CS 763: Assignment 2

Ayush Baid (12D100002)
Niranjan Thakurdesai (12D100007)
Jainesh Doshi (12D070014)

February 17, 2016

Problem 1

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i = \sum_{i=1}^{N} p_i \log \frac{1}{p_i}$$

Since $0 \leq p_i \leq 1$, $1 \leq \frac{1}{p_i} < \infty$ (assuming $\log 0 = 0$). Hence, each term in the above sum is non-negative. Thus, $H(X) \geq 0$.

$$J(X) = -\sum_{i=1}^{N} p_i \log p_i - \lambda \left(\sum_{i=1}^{N} p_i - 1 \right)$$

To find a stationary point of $J(X)$, $\frac{\partial J}{\partial p_i} = 0 \ \forall i$. Hence,

$$\begin{aligned} -1 - \log p_i - \lambda &= 0 \\ \implies \lambda &= -(1 + \log p_i) \end{aligned}$$

As λ is constant $\forall i$, $p_1 = p_2 = \cdots = p_N$. As $\sum_{i=1}^{N} p_i = 1$, $p_i = \frac{1}{N} \ \forall i$. Hence, a uniform distribution maximises the Shannon entropy.

Problem 2

Q-2.

Gurukul Page No.:
Date: / /

Note: dropping subscripts (k,l) from I_x, I_y & I_t for readability

$$J = \sum_{i=1}^{N}\sum_{j=1}^{N} (I_x u_{i,j} + I_y v_{i,j} + I_t)^2 + \lambda\Big((u_{i,j+1} - u_{i,j})^2 + (u_{i+1,j} - u_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2 + (v_{i+1,j} - v_{i,j})^2 \Big)$$

Considering cost for pixel (k,l)

$$J_{k,l} = \overbrace{(I_x u_{k,l} + I_y v_{k,l} + I_t)^2}^{T1} + \lambda \overbrace{(u_{k,l+1} - u_{k,l})^2}^{T2} + \lambda\underbrace{(u_{k+1,l} - u_{k,l})^2}_{T3} + \lambda\underbrace{(v_{k,l+1} - v_{k,l})^2}_{T4} + \lambda\underbrace{(v_{k+1,l} - v_{k,l})^2}_{T5}$$

Note that $u_{k,l}$ occurs in T1 for only $J_{k,l}$
in T2 for $J_{k,l-1}$ and $J_{k,l}$
in T3 for $J_{k-1,l}$ and $J_{k,l}$.

Using this fact,

$$\frac{\partial J}{\partial u_{k,l}} = 2(I_x u_{k,l} + I_y v_{k,l} + I_t)I_x + \lambda\Big[2(u_{k,l+1} - u_{k,l})(-1) + 2(u_{k,l} - u_{k,l-1})(1) + 2(u_{k+1,l} - u_{k,l})(-1) + 2(u_{k,l} - u_{k-1,l})(1)\Big]$$

For minimization, $\frac{\partial J}{\partial u_{k,l}} = 0 \quad \forall k,l$

Gurukul Page No.: Date: / /

$$\Rightarrow (I_x^2+4\lambda)\, u_{k,l} + I_x I_y v_{k,l} = 4\lambda \bar{u}_{k,l} - I_x I_t \quad \ldots (a)$$

where $\bar{u}_{k,l} = \dfrac{u_{k+1,l} + u_{k-1,l} + u_{k,l+1} + u_{k,l-1}}{4}$

Following the same steps for $v_{k,l}$, we get

$$(I_y^2+4\lambda)\, v_{kl} + I_x I_y u_{k,l} = 4\lambda \bar{v}_{k,l} - I_y I_t \quad \ldots (b)$$

Solving (a) & (b) simultaneously,

$$u_{kl} = \bar{u}_{k,l} - \frac{I_x (I_x \bar{u}_{kl} + I_y \bar{v}_{k,l} + I_t)}{I_x^2 + I_y^2 + 4\lambda} \quad \ldots (1)$$

and

$$v_{kl} = \bar{v}_{k,l} - \frac{I_y (I_x \bar{u}_{k,l} + I_y \bar{v}_{k,l} + I_t)}{I_x^2 + I_y^2 + 4\lambda} \quad \ldots (2)$$

we can write the above system of equations as

$Ax = b$ where $x = \begin{bmatrix} u_{11} \\ v_{11} \\ u_{12} \\ v_{12} \\ \vdots \\ u_{21} \\ v_{21} \\ u_{22} \\ v_{22} \\ \vdots \end{bmatrix}$

Date: / /

Consider arrangement of equation such that the row for which $b_i = u_{k,l}$, we have eq^n (1) at k,l and the row for which $b_i = v_{k,l}$, we have eq^n (2) at k,l

Also, for Jacobi's method, we need $A = D + R$, where D = diagonal matrix

$$Ax = b$$
$$\Rightarrow Dx + Rx = b$$

Consider the eq^n (1) for k,l,

~~we have $d_a u_{kl}$ + $R x$~~

we have contribution for $u_{k,l}$ coming from D and $\bar{u}_{kl}$ & $\bar{v}_{k,l}$ coming from $R_{i,j}$ $i \neq j$.

$\Rightarrow$ coefficients of D are 1 on diagonal and R has zero on diagonal

using $x_i^{(t+1)} = \dfrac{b_i - \sum_{j \neq i} A_{ij} x_j^{(t)}}{A_{ii}}$ where $A_{ii} = D_{ii} = 1$.

$$\Rightarrow u_{k,l}^{(t+1)} = \frac{-I_x I_t}{I_x^2 + I_y^2 + 4\lambda} - \left\{ -\bar{u}_{kl}^{(t)} \left(1 - \frac{I_x^2}{I_x^2 + I_y^2 + 4\lambda} \right) + \bar{v}_{k,l}^{(t)} \left(\frac{I_x I_y}{I_x + I_y^2 + 4\lambda} \right) \right\}$$

rearranging, we get (3)

and repeating for $v_{k,l}$, we get (4).

Problem 3

1. **Horn-Schunck**: Because, Horn-Schunck is a global method, if there's an error in the optical flow computation for any pixel, it will also adversely affect the computation at other pixels. The Horn-Schunck equations at any pixel are as follows:

$$u_{kl} - \bar{u}_{kl}\left(1 - \frac{I_x^2}{I_x^2 + I_y^2 + 4\lambda}\right) + \bar{v}_{kl}\left(1 - \frac{I_x I_y}{I_x^2 + I_y^2 + 4\lambda}\right) = -\frac{I_x I_t}{I_x^2 + I_y^2 + 4\lambda}$$
$$v_{kl} - \bar{v}_{kl}\left(1 - \frac{I_y^2}{I_x^2 + I_y^2 + 4\lambda}\right) + \bar{u}_{kl}\left(1 - \frac{I_x I_y}{I_x^2 + I_y^2 + 4\lambda}\right) = -\frac{I_y I_t}{I_x^2 + I_y^2 + 4\lambda}$$

   We can collect these equations at every pixel and rewrite them in the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a $2MN \times 2MN$ matrix, vector $\mathbf{x}$ contains all the unknowns, i.e. u and v values at all pixels and vector $\mathbf{b}$ contains the terms on the RHS of the above equations.

   Now we use the LMedS algorithm which is robust to outliers on this system of equations. As we know that the number of outliers (in this case, pixels where the brightness constancy assumption is violated) is $< 50\%$, this method is guaranteed to work.

2. **Lucas-Kanade**: Lucas-Kanade is a local method. Therefore, errors at some pixels do not affect other pixels. We'll use this to our advantage while dealing with outliers.

   First compute the Lucas-Kanade optical flow at all pixels using an appropriate window size. After this, calculate the mean square error (MSE) between physically corresponding pixels for the entire image. For the patches for which the MSE of the patch is greater than the MSE of the entire image, apply LMedS and recompute optical flow of such patches.

Problem 4

$\hat{R}$ is the estimated rotation matrix by the first camera calibration method. However, we do not have the guaranty of $\hat{R}$ being a Rotation matrix as we have not yet imposed the condition for the Rotation matrix that is as follows:
For any Rotation matrix $\mathbf{R}$; $\mathbf{R^T R = I}$
SVD allows us to find the closest matrix to the required matrix $\hat{\mathbf{R}}$ in sense of the Frobenius norm and that satisfies the constraints exactly, cause if not further computations in the algorithm would be severely affected if this result were to not hold true.
The Frobenius norm of a matrix A is given by

$$\|A\|_F = \sum_i \sigma_i^2$$

where σ_i's are the singular values of the SVD of A, thus this gives us a distance measure between matrices. Thus we try to minimize the difference between the corresponding singular values of the matrix $\hat{\mathbf{R}}$ and another that obeys the orthonormality constraint for the rotation matrix as in $\mathbf{R^T R} = \mathbf{I}$ to find the optimum solution for $\mathbf{R}$. Thus We have

$$\tilde{\mathbf{R}} = \operatorname{argmin}_{\mathbf{R}} \left\| \mathbf{R} - \hat{\mathbf{R}} \right\|^2$$

Where $\mathbf{R}$ satisfies the condition $\mathbf{R^T R} = \mathbf{I}$.
However while applying this result we did not impose the condition for $\det(\mathbf{R}) = 1$ explicitly. This is needed so that the estimated rotation matrix doesn't contain a reflection within itself.

Problem 5

If the object motion were 3D Affine instead of 3D rigid, the point coordinates and the object motion can be determined uniquely only upto a 3×3 invertible matrix $\mathbf{Q}$. The necessary equations are as follows:

$$\begin{aligned} x_{ij} &= \mathbf{A}_{i1}^T \mathbf{P}_j + t_{xi} \\ y_{ij} &= \mathbf{A}_{i2}^T \mathbf{P}_j + t_{yi} \\ \tilde{x}_{ij} &= x_{ij} - \bar{x}_i, \bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij} \\ \tilde{y}_{ij} &= y_{ij} - \bar{y}_i, \bar{y}_i = \frac{1}{n} \sum_{j=1}^{n} y_{ij} \end{aligned}$$

We assume that the centroid of the 3D object is the origin of the world coordinate system. Therefore,

$$\frac{1}{n} \sum_{j=1}^{n} \mathbf{P}_j = 0$$

Combining these equations gives

$$\begin{aligned} \tilde{x}_{ij} &= \mathbf{A}_{i1}^T \mathbf{P}_j + t_{xi} - \frac{1}{n} \sum_{m=1}^{n} (\mathbf{A}_{i1}^T \mathbf{P}_j + t_{xi}) = \mathbf{A}_{i1}^T \mathbf{P}_j \\ \tilde{y}_{ij} &= \mathbf{A}_{i2}^T \mathbf{P}_j + t_{yi} - \frac{1}{n} \sum_{m=1}^{n} (\mathbf{A}_{i2}^T \mathbf{P}_j + t_{yi}) = \mathbf{A}_{i2}^T \mathbf{P}_j \end{aligned}$$

The matrix $\tilde{\mathbf{W}}$ can be written as

$$\tilde{\mathbf{W}} = \mathbf{RS}$$
$$\mathbf{R} = \begin{pmatrix} \mathbf{A}_{11}^T \\ \mathbf{A}_{21}^T \\ \vdots \\ \mathbf{A}_{N1}^T \\ \mathbf{A}_{12}^T \\ \mathbf{A}_{22}^T \\ \vdots \\ \mathbf{A}_{N2}^T \end{pmatrix}, \mathbf{S} = \begin{pmatrix} \mathbf{P}_1 & \mathbf{P}_2 \ldots \mathbf{P}_n \end{pmatrix}$$

Now, we compute the SVD of $\tilde{\mathbf{W}}$:

$$\tilde{\mathbf{W}} = \mathbf{UDV}^T = \mathbf{UD}^{1/2}\mathbf{D}^{1/2}\mathbf{V}^T$$
$$\mathbf{R} = \mathbf{UD}^{1/2}, \mathbf{S} = \mathbf{D}^{1/2}\mathbf{V}^T$$

The rows of $\mathbf{R}$ give the 3D Affine motion and the columns of $\mathbf{S}$ give the 3D point coordinates. However, $\mathbf{R}$ and $\mathbf{S}$ are not unique because for any invertible 3×3 matrix $\mathbf{Q}$, we have

$$\tilde{\mathbf{W}} = \mathbf{RS} = \mathbf{RQQ}^{-1}\mathbf{S}$$

Thus, replacing $\mathbf{R}$ by $\mathbf{RQ}$ and $\mathbf{S}$ by $\mathbf{Q}^{-1}\mathbf{S}$ won't change $\tilde{\mathbf{W}}$ In this case, we can't apply the metric properties as the motion is Affine.