Optimization of the Deployment of Proxies in WebFVV

Haipeng Zhang

November 10, 2022

1 Optimal Number of Proxies

In WebFVV, We denote the number of the cameras in the acquisition protocol as V, which means there are at most V original video streams generated by cameras. For a specific virtual viewpoint, the proxy needs to receive its left and right neighboring reference viewpoints to perform the synthesis. When a proxy is synthesizing virtual views, according to users' requests, only part of those V original video streams are needed as reference videos. The needed videos will be transmitted from the server to the proxy and cached. We denote the set of original videos cached at a specific proxy server as S, and the expectation of size of S is M. To further calculate the processing time, we denote the number of users served by a proxy as U. Suppose for an original video stream, the proxy needs to download a unit size of the file to start the synthesis task, we denote the proxy's bandwidth as B. In the synthesis process, suppose a proxy can concurrently perform G virtual view synthesis tasks, and the time the proxy needs to finish one synthesis task is T_v.

The expectation of processing time T for a user's request can be divided into two parts: the download time and the synthesis time, which can be denoted as $T = t_d + t_s,$, where t_d represents the download time and t_s represents the synthesis time.

Suppose there are U_{all} users in total. Given the number of proxies N_p, averagely, the number of users that each proxy may serve is

$$U = \frac{U_{all}}{N_P} \tag{1}$$

Lemma 1. *The expectation of the number of original videos cached at a proxy is* $V(1-(1-\frac{2}{V})^U)$

Proof. For a single original video stream A_i, it is not used by a specific user only if the user is not in a virtual viewpoint on its left or right. Therefore the probability that it is not used by a specific user is

$$P_{single} = 1 - \frac{2}{V} \tag{2}$$

For all the U users the proxy server is serving, the viewpoint they watch is independent of each other. Therefore, the probability that A_i is not used by all the users is

$$P_u = (P_{single})^U = (1 - \frac{2}{V})^U \tag{3}$$

Denote the expectation of the size of set S is M, we can have

$$\begin{aligned} M &= E(\sum_{i=0}^{V-1} A_i) = \sum_{i=0}^{V-1}(E(A_i)) \\ &= V(1-P_u) \\ &= V(1-(1-\frac{2}{V})^U) \end{aligned} \tag{4}$$

□

Lemma 2. *The expectation of the download time* t_d *can be represented as* $\frac{(V-M)(M^2+2V+VM)}{BV^2}$

Proof. For a proxy server that already has M video streams in transmitting, and wants to transmit N new video streams. The new streams have to share the bandwidth equally with M video streams, we can have the download time

$$t_d = \frac{M+N}{B} \tag{5}$$

When a new specific video stream v is needed by the proxy server, if v is in S, then the proxy server can use the cache and directly start the synthesis task. Otherwise, the proxy server needs to apply v from the main server and increase the processing time. We denote the probability of only one original video stream needing to be downloaded as P_1, and the probability of both the two video streams needing to be downloaded as P_2. After the needed video frames are ready, the proxy server will synthesize the virtual view taking t_s time. We can have the expectation of the processing time

$$T = P_1 * \frac{M+1}{B} + P_2 * \frac{M+2}{B} + t_s \tag{6}$$

For a new synthesis task, the probability that only one new stream is needed to be downloaded is

$$P_1 = 2(1-P_u)P_u \tag{7}$$

The probability that two new streams are needed to be downloaded is

$$P_2 = (P_u)^2 \tag{8}$$

The expectation of the time need to download new original videos is

$$\begin{aligned} t_d &= P_0 * 0 + P_1 * \frac{M+1}{B} + P_2 * \frac{M+2}{B} \\ &= \frac{P_u(2(M+1) - MP_u)}{B} \\ &= \frac{(V-M)(M^2+2V+VM)}{BV^2} \end{aligned} \tag{9}$$

□

Proposition 1. *The expectation of the processing time can be represented as* $\frac{(V-M)(M^2+2V+VM)}{BV^2} + (\frac{U}{G}+1) * T_v$.

When a new synthesis task comes, we can estimate the time this task needs to wait is

$$t_{wait} = \frac{U}{G} * T_v \tag{10}$$

We can estimate the total synthesis time t_s as

$$t_s = (\frac{U}{G}+1) * T_v \tag{11}$$

Therefore the expectation of the processing time is

$$\begin{aligned} T &= t_d + t_s \\ &= \frac{P_u(2(M+1) - MP_u)}{B} + (\frac{U}{G}+1) * T_v \\ &= \frac{(V-M)(M^2+2V+VM)}{BV^2} + (\frac{U}{G}+1) * T_v \end{aligned} \tag{12}$$

where $M = V(1-(1-\frac{2}{V})^U)$, $U = \frac{U_{all}}{N_P}$, and B, G, V, U_{all}, T_v depends on the scene.

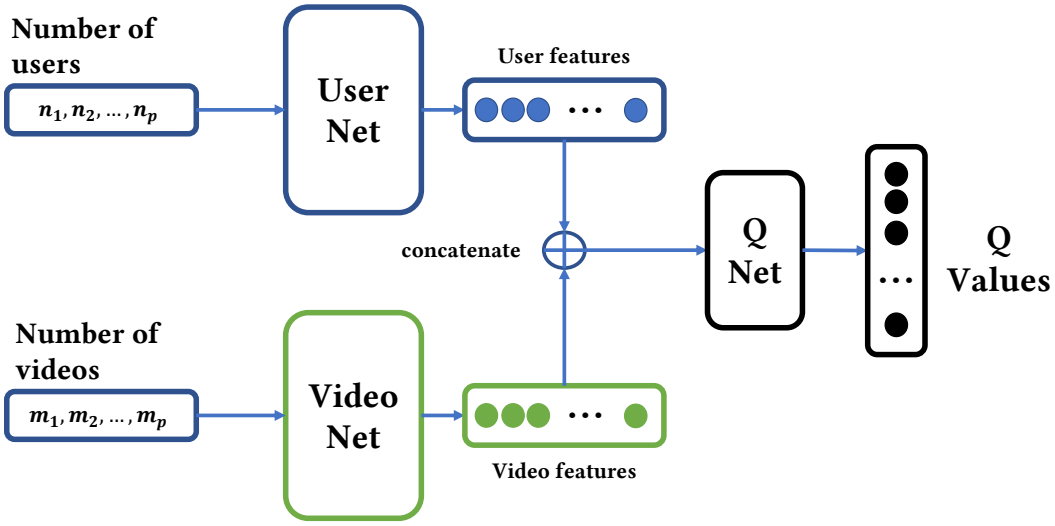

Figure 1: Architecture of DQN Network

2 The Architecture of DQN in WebFVV

The architecture of our neural network is illustrated in fig 1. To consider the relationship between choosing different proxy servers, our network first employs two distinct networks: the UserNet and the VideoNet, to separately calculate the feature of the number of users and the number of required videos. These two networks are the same in architecture but have different parameters. Each network consists of fully-connected layers to integrate the influence of each proxy server. The outputs of UserNet and VideoNet have the same number of channels as the input, and each channel represents the feature of one proxy server. Then the outputs of the UserNet and the VideoNet will be concatenated as the proxy features. These features will be input into a Q network to estimate the Q value of each proxy server. During the training process, we employed Double DQN with experience replay to better train our model.