当前帧中的2x2像素块如图4所示，

其在参考帧中的同位置块如图5中的阴影区域所示。

给定一个1像素的搜索窗口，

如果失真标准是均方误差（Mean Square Error， MSE），
则在图5中找到最匹配的运动矢量和相应的块。

3. (a) A block of 2×2 pixels in the current frame is shown in Figure 4 and its co-located block in the reference frame is shown by the shaded area in Figure 5. Given a search window of ± 1 pixels, find the best-matched motion vector and the corresponding block in Figure 5, if the distortion criterion is Mean Square Error (MSE).

70	85
60	75

Figure 4

80	70	50	60
60	55	70	80
60	60	70	60
70	85	70	60

Figure 5

(10 Marks)

(b) In motion estimation, explain the main reason why half-pel accurate motion estimation could achieve better prediction performance than integer-pel accurate motion estimation. With the help of a simple diagram, explain the bilinear interpolation method used to obtain half-pel values in a 2×2 image block.

在运动估计中，解释半精度运动估计比整数精度运动估计预测性能更好的主要原因。
借助一个简单的图表，解释在2x2图像块中获取半像素值的双线性插值方法。

(10 Marks)

Answer to Problem 3(a):问题3(a)的答案:

Objective: Find the best-matched motion vector and the corresponding block in Figure 5, given the block in Figure 4 and a search window of ±1 pixel, using the Mean Square Error (MSE) criterion.

目标：在给定图 4 中的块和 ±1 像素的搜索窗口的情况下，使用均方误差 (MSE) 标准找到最匹配的运动矢量和图 5 中的相应块。

Given:鉴于:

- **Current Frame Block (Figure 4):当前帧块（图 4）：**

$$\begin{bmatrix} 70 & 85 \\ 60 & 75 \end{bmatrix}$$

- **Reference Frame (Figure 5):参考系（图5）：**

$$\begin{bmatrix} 80 & 70 & 50 & 60 \\ 60 & 55 & 70 & 80 \\ 60 & 60 & 70 & 60 \\ 70 & 85 & 70 & 60 \end{bmatrix}$$

- **Co-located Block in Reference Frame (Shaded Area):参考系中的同位块（阴影区域）：**

$$\begin{bmatrix} 55 & 70 \\ 60 & 70 \end{bmatrix}$$

Steps:步骤:

1. **Identify the Co-located Block Position:确定共置块位置:**

   - The co-located block starts at row 2, column 2 in the reference frame.

     共置块从参考系中的第 2 行、第 2 列开始。

   - This block spans from positions (2,2) to (3,3).该块从位置 (2,2) 跨越到 (3,3)。

2. **Define the Search Window:定义搜索窗口:**

   - With a search window of ±1 pixel, the search area includes positions from row 1 to 3 and column 1 to 3.

     搜索窗口为 ±1 像素，搜索区域包括第 1 行至第 3 行和第 1 至第 3 列的位置。

   - Possible starting positions for candidate blocks are (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), and (3,3).

     候选块的可能起始位置为 (1,1)、(1,2)、(1,3)、(2,1)、(2,2)、(2,3)、(3,1)、(3,2) 和 (3,3)。

3. **List Candidate Blocks and Compute MSE:**
   - **Candidate Blocks:**
     - **Block 1 (1,1):** $\begin{bmatrix} 80 & 70 \\ 60 & 55 \end{bmatrix}$
     - **Block 2 (1,2):** $\begin{bmatrix} 70 & 50 \\ 55 & 70 \end{bmatrix}$
     - **Block 3 (1,3):** $\begin{bmatrix} 50 & 60 \\ 70 & 80 \end{bmatrix}$
     - **Block 4 (2,1):** $\begin{bmatrix} 60 & 55 \\ 60 & 60 \end{bmatrix}$
     - **Block 5 (2,2):** $\begin{bmatrix} 55 & 70 \\ 60 & 70 \end{bmatrix}$ (Co-located Block)
     - **Block 6 (2,3):** $\begin{bmatrix} 70 & 80 \\ 70 & 60 \end{bmatrix}$
     - **Block 7 (3,1):** $\begin{bmatrix} 60 & 60 \\ 70 & 85 \end{bmatrix}$
     - **Block 8 (3,2):** $\begin{bmatrix} 60 & 70 \\ 85 & 70 \end{bmatrix}$
     - **Block 9 (3,3):** $\begin{bmatrix} 70 & 60 \\ 70 & 60 \end{bmatrix}$
   - **Compute MSE for Each Block:**

     The MSE is calculated using:

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\mathrm{CurrentFramePixel}_i - \mathrm{ReferenceFramePixel}_i)^2$$

     where $N = 4$ (total pixels in the block).

- **Calculations:**

Block	Differences (Current - Reference)	Squared Differences	Sum	MSE
1	$[-10, 15, 0, 20]$	$[100, 225, 0, 400]$	725	181.25
2	$[0, 35, 5, 5]$	$[0, 1225, 25, 25]$	1275	318.75
3	$[20, 25, -10, -5]$	$[400, 625, 100, 25]$	1150	287.5
4	$[10, 30, 0, 15]$	$[100, 900, 0, 225]$	1225	306.25
5	$[15, 15, 0, 5]$	$[225, 225, 0, 25]$	475	118.75
6	$[0, 5, -10, 15]$	$[0, 25, 100, 225]$	350	**87.5**
7	$[10, 25, -10, -10]$	$[100, 625, 100, 100]$	925	231.25
8	$[10, 15, -25, 5]$	$[100, 225, 625, 25]$	975	243.75
9	$[0, 25, -10, 15]$	$[0, 625, 100, 225]$	950	237.5

4. **Determine the Best Match:**
   - **Block 6** has the lowest MSE of **87.5**.
   - **Motion Vector:** The displacement from the co-located block (2,2) to Block 6 (2,3) is $(\Delta x, \Delta y) = (1, 0)$.

(2,1)

MV=(1,0)

x

80	70	50	60
60	55	70	80
60	60	70	60
70	85	70	60

y

Figure 5

Answer:

- **Best-Matched Motion Vector:** $(1, 0)$ pixels (right by 1 pixel, no vertica
- **Corresponding Block in Figure 5:**

$$\begin{bmatrix} 70 & 80 \\ 70 & 60 \end{bmatrix}$$

(Starting at row 2, column 3 in Figure 5.)

x

	0	1	2	3
0	80	70	50	60
1	60	55	70	80
2	60	60	70	60
3	70	85	70	60

Figure 5

MV=(2,1)

MV=(-1,0)

x

80	70	50	60
60	55	70	80
60	60	70	60
70	85	70	60

Figure 5

y

Answer to Problem 3(b):问题 3(b) 的答案:

Explanation:解释:

1. **Why Half-Pel Accurate Motion Estimation is Better:为什么半像素精确运动估计更好:**
   - **Sub-Pixel Motion Capture:** Objects in video frames often move by less than one pixel between frames due to continuous motion and the discrete nature of frame sampling. Integer-pixel accuracy may miss these subtle movements.

     子像素运动捕捉： 由于连续运动和帧采样的离散性质，视频帧中的对象在帧之间移动的距离通常小于 1 个像素。整数像素精度可能会错过这些微妙的运动。

   - **Reduced Prediction Error:** Half-pel accuracy allows for finer motion vector adjustments, aligning the predicted block more closely with the actual movement, thereby reducing residual errors.

     减少预测误差： 半像素精度允许更精细的运动矢量调整，使预测块与实际运动更紧密地对齐，从而减少残余误差。

   - **Improved Compression Efficiency:** A better motion estimation leads to smaller differences between the predicted and actual blocks, resulting in more efficient encoding and better compression ratios.

     提高压缩效率： 更好的运动估计可以使预测块和实际块之间的差异更小，从而实现更高效的编码和更好的压缩比。

2. **Bilinear Interpolation Method in a 2×2 Image Block:2×2 图像块中的双线性插值方法:**

- **Concept:** Bilinear interpolation estimates pixel values at non-integer (half-pixel) positions by performing linear interpolation first in one direction (e.g., horizontal) and then in the other (vertical).

  概念：双线性插值通过首先在一个方向（例如水平）然后在另一个方向（垂直）执行线性插值来估计非整数（半像素）位置的像素值。

- **Process:过程:**
  - **Given Pixel Values:给定像素值:**

    $$\begin{bmatrix} P_{00} & P_{10} \\ P_{01} & P_{11} \end{bmatrix}$$

    at integer positions $(0,0),(1,0),(0,1),(1,1)$.

    在整数位置 $(0,0),(1,0),(0,1),(1,1)$ 。

  - **Interpolate Horizontal Half-Pixels:插值水平半像素:**
    - At $(0.5,0)$: $P_{0.5,0}=\frac{P_{00}+P_{10}}{2}$在 $(0.5,0)$: $P_{0.5,0}=\frac{P_{00}+P_{10}}{2}$
    - At $(0.5,1)$: $P_{0.5,1}=\frac{P_{01}+P_{11}}{2}$在 $(0.5,1)$: $P_{0.5,1}=\frac{P_{01}+P_{11}}{2}$
  - **Interpolate Vertical Half-Pixels:插值垂直半像素:**
    - At $(0,0.5)$: $P_{0,0.5}=\frac{P_{00}+P_{01}}{2}$在 $(0,0.5)$: $P_{0,0.5}=\frac{P_{00}+P_{01}}{2}$
    - At $(1,0.5)$: $P_{1,0.5}=\frac{P_{10}+P_{11}}{2}$在 $(1,0.5)$: $P_{1,0.5}=\frac{P_{10}+P_{11}}{2}$
  - **Interpolate Center (Half-Pel in Both Directions):插值中心（双向半像素）：**
    - At $(0.5,0.5)$:在 $(0.5,0.5)$:

      $$P_{0.5,0.5}=\frac{P_{00}+P_{10}+P_{01}+P_{11}}{4}$$

  - **Diagram Explanation:图表说明:**

    Imagine a 2×2 grid of pixels:想象一个 2×2 的像素网格:

    scss 复制代码

```scss
(0,1)     (1,1)
  P01-------P11
   |         |
   |         |
(0,0)     (1,0)
  P00-------P10
```

    - The values at the half-pixel positions (dotted lines) are calculated using the average of neighboring pixels.

      半像素位置（虚线）的值是使用相邻像素的平均值计算的。

Answer:回答:

- **Main Reason for Improved Prediction:预测改进的主要原因:**
  - **Half-pel accurate motion estimation provides a finer resolution in capturing motion, allowing for more precise alignment between the current and reference frames.** This reduces the residual errors and improves prediction performance compared to integer-pel motion estimation.

    半像素精确运动估计在捕捉运动时提供了更精细的分辨率，从而允许当前帧和参考帧之间更精确的对齐。与整像素运动估计相比，这减少了残余误差并提高了预测性能。

- **Bilinear Interpolation in a 2×2 Block:2×2 块中的双线性插值:**
  - **Method:方法:**
    - **Horizontal Interpolation:** Compute averages between horizontally adjacent pixels to find intermediate values.

      水平插值: 计算水平相邻像素之间的平均值以找到中间值。
    - **Vertical Interpolation:** Compute averages between vertically adjacent pixels.

      垂直插值: 计算垂直相邻像素之间的平均值。
    - **Center Interpolation:** Average all four surrounding pixels to estimate the value at the center (both half-pel horizontally and vertically).

      中心插值: 平均所有四个周围像素以估计中心值（水平和垂直半像素）。
  - **Illustration:插图:**
    - For pixel positions (x, y) with integer pixels at $(0,0),(1,0),(0,1),(1,1)$, the half-pel values are:

      对于像素位置 (x, y) 整数像素位于 $(0,0),(1,0),(0,1),(1,1)$ ，半像素值为:
      - $P_{0.5,0} = \frac{P_{00}+P_{10}}{2}$
      - $P_{0,0.5} = \frac{P_{00}+P_{01}}{2}$
      - $P_{0.5,0.5} = \frac{P_{00}+P_{10}+P_{01}+P_{11}}{4}$

By using bilinear interpolation, motion estimation algorithms can access and utilize pixel values at half-pixel positions, enabling more accurate motion vectors and better prediction performance.

通过使用双线性插值，运动估计算法可以访问和利用半像素位置的像素值，从而实现更准确的运动矢量和更好的预测性能。