HM" = 1')
arcpy.Merge_management([tempGDBPfad + "Feuchtgebiet_f", tempGDBPfad +
"Inventare_f"], tempGDBPfad + "Feuchtgebiet_alle")
arcpy.Buffer_analysis(tempGDBPfad + "Feuchtgebiet_alle", tempGDBPfad +
"Feuchtgebiet", "1", "FULL", "ROUND", "ALL")
arcpy.Clip_analysis(tempGDBPfad + "Feuchtgebiet", tempGDBPfad + "Ufer_3m",
tempGDBPfad + "Feuchtgebiet_3m_multi")
arcpy.MultipartToSinglepart_management(tempGDBPfad + "Feuchtgebiet_3m_multi",
tempGDBPfad + "Feuchtgebiet_3m")
arcpy.PolygonToLine_management(tempGDBPfad + "Feuchtgebiet_3m", tempGDBPfad +
"Feuchtgebiet_3m_l", "IGNORE_NEIGHBORS")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_3m_l", inputPath +
"10_Perimeter\uferv.gdb\general_" + See], tempGDBPfad +
"Feuchtgebiet_innen_l")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_innen_l",
inputPath + "10_Perimeter\uferv.gdb\route_" + See, "RID", "1", tempMDBPfad +
"Attr08_2_i", "RID LINE VON BIS", "ALL", "NO_DISTANCE", "NO_ZERO", "NO_FIELDS")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_3m", tempGDBPfad +
"Ufer_3m_l"], tempGDBPfad + "Feuchtgebiet_aussen_3m_multi_l")
arcpy.MultipartToSinglepart_management(tempGDBPfad +
"Feuchtgebiet_aussen_3m_multi_l", tempGDBPfad + "Feuchtgebiet_aussen_3m_l")
arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Feuchtgebiet_aussen_3m_l",
tempGDBPfad + "Feuchtgebiet_aussen_3m_p", "Node")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_aussen_3m_p",
inputPath + "10_Perimeter\uferv.gdb\route_" + See, "RID", "10", tempGDBPfad
+ "Attr08_2_a_temp", "RID POINT POSITION")

arcpy.CreateTable_management(tempMDBPfad, "Attr08_1_a", outputPath + "template")
arcpy.CreateTable_management(tempMDBPfad, "Attr08_2_a", outputPath + "template")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr08_1_a_temp", tempMDBPfad,
"Attr08_1_a_temp")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr08_2_a_temp", tempMDBPfad,
"Attr08_2_a_temp")
arcpy.TableToTable_conversion(inputPath + "10_Perimeter\uferv.gdb\route_" +
See, tempMDBPfad, "Attr08_5")

arcpy.AddField_management(tempMDBPfad + "Attr08_1_i", "TYP1i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr08_2_i", "TYP2i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr08_1_a", "TYP1a", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr08_2_a", "TYP2a", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr08_5", "VON", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr08_5", "BIS", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr08_5", "TYP5", "DOUBLE")

constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
conn = adodbapi.connect(constr)
cur = conn.cursor()
cur.execute("UPDATE Attr08_5 SET VON = 0, BIS = round(Shape_Length, 0), TYP5 =
8.05")
cur.execute("UPDATE Attr08_1_i SET TYP1i = 8.01")
cur.execute("UPDATE Attr08_2_i SET TYP2i = 8.02")
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr08_1_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
i = int(item[0])
j = int(item[1])
k = int(item[2])
l = 8.01
cur.execute("INSERT INTO Attr08_1_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
h = h+1
conn.commit()

```

```

cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr08_2_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 8.02
    cur.execute("INSERT INTO Attr08_2_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
    h = h+1
    conn.commit()
cur.close()
conn.close()

arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr08_1_i", "RID LINE VON BIS",
tempMDBPfad + "Attr08_1_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr08_1", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr08_1", "RID LINE VON BIS",
tempMDBPfad + "Attr08_2_i", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr08_12_i", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr08_12_i", "RID LINE VON BIS",
tempMDBPfad + "Attr08_2_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr08_12", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr08_12", "RID LINE VON BIS",
tempMDBPfad + "Attr08_5", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr08_15", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")

arcpy.AddField_management(tempMDBPfad + "Attr08_15", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr08_15")
for row in rows:
    if row.TYP2i == 8.02:
        if row.TYP2a == 8.02:
            row.TYP = 8.02
        else:
            row.TYP = 8.04
    elif row.TYP1i == 8.01:
        if row.TYP1a == 8.01:
            row.TYP = 8.01
        else:
            row.TYP = 8.03
    elif row.TYP2a == 8.02:
        row.TYP = 8.04
    elif row.TYP1a == 8.01:
        row.TYP = 8.03
    else:
        row.TYP = 8.05
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr08_15", "RID LINE VON BIS",
"TYP", outputPath + "Attr08_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der Ufersaum-Vegetation aufgetreten."
    print arcpy.GetMessages()

print "Ufersaum-Vegetation erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

**Anhang 3.04: Uferverbau und künstliche Morphologie**

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
#   Uferverbau und kuenstliche Morphologie
#   Resultate: Ein Event Table pro See
#   - Attr09 (Attribut Uferverbau und kuenstliche Morphologie Nr. 9)

try:
    for See in listeSeen:
        arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
            See, tempMDBPfad, "route")
        arcpy.CreateTable_management(tempMDBPfad, "Attr09_01", outputPath + "template")
        arcpy.AddField_management(tempMDBPfad + "Attr09_01", "TYP01", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
            Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
            oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT round(Shape_Length, 0) FROM route")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = 1
            j = 0
            k = int(item[0])
            l = 9.01
        cur.execute("INSERT INTO Attr09_01 VALUES (?, ?, ?, ?, ?)", (h, i, j, k, l))
        conn.commit()
        cur.close()
        conn.close()

    if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
        "_VERBAUUNG MAUER"):
        arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_VERBAUUNG MAUER", tempGDBPfad + "Verbauung_aller", "OBJEKTART" =
            '\\Gewaesserverbauung\\')
        arcpy.Clip_analysis(tempGDBPfad + "Verbauung_aller", inputPath +
            "30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Verbauung_multi")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Verbauung_multi",
            tempGDBPfad + "Verbauung")
        rows = arcpy.SearchCursor(tempGDBPfad + "Verbauung")
        if rows.next():
            arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Verbauung", tempGDBPfad +
                "Verbauung_p", "Node")

```

```

        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Verbauung_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr09_05_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Verbauung")
        if arcpy.Exists(tempGDBPfad + "Attr09_05_temp"):
            arcpy.TableToTable_conversion(tempGDBPfad + "Attr09_05_temp",
tempMDBPfad, "Attr09_05_temp")
            arcpy.CreateTable_management(tempMDBPfad, "Attr09_05", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr09_05", "TYP05", "DOUBLE")
            constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
            conn = adodbapi.connect(constr)
            cur = conn.cursor()
            cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr09_05_temp GROUP BY ET_IDP, RID")
            result = cur.fetchall()
            h = 1
            for item in result:
                i = int(item[0])
                j = int(item[1])
                k = int(item[2])
                l = 9.05
                cur.execute("INSERT INTO Attr09_05 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
                h = h+1
            conn.commit()
            cur.close()
            conn.close()
            arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr09_01", "RID LINE VON
BIS", tempMDBPfad + "Attr09_05", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr09", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
            arcpy.AddField_management(tempMDBPfad + "Attr09", "TYP", "DOUBLE")
            rows = arcpy.UpdateCursor(tempMDBPfad + "Attr09")
            for row in rows:
                if row.TYP05 == 9.05:
                    row.TYP = 9.05
                elif row.TYP01 == 9.01:
                    row.TYP = 9.01
                else:
                    row.TYP = 9
                rows.updateRow(row)
            del row
            del rows
            arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr09", "RID LINE VON BIS",
"TYP", outputPath + "Attr09_" + See, "RID LINE VON BIS")
        else:
            arcpy.TableToTable_conversion(tempMDBPfad + "Attr09_01", outputPath +
"Attr09_" + See)
except:
    print "Ein Fehler beim Erstellen Uferverbau und kuenstliche Morphologie
aufgetreten."
    print arcpy.GetMessages()

print "Uferverbau und kuenstliche Morphologie erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

**Anhang 3.05: Fließgewässerranbindung (Mündung / Abfluss)**

```

# Libraries/allgemeine Einstellungen
import arcpy, os
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
path = os.path.abspath("../")
path = path.replace("\\", r"\\")
inputPath = path + "\\20_bearbeitete_Daten\\"
outputPath = path + "\\20_bearbeitete_Daten\\40_Attribute\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende gelöscht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
#   Ufertyp
#   Resultate: Ein Event Table pro See
#   - Attr10 (Attribut Fließgewässerranbindung (Mündung / Abfluss) Nr. 10)

try:
    for See in listeSeen:
        arcpy.Clip_analysis(inputPath + "20_Geodaten\\OEKMO.gdb\\Abstuerze", inputPath +
            "30_Raeume\\raeume.gdb\\ufer_" + See, outputPath +
            "tempGDB.gdb\\oekmo_alle_Abstuerze_" + See)
        arcpy.Select_analysis(outputPath + "tempGDB.gdb\\oekmo_alle_Abstuerze_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Abstuerze_" + See, '"ABSTHOEH" >= 20')
        arcpy.Clip_analysis(inputPath + "20_Geodaten\\OEKMO.gdb\\Bauwerke", inputPath +
            "30_Raeume\\raeume.gdb\\ufer_" + See, outputPath +
            "tempGDB.gdb\\oekmo_alle_Bauwerke_" + See)
        arcpy.Select_analysis(outputPath + "tempGDB.gdb\\oekmo_alle_Bauwerke_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Bauwerke_" + See, 'NOT "BAUWTYP" in (7, 10,
            11)')
        arcpy.Clip_analysis(inputPath + "20_Geodaten\\OEKMO.gdb\\Abschnitte", inputPath +
            "30_Raeume\\raeume.gdb\\ufer_" + See, outputPath +
            "tempGDB.gdb\\oekmo_Abschnitte_" + See)
        arcpy.JoinField_management(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_" + See,
            "BACHNR", outputPath + "tempGDB.gdb\\oekmo_Abstuerze_" + See, "BACHNR")
        arcpy.JoinField_management(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_" + See,
            "BACHNR", outputPath + "tempGDB.gdb\\oekmo_Bauwerke_" + See, "BACHNR")
        arcpy.gp.ET_GPPolylineToPoints(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_" +
            See, outputPath + "tempGDB.gdb\\oekmo_Abschnitte_p_" + See, "Node", "True")
        arcpy.Clip_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_FLIESSGEWAESSER", inputPath + "30_Raeume\\raeume.gdb\\ufer_" + See,
            outputPath + "tempGDB.gdb\\fliessgewaesser_alle_" + See)
        arcpy.Select_analysis(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Abschnitte_offen_" + See, '"EINDOL" = 0')
        arcpy.Buffer_analysis(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_offen_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buf1_" + See, "10")
        arcpy.Select_analysis(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Abschnitte_eind_" + See, '"EINDOL" = 1')
        arcpy.Buffer_analysis(outputPath + "tempGDB.gdb\\oekmo_Abschnitte_eind_" + See,
            outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buf2_" + See, "30")
        arcpy.Merge_management([outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buf1_" +
            See, outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buf2_" + See], outputPath +
            "tempGDB.gdb\\oekmo_Abschnitte_buffer_" + See)
        arcpy.MakeFeatureLayer_management(outputPath +
            "tempGDB.gdb\\fliessgewaesser_alle_" + See, "lyr_fliessgewaesser")

```

```

arcpy.SelectLayerByLocation_management("lyr_fliessgewaesser", "INTERSECT",
outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buffer_" + See)
arcpy.SelectLayerByLocation_management("lyr_fliessgewaesser", "INTERSECT",
outputPath + "tempGDB.gdb\\oekmo_Abschnitte_buffer_" + See, "",
"SWITCH_SELECTION")
arcpy.CopyFeatures_management("lyr_fliessgewaesser", outputPath +
"tempGDB.gdb\\fliessgewaesser_" + See)
arcpy.gp.ET_GPPolylineToPoints(outputPath + "tempGDB.gdb\\fliessgewaesser_" +
See, outputPath + "tempGDB.gdb\\fliessgewaesser_p_" + See, "Node", "True")
arcpy.Merge_management([outputPath + "tempGDB.gdb\\oekmo_Abschnitte_p_" + See,
outputPath + "tempGDB.gdb\\fliessgewaesser_p_" + See], outputPath +
"tempGDB.gdb\\muendungen_p_" + See)
arcpy.AddField_management(outputPath + "tempGDB.gdb\\muendungen_p_" + See,
"Typ", "DOUBLE")
arcpy.LocateFeaturesAlongRoutes_lr(outputPath + "tempGDB.gdb\\muendungen_p_" +
See, inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "10",
outputPath + "tempGDB.gdb\\Attr10_" + See, "RID POINT POSITION")
rows = arcpy.UpdateCursor(outputPath + "tempGDB.gdb\\Attr10_" + See)
for row in rows:
    if row.GSBREITE > 10:
        row.VON = round(row.POSITION, 0) - round(row.GSBREITE/2, 0)
        row.BIS = round(row.POSITION, 0) + round(row.GSBREITE/2, 0)
    else:
        row.VON = round(row.POSITION, 0) - 5
        row.BIS = round(row.POSITION, 0) + 5
    rows.updateRow(row)
del row
del rows
rows = arcpy.UpdateCursor(outputPath + "tempGDB.gdb\\Attr10_" + See)
for row in rows:
    if row.KLASSEZH == 5:
        row.Typ = 10.05
    elif row.KLASSEZH in (3, 4):
        if row.ABSTNR > 0:
            row.Typ = 10.04
        elif row.BAUWNR > 0:
            row.Typ = 10.04
        else:
            row.Typ = 10.03
    elif row.KLASSEZH in (1, 2):
        if row.ABSTNR > 0:
            row.Typ = 10.02
        elif row.BAUWNR > 0:
            row.Typ = 10.02
        else:
            row.Typ = 10.01
    else:
        row.Typ = 10.00
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(outputPath + "tempGDB.gdb\\Attr10_" + See, "RID
LINE VON BIS", "Typ", outputPath + "attribute.gdb\\Attr10_" + See, "RID LINE
VON BIS")

except:
    print "Ein Fehler beim Erstellen der Fliessgewaesseranbindung aufgetreten."
    print arcpy.GetMessages()

print "Fliessgewaesseranbindung erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

**Anhang 3.06: Land- und forstwirtschaftliche Nutzung im Uferstreifen**

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
# Land- und forstwirtschaftliche Nutzung im Uferstreifen
# Resultate: Ein Event Table pro See
# - Attr11 (Attribut Land- und forstwirtschaftliche Nutzung im Uferstreifen Nr. 11)

try:
    for See in listeSeen:
        if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_BODENBEDECKUNG"):
            arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_BODENBEDECKUNG", tempGDBPfad + "Wald_alles", 'OBJEKTART = \'Wald\'')
            arcpy.Clip_analysis(tempGDBPfad + "Wald_alles", inputPath +
                "30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Wald_multi")
            arcpy.Dissolve_management(tempGDBPfad + "Wald_multi", tempGDBPfad + "Wald",
                "", "SINGLE_PART")
            rows = arcpy.SearchCursor(tempGDBPfad + "Wald")
            if rows.next():
                arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Wald", tempGDBPfad +
                    "Wald_p", "Vertex")
                arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Wald_p", inputPath +
                    "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad +
                    "Attr11_2_temp", "RID POINT POSITION")
            else:
                arcpy.Delete_management(tempGDBPfad + "Wald")
                if arcpy.Exists(tempGDBPfad + "Attr11_2_temp"):
                    arcpy.TableToTable_conversion(tempGDBPfad + "Attr11_2_temp",
                        tempMDBPfad, "Attr11_2_temp")
                    arcpy.CreateTable_management(tempMDBPfad, "Attr11_2", outputPath +
                        "template")
                    arcpy.AddField_management(tempMDBPfad + "Attr11_2", "TYP2", "DOUBLE")
                    constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
                        Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
                        oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
                    conn = adodbapi.connect(constr)
                    cur = conn.cursor()
                    cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
                        0) FROM Attr11_2_temp GROUP BY ET_IDP, RID")
                    result = cur.fetchall()
                    h = 1
                    for item in result:
                        i = int(item[0])
                        j = int(item[1])
                        k = int(item[2])

```

```

        l = 11.02
        cur.execute("INSERT INTO Attr11_2 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
        h = h+1
        conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr11_2", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr11_2", "TYP2", "DOUBLE")

if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_NUTZUNGSAREAL"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"NUTZUNGSAREAL", tempGDBPfad + "Sonderkultur_alles", "OBJEKTART" in
('\\Obstanlagen\\', '\\Reben\\', '\\Baumschule\\'))
    arcpy.Clip_analysis(tempGDBPfad + "Sonderkultur_alles", inputPath +
"30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Sonderkultur_multi")
    arcpy.Dissolve_management(tempGDBPfad + "Sonderkultur_multi", tempGDBPfad +
"Sonderkultur", "", "", "SINGLE_PART")
    rows = arcpy.SearchCursor(tempGDBPfad + "Sonderkultur")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Sonderkultur", tempGDBPfad
+ "Sonderkultur_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Sonderkultur_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr11_3_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Sonderkultur")
    if arcpy.Exists(tempGDBPfad + "Attr11_3_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr11_3_temp",
tempMDBPfad, "Attr11_3_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr11_3", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr11_3", "TYP3", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr11_3_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 11.03
            cur.execute("INSERT INTO Attr11_3 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
            h = h+1
            conn.commit()
            cur.close()
            conn.close()
        else:
            arcpy.CreateTable_management(tempMDBPfad, "Attr11_5", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr11_5", "TYP5", "DOUBLE")

arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
See, tempMDBPfad, "route")
arcpy.CreateTable_management(tempMDBPfad, "Attr11_1", outputPath + "template")
arcpy.AddField_management(tempMDBPfad + "Attr11_1", "TYP1", "DOUBLE")
constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
conn = adodbapi.connect(constr)
cur = conn.cursor()
cur.execute("SELECT round(Shape_Length, 0) FROM route")
result = cur.fetchall()
h = 1
for item in result:
    i = 1
    j = 0

```

```
k = int(item[0])
l = 11.01
cur.execute("INSERT INTO Attr11_1 VALUES (?, ?, ?, ?, ?)", (h, i, j, k, l))
conn.commit()
cur.close()
conn.close()

arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr11_1", "RID LINE VON BIS",
tempMDBPfad + "Attr11_2", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr11_12", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr11_12", "RID LINE VON BIS",
tempMDBPfad + "Attr11_3", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr11_13", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.AddField_management(tempMDBPfad + "Attr11_13", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr11_13")
for row in rows:
    if row.TYP3 == 11.03:
        row.TYP = 11.03
    elif row.TYP2 == 11.02:
        row.TYP = 11.02
    elif row.TYP1 == 11.01:
        row.TYP = 11.01
    else:
        row.TYP = 11
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr11_13", "RID LINE VON BIS",
"TYP", outputPath + "Attr11_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der land- und forstwirtschaftlichen Nutzung
aufgetreten."
    print arcpy.GetMessages()

print "Land- und forstwirtschaftlichen Nutzung erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")
```

**Anhang 3.07: Siedlung, Gewerbe, Industrie im Uferstreifen**

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
from decimal import Decimal
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
#   Siedlung, Gewerbe, Industrie im Uferstreifen
#   Resultate: Ein Event Table pro See
#   - Attr11 (Attribut Siedlung, Gewerbe, Industrie im Uferstreifen im Uferstreifen Nr.
    13)

try:
    for See in listeSeen:
        if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_GEBAEUDE_FOOTPRINT"):
            _arcpy.Clip_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_GEBAEUDE_FOOTPRINT", inputPath + "30_Raeume\\raeume.gdb\\ufer_" + See,
                tempGDBPfad + "Gebaeude")
            arcpy.Buffer_analysis(tempGDBPfad + "Gebaeude", tempGDBPfad +
                "Gebaeude_alle_Buffer", "15", "", "", "ALL")
            arcpy.Clip_analysis(tempGDBPfad + "Gebaeude_alle_Buffer", inputPath +
                "30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Gebaeude_multi_buffer")
            arcpy.MultipartToSinglepart_management(tempGDBPfad +
                "Gebaeude_multi_buffer", tempGDBPfad + "Gebaeude_buffer")
            arcpy.Union_analysis([tempGDBPfad + "Gebaeude_buffer", tempGDBPfad +
                "Gebaeude"], tempGDBPfad + "Siedlung")
            arcpy.Select_analysis(tempGDBPfad + "Siedlung", tempGDBPfad +
                "Siedlung_Geb_multi", '"FID_Gebaeude" > 0')
            arcpy.Dissolve_management(tempGDBPfad + "Siedlung_Geb_multi", tempGDBPfad +
                "Siedlung_Geb_Fl", "FID_Gebaeude_buffer", "FID_Gebaeude_buffer COUNT;SHAPE_AREA
                SUM", "MULTI_PART")
            arcpy.AddField_management(tempGDBPfad + "Siedlung_Geb_Fl", "Anz_geb",
                "DOUBLE")
            arcpy.AddField_management(tempGDBPfad + "Siedlung_Geb_Fl", "Fl_geb",
                "DOUBLE")
            rows = arcpy.UpdateCursor(tempGDBPfad + "Siedlung_Geb_Fl")
            for row in rows:
                row.Anz_geb = row.COUNT_FID_Gebaeude_buffer
                row.Fl_geb = row.SUM_Shape_Area
                rows.updateRow(row)
            del row
            del rows

            arcpy.Union_analysis([tempGDBPfad + "Siedlung", tempGDBPfad +
                "Siedlung_Geb_Fl"], tempGDBPfad + "Siedlung_Attr_multi")

```

```

    arcpy.Dissolve_management(tempGDBPfad + "Siedlung_Attr_multi", tempGDBPfad +
"Siedlung_Attr", "FID_Gebaeude_buffer", "Anz_geb MAX; Fl_geb MAX",
"MULTI_PART")
    arcpy.AddField_management(tempGDBPfad + "Siedlung_Attr" , "TYP", "DOUBLE")
    rows = arcpy.UpdateCursor(tempGDBPfad + "Siedlung_Attr")
    for row in rows:
        if row.Max_Anz_geb == 1:
            if row.Max_Fl_geb > 50:
                row.TYP = 13.03
            else:
                row.TYP = 13.02
        elif row.Max_Anz_geb > 1:
            if row.Max_Fl_geb > row.Shape_Area:
                row.TYP = 13.05
            else:
                row.TYP = 13.04
        else:
            row.TYP = 13.00
    rows.updateRow(row)
del row
del rows

    arcpy.Select_analysis(tempGDBPfad + "Siedlung_Attr", tempGDBPfad +
"Siedlung_Attr05", '"Typ" = 13.05')
    arcpy.Select_analysis(tempGDBPfad + "Siedlung_Attr", tempGDBPfad +
"Siedlung_Attr04", '"Typ" = 13.04')
    arcpy.Select_analysis(tempGDBPfad + "Siedlung_Attr", tempGDBPfad +
"Siedlung_Attr03", '"Typ" = 13.03')
    arcpy.Select_analysis(tempGDBPfad + "Siedlung_Attr", tempGDBPfad +
"Siedlung_Attr02", '"Typ" = 13.02')

    rows = arcpy.SearchCursor(tempGDBPfad + "Siedlung_Attr05")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Siedlung_Attr05",
tempGDBPfad + "Siedlung_Attr05_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Siedlung_Attr05_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr13_05_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Siedlung_Attr05")
    rows = arcpy.SearchCursor(tempGDBPfad + "Siedlung_Attr04")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Siedlung_Attr04",
tempGDBPfad + "Siedlung_Attr04_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Siedlung_Attr04_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr13_04_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Siedlung_Attr04")
    rows = arcpy.SearchCursor(tempGDBPfad + "Siedlung_Attr03")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Siedlung_Attr03",
tempGDBPfad + "Siedlung_Attr03_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Siedlung_Attr03_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr13_03_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Siedlung_Attr03")
    rows = arcpy.SearchCursor(tempGDBPfad + "Siedlung_Attr02")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Siedlung_Attr02",
tempGDBPfad + "Siedlung_Attr02_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Siedlung_Attr02_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr13_02_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Siedlung_Attr02")

    if arcpy.Exists(tempGDBPfad + "Attr13_05_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr13_05_temp",
tempMDBPfad, "Attr13_05_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_05", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_05", "TYP05", "DOUBLE")

```

```

        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr13_05_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 13.05
            cur.execute("INSERT INTO Attr13_05 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
            h = h+1
            conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_05", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_05", "TYP05", "DOUBLE")

    if arcpy.Exists(tempGDBPfad + "Attr13_04_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr13_04_temp",
tempMDBPfad, "Attr13_04_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_04", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_04", "TYP04", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr13_04_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 13.04
            cur.execute("INSERT INTO Attr13_04 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
            h = h+1
            conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_04", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_04", "TYP04", "DOUBLE")

    if arcpy.Exists(tempGDBPfad + "Attr13_03_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr13_03_temp",
tempMDBPfad, "Attr13_03_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_03", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_03", "TYP03", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr13_03_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])

```

```

        l = 13.03
        cur.execute("INSERT INTO Attr13_03 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
        h = h+1
        conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr13_03", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr13_03", "TYP03", "DOUBLE")

        if arcpy.Exists(tempGDBPfad + "Attr13_02_temp"):
            arcpy.TableToTable_conversion(tempGDBPfad + "Attr13_02_temp",
tempMDBPfad, "Attr13_02_temp")
            arcpy.CreateTable_management(tempMDBPfad, "Attr13_02", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr13_02", "TYP02", "DOUBLE")
            constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
            conn = adodbapi.connect(constr)
            cur = conn.cursor()
            cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr13_02_temp GROUP BY ET_IDP, RID")
            result = cur.fetchall()
            h = 1
            for item in result:
                i = int(item[0])
                j = int(item[1])
                k = int(item[2])
                l = 13.02
                cur.execute("INSERT INTO Attr13_02 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
                h = h+1
                conn.commit()
            cur.close()
            conn.close()
        else:
            arcpy.CreateTable_management(tempMDBPfad, "Attr13_02", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr13_02", "TYP02", "DOUBLE")

            arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_"
+ See, tempMDBPfad, "route")
            arcpy.CreateTable_management(tempMDBPfad, "Attr13_01", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr13_01", "TYP01", "DOUBLE")
            constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
            conn = adodbapi.connect(constr)
            cur = conn.cursor()
            cur.execute("SELECT round(Shape_Length, 0) FROM route")
            result = cur.fetchall()
            h = 1
            for item in result:
                i = 1
                j = 0
                k = int(item[0])
                l = 13.01
                cur.execute("INSERT INTO Attr13_01 VALUES (?, ?, ?, ?, ?)", (h, i, j, k, l))
            conn.commit()
            cur.close()
            conn.close()

arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr13_01", "RID LINE VON BIS",
tempMDBPfad + "Attr13_02", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr13_12", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr13_12", "RID LINE VON BIS",
tempMDBPfad + "Attr13_03", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr13_13", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr13_13", "RID LINE VON BIS",
tempMDBPfad + "Attr13_04", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr13_14", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")

```

```
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr13_14", "RID LINE VON BIS",
tempMDBPfad + "Attr13_05", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr13_15", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.AddField_management(tempMDBPfad + "Attr13_15", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr13_15")
for row in rows:
    if row.TYP05 == 13.05:
        row.TYP = 13.05
    elif row.TYP04 == 13.04:
        row.TYP = 13.04
    elif row.TYP03 == 13.03:
        row.TYP = 13.03
    elif row.TYP02 == 13.02:
        row.TYP = 13.02
    elif row.TYP01 == 13.01:
        row.TYP = 13.01
    else:
        row.TYP = 13
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr13_15", "RID LINE VON BIS",
"TYP", outputPath + "Attr13_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der Siedlung, Gewerbe, Industrie im Uferstreifen
aufgetreten."
    print arcpy.GetMessages()

print "Siedlung, Gewerbe, Industrie im Uferstreifen erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")
```

## Anhang 3.08: Verkehrswege und -flächen im Uferstreifen

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende gelöscht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
# Verkehrswege und -flaechen im Uferstreifen
# Resultate: Ein Event Table pro See
# - Attr14 (Attribut Verkehrswege und -flaechen im Uferbereich Nr. 14)

try:
    for See in listeSeen:
        if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_EISENBAHN"):
            arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_EISENBAHN", tempGDBPfad + "Bahn_normal", "OBJEKTART" = \'Normalspur\')
            arcpy.Buffer_analysis(tempGDBPfad + "Bahn_normal", tempGDBPfad +
                "Bahn_normal_buf", "15")
            arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
                "_EISENBAHN", tempGDBPfad + "Bahn_schmal", "OBJEKTART" = \'Schmalspur\')
            arcpy.Buffer_analysis(tempGDBPfad + "Bahn_schmal", tempGDBPfad +
                "Bahn_schmal_buf", "10")
            arcpy.Merge_management([tempGDBPfad + "Bahn_normal_buf", tempGDBPfad +
                "Bahn_schmal_buf"], tempGDBPfad + "Bahn_alle_buffer")
            arcpy.Dissolve_management(tempGDBPfad + "Bahn_alle_buffer", tempGDBPfad +
                "Bahn_buffer_diss", "", "", "SINGLE_PART")
            arcpy.Clip_analysis(tempGDBPfad + "Bahn_buffer_diss", inputPath +
                "30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Bahn_buffer_multi")
            arcpy.MultipartToSinglepart_management(tempGDBPfad + "Bahn_buffer_multi",
                tempGDBPfad + "Bahn_buffer")
            rows = arcpy.SearchCursor(tempGDBPfad + "Bahn_buffer")
            if rows.next():
                arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Bahn_buffer", tempGDBPfad +
                    "Bahn_buffer_p", "Vertex")
                arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Bahn_buffer_p",
                    inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
                    + "Attr14_11_temp", "RID POINT POSITION")
            else:
                arcpy.Delete_management(tempGDBPfad + "Bahn_buffer")
                if arcpy.Exists(tempGDBPfad + "Attr14_11_temp"):
                    arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_11_temp",
                        tempMDBPfad, "Attr14_11_temp")
                    arcpy.CreateTable_management(tempMDBPfad, "Attr14_11", outputPath +
                        "template")
                    arcpy.AddField_management(tempMDBPfad + "Attr14_11", "TYP11", "DOUBLE")
                    constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
                        Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
                        oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"

```

```

        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_11_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 14.11
            cur.execute("INSERT INTO Attr14_11 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
            h = h+1
        conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr14_11", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr14_11", "TYP11", "DOUBLE")

if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_STRASSE"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Autobahn", '"OBJEKTART" = \'Autobahn\'')
    arcpy.Buffer_analysis(tempGDBPfad + "Autobahn", tempGDBPfad +
"Autobahn_buf", "20")
    arcpy.Dissolve_management(tempGDBPfad + "Autobahn_buf", tempGDBPfad +
"Autobahn_buffer_diss", "", "", "SINGLE_PART")
    arcpy.Clip_analysis(tempGDBPfad + "Autobahn_buffer_diss", inputPath +
"30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Autobahn_buffer_multi")
    arcpy.MultipartToSinglepart_management(tempGDBPfad +
"Autobahn_buffer_multi", tempGDBPfad + "Autobahn_buffer")
    rows = arcpy.SearchCursor(tempGDBPfad + "Autobahn_buffer")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Autobahn_buffer",
tempGDBPfad + "Autobahn_buffer_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Autobahn_buffer_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr14_10_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Strasse_buffer")
        if arcpy.Exists(tempGDBPfad + "Attr14_10_temp"):
            arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_10_temp",
tempMDBPfad, "Attr14_10_temp")
            arcpy.CreateTable_management(tempMDBPfad, "Attr14_10", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr14_10", "TYP10", "DOUBLE")
            constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
            conn = adodbapi.connect(constr)
            cur = conn.cursor()
            cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_10_temp GROUP BY ET_IDP, RID")
            result = cur.fetchall()
            h = 1
            for item in result:
                i = int(item[0])
                j = int(item[1])
                k = int(item[2])
                l = 14.10
                cur.execute("INSERT INTO Attr14_10 VALUES (?, ?, ?, ?, ?)", (h, i,
j, k, l))
                h = h+1
            conn.commit()
            cur.close()
            conn.close()
        else:
            arcpy.CreateTable_management(tempMDBPfad, "Attr14_10", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr14_10", "TYP10", "DOUBLE")
if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_STRASSE"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Autostrasse", '"OBJEKTART" = \'Autostrasse\'')

```

```

    arcpy.Buffer_analysis(tempGDBPfad + "Autostrasse", tempGDBPfad +
"Autostrasse_buf", "7.5")
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Strassel0m", '"OBJEKTART" in (\'10m Strasse\',
\'Verbindung\', \'Einfahrt\', \'Ausfahrt\')')
    arcpy.Buffer_analysis(tempGDBPfad + "Strassel0m", tempGDBPfad +
"Strassel0m_buf", "5")
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Strasse6m", '"OBJEKTART" = \'6m Strasse\')
    arcpy.Buffer_analysis(tempGDBPfad + "Strasse6m", tempGDBPfad +
"Strasse6m_buf", "3")
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Strasse4m", '"OBJEKTART" in (\'4m Strasse\',
\'Zufahrt\', \'Dienstzufahrt\')')
    arcpy.Buffer_analysis(tempGDBPfad + "Strasse4m", tempGDBPfad +
"Strasse4m_buf", "2")
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Strasse3m", '"OBJEKTART" = \'3m Strasse\')
    arcpy.Buffer_analysis(tempGDBPfad + "Strasse3m", tempGDBPfad +
"Strasse3m_buf", "1.5")
    arcpy.Merge_management([tempGDBPfad + "Autostrasse_buf", tempGDBPfad +
"Strassel0m_buf", tempGDBPfad + "Strasse6m_buf", tempGDBPfad + "Strasse4m_buf",
tempGDBPfad + "Strasse3m_buf", ], tempGDBPfad + "Strasse_alle_buffer")
    arcpy.Dissolve_management(tempGDBPfad + "Strasse_alle_buffer", tempGDBPfad +
"Strasse_buffer_diss", "", "", "SINGLE_PART")
    arcpy.Clip_analysis(tempGDBPfad + "Strasse_buffer_diss", inputPath +
"30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Strasse_buffer_multi")
    arcpy.MultipartToSinglepart_management(tempGDBPfad + "Strasse_buffer_multi",
tempGDBPfad + "Strasse_buffer")
    rows = arcpy.SearchCursor(tempGDBPfad + "Strasse_buffer")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Strasse_buffer",
tempGDBPfad + "Strasse_buffer_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Strasse_buffer_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr14_9_temp", "RID POINT POSITION")
        else:
            arcpy.Delete_management(tempGDBPfad + "Strasse_buffer")
            if arcpy.Exists(tempGDBPfad + "Attr14_9_temp"):
                arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_9_temp",
tempMDBPfad, "Attr14_9_temp")
                arcpy.CreateTable_management(tempMDBPfad, "Attr14_9", outputPath +
"template")
                arcpy.AddField_management(tempMDBPfad + "Attr14_9", "TYP9", "DOUBLE")
                constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
                conn = adodbapi.connect(constr)
                cur = conn.cursor()
                cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_9_temp GROUP BY ET_IDP, RID")
                result = cur.fetchall()
                h = 1
                for item in result:
                    i = int(item[0])
                    j = int(item[1])
                    k = int(item[2])
                    l = 14.09
                    cur.execute("INSERT INTO Attr14_9 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
                    h = h+1
                    conn.commit()
                cur.close()
                conn.close()
            else:
                arcpy.CreateTable_management(tempMDBPfad, "Attr14_9", outputPath +
"template")
                arcpy.AddField_management(tempMDBPfad + "Attr14_9", "TYP9", "DOUBLE")
if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_STRASSE"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Platz", '"OBJEKTART" = \'Platz\')
    rows = arcpy.SearchCursor(tempGDBPfad + "Platz")
    if rows.next():

```

```

        arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Platz", tempGDBPfad +
"Platz_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Platz_p", inputPath +
"10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad +
"Attr14_7_temp" , "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Platz")
    if arcpy.Exists(tempGDBPfad + "Attr14_7_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_7_temp",
tempMDBPfad, "Attr14_7_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr14_7", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr14_7", "TYP7", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_7_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 14.05
            cur.execute("INSERT INTO Attr14_7 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
            h = h+1
            conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr14_7", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr14_7", "TYP7", "DOUBLE")

if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_STRASSE"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Weg2m", '"OBJEKTART" in (\'2m Weg\', \'2m Weg
Wegfragment\')')
    arcpy.Buffer_analysis(tempGDBPfad + "Weg2m", tempGDBPfad +
"Weg2m_buffer_diss", "1", "FULL", "ROUND", "ALL")
    arcpy.Clip_analysis(tempGDBPfad + "Weg2m_buffer_diss", inputPath +
"30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Weg2m_buffer_multi")
    arcpy.MultipartToSinglepart_management(tempGDBPfad + "Weg2m_buffer_multi",
tempGDBPfad + "Weg2m_buffer")
    rows = arcpy.SearchCursor(tempGDBPfad + "Weg2m_buffer")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Weg2m_buffer", tempGDBPfad
+ "Weg2m_buffer_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Weg2m_buffer_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr14_4_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Weg2m_buffer")
    if arcpy.Exists(tempGDBPfad + "Attr14_4_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_4_temp",
tempMDBPfad, "Attr14_4_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr14_4", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr14_4", "TYP4", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_4_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])

```

```

        k = int(item[2])
        l = 14.04
        cur.execute("INSERT INTO Attr14_4 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
        h = h+1
        conn.commit()
        cur.close()
        conn.close()
    else:
        arcpy.CreateTable_management(tempMDBPfad, "Attr14_4", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr14_4", "TYP4", "DOUBLE")

if arcpy.Exists(inputPath + "20_Geodaten\\TLM.gdb\\" + See + "_STRASSE"):
    arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
"_STRASSE", tempGDBPfad + "Weglm", '"OBJEKTART" in (\\'lm Weg\\', \\'lm Weg
Wegfragment\\', \\'markierte Spur\\')')
    arcpy.Buffer_analysis(tempGDBPfad + "Weglm", tempGDBPfad +
"Weglm_buffer_diss", "0.5", "FULL", "ROUND", "ALL")
    arcpy.Clip_analysis(tempGDBPfad + "Weglm_buffer_diss", inputPath +
"30_Raeume\\raeume.gdb\\ufer_" + See, tempGDBPfad + "Weglm_buffer_multi")
    arcpy.MultipartToSinglepart_management(tempGDBPfad + "Weglm_buffer_multi",
tempGDBPfad + "Weglm_buffer")
    rows = arcpy.SearchCursor(tempGDBPfad + "Weglm_buffer")
    if rows.next():
        arcpy.gp.ET_GPPolygonToPoints(tempGDBPfad + "Weglm_buffer", tempGDBPfad
+ "Weglm_buffer_p", "Vertex")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Weglm_buffer_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "20", tempGDBPfad
+ "Attr14_3_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Weglm_buffer")
        if arcpy.Exists(tempGDBPfad + "Attr14_3_temp"):
            arcpy.TableToTable_conversion(tempGDBPfad + "Attr14_3_temp",
tempMDBPfad, "Attr14_3_temp")
            arcpy.CreateTable_management(tempMDBPfad, "Attr14_3", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr14_3", "TYP3", "DOUBLE")
            constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
            conn = adodbapi.connect(constr)
            cur = conn.cursor()
            cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_),
0) FROM Attr14_3_temp GROUP BY ET_IDP, RID")
            result = cur.fetchall()
            h = 1
            for item in result:
                i = int(item[0])
                j = int(item[1])
                k = int(item[2])
                l = 14.03
                cur.execute("INSERT INTO Attr14_3 VALUES (?, ?, ?, ?, ?)", (h, i, j,
k, l))
                h = h+1
                conn.commit()
                cur.close()
                conn.close()
        else:
            arcpy.CreateTable_management(tempMDBPfad, "Attr14_3", outputPath +
"template")
            arcpy.AddField_management(tempMDBPfad + "Attr14_3", "TYP3", "DOUBLE")

arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
See, tempMDBPfad, "route")
arcpy.CreateTable_management(tempMDBPfad, "Attr14_1", outputPath + "template")
arcpy.AddField_management(tempMDBPfad + "Attr14_1", "TYP1", "DOUBLE")
constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
conn = adodbapi.connect(constr)
cur = conn.cursor()
cur.execute("SELECT round(Shape_Length, 0) FROM route")
result = cur.fetchall()
h = 1

```

```

for item in result:
    i = 1
    j = 0
    k = int(item[0])
    l = 14.01
cur.execute("INSERT INTO Attr14_1 VALUES (?, ?, ?, ?, ?)", (h, i, j, k, l))
conn.commit()
cur.close()
conn.close()

arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_1", "RID LINE VON BIS",
tempMDBPfad + "Attr14_3", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_13", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_13", "RID LINE VON BIS",
tempMDBPfad + "Attr14_4", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_14", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_14", "RID LINE VON BIS",
tempMDBPfad + "Attr14_7", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_17", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_17", "RID LINE VON BIS",
tempMDBPfad + "Attr14_9", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_19", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_19", "RID LINE VON BIS",
tempMDBPfad + "Attr14_10", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_110", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr14_110", "RID LINE VON BIS",
tempMDBPfad + "Attr14_11", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr14_111", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.AddField_management(tempMDBPfad + "Attr14_111", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr14_111")
for row in rows:
    if row.TYP11 == 14.11:
        row.TYP = 14.11
    elif row.TYP10 == 14.10:
        row.TYP = 14.10
    elif row.TYP9 == 14.09:
        row.TYP = 14.09
    elif row.TYP7 == 14.07:
        row.TYP = 14.07
    elif row.TYP4 == 14.04:
        row.TYP = 14.04
    elif row.TYP3 == 14.03:
        row.TYP = 14.03
    elif row.TYP1 == 14.01:
        row.TYP = 14.01
    else:
        row.TYP = 14
    rows.updateRow(row)
del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr14_111", "RID LINE VON BIS",
"TYP", outputPath + "Attr14_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der Verkehrswege und -flaechen aufgetreten."
    print arcpy.GetMessages()

print "Verkehrswege und -flaechen erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

**Anhang 3.09: Ausdehnung der Flachwasserzone**

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
# Ausdehnung der Flachwasserzone
# Resultate: Ein Event Table pro See
# - Attr04 (Attribut Ausdehnung der Flachwasserzone Nr. 4)

try:
    for See in listeSeen:
        arcpy.gp.ET_GPStationPoints(inputPath + "10_Perimeter\\ufer_v.gdb\\general_" +
            See, tempGDBPfad + "aussen_10m", 10)
        arcpy.gp.ET_GPStationPoints(tempGDBPfad + "aussen_10m", inputPath +
            "30_Raeume\\raeume.gdb\\flachwasser_" + See + "_innen_1", tempGDBPfad +
            "innen_10m", 10000, False, True)
        arcpy.Merge_management([tempGDBPfad + "aussen_10m", tempGDBPfad + "innen_10m"],
            tempGDBPfad + "punkte_10m")
        arcpy.gp.ET_GPointToPolyline(tempGDBPfad + "punkte_10m", tempGDBPfad +
            "Attr04_1", "ET_STATION")
        arcpy.AddField_management(tempGDBPfad + "Attr04_1", "RID", "SHORT")
        arcpy.AddField_management(tempGDBPfad + "Attr04_1", "VON", "LONG")
        arcpy.AddField_management(tempGDBPfad + "Attr04_1", "BIS", "LONG")
        arcpy.AddField_management(tempGDBPfad + "Attr04_1", "Typ", "DOUBLE")
        rows = arcpy.UpdateCursor(tempGDBPfad + "Attr04_1")
        for row in rows:
            row.RID = 1
            row.VON = int(row.ET_ID) - 5
            row.BIS = int(row.ET_ID) + 5
            if round(row.Shape_Length, 0) < 10:
                row.Typ = 4.01
            elif round(row.Shape_Length, 0) < 50:
                row.Typ = 4.02
            elif round(row.Shape_Length, 0) < 100:
                row.Typ = 4.03
            elif round(row.Shape_Length, 0) < 250:
                row.Typ = 4.04
            elif round(row.Shape_Length, 0) >= 250:
                row.Typ = 4.05
            else:
                row.Typ = 4
            rows.updateRow(row)
        del row
        del rows
        arcpy.DissolveRouteEvents_lr(tempGDBPfad + "Attr04_1", "RID LINE VON BIS",
            "Typ", outputPath + "Attr04_" + See, "RID LINE VON BIS")

```

```
except:
    print "Ein Fehler beim Erstellen der Ausdehnung der Flachwasserzone aufgetreten."
    print arcpy.GetMessages()

print "Ausdehnung der Flachwasserzone erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")
```

**Anhang 3.10: Strukturen / Anlagen in der Flachwasserzone**

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
#   Strukturen/Anlagen in der Flachwasserzone
#   Resultate: Ein Event Table pro See
#   - Attr15 (Attribut Strukturen/Anlagen in der Flachwasserzone Nr. 15)

try:
    for See in listeSeen:
        arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
            See, tempMDBPfad, "route")
        arcpy.CreateTable_management(tempMDBPfad, "Attr15_01", outputPath + "template")
        arcpy.AddField_management(tempMDBPfad + "Attr15_01", "TYP01", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
            Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
            oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT round(Shape_Length, 0) FROM route")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = 1
            j = 0
            k = int(item[0])
            l = 15.01
        cur.execute("INSERT INTO Attr15_01 VALUES (?, ?, ?, ?, ?)", (h, i, j, k, l))
        conn.commit()
        cur.close()
        conn.close()

        arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_VERBAUUNG MAUER", tempGDBPfad + "Verbauung_allles", "OBJEKTART" =
            '\\Gewaesserverbauung\\')
        arcpy.Clip_analysis(tempGDBPfad + "Verbauung_allles", inputPath +
            "30_Raeume\\raeume.gdb\\flachwasser_" + See, tempGDBPfad + "Verbauung_multi")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Verbauung_multi",
            tempGDBPfad + "Verbauung")
        rows = arcpy.SearchCursor(tempGDBPfad + "Verbauung")
        if rows.next():
            arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Verbauung", tempGDBPfad +
                "Verbauung_p", "Node")

```

```

        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Verbauung_p", inputPath +
"10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "200", tempGDBPfad +
"Attr15_05_temp", "RID POINT POSITION")
    else:
        arcpy.Delete_management(tempGDBPfad + "Verbauung")
    if arcpy.Exists(tempGDBPfad + "Attr15_05_temp"):
        arcpy.TableToTable_conversion(tempGDBPfad + "Attr15_05_temp", tempMDBPfad,
"Attr15_05_temp")
        arcpy.CreateTable_management(tempMDBPfad, "Attr15_05", outputPath +
"template")
        arcpy.AddField_management(tempMDBPfad + "Attr15_05", "TYP05", "DOUBLE")
        constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
        conn = adodbapi.connect(constr)
        cur = conn.cursor()
        cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0)
FROM Attr15_05_temp GROUP BY ET_IDP, RID")
        result = cur.fetchall()
        h = 1
        for item in result:
            i = int(item[0])
            j = int(item[1])
            k = int(item[2])
            l = 15.05
            cur.execute("INSERT INTO Attr15_05 VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
            h = h+1
            conn.commit()
        cur.close()
        conn.close()
        arcpy.OverlayRouteEvents_lr (tempMDBPfad + "Attr15_01", "RID LINE VON BIS",
tempMDBPfad + "Attr15_05", "RID LINE VON BIS", "UNION", tempMDBPfad + "Attr15",
"RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
        arcpy.AddField_management(tempMDBPfad + "Attr15", "TYP", "DOUBLE")
        rows = arcpy.UpdateCursor(tempMDBPfad + "Attr15")
        for row in rows:
            if row.TYP05 == 15.05:
                row.TYP = 15.05
            elif row.TYP01 == 15.01:
                row.TYP = 15.01
            else:
                row.TYP = 15
            rows.updateRow(row)
        del row
        del rows
        arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr15", "RID LINE VON BIS",
"TYP", outputPath + "Attr15_" + See, "RID LINE VON BIS")
    else:
        arcpy.TableToTable_conversion(tempMDBPfad + "Attr15_01", outputPath +
"Attr15_" + See)

except:
    print "Ein Fehler beim Erstellen der Strukturen/Anlagen in der Flachwasserzone
aufgetreten."
    print arcpy.GetMessages()

print "Strukturen/Anlagen in der Flachwasserzonee erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")

```

### Anhang 3.11: Hinterland-Übergangsvegetation

```

# Libraries/allgemeine Einstellungen
import arcpy, os, adodbapi
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")
arcpy.ImportToolbox(r"C:\Program Files (x86)\ET SpatialTechniques\ET GeoWizards 10.2 for
    ArcGIS 10.0\ET GeoWizards.tbx")
env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = "Coordinate Systems/Projected Coordinate
    Systems/National Grids/Europe/CH1903 LV03.prj"

# Pfade
pfad = os.path.abspath("../")
pfad = pfad.replace("\\", r"\\")
inputPath = pfad + "\\20_bearbeitete_Daten\\"
outputPath = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\attribute.gdb\\"
tempGDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempGDB.gdb\\"
tempMDBPfad = pfad + "\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb\\"

# Listen/Variablen
listeSeen = ['b', 'm', 'n']

# file geodatabase erstellen (attribute.gdb wurde bei Attribut 3 erstellt;
    tempGDB.gdb/tempMDB.mdb werden am Ende geloescht)
arcpy.CreateFileGDB_management(inputPath + "40_Attribute\\", "tempGDB.gdb", "10.0")
arcpy.CreatePersonalGDB_management(inputPath + "40_Attribute\\", "tempMDB.mdb", "10.0")

#####
# Hinterland-Uebergangsvegetation
# Resultate: Ein Event Table pro See
# - Attr17 (Attribut Hinterland-Uebergangsvegetation Nr. 17)

try:
    for See in listeSeen:
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", tempGDBPfad + "See_15m", "15")
        arcpy.Buffer_analysis(inputPath + "10_Perimeter\\see_v.gdb\\general_" + See +
            "_f", tempGDBPfad + "See_30m", "30")
        arcpy.Erase_analysis(tempGDBPfad + "See_30m", tempGDBPfad + "See_15m",
            tempGDBPfad + "Hinterland_15m")
        arcpy.PolygonToLine_management(tempGDBPfad + "See_15m", tempGDBPfad +
            "See_15m_1", "IGNORE_NEIGHBORS")
        arcpy.PolygonToLine_management(tempGDBPfad + "See_30m", tempGDBPfad +
            "See_30m_1", "IGNORE_NEIGHBORS")

        arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
            "_BODENBEDECKUNG", tempGDBPfad + "Gehoelz", "OBJEKTART" in ('Wald', 'Wald
            offen', 'Gebueschwald'))
        arcpy.Clip_analysis(tempGDBPfad + "Gehoelz", tempGDBPfad + "Hinterland_15m",
            tempGDBPfad + "Gehoelz_15m_multi")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Gehoelz_15m_multi",
            tempGDBPfad + "Gehoelz_15m")
        arcpy.PolygonToLine_management(tempGDBPfad + "Gehoelz_15m", tempGDBPfad +
            "Gehoelz_15m_1", "IGNORE_NEIGHBORS")
        arcpy.Intersect_analysis([tempGDBPfad + "Gehoelz_15m_1", tempGDBPfad +
            "See_15m_1"], tempGDBPfad + "Gehoelz_innen_multi_1")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Gehoelz_innen_multi_1",
            tempGDBPfad + "Gehoelz_innen_1")
        arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Gehoelz_innen_1", tempGDBPfad +
            "Gehoelz_innen_p", "Node")
        arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Gehoelz_innen_p", inputPath +
            "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "60 Meters", tempGDBPfad +
            "Attr17_1_i_temp", "RID POINT POSITION")
        arcpy.Intersect_analysis([tempGDBPfad + "Gehoelz_15m", tempGDBPfad +
            "See_30m_1"], tempGDBPfad + "Gehoelz_aussen_multi_1")
        arcpy.MultipartToSinglepart_management(tempGDBPfad + "Gehoelz_aussen_multi_1",
            tempGDBPfad + "Gehoelz_aussen_1")
        arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Gehoelz_aussen_1", tempGDBPfad +
            "Gehoelz_aussen_p", "Node")

```

```

arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Gehoelz_aussen_p", inputPath +
"10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "100 Meters", tempGDBPfad +
"Attr17_1_a_temp", "RID POINT POSITION")

arcpy.Select_analysis(inputPath + "20_Geodaten\\TLM.gdb\\" + See +
" _BODENBEDECKUNG", tempGDBPfad + "Feuchtgebiet_f", "'OBJEKTART" in
(\\'Feuchtgebiet\\'))
arcpy.Select_analysis(inputPath + "20_Geodaten\\INV.gdb\\" + See + "_INV",
tempGDBPfad + "Inventare_f", "'AU" = 1 OR "FM" = 1 OR "HM" = 1')
arcpy.Merge_management([tempGDBPfad + "Feuchtgebiet_f", tempGDBPfad +
"Inventare_f"], tempGDBPfad + "Feuchtgebiet")
arcpy.Clip_analysis(tempGDBPfad + "Feuchtgebiet", tempGDBPfad +
"Hinterland_15m", tempGDBPfad + "Feuchtgebiet_15m_multi")
arcpy.MultipartToSinglepart_management(tempGDBPfad + "Feuchtgebiet_15m_multi",
tempGDBPfad + "Feuchtgebiet_15m")
arcpy.PolygonToLine_management(tempGDBPfad + "Feuchtgebiet_15m", tempGDBPfad +
"Feuchtgebiet_15m_l", "IGNORE_NEIGHBORS")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_15m_l", tempGDBPfad +
"See_15m_l"], tempGDBPfad + "Feuchtgebiet_innen_multi_l")
arcpy.MultipartToSinglepart_management(tempGDBPfad +
"Feuchtgebiet_innen_multi_l", tempGDBPfad + "Feuchtgebiet_innen_l")
arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Feuchtgebiet_innen_l", tempGDBPfad
+ "Feuchtgebiet_innen_p", "Node")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_innen_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "60 Meters",
tempGDBPfad + "Attr17_2_i_temp", "RID POINT POSITION")
arcpy.Intersect_analysis([tempGDBPfad + "Feuchtgebiet_15m", tempGDBPfad +
"See_30m_l"], tempGDBPfad + "Feuchtgebiet_aussen_multi_l")
arcpy.MultipartToSinglepart_management(tempGDBPfad +
"Feuchtgebiet_aussen_multi_l", tempGDBPfad + "Feuchtgebiet_aussen_l")
arcpy.gp.ET_GPPolylineToPoints(tempGDBPfad + "Feuchtgebiet_aussen_l",
tempGDBPfad + "Feuchtgebiet_aussen_p", "Node")
arcpy.LocateFeaturesAlongRoutes_lr(tempGDBPfad + "Feuchtgebiet_aussen_p",
inputPath + "10_Perimeter\\ufer_v.gdb\\route_" + See, "RID", "100 Meters",
tempGDBPfad + "Attr17_2_a_temp", "RID POINT POSITION")

arcpy.CreateTable_management(tempMDBPfad, "Attr17_1_i", outputPath + "template")
arcpy.CreateTable_management(tempMDBPfad, "Attr17_1_a", outputPath + "template")
arcpy.CreateTable_management(tempMDBPfad, "Attr17_2_i", outputPath + "template")
arcpy.CreateTable_management(tempMDBPfad, "Attr17_2_a", outputPath + "template")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr17_1_i_temp", tempMDBPfad,
"Attr17_1_i_temp")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr17_1_a_temp", tempMDBPfad,
"Attr17_1_a_temp")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr17_2_i_temp", tempMDBPfad,
"Attr17_2_i_temp")
arcpy.TableToTable_conversion(tempGDBPfad + "Attr17_2_a_temp", tempMDBPfad,
"Attr17_2_a_temp")
arcpy.TableToTable_conversion(inputPath + "10_Perimeter\\ufer_v.gdb\\route_" +
See, tempMDBPfad, "Attr17_5")

arcpy.AddField_management(tempMDBPfad + "Attr17_1_i", "TYP1i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr17_2_i", "TYP2i", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr17_1_a", "TYP1a", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr17_2_a", "TYP2a", "DOUBLE")
arcpy.AddField_management(tempMDBPfad + "Attr17_5", "VON", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr17_5", "BIS", "LONG")
arcpy.AddField_management(tempMDBPfad + "Attr17_5", "TYP5", "DOUBLE")

constr = "Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=C:\\Users\\Anita\\Documents\\UNIGIS\\Masterarbeit\\20_MasterThesis\\30_L
oesungsansatz\\20_bearbeitete_Daten\\40_Attribute\\tempMDB.mdb"
conn = adodbapi.connect(constr)
cur = conn.cursor()
cur.execute("UPDATE Attr17_5 SET VON = 0, BIS = round(Shape_Length, 0), TYP5 =
17.05")
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr17_1_i_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
i = int(item[0])
j = int(item[1])
k = int(item[2])
l = 17.01

```

```

        cur.execute("INSERT INTO Attr17_1_i VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
        h = h+1
        conn.commit()
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr17_1_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 17.01
    cur.execute("INSERT INTO Attr17_1_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
    h = h+1
    conn.commit()
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr17_2_i_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 17.02
    cur.execute("INSERT INTO Attr17_2_i VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
    h = h+1
    conn.commit()
cur.execute("SELECT RID, round(min(POSITION_), 0), round(max(POSITION_), 0) FROM
Attr17_2_a_temp GROUP BY ET_ID, RID")
result = cur.fetchall()
h = 1
for item in result:
    i = int(item[0])
    j = int(item[1])
    k = int(item[2])
    l = 17.02
    cur.execute("INSERT INTO Attr17_2_a VALUES (?, ?, ?, ?, ?)", (h, i, j, k,
1))
    h = h+1
    conn.commit()
cur.close()
conn.close()

arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr17_1_i", "RID LINE VON BIS",
tempMDBPfad + "Attr17_1_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr17_1", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr17_1", "RID LINE VON BIS",
tempMDBPfad + "Attr17_2_i", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr17_12_i", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr17_12_i", "RID LINE VON BIS",
tempMDBPfad + "Attr17_2_a", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr17_12", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")
arcpy.OverlayRouteEvents_lr(tempMDBPfad + "Attr17_12", "RID LINE VON BIS",
tempMDBPfad + "Attr17_5", "RID LINE VON BIS", "UNION", tempMDBPfad +
"Attr17_15", "RID LINE VON BIS", "NO_ZERO", "FIELDS", "INDEX")

arcpy.AddField_management(tempMDBPfad + "Attr17_15", "TYP", "DOUBLE")

rows = arcpy.UpdateCursor(tempMDBPfad + "Attr17_15")
for row in rows:
    if row.TYP2i == 17.02:
        if row.TYP2a == 17.02:
            row.TYP = 17.02
        else:
            row.TYP = 17.04
    elif row.TYP1i == 17.01:
        if row.TYP1a == 17.01:
            row.TYP = 17.01
        else:
            row.TYP = 17.03
    else:
        row.TYP = 17.05

```

```
        rows.updateRow(row)
    del row
del rows
arcpy.DissolveRouteEvents_lr(tempMDBPfad + "Attr17_15", "RID LINE VON BIS",
    "TYP", outputPath + "Attr17_" + See, "RID LINE VON BIS")

except:
    print "Ein Fehler beim Erstellen der Ufersaum-Vegetation aufgetreten."
    print arcpy.GetMessages()

print "Ufersaum-Vegetation erstellt."

arcpy.Delete_management(inputPath + "40_Attribute\\tempGDB.gdb")
arcpy.Delete_management(inputPath + "40_Attribute\\tempMDB.mdb")
```