

Multimedia-Technik WS 2001/2002

Musterlösung Übungsblatt 4

Aufgabe 1: Videokompression mit MPEG

Gegeben sind die folgenden aufeinanderfolgenden Bilder einer Videosequenz (Bild 84, Bild 85).



Berechnen Sie die Bewegungsvektoren von vier 16×16 -Makroblöcken in Bild 85 in Bezug auf Bild 84. Die jeweilige obere linke Ecke der Blöcke befindet sich an den Bildpositionen $(32, 32)$, $(32, 128)$, $(256, 32)$ und $(256, 128)$.

Verwenden Sie das Full-Search-Verfahren mit einem Suchradius von 15 Pixel.

Aufgabe 1: Abstandsmaß für Makroblöcke

Gegeben sind zwei $N \times N$ -Makroblöcke A und B. Damit errechnet sich die Distanz dieser Makroblöcke folgendermaßen:

$$E = \frac{1}{N^2} \sum_{i=0}^N \sum_{j=0}^N |A(i, j) - B(i, j)|$$

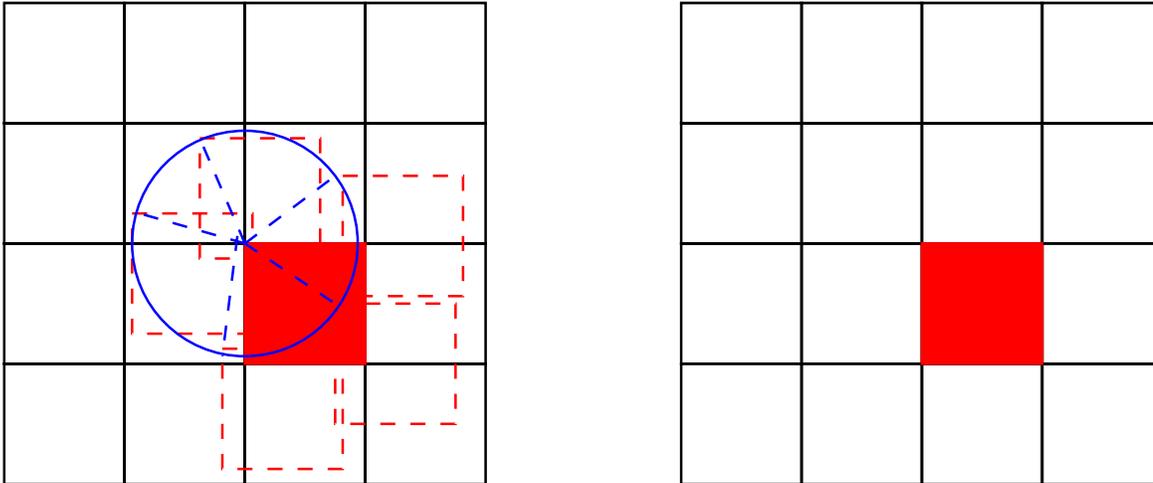
```
double match(Frame& previousFrame, int pFx, int pFy,
             Frame& currentFrame, int cFx, int cFy)
{
    double distance = 0.0;

    for (int x = 0; x < BLOCKSIZE; x++) {
        for (int y = 0; y < BLOCKSIZE; y++) {
            distance +=
                abs(previousFrame.getPixel(pFx+x, pFy+y) -
                  currentFrame.getPixel(cFx+x, cFy+y));
        }
    }

    return 1.0 / (BLOCKSIZE * BLOCKSIZE) * distance;
}
```

Aufgabe 1: Full search

Suche nach dem ähnlichsten Block innerhalb eines bestimmten Suchradius.



```
void fullSearch (...)
{
    double minDistance = double(INT_MAX);
    for (int dx = -radius; dx <= radius; dx++) {
        for (int dy = -radius; dy <= radius; dy++) {
            double distance =
                match(previousFrame,
                    blockX+dx, blockY+dy,
                    currentFrame,
                    blockX, blockY);
            if (distance < minDistance) {
                minDistance = distance;
                vectorX = dx;
                vectorY = dy;
            }
        }
    }
}
```

}

}

Aufgabe 1: Ergebnisse Teilaufgabe a

1. Position für den Block (32, 32): (39,32)
2. Position für den Block (32, 128): (38,128)
3. Position für den Block (256, 32): (263,32)
4. Position für den Block (256, 128): (263,128)

Aufgabe 1: Kodierung von P-Frames (1)

Originalbild aus einer Videosequenz:



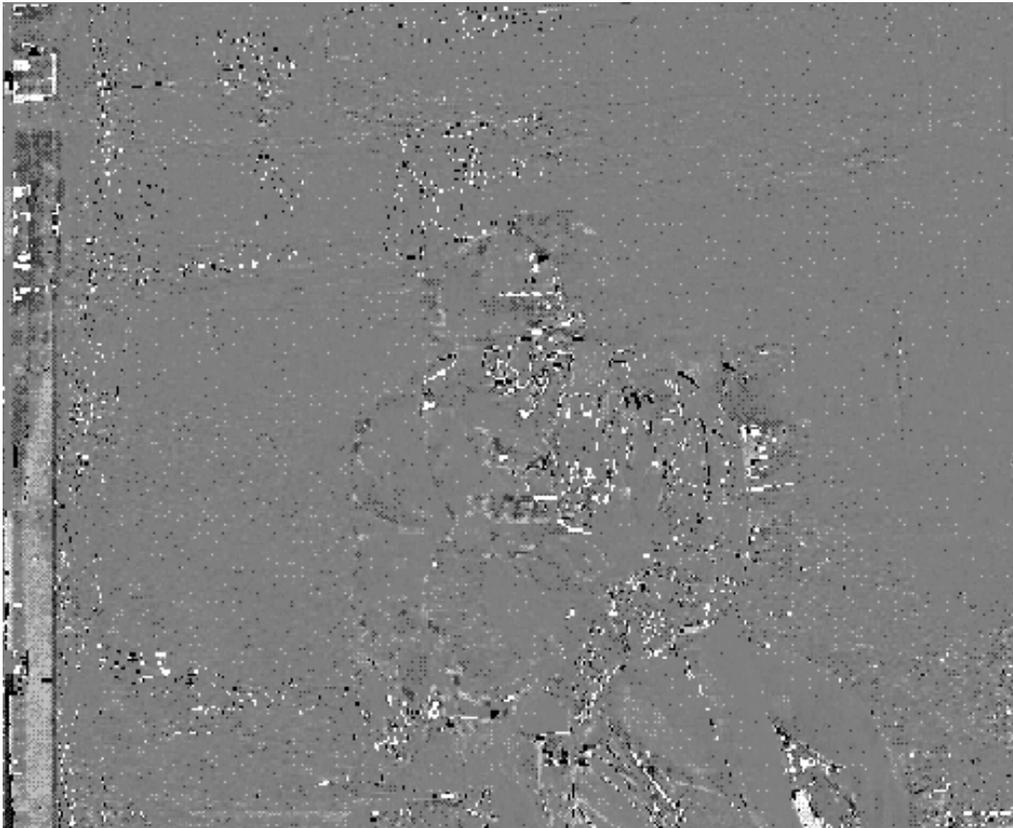
Aufgabe 1: Kodierung von P-Frames (2)

Differenzbild D aus zwei aufeinanderfolgenden Bildern F und G, d.h. $D(i, j) = |F(i, j) - G(i, j)|$:



Aufgabe 1: Kodierung von P-Frames (3)

Differenzbild D aus zwei Bildern F und G, wobei G aus F unter Verwendung der Bewegungsvektoren errechnet wurde:

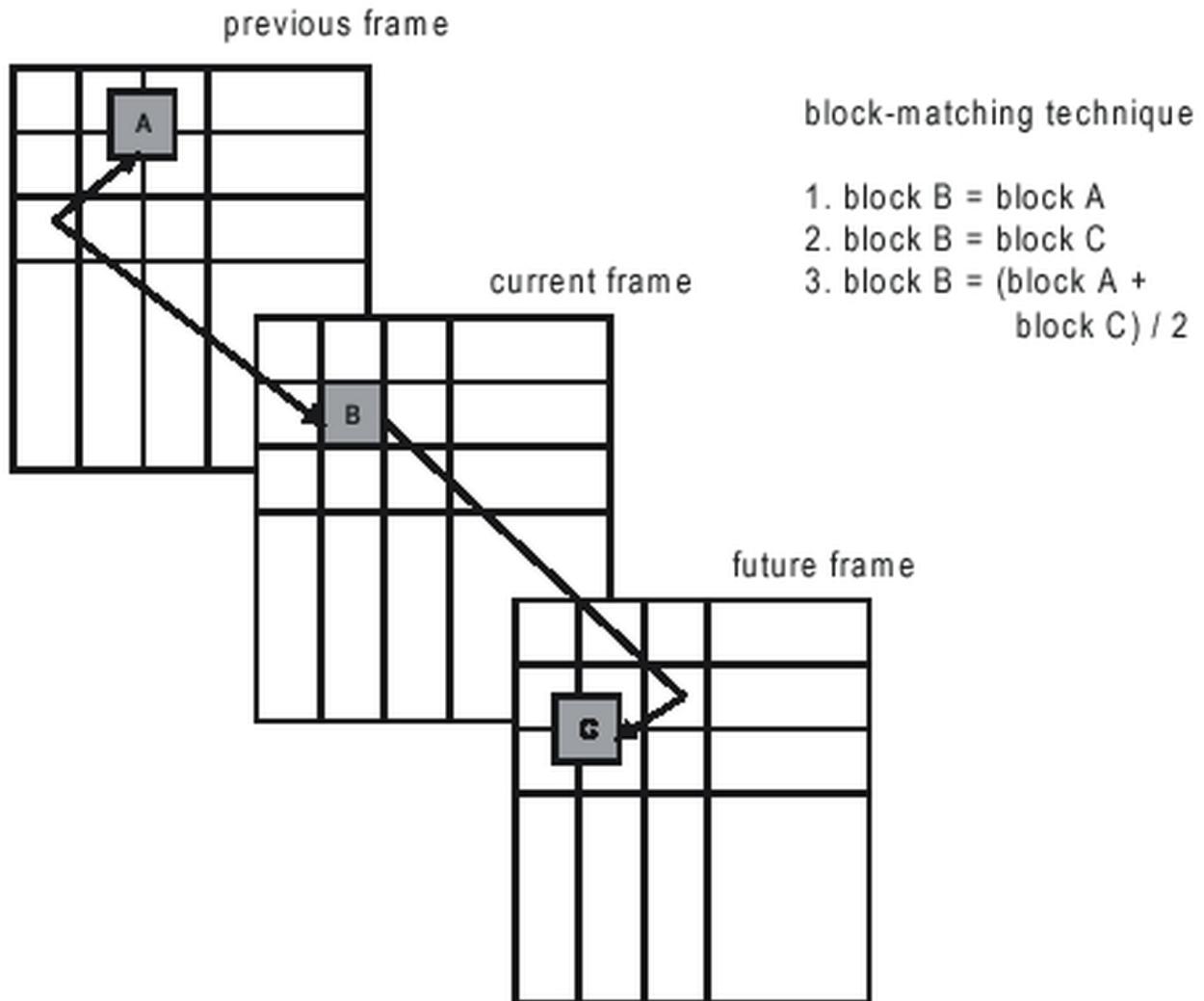


Dieses Differenzbild wird zusätzlich zu den Bewegungsvektoren kodiert, d.h. der übliche Kodierungsprozeß aus DCT, Quantisierung, Lauflängen- und Entropiekodierung wird verwendet.

Was passiert, wenn sich kein „geeigneter“ Bewegungsvektor finden läßt?

Aufgabe 1: Kodierung von B-Frames

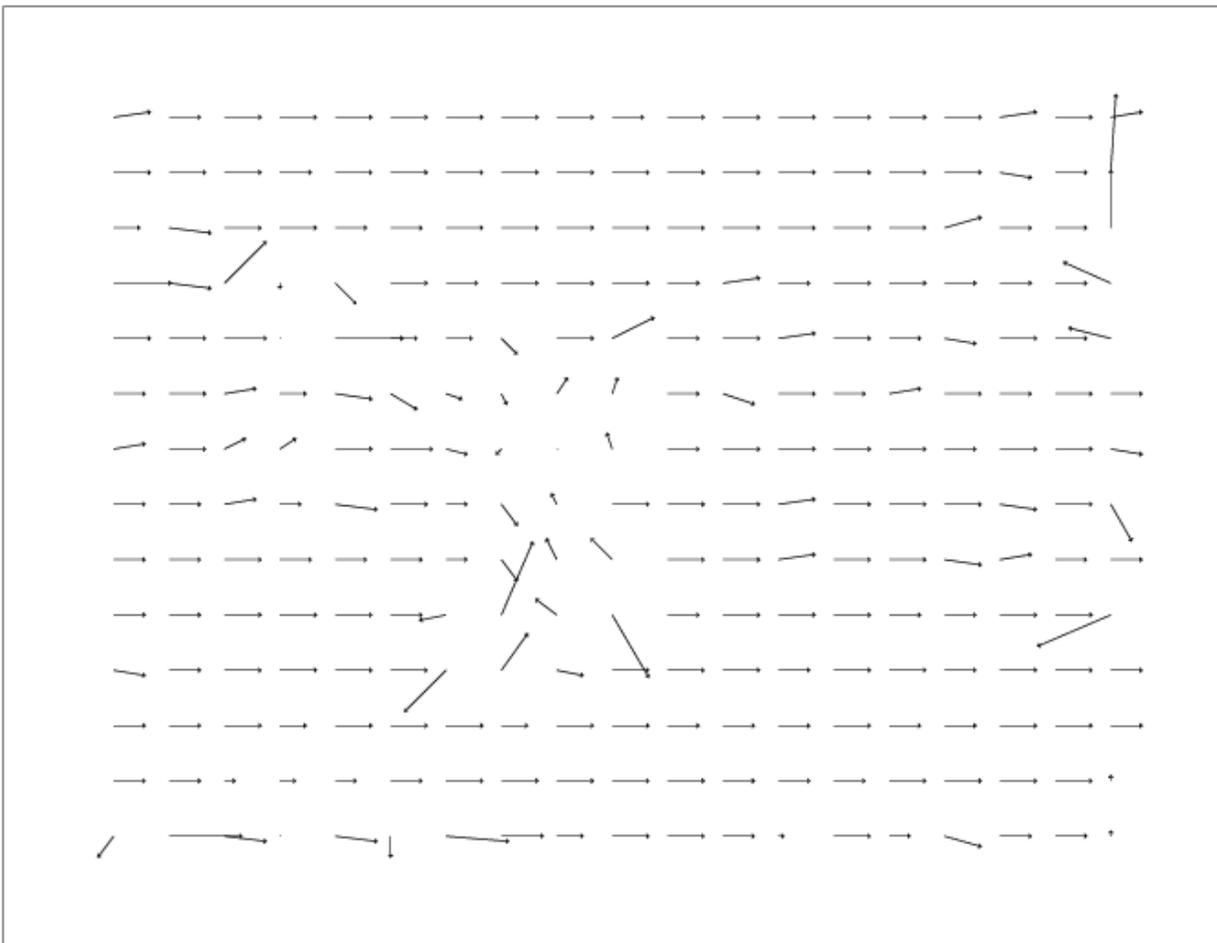
B-Frames können sowohl Referenzen in die Zukunft als auch in die Vergangenheit enthalten.



Aufgabe 1: Kamerabewegungen

Aus der Konstellation der Bewegungsvektoren können Rückschlüsse auf die Kamerabewegung gezogen werden.

Z.B. Bewegungsvektoren der „Mogli“-Bilder weisen auf einen Kameraschwenk (pan) hin.



Die Kamerabewegungen “tilt”(vertikaler Schwenk), “zoom in” und “zoom out” lassen sich ebenfalls auf diese Weise indentifizieren.

Aufgabe 2: Wavelet-Kodierung

Gegeben ist ein Audiostrom. Die einzelnen Einträge sind dabei 8-Bit-Samples. Zerlegen Sie dieses Audiosignal mittels der Wavelet-Transformation in Hochpass- und Tiefpassanteile (eine Iteration). Verwenden Sie dazu ein Daubechies-2-Wavelet, das durch die folgenden vier Filter charakterisiert wird:

1. Approximationsfilter für den Mittelwert:

$$\begin{aligned} L &= \{L_1, L_2, L_3, L_4\} \\ &= \{0.482963, 0.836516, 0.224144, -0.12941\} \end{aligned}$$

2. Detailfilter für Abweichung:

$$\begin{aligned} H &= \{H_1, H_2, H_3, H_4\} \\ &= \{-0.12941, -0.224144, 0.836516, -0.482963\} \end{aligned}$$

3. Synthesefilter für den 1. Wert:

$$\begin{aligned} IL &= \{IL_1, IL_2, IL_3, IL_4\} \\ &= \{0.224144, 0.836516, 0.482963, -0.12941\} \end{aligned}$$

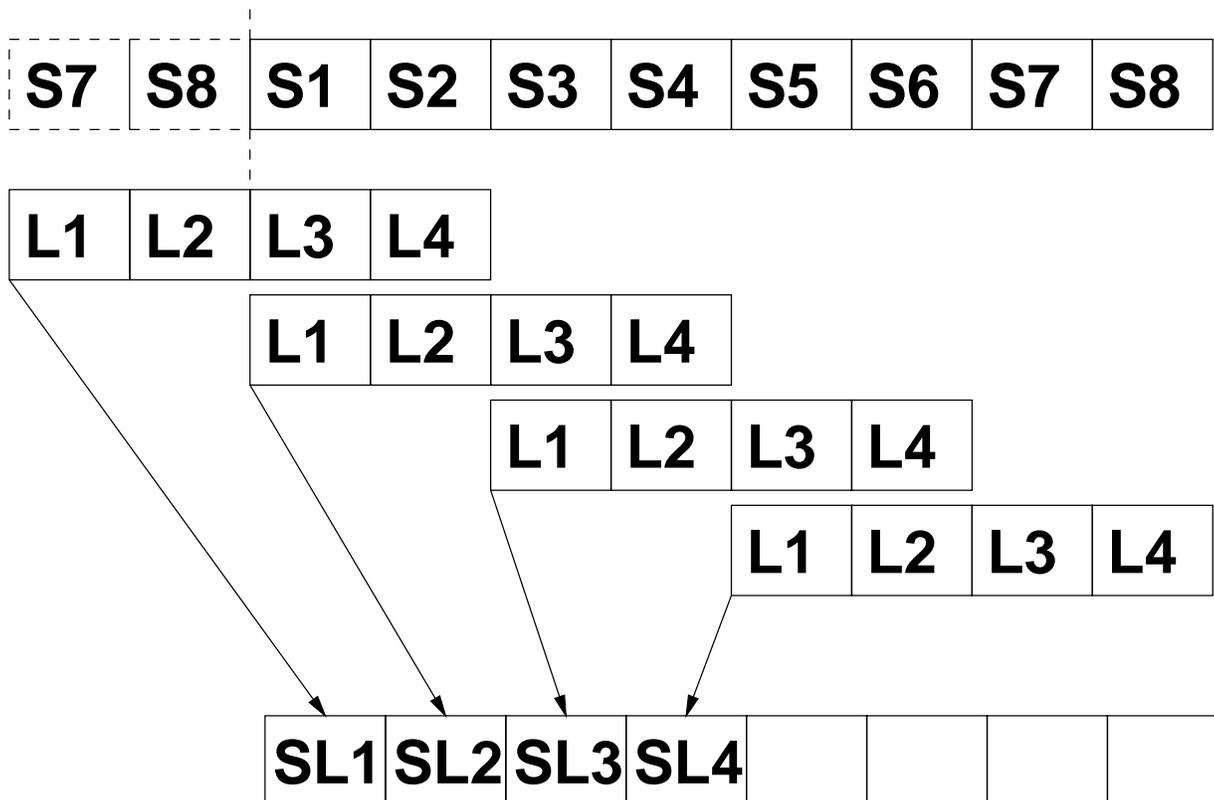
4. Synthesefilter für den 2. Wert:

$$\begin{aligned} IH &= \{IH_1, IH_2, IH_3, IH_4\} \\ &= \{-0.12941, -0.482963, 0.836516, -0.224144\} \end{aligned}$$

Aufgabe 2: Analyse - Lowpass-Filter

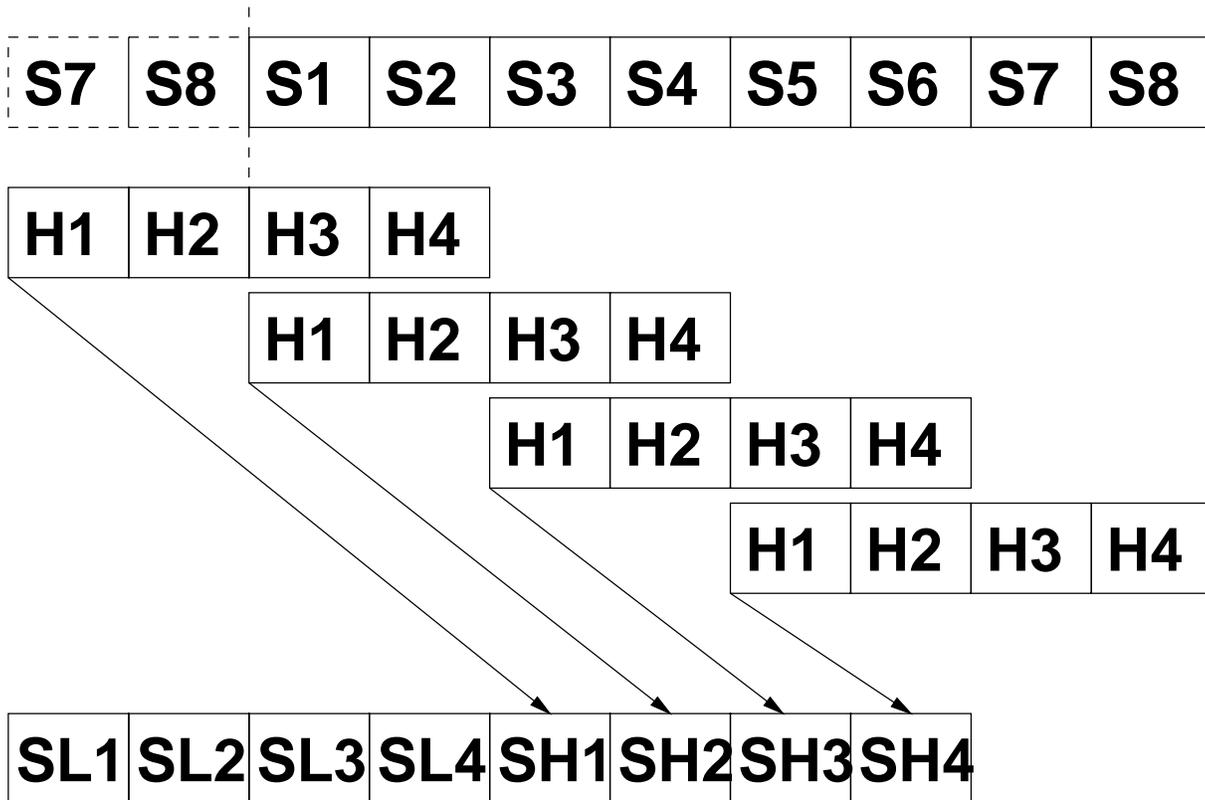
Berechnung der Lowpass-Anteile des Signals S_1, S_2, \dots, S_8 mit dem Analyse-Filter L_1, L_2, L_3, L_4 .

- Periodische Fortsetzung des Signals, d.h. $S_0 = S_8, S_{-1} = S_7$
- Alternativen: Spiegeln, Fortsetzen oder Auffüllen der Randwerte



Aufgabe 2: Analyse- Highpass-Filter

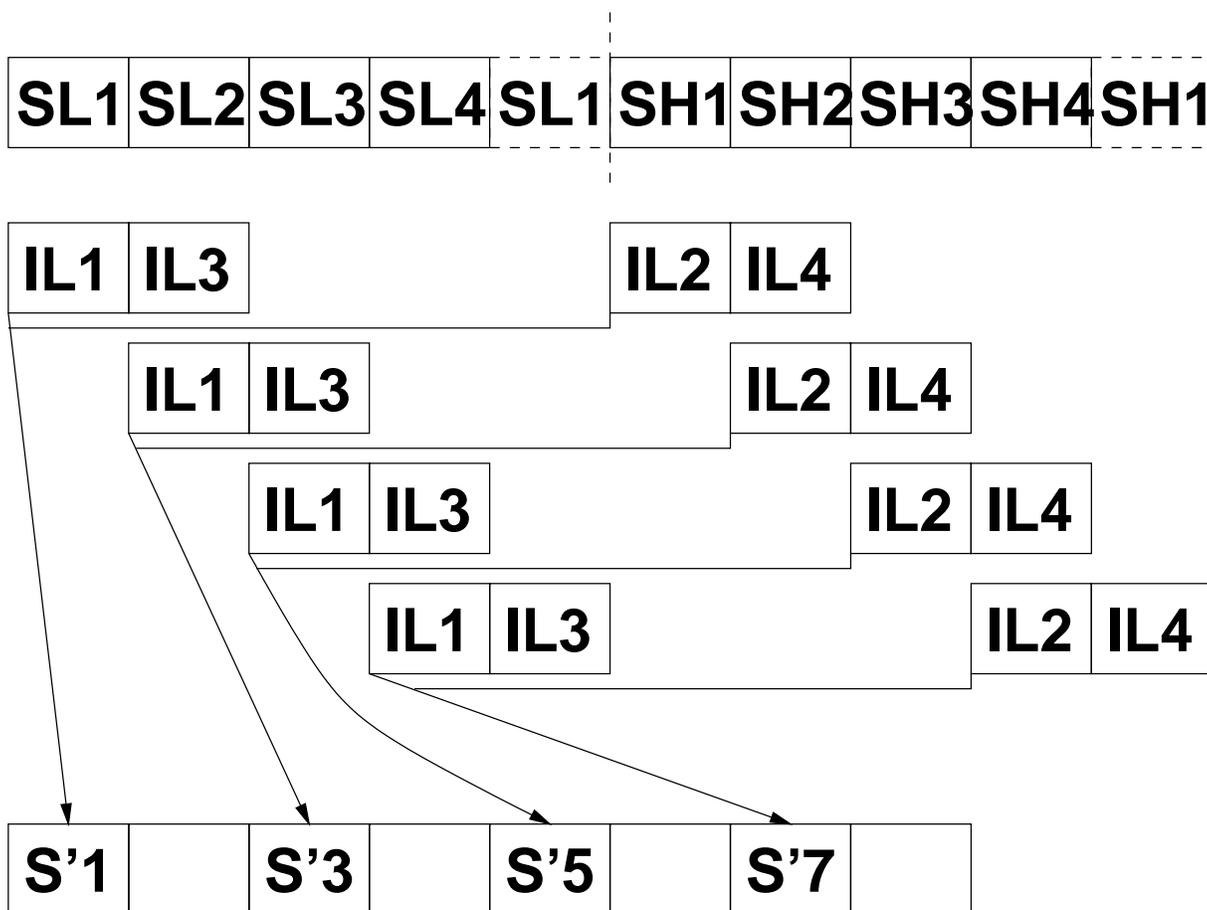
Berechnung der Highpass-Anteile des Signals S_1, S_2, \dots, S_8 mit dem Analyse-Filter H_1, H_2, H_3, H_4 .



Aufgabe 2: Synthese - Lowpass-Filter

Synthetisierung des Signals S'_1, S'_3, S'_5, S'_7 mit dem Synthese-Filter IL_1, IL_2, IL_3, IL_4 .

- Periodische Fortsetzung der Lowpass- und Highpass-Koeffizienten



Aufgabe 2: Synthese - Highpass-Filter

Synthetisierung des Signals S'_2, S'_4, S'_6, S'_8 mit dem Synthese-Filter IH_1, IH_2, IH_3, IH_4 .

