

6.2 Zusammenfassung der Pfadkommandos

Kommando	Englische Bezeichnung	Erklärung	Beispiel
M, m	<i>moveTo</i>	Setzt den Startpunkt des Pfads fest. Ein Pfad muss zwingend damit beginnen. Befindet sich diese Anweisung im Verlauf der Pfaddefinition, wird der Pfad abgesetzt und startet mit den neuen Koordinaten wieder.	M124 412 M124 412 ... z m12 77
Z, z	<i>closePath</i>	Schließt die Form. Groß-/Kleinschreibung irrelevant.	Z
L, l	<i>lineTo</i>	Zeichnet eine Segment vom aktuellen zum angegebenen Punkt.	L2352 1234
H, h	<i>horizontal lineTo</i>	Zeichnet ein horizontales Segment vom aktuellen Punkt aus.	H234
V, v	<i>vertical lineTo</i>	Zeichnet ein vertikales Segment vom aktuellen Punkt aus.	v89
C, c	<i>curveTo</i>	Zeichnet eine kubische Bézierkurve vom aktuellen Punkt aus.	c0, 0, 3. 8-6. 1, 4. 8-6. 9
S, s	<i>smooth curveTo</i>	Kurzform einer kubischen Bézierkurve, jedoch wird als erster Kontrollpunkt der letzte des vorangegangenen Kommandos genommen.	S 300 10 380 110
Q, q	<i>quadratic curveTo</i>	Zeichnet eine quadratische Bézierkurve .	Q 120 5 230 130
T, t	<i>smooth quadratic Bézier curveTo</i>	Kurzform einer quadratischen Bézierkurve, jedoch wird als erster Kontrollpunkt der letzte des vorangegangenen Kommandos genommen.	T 23 50
A, a	<i>elliptical arc</i>	Zeichnet eine Bogenkurve	A 50 50 0 0 1 4 6

6.3 Komplexe Liniensignaturen

In Kapitel 5.11 haben wir bereits gezeigt, dass es möglich ist, Text am Pfad auszurichten. Die dabei verwendete Technik erlaubt es uns jedoch, komplexe Pfadsignaturen zu gestalten, die mit einem normalen Pfad nicht möglich wären.

Unter komplexen Liniensignaturen verstehen wir Signaturen für linienhafte Elemente, die nicht nur aus einer einfachen Linie wie bei den bisher dargestellten Flüssen oder einer doppelt dargestellten Linie, wie es bei den Straßen der Fall war, bestehen, sondern die zusätzliche Symbole oder Asymmetrien benötigen wie etwa bei der Darstellung von Zäunen oder Kanälen. Das <path>-Element hat in SVG 1.1 leider keine zusätzlichen Attribute, in denen man solche Darstellungen einbinden könnte, deshalb muss man sich mit einem kleinen Trick helfen: der Anordnung von Text am Pfad. Um beispielsweise einen Zaun darzustellen, könnte man eine Reihe von „v“ an einem Pfad entlangreihen.

39710.svg

```
<path id="myPlot" fill="none" stroke="black" stroke-width="1"
      d="M20 20 v 300 h 400 v -300 z"/>
<text id="myFence">
  <textPath xlink:href="#myPlot"> v          v    ...    v          v
  </textPath>
</text>
```

Dafür wird zunächst ein Pfad definiert, der in diesem Fall unsere Grundstücksgrenze darstellt. Als Kindelement des <text>-Elements wird nun ein <textPath>-Element erstellt, welches als Referenz mit dem Attribut `xlink:href` den zuvor erstellten Pfad enthält. Alternativ hätte man den Pfad auch direkt im <textPath>-Element mit dem `d`-Attribut definieren können; dies macht das Dokument aber meist nur unnötig unübersichtlich. Wichtig bei der Definition des Pfads ist die Richtung. In diesem Beispiel ist der Pfad von links oben beginnend nach links unten und weiter gegen den Uhrzeigersinn gezeichnet worden, damit später die „Zaunecken“ in der Fläche liegen. Wäre der Pfad anders herum gezeichnet worden, hätte man die Buchstaben aber einfach nur umzudrehen brauchen. Wenn man das beeinflussen kann, bietet es sich aber an, die Pfade gleich richtig herum zu definieren.

Besonders kompliziert war diese Liniensignatur noch nicht. Anspruchsvoller wäre die Zaunsignatur, wie sie zum Teil für topographische Karten verwendet wird, welche nur aus senkrecht zur Grenzlinie mittig stehenden Strichen besteht. Wenn wir uns diese ebenfalls als Umsetzung mithilfe von einfachen Zeichen vorstellen, könnten die Striche jeweils aus „|“ (Pipe-Zeichen) bestehen, die durch das Attribut `baseline-shift` nach unten gesetzt wurden:

39711.svg

```
<text id="myFence" baseline-shift="-3">
  <textPath xlink:href="#myPlot"> |          |    ...    |          |
  </textPath>
</text>
```

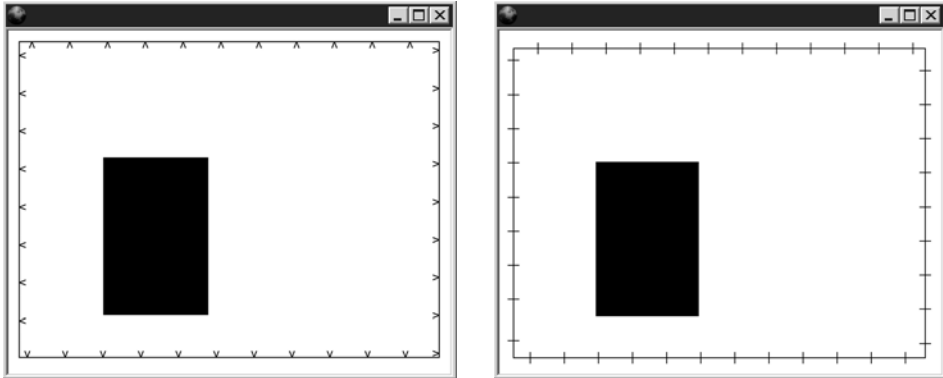


Abb. 6.6: Darstellung einer Zaunsignatur für großmaßstäbige topographische Karten, z. B. 1:5000 (links) und 1:25 000 (rechts). Eine Reihe von „v“ beziehungsweise „|“ wurde am Pfad entlanggeführt und im zweiten Fall um etwas mehr als die Hälfte nach unten, beziehungsweise außen, verschoben (vergleiche Beispiele 39710.svg und 39711.svg).

Eine kleine Herausforderung wäre schon die Signatur für einen Kanal, indem kleine Quadrate entlang des Kanalverlaufs mitgeführt werden. Dafür kann man nun keinen regulären Buchstaben mehr nehmen, aber wie Sie in Kapitel 5.4 erfahren haben, können eigene Zeichen, Graphiken oder Symbole als Schriftzeichen definiert werden. Wir definieren also ein Quadrat so, dass es mit einem bestimmten Buchstaben in einer bestimmten Schriftart aufgerufen wird, also beispielsweise ein q in der Schriftart myFont. Der Buchstabe wird innerhalb des <defs>-Bereichs mithilfe der Elemente (das äußerste Element, mit dem die Schriftart eingeleitet wird), <font-face> (hier wird die Schriftart genauer definiert) und <glyph> (jetzt kommt der eigentliche Buchstabe) definiert (vergleiche Kapitel 5.4). In diesem Beispiel wird der Buchstabe q durch ein **blaues Rechteck** definiert und anschließend am Pfad ausgerichtet.

39712.svg

```
<defs>
  <font>
    <font-face font-family="myFont" units-per-em="100"/>
    <missing-glyph horiz-adv-x="100"/>
    <glyph unicode="q" horiz-adv-x="100" >
      <rect id="myRect" x="35" y="0" width="30" height="30"
        fill="blue"/>
    </glyph>
  </font>
</defs>
```


über hinaus ist zu berücksichtigen, dass die Rechtecke selbst relativ schmal definiert werden sollten, damit sich keine Ecken bei der Anordnung am Pfad abzeichnen, wie es nachfolgende Abbildung 6.8 (rechts, vergleiche 39716.svg), zugegebenermaßen etwas übertrieben, zeigt.

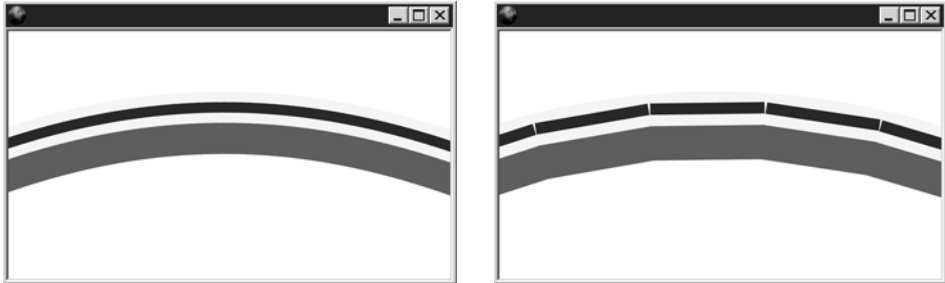


Abb. 6.8: Unregelmäßige Liniensignaturen. Links wurden die Rechtecke schmal genug definiert, damit sie nicht zu sehen sind (vergleiche Beispiel 39715.svg). Hingegen wurden sie rechts zu breit (`width="100"`) definiert: Die einzelnen Rechtecke sind deutlich zu sehen (vergleiche Beispiel 39716.svg).

SVG eignet sich auch für die Darstellung von Wetterkarten, z. B. mit Kalt- und Warmfronten, wie nachfolgendes Beispiel (52361.svg) zeigt. Hier wird noch einmal deutlich, worauf bei der Definition der Glyphen besonders geachtet werden sollte:

- Die Glyphen müssen richtig orientiert sein, da das Umkehren der Pfadrichtung im Nachhinein schwieriger ist als das Umrorientieren der Glyphen. Deshalb wurde hier die Glyphe für die Kaltfront „hängend“ definiert. Gegebenenfalls sollte jeweils eine eigene Glyphe für die verschiedenen Richtungen definiert werden.
- Der gewünschte Abstand zwischen den Symbolen muss im `<glyph>`-Element mit dem Attribut `horiz-adv-x` definiert werden. Dabei muss berücksichtigt werden, dass die Platzierung am Pfad über die untere Mitte der Glyphe erfolgt. Entsprechend sollte das Symbol mittig im angegebenen Abstand (`horiz-adv-x`) platziert werden (vergleiche Beispiel 52362.svg). Manche Viewer richten sich allerdings trotzdem nach der Größe des definierten Symbols. Deshalb ist es ratsam, ein unsichtbares Rechteck mit der gewünschten Breite des Zeichens zusätzlich zu definieren, damit die Platzierung richtig erfolgt.
- Liniensignaturen mit Fonts zu gestalten öffnet einen großen Spielraum, bedenken Sie aber, dass dadurch der Rechenaufwand zum Rendern am Bildschirm dramatisch ansteigt. In Beispiel 52361.svg mit der Wetterkarte ist die Anwendung zulässig. Ähnliches auf das Gewässernetz angewandt wäre hingegen bereits problematisch.

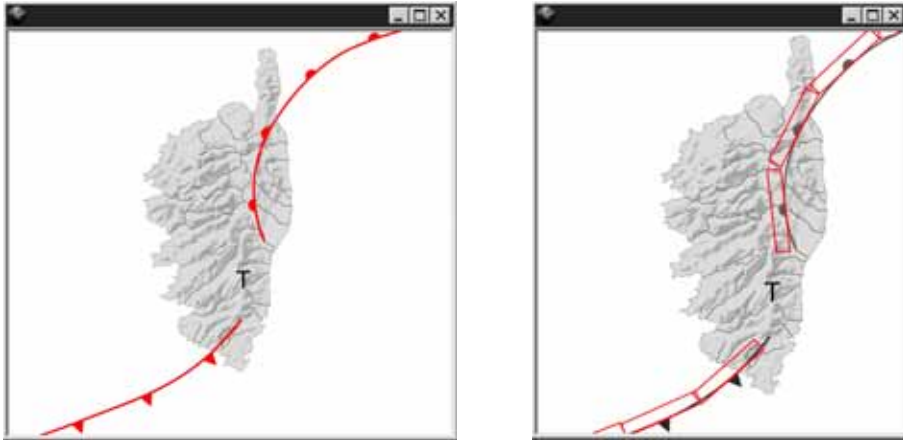


Abb. 6.9: Selbst definierte Glyphen eignen sich auch für Wetterkarten. Hierbei ist allerdings zu beachten, dass die untere Mitte des Glyphen als Ankerpunkt an der Linie verwendet wird (vergleiche Beispiel 52361.svg und 52362.svg).

6.4 Marker für Koordinatenachsen

Koordinatensysteme für Diagramme werden im Allgemeinen kartesisch mit zwei Koordinatenachsen dargestellt, die jeweils verschiedene Einheiten darstellen. Häufig wird die x-Achse (in der Waagerechten) für die Darstellung verschiedener Zeitabschnitte verwendet, während auf der y-Achse (in der Senkrechten) die jeweiligen Werte dargestellt werden (vergleiche unser erstes Diagramm auf Seite 56). Diese Achsen werden mit dem `<line>`-Element oder auch mit Pfaden definiert. Soll symbolisiert werden, dass es sich um Einheiten handelt, die theoretisch weiter vergrößert werden können (wie das Aufführen verschiedener Zeitpunkte), so ist es üblich, die jeweils ins „Unendliche“ weisenden Enden der Achsen mit Pfeilspitzen zu versehen. Und wie Sie bereits in verschiedenen Beispielen gesehen haben, gibt es auch in SVG die Möglichkeit, Linien mit solchen Pfeilspitzen auszustatten. Allerdings müssen das nicht immer Pfeilspitzen sein, es könnten auf die gleiche Weise beispielsweise auch die Zeitpunkte in einem Liniendiagramm als Stützpunkte dargestellt werden. Zunächst aber zum Koordinatensystem.

6.4.1 Pfeile am Koordinatensystem

Bei Koordinatensystemen werden die Pfeilspitzen üblicherweise mit dem Attribut `marker-end` ausgeführt. In ihm wird die zu verwendende Pfeilspitze z. B. mit dem Wert `url(#myTriangle)` angegeben, wobei `myTriangle` den Namen des `<marker>`-Elements angibt. Das Gleiche gilt für die Attribute `marker-start` und `marker-mid`. Wo das Ende oder der Anfang einer Linie ist, ergibt sich durch die Definition der jeweiligen Elemente. Mit dem Attribut `marker-mid` wird an jedem Stützpunkt der