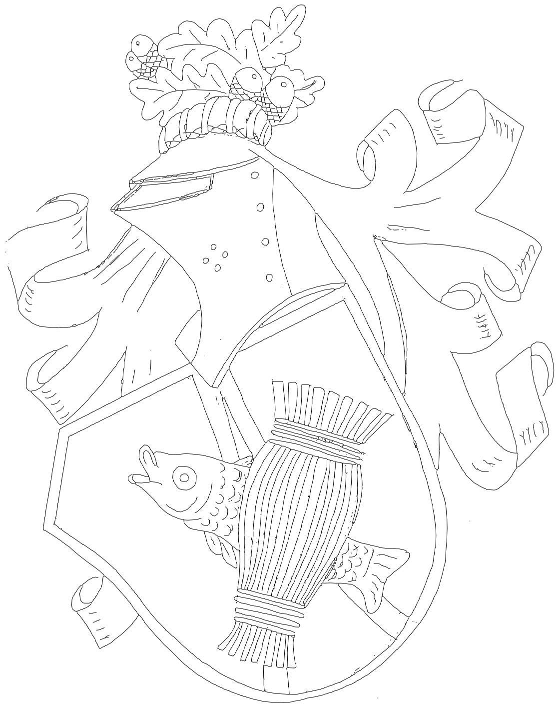

Dieses Bild habe ich von einer Bleistiftzeichnung eingescannt (mit 300 dpi, wenn es jemanden interessiert) und dann als PCX-Bild (ist bloß noch so ein Rasterbildformat) abgespeichert. Da brauchte es dann 300 KB Speicher. Ganz ordentlich. Wenn man es nun in ein PostScript-Rasterbild konvertiert[47], so frißt das entstehende PostScript-Rasterbild nicht weniger als 2500 KB, braucht also etwa achtmal so viel Speicher wie das PCX-Rasterbild. Da platzt einem doch die Platte! Richtig, und wenn Sie sich das Bild mal ansehen, werden Sie feststellen, daß es nur aus Linien besteht. Keine ausgefüllten Flächen, nicht bunt, also ⟶ vektorisieren! In der Tat ist obiges Beispiel auch nicht das direkt konvertierte Rasterbild, sondern ein Rasterbild, das mit LS-Vektor vektorisiert wurde[48]. Nach der Vektorisierung brauchte es nur noch 160 KB Speicher und sieht noch genauso gut aus wie die originale Bleistiftzeichnung.

Mit diesen wenigen, grundlegenden Bemerkungen über Bilder, Bildkonvertierung und Einbindung von Bildern in TEX will ich es dann auch mal gut sein lassen. Für weitere Bildbeispiele und Anwendungsbeispiele verweise ich auf einen Artikel, den Sie als Anhang dieser Anleitung finden.

Nur eine Warnung will ich noch aussprechen: Unterschätzen Sie niemals den Speicherplatzbedarf der Rasterbildverarbeitung mit PostScript. Als Beipspiel will ich dieses Kapitel hier

[47] In diesem Falle wurde die probeweise Konvertierung mit dem kommerziellen Programm HiJaak ausgeführt, da ALCHEMY in der ShareWare-Version aufgrund der eingebauten Beschränkungen streikt und GraphicWorkshop für die Konvertierung eines solchen Rasterbildmonstrums etwa 25 Minuten bräuchte (Zeit gemessen bei einem vergleichbaren Bild auf einem 386DX-25-Rechner). Man hätte es natürlich auch mit der Vollversion von ALCHEMY konvertieren können.

[48] Genaue Prozeßbeschreibung: Das eingescannte PCX-Bild wurde in LS-Vektor zunächst einfach gefiltert (9 Minuten), dann skelettiert (21 Minuten), danach mit dem Spezialfilterprogramm in LS-Vektor behandelt (16 Minuten) und dann schließlich vektorisiert (11 Minuten) und anschließend zweimal mit Schwellenwert 0.5 gepackt (6 Minuten). Dieses Vorgehen kann ich als universell anwendbar weiterempfehlen. Die Zeiten wurden gemessen auf einem 386DX-25-Rechner.