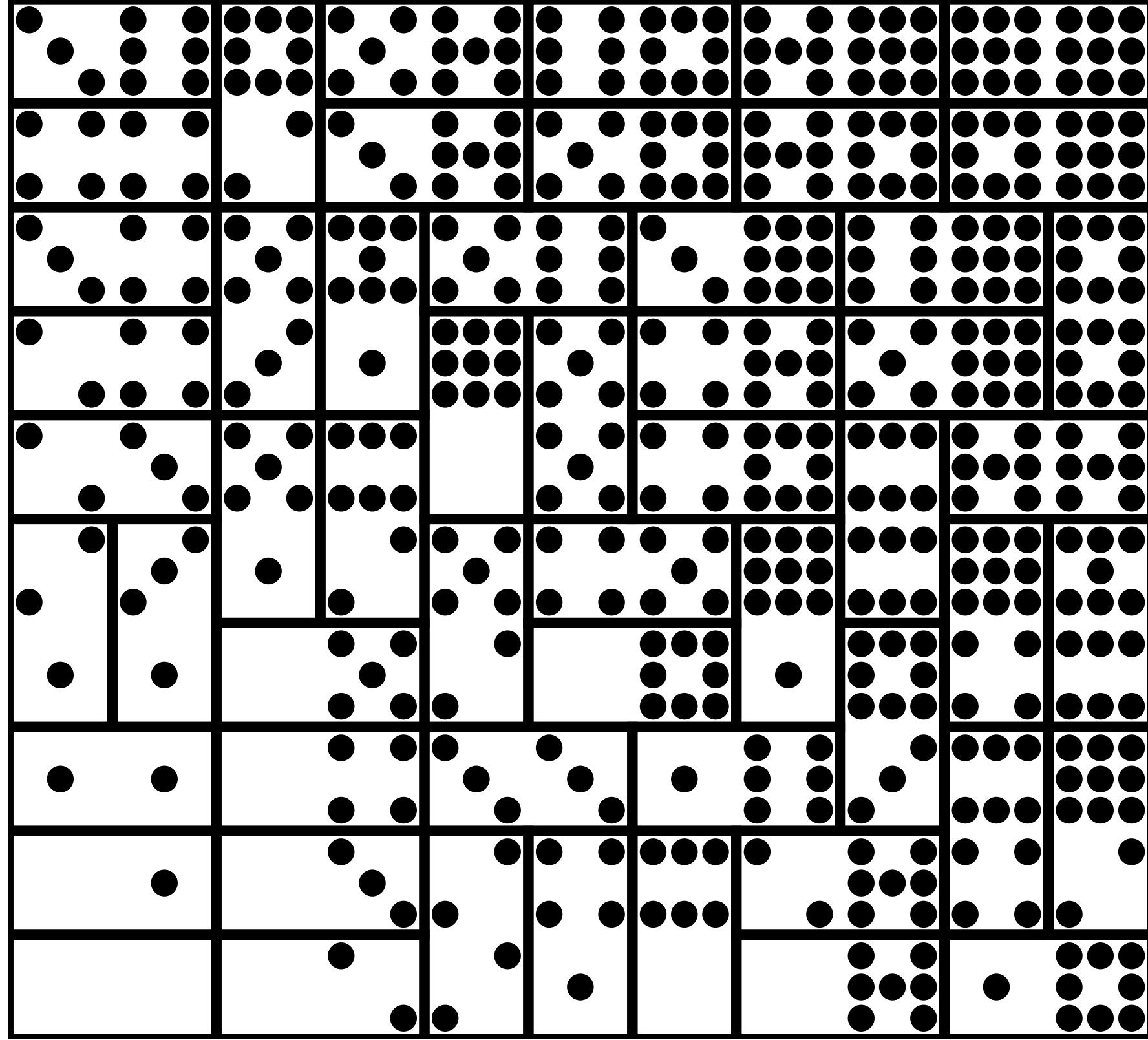


# Wie ein Dominobild entsteht

ANJA FISCHER, FRANK FISCHER

## IDEE

Ein großes Dominoset besteht aus 55 Dominosteinen von  $0 \times 0$ , über  $5 \times 5$  bis  $9 \times 9$ , wobei sich jeder Stein aus zwei quadratischen Teilen zusammensetzt.



Wir wollen jetzt die Dominosteine so anordnen, dass aus einiger Entfernung betrachtet, die Punkte der Dominosteine unser Porträt oder Bild darstellen. Dabei nutzen wir aus, dass bei weißen Steinen mit schwarzen Punkten z. B. eine 9 sehr dunkel, eine 5 mittelhell und eine 0 sehr hell erscheint. Insgesamt wollen wir vollständige Dominosets verwenden, d. h., von jedem Dominosteintyp gibt es gleich viele. Bei einer Bildgröße von  $88 \times 110$  nutzen wir z. B. 88 vollständige Dominosets, da jedes Set genau 110 Pixel des Bildes abdeckt.

## MATHEMATISCHE SICHT

Sei  $m \times n$  die Größe des Porträts und  $k$  die Anzahl der Dominosets, bestehend aus 55 Dominos. Dann wollen wir für jedes der  $k \cdot 55$  Dominos festlegen, an welcher Position sich seine linke untere Ecke befindet und wie es gedreht ist ( $\begin{smallmatrix} \square & * \\ * & \square \end{smallmatrix}$  oder  $\begin{smallmatrix} * & \square \\ \square & * \end{smallmatrix}$  oder  $\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}$ ). Wir bewerten die Helligkeitsabweichung der entsprechenden Pixel zu den beiden Feldern eines Dominosteins so, dass geringe Abweichungen in der Helligkeit wenig, hohe Abweichungen aber stark (quadratisch) bestraft werden. Wir suchen also eine Dominolösung mit einer minimalen quadratischen Graustufenabweichung, sodass

- jeder der 55 Dominosteine genau  $k$  Mal verwendet wird;
- jedes Pixel des Porträts von einer Hälfte eines Dominosteins genau einmal überdeckt wird.

Damit das Problem eine Lösung besitzt, muss für die Gesamtpixelzahl des Bildes  $m \times n = 2 \cdot 55 \cdot k$  gelten.

Diese Fragestellung lässt sich als mathematisches Optimierungsproblem schreiben, welches wir mithilfe von moderner Optimierungssoftware lösen können. Die eingesetzten Algorithmen liefern uns dann automatisch eine bestmögliche Lösung des Modells, welche wir zum Schluss wieder in ein Dominobild rückübersetzen können.

## MATHEMATISCHES MODELL

**Mengen und Parameter:**

- Arten an Dominosteinen  $D = \{1, \dots, 55\}$ ;
- Orientierungen eines Steins  $O = \{1, 2, 3, 4\}$ ;
- Menge an Positionen/Pixeln  $P = \{(i, j) : i \in \{1, \dots, m\}, j \in \{1, \dots, n\}\}$  (linke untere Ecke eines Steins bzw. Name eines Pixels);
- $c_{dop}, d \in D, o \in O, p \in P \dots$  Bewertung der Graustufenabweichung, falls Stein  $d$  in Orientierung  $o$  an Position  $p$ ;
- $u_{dopp'} \in \{0, 1\}, d \in D, o \in O, p, p' \in P$ , mit  $u_{dopp'} = 1$  genau dann, wenn Stein  $d$  in Orientierung  $o$  an Position  $p$  Pixel  $p'$  überdeckt;
- $v_{dop} \in \{0, 1\}, d \in D, o \in O, p \in P$ , mit  $v_{dop} = 1$  genau dann, wenn Stein  $d$  in Orientierung  $o$  an Position  $p$  vollständig im Bild enthalten ist.

**Variablen:**  $x_{dop} \in \{0, 1\}, d \in D, o \in O, p \in P$ , mit  $x_{dop} = 1$  genau dann, wenn sich Stein  $d$  in Orientierung  $o$  an Position  $p$  befindet, sonst  $x_{dop} = 0$ .

**Modell:**

$$\begin{aligned} \min & \sum_{d \in D} \sum_{o \in O} \sum_{p \in P} c_{dop} x_{dop} \\ \text{s. t.} & \sum_{o \in O} \sum_{p \in P} x_{dop} = k, \quad d \in D, \\ & \sum_{d \in D} \sum_{o \in O} \sum_{p \in P} u_{dopp'} x_{dop} = 1, \quad p' \in P, \\ & x_{dop} \in \{0, v_{dop}\}, \quad (d, o, p) \in D \times O \times P \end{aligned}$$

## VISUALISIERUNG DER SCHRITTE

