计算机科学
COMPUTER SCIENCE

基于LpTransformer网络的手语动画拼接模型

黄涵强, 邢云冰, 沈建飞, 范非易

引用本文

黄涵强, 邢云冰, 沈建飞, 范非易. 基于LpTransformer网络的手语动画拼接模型[J]. 计算机科学, 2023, 50(9): 184-191.

HUANG Hanqiang, XING Yunbing, SHEN Jianfei, FAN Feiyi. Sign Language Animation Splicing Model Based on LpTransformer Network [J]. Computer Science, 2023, 50(9): 184-191.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221100043

基于 LpTransformer 网络的手语动画拼接模型

黄涵强[1,2]　邢云冰[2,3]　沈建飞[2,3]　范非易[2]
1 郑州大学河南先进技术研究院　郑州 450000
2 中国科学院计算技术研究所　北京 100000
3 山东产业技术研究院智能计算研究院　济南 250000
(893586949@qq.com)

摘　要　手语动画拼接是一个热门话题。随着机器学习技术的不断发展,尤其是深度学习相关技术的逐渐成熟,手语动画拼接的速度和质量不断提高。将手语单词拼接成句子时,相应的动画也需要拼接。传统的算法在拼接动画时采取距离损失的方式寻找最佳拼接点,使用线性或球面插值的方式生成过渡帧,这种拼接算法不仅在效率和灵活性方面存在明显缺陷,而且生成的过渡帧也不自然。为解决上述问题,提出了 LpTransformer 模型来预测拼接位置和生成过渡帧。实验表明,LpTransformer 的过渡帧预测精度达到 99%,优于 ConvS2S,LSTM 和 Transformer 模型,且其拼接速度较 Transformer 快 5 倍。因此,所提模型能够实现实时性拼接。

关键词:手语动画拼接;深度学习;LpTransformer;拼接位置;过渡帧
中图法分类号　TP183

Sign Language Animation Splicing Model Based on LpTransformer Network

HUANG Hanqiang[1,2],XING Yunbing[2,3],SHEN Jianfei[2,3] and FAN Feiyi[2]
1 Henan Institute of Advanced Technology,Zhengzhou University,Zhengzhou 450000,China
2 Institute of Computing Technology,Chinese Academy of Sciences,Beijing 100000,China
3 Shandong Industrial Technology Research Institute Intelligent Computing Research Institute,Jinan 250000,China

Abstract　Sign language animation splicing is a hot topic. With the continuous development of machine learning technology,especially the gradual maturity of deep learning related technologies,the speed and quality of sign language animation splicing are constantly improving. When splicing sign language words into sentences,the corresponding animation also needs to be spliced. Traditional algorithms use distance loss to find the best splicing position when splicing animation,and use linear or spherical interpolation to generate transition frames. This splicing algorithm not only has obvious defects in efficiency and flexibility,but also generates unnatural sign language animation. In order to solve the above problems,LpTransformer model is proposed to predict the splicing position and generate transition frames. Experiment results show that the prediction accuracy of LpTransformer's transition frames reaches 99%,which is superior to ConvS2S,LSTM and Transformer,and its splicing speed is five times faster than Transformer,so it can achieve real-time splicing.

Keywords　Sign language animation splicing,Deep learning,LpTransformer,Splicing position,Transition frames

1　引言

手语(Sign Language)是一种视觉语言,其以肢体动作、手指手势、面部表情和口型唇动表意和交流,使用者主要是听觉或言语功能障碍者,即聋哑人。类似于汉语等听觉语言,手语句也是由各个手语词按序连接而成,最终的呈现形式是视频或 3D 动画。在按序连接手语词时,代表着对应手语词的视频片段或 3D 动画片段也需要进行拼接,这就涉及到拼接位置的搜索和过渡帧的生成。传统方法使用欧氏距离寻找拼接点,耗时较多;同时,传统方法使用线性或球面插值的方式生成过渡帧,导致过渡不平滑。现如今,手语动画生成广泛使用深度学习模型。将深度学习模型用于手语动画拼接面临许多挑战,传统的 LSTM 模型预测的过渡帧误差大,导致手势产生变形;并且随着编码层数量的增加,其遗忘程度也会增加,导致模型无法充分利用输入数据进行预测。由于手语动作的整体关联性较强,各个时间点的手势位置对拼接点的预测存在影响,因此会造成拼接点预测错误,进而导致手势过渡不自然、不连贯。ConS2S 模型虽然能提高预测速度,但存在

到稿日期:2022-11-07　返修日期:2023-02-28
基金项目:国家重点研发计划(2018YFC2002603)
This work was supported by the National Key Research and Development Program of China(2018YFC2002603).
通信作者:邢云冰(xingyunbing@ict.ac.cn)

遗漏过渡帧的严重问题。Transformer 模型在过渡帧的对齐方面效果不佳，预测出的过渡帧位置与真实的情况相差较远，使得绝对误差比 LSTM 模型更大，且预测时间随输入序列的增加而增加，无法实现实时预测。

LpTransformer 模型解决了上述问题，注意力机制的引入使模型可根据各个时间点的手势位置预测拼接位置。与此同时，将拼接位置输入 Transformer 网络可减少其输入序列的数量，进而大幅缩短模型的预测时间，实现实时预测。除此之外，模型生成的过渡帧误差小，对齐效果好，更接近自然手势动作。如图 1 所示，手语动画拼接过程包括拼接位置搜索和过渡帧生成两部分，将前后两段手语动画首尾相接作为模型的原始输入。拼接位置搜索网络（Splicing Position Search Network，SPSN）用于搜索相邻的拼接位置，过渡帧生成网络（Transition Frames GenerationNetwork，TFGN）用于生成两段手语动画间的过渡帧动画。最后，过滤层（Filter Layer）分割原始输入并插入生成的过渡帧作为模型的输出。

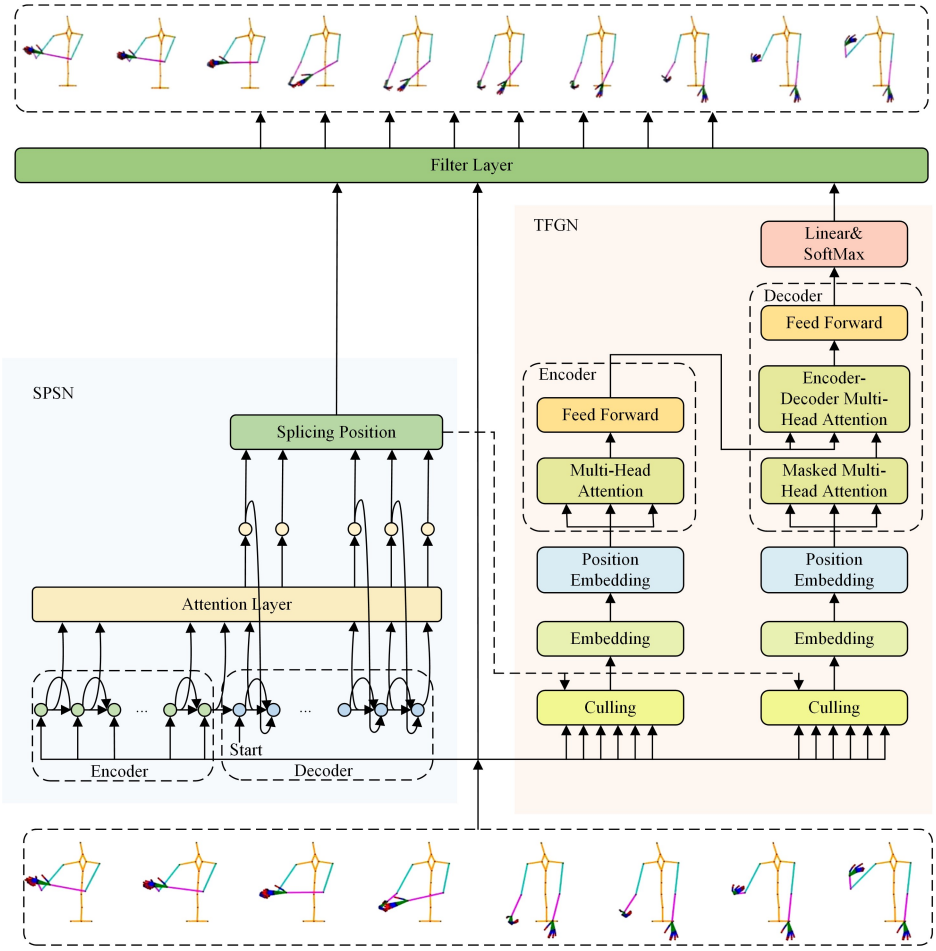


图 1 LpTransformer 总体架构图

Fig. 1 Overall architecture of LpTransformer

本文第 2 章和第 3 章介绍了手语动画拼接的相关工作，并详细介绍了 LpTransformer 的工作原理；第 4 章将 LpTransformer 与现有的模型进行比较实验并分析实验结果，最终提供了模型预测的可视化结果；最后得出结论并简要讨论了未来的潜在研究方向。

2 相关工作

相关工作分为骨骼动画拼接和视频拼接两部分。

骨骼手语动画的拼接一般使用传统方法搜索拼接点。Zhu 等[1]利用手部空间的欧氏距离和手部的角度转移量构造拼接成本函数，通过最小化拼接成本函数的方式确定拼接点，并利用基于纹理模板的手纹理合成方法优化过渡帧的纹理细节，进而生成过渡帧。Chen 等[2]也利用相同的方式确定拼接点，为了实现更自然平滑的过渡帧生成，他们用图像技术对形变操作得到的中间结果进行融合，并利用块匹配技术整合对应形变的图像，最后得到过渡帧。Zhao 等[3]通过最小化两帧之间位置和速度的变化幅度来确认拼接点，并采用改进的 Hermite 插值算法和 VRML 生成过渡帧，最后完成手语拼接。以上拼接手语动画的方式都是采取传统的方法搜索拼接点，这一过程费时费力，且采用传统的图像融合方式生成过渡帧，容易产生过渡帧的重影。骨骼手语动画拼接过程中拼接点的搜索均采取最小化距离损失的方式确定。近些年的研究成果也都忽略了拼接点的搜索，而集中在提升过渡帧的质量方面。Duarte 等[4]提出了一种具体的神经网络模型来实现语音到手语动画的转换，其中就涉及过渡帧的生成。Kapoor 等[5]提出了一种 multi-ask Transformer 网络，它将语音转换为文本作为辅助任务，并添加跨模式鉴别器，以生成序列连续的手语过渡帧。Xiao 等[6]提出了一种基于 RNN 网络的汉语手语识别与生成框架来解决手语表演者和手语听众之间的双向通信问题。该模型不仅可以完成手语翻译，并且能根据

文字生成专业的手语过渡帧手势。与此类似的工作还有文献[7-8]。基于上述网络结构，为了解决手语过渡帧生成中的延迟问题，Saunders 等[9]采用一种对立的训练机制和混合密度网络(Mixed Density Network，MDN)来生成真实手语过渡帧序列。与上述模型类似，Huang 等[10]提出了一种新型的非自回归(Non-autoregressive，NAT)模型用于并行生成手语过渡帧序列。Zhou 等[11]提出了一种卷积神经网络的深部运动插值模型用于生成手语过渡帧动画。其他方法，如骨架图自注意力机制[12](Skeleton Graph Self-attention，SGSA)和生成对抗网络等，也被广泛使用在手语过渡帧动画生成领域。以上相关研究生成的动画过渡不自然，耗时长，且存在手势变形现象。

传统的视频拼接采用摄像机直接记录过渡帧，并在断点处插入完成拼接[13-14]。尺度不变特征变换(Scale Invariant Feature Transform，SIFT)算法被广泛用于搜索视频拼接中的拼接位置。Vasuhi 等[15]提取所有摄像机的 SIFT 关键点，并使用随机样本一致性(Random Sample Consistency，RANSAC)来识别所有匹配点之间的对应关系，进而确定拼接位置，然后基于前景提取技术和人体质心点绘图技术生成过渡帧。Cao 等[16]改进了 SIFT 技术，将其与矢量相关系数相结合并应用于无人机视频图像拼接位置的搜索，然后采用单应矩阵估计模型和视频传感器完成过渡帧的生成和拼接。同样，深度学习技术也被用于视频拼接领域。Das 等[17]提出了 CAM-STICH 算法，其在已知拼接位置的情况下使用 LSTM 网络生成过渡帧并完成拼接。这些工作独立搜索拼接位置和生成过渡帧，无法同时解决二者的联动问题，且拼接的视频多为物体或者行人，生成的图像也存在扭曲不规整的现象。

3 LpTransformer

图 1 给出了 LpTransformer 的总体结构。原始输入可通过以下方式定义：

$$\boldsymbol{P}=[\boldsymbol{P}_1,\boldsymbol{P}_2,\boldsymbol{P}_3,\cdots,\boldsymbol{P}_N] \tag{1}$$

$$\boldsymbol{Q}=[\boldsymbol{Q}_1,\boldsymbol{Q}_2,\boldsymbol{Q}_3,\cdots,\boldsymbol{Q}_M] \tag{2}$$

其中，$\boldsymbol{P}_i$ 和 $\boldsymbol{Q}_i$ 是骨骼坐标向量，$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 是前段手语序列向量和后段手语序列向量，N 和 M 是序列的总帧数。通过串联 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 来构建输入序列 $\boldsymbol{X}$。

$$\boldsymbol{X}=[\boldsymbol{P}_1,\boldsymbol{P}_2,\boldsymbol{P}_3,\cdots,\boldsymbol{P}_N,\boldsymbol{Q}_1,\boldsymbol{Q}_2,\boldsymbol{Q}_3,\cdots,\boldsymbol{Q}_M] \tag{3}$$

LpTransformer 的架构由 3 部分组成：基于 LSTM 的拼接位置搜索网络(SPSN)，用于搜索手语序列 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的拼接位置；基于 Transformer 的过渡帧生成网络(TFGN)，用于生成过渡帧；以及用于处理位置信息和过渡帧信息的过滤层(Filter Layer)。

3.1 SPSN

基于 LSTM 的 SPSN 包括编解码层和注意力层。

编解码层：如图 2 所示，LSTM 由遗忘门、输入门和输出门组成。3 个门控单元的输入和输出表达式定义如下：

$$\begin{cases}\boldsymbol{I}_t=\boldsymbol{\sigma}(\boldsymbol{X}_t\boldsymbol{W}_{xi}+\boldsymbol{H}_{t-1}\boldsymbol{W}_{hi}+\boldsymbol{b}_i)\\ \boldsymbol{F}_t=\boldsymbol{\sigma}(\boldsymbol{X}_t\boldsymbol{W}_{xf}+\boldsymbol{H}_{t-1}\boldsymbol{W}_{hf}+\boldsymbol{b}_f)\\ \boldsymbol{O}_t=\boldsymbol{\sigma}(\boldsymbol{X}_t\boldsymbol{W}_{xo}+H_{t-1}\boldsymbol{W}_{ho}+\boldsymbol{b}_o)\end{cases} \tag{4}$$

其中，$\boldsymbol{X}_t$ 是输入序列 X 的第 t 帧向量；$\boldsymbol{H}_{t-1}$ 为隐藏层状态向量；$\boldsymbol{I}_t$，$\boldsymbol{F}_t$ 和 $\boldsymbol{O}_t$ 分别为输入、遗忘和输出门的输出；$\boldsymbol{W}$ 和 $\boldsymbol{b}$ 分别表示权重向量和偏置向量。*LSTM* 的输出如下：

$$\begin{cases}\widetilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{X}_t\boldsymbol{W}_{xc}+\boldsymbol{H}_{t-1}\boldsymbol{W}_{hc}+\boldsymbol{b}_c)\\ \boldsymbol{C}_t=\boldsymbol{F}_t\odot\boldsymbol{W}_{xc}+I_t\odot\widetilde{\boldsymbol{C}}_t\\ \boldsymbol{H}_t=\boldsymbol{O}_t\odot\tanh(\boldsymbol{C}_t)\end{cases} \tag{5}$$

其中，$\widetilde{\boldsymbol{C}}_t$ 是候选神经元的输出。

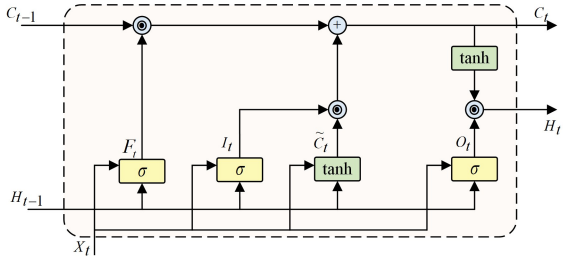


图 2 LSTM 网络结构

Fig. 2 LSTM network structure

注意力层：为解决 LSTM 网络预测长手语序列时的遗忘问题，本文引入了注意力机制。当预测时刻 t 的手语动作时，根据编码层和解码层的隐藏层状态生成注意力分数，然后重构编码信息，最后与 t 时刻的隐藏层状态连接，如图 3 所示。

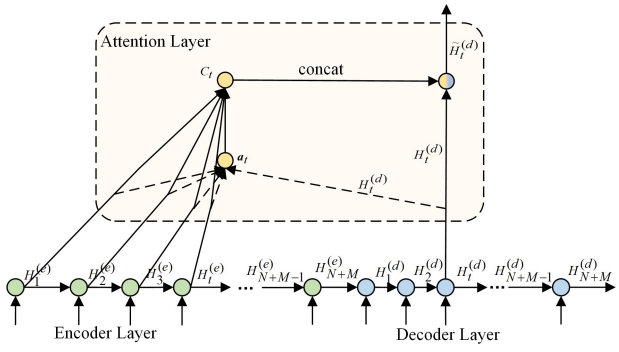


图 3 注意力层结构

Fig. 3 Attention layer structure

首先，利用 score 方法计算 t 时刻的解码结果和任意时刻编码结果的均方误差，然后通过归一化得到单个注意力分数 $a_{t,k}$，最后拼接生成注意力分数 $\boldsymbol{a}_t$。

$$score(\boldsymbol{H}_t^{(d)},\boldsymbol{H}_k^{(e)})=\frac{1}{h}\sum_{i=1}^{h}(H_{t,i}^{(d)}-H_{k,i}^{(e)})^2 \tag{6}$$

where $\boldsymbol{H}_t^{(d)}\in R^{h\times 1}$, $\boldsymbol{H}_k^{(e)}\in R^{h\times 1}$

$$a_{t,k}=1-\frac{\exp(score(\boldsymbol{H}_t^{(d)},\boldsymbol{H}_k^{(e)}))}{\sum_{k=1}^{N+M}\exp(score(\boldsymbol{H}_t^{(d)},\boldsymbol{H}_k^{(e)}))} \tag{7}$$

$$\boldsymbol{a}_t=[\boldsymbol{a}_{t,1},\boldsymbol{a}_{t,2},\cdots,\boldsymbol{a}_{t,N+M}]^{\mathrm{T}} \tag{8}$$

其中，$\boldsymbol{H}_t^{(d)}$ 为 t 时刻的解码结果，$\boldsymbol{H}_k^{(e)}$ 为第 k 层编码层的编码结果，h 表示隐藏层矩阵的维度；$a_{t,k}$ 表示 t 时刻解码结果对第 k 层编码层的注意力分数。最后利用注意力分数重构编码结果得到 $\boldsymbol{C}_t$，并连接 $\boldsymbol{C}_t$ 与 $\boldsymbol{H}_t^{(d)}$，通过引入线性变换矩阵 $\boldsymbol{W}_c$ 和非线性变换函数 tanh 生成拼接的隐藏层结果 $\widetilde{\boldsymbol{H}}_t^{(d)}$。以上描述可被定义为以下公式：

$$\begin{cases}\boldsymbol{H}^{(e)}=[\boldsymbol{H}_1^{(e)},\boldsymbol{H}_2^{(e)},\cdots,\boldsymbol{H}_{N+M}^{(e)}]\\ \boldsymbol{C}_t=\boldsymbol{a}_t\boldsymbol{H}^{(e)}\\ \widetilde{\boldsymbol{H}}_t^{(d)}=\tanh(\boldsymbol{W}_c\cdot[\boldsymbol{C}_t,\boldsymbol{H}_t^{(d)}])\\ \boldsymbol{Y}_t=\boldsymbol{W}_o\widetilde{\boldsymbol{H}}_t^{d}+\boldsymbol{b}_o\end{cases} \tag{9}$$

其中，$\boldsymbol{Y}_t$ 是注意力层的输出向量。分别设置两个标志 ***prePositionToken*** 和 ***postPositionToken*** 表示前段手语动画的拼接点和后段手语动画的拼接点。当 $\boldsymbol{Y}_t$ 与标志相等时，手语拼接点的位置即为 $\boldsymbol{Y}_t$ 的当前帧 t。

$$\begin{cases} prePosition=t_1, & \text{if } \boldsymbol{Y}_{t_1}=\boldsymbol{prePositionToken} \\ postPosition=t_2, & \text{if } \boldsymbol{Y}_{t_2}=\boldsymbol{postPositionToken} \end{cases} \tag{10}$$

3.2 TFGN

如图 1 所示，Transformer 的 TFGN 包括选择(Culling)层、编码(Embedding)层、位置编码(Position Embedding)层、编码器和解码器(Encoder&Decoder)层以及输出层(Linear&Softmax)层。

选择层：由于 Transformer 网络的训练和预测时间随输入序列的增加而增加，为降低输入数据数量，减少训练和预测时间，在 Transformer 网络结构里加入 Culling 层，其接收 SPSN 的输出，截取输入序列中拼接位置前后各 fr 帧作为 Transformer 模型的输入。实验表明，随着 fr 的增加，生成过渡帧所需的时间也相应增加，但生成的过渡帧质量可能不会提高(详见图 4(b))，因此通过实验确定了一个合理的 fr 值，用于确保过渡帧质量的同时缩短模型的预测时间。具体计算公式如下：

$$\boldsymbol{X}_c=[\boldsymbol{P}_{\text{prePosition}-fr+1},\cdots,\boldsymbol{P}_{\text{prePosition}},\boldsymbol{Q}_{\text{postPosition}},\cdots,\boldsymbol{Q}_{\text{postPosition}-fr+1}] \tag{11}$$

编码层：编码层将 $\boldsymbol{X}_c$ 输出至多维空间。

位置编码层：为了解决位置信息丢失问题，Transformer 将编码层的输出输入到位置编码层来获取位置信息。该层采用固定位置编码方法，具体编码函数定义如下：

$$\begin{cases} PE(pos,2i)=\sin(pos/10\,000^{2i/d_{\text{model}}}) \\ PE(pos,2i+1)=\sin(pos/10000^{2i+1/d_{\text{model}}}) \end{cases} \tag{12}$$

其中，d_{model} 是编码层的数据维度，pos 是帧位置，i 表示[0，$d_{\text{model}}/2$]中的维度信息。

编码器和解码器层：编码器层和解码器层有两个相似的模块，一个是多头注意力机制(Mul-ti-Head Attention，MHA)模块，另一个是前馈网络(Feed Forward Network)模块。

多头注意力机制模块可通过以下公式定义：

$$\begin{cases} \boldsymbol{Z}=MHA(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=[\boldsymbol{head}_1,\cdots,\boldsymbol{head}_n]\cdot\boldsymbol{W}^o \\ head_i=Attention(\boldsymbol{QW}_i^Q,\boldsymbol{KW}_i^K,\boldsymbol{VW}_i^V) \\ Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\dfrac{\boldsymbol{QK}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \end{cases} \tag{13}$$

其中，$\boldsymbol{W}_i^Q$，$\boldsymbol{W}_i^K$，$\boldsymbol{W}_i^V$ 是根据输入随机生成的权重矩阵向量；$\boldsymbol{Q}$，$\boldsymbol{K}$，$\boldsymbol{V}$ 是随机生成的 3 个矩阵向量；$\boldsymbol{W}^O$ 是可学习的线性变换。为确保结果的稳定性，Transformer 引入了缩放系数 d_k，其为向量的维度。最后，Transformer 将结果归一化并乘以矩阵 $\boldsymbol{V}$，以获得注意力权重的求和表示。

前馈网络模块提供非线性变换，并表示为：

$$\boldsymbol{O}^d=\text{Relu}(\boldsymbol{ZW}_1+\boldsymbol{b}_1)\boldsymbol{W}_3+\boldsymbol{b}_3+\boldsymbol{Z} \tag{14}$$

其中，$\boldsymbol{W}_1$ 和 $\boldsymbol{W}_3$ 是权重参数，$\boldsymbol{b}_1$ 和 $\boldsymbol{b}_3$ 是偏差参数。

输出层：对输入 $\boldsymbol{O}^d$ 做线性变换，最后通过 Softmax 函数输出最后的结果 $\boldsymbol{T}_i$。

$$\boldsymbol{T}_i=\text{softmax}(\boldsymbol{O}^d,\boldsymbol{W}_L) \tag{15}$$

其中，$\boldsymbol{T}_i$ 表示由 TFGN 预测的第 i 个过渡帧，$\boldsymbol{W}_L$ 是全连接层的矩阵向量。

与 SPSN 的输出类似，在 Linear &Softmax 层之后，我们设置了两个标志 ***preIndexToken*** 和 ***postIndexToken*** 分别表示 TFGN 生成的过渡帧的第一帧和最后一帧，当 T_i 与标志相等时，生成的手语过渡帧即为 $\boldsymbol{T}_i$ 的当前帧 i。最后，组合[$preIndex$，$postIndex$]的所有过渡帧，生成 TFGN 的输出 $\boldsymbol{O}$。

$$\begin{cases} \boldsymbol{T}_i=\text{softmax}(\boldsymbol{O}^d,\boldsymbol{W}_L) \\ preIndex=t_1, & \text{if } \boldsymbol{T}_{t_1}=\boldsymbol{preIndexToken} \\ postIndex=t_2, & \text{if } \boldsymbol{T}_{t_2}=\boldsymbol{postIndexToken} \\ \boldsymbol{O}=[\boldsymbol{T}_{\text{preIndex}},\cdots,\boldsymbol{T}_{\text{postIndex}}] \end{cases} \tag{16}$$

3.3 Filter Layer

如图 1 所示，Filter Layer 具有 4 个输入：原始输入 X、SPSN 的输出结果 $prePosition$ 和 $postPosition$，以及 TFGN 的输出结果 $\boldsymbol{O}$。过滤层分割原始输入 $\boldsymbol{X}$，并在[$prePosition$，$postPosition$]之间插入生成的过渡帧 $\boldsymbol{O}$，最终生成 LpTransformer 模型的输出 $\boldsymbol{Y}$。

$$\boldsymbol{Y}=[\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_{\text{prePosition}},\boldsymbol{O},\boldsymbol{Q}_{\text{postPosition}},\cdots,\boldsymbol{Q}_M] \tag{17}$$

4 实验结果

为验证模型的有效性，本节在国家通用手语 3D 数据库中进行对比实验。实验主要展示不同神经网络框架的预测结果，并进行对比分析和消融实验。

4.1 数据集

手语 3D 数据来源于《国家通用手语词典》，采用动捕设备采集。数据集包括 6 707 个手语词，每个手语词是人体 53 个层级骨骼的旋转数据，以双臂自然下落开始和结束，并表示为 bvh 骨架动画的形式。实验使用其中 100 个手语词组成的 10 000 条拼接词，首先将每帧手语动画的三维旋转数据转换成位置坐标，然后将坐标数据归一到[−0.5，0.5]的区间，最后将 53 个骨骼坐标展开，形成 159 维的向量。

4.2 评估细则

Accuary：使用如下公式定义模型精度

$$\text{if } Accuracy_k=\begin{cases} \dfrac{\sum_{i=1}^{Len_k^T}\sum_{j=1}^{159}E_M-(T_{ij}-L_{ij})}{E_M}, & Len_k^T=Len_k^L \\ \dfrac{E_M-|Len_k^T-Len_k^L|*\sigma}{E_M}, & Len_k^T\neq Len_k^L \end{cases} \tag{18}$$

$$\text{if } Accuary=\sum_{k=1}^{K}Accuary_k \tag{19}$$

当预测帧的总数和实际帧的总数不同时，采用固定误差系数 σ 的方式。其中，E_M 是数据集允许的最大误差，可根据不同的数据集动态调整，在我们的数据集中将其定义为 1；K 是测试集的大小；Len_k^T 表示第 k 个数据通过 LpTransformer 模型生成的过渡帧序列的长度；Len_k^L 表示其真实的过渡帧序列长度；T_{ij} 和 L_{ij} 分别代表预测序列和实际序列的第 i 帧第 j 个点。

MSE 误差：在过渡帧的预测中存在极端误差，如果预测结果与实际结果相差较大就会产生手势的变形，因此用 MSE 误差衡量极端预测点的影响。

$$MSE_k=\begin{cases}\dfrac{\sum_{i=1}^{Len_k^T}\sum_{j=1}^{159}(T_{ij}-L_{ij})^2}{159}, & Len_k^T=Len_k^L\\ \dfrac{(|Len_k^T-Len_k^L|*\lambda)^2}{159}, & Len_k^T\neq Len_k^L\end{cases} \tag{20}$$

$$MSE=\sum_{k=1}^{K}MSE_k \tag{21}$$

双语评估评分(Bilingual Evaluation Understudy, BLEU):用于在自然语言处理任务中衡量候选句子和参考句子之间的相似性。根据文献[7],我们用它衡量预测过渡帧和实际过渡帧之间的重合程度。当评估 BLEU 分数时,划分预测帧序列为不同长度 n 的子序列,并计算预测的序列在真实序列中出现的比例。例如存在预测序列$[\boldsymbol{T}_1,\boldsymbol{T}_2,\boldsymbol{T}_3,\cdots,\boldsymbol{T}_m]$,真实序列$[\boldsymbol{L}_1,\boldsymbol{L}_2,\boldsymbol{L}_3,\cdots,\boldsymbol{L}_m]$,计算 BELU-2 分数时,可将预测序列分为 $m-n+1$ 个子序列,即 $m-1$ 个子序列$[\boldsymbol{T}_1,\boldsymbol{T}_2]$,$[\boldsymbol{T}_2,\boldsymbol{T}_3]$,$\cdots$,$[\boldsymbol{T}_{m-1},\boldsymbol{T}_m]$,当存在$[\boldsymbol{L}_i,\boldsymbol{L}_j]$使其等于其中某个预测子序列时,认为该预测子序列在真实子序列中出现。在本文中,测量的 BLEU 范围 $n\in[1,\cdots,4]$,对应的结果表示为 BLEU-n。

成功预测拼接位置数据占全部数据的比例(The Proportion of Successfully Predicted Splicing Position Data to All Data, PTA):在这项工作中,模型能否预测出拼接位置十分重要,所以我们提出了如下衡量标准。

$$PTA=\frac{N_P}{N_T} \tag{22}$$

其中,N_P 是符合式(10)条件的测试集样本数量,N_T 是测试集所有数据的数量。

预测拼接位置与实际位置对齐的比值(The Ratio of the Predicted Splicing Position is Aligned with the Actual Position, PAA):即使某些模型可以预测出拼接位置,预测的拼接位置与实际拼接位置也并不相同。因此,我们提出了衡量标准:

$$PAA=\frac{N_A}{N_T} \tag{23}$$

其中,N_A 表示预测拼接位置和实际拼接位置相同的测试集样本数量。

训练细节:实验在英伟达 GeForce RTX 3090 的 PC 机上进行。ConvS2S[18],LSTM 和 Trans-former[19] 都适用于实验环境。ConvS2S 模型的编解码器由 20 个编解码层构成,卷积核宽度均为 3。Transformer 的编解码器由 $N=6$ 个编解码层堆叠组成。此外,参考原始论文[19],将 MHA 的头数设置为 8。设定所有模型的编码器和解码器的神经元数量为 512,并且使用 SGD 方式训练模型,初始学习率为 1×10^{-3}。

4.3 对比实验

根据 3.2 节,尝试使用不同的 fr 值来寻找 Culling 层的最佳参数。Accuary 随 fr 值增加的变化关系如图 4(a)所示。在训练结束时,$fr=6$ 模型由于过拟合,Accuary 急剧下降;而 $fr=8$ 模型的 Accuary 接近 0.99。图 4(b)展示了不同 fr 值的平均预测时间和 MSE 误差的关系。虽然 $fr=6$ 与 $fr=8$ 模型的平均预测时间相似,但 $fr=8$ 模型的 MSE 误差比 $fr=6$ 模型的更低。因此,实验选择 $fr=8$ 作为 Culling 层的最佳参数。

实验将 LpTransformer 与其他 3 种模型进行比较。Accuary 随迭代的变化如图 4(c)所示。显然,LpTransformer 的 Accuary 优于其他 3 个模型。同样,图 4(d)中给出了不同模型的 MSE 误差和平均预测时间的关系,从图中可以看出 ConvS2S 速度最快,因为它使用了相同的卷积结构;而 Transformer 的速度最慢,因为其输入序列较长且 MHA 具有大量的计算参数。根据主观测评,0.002 以内的 MSE 误差基本可以忽略预测过渡帧和实际过渡帧之间的视觉误差。因此,我们将 0.002 定义为 MSE 误差的阈值。其他 3 个模型的 MSE 误差远超误差阈值。LpTransformer 的 MSE 误差为 0.0015,且其预测速度比 Transformer 快 5 倍,因此 LpTransformer 比其他 3 个模型具有更低的预测误差和更快的预测速度。

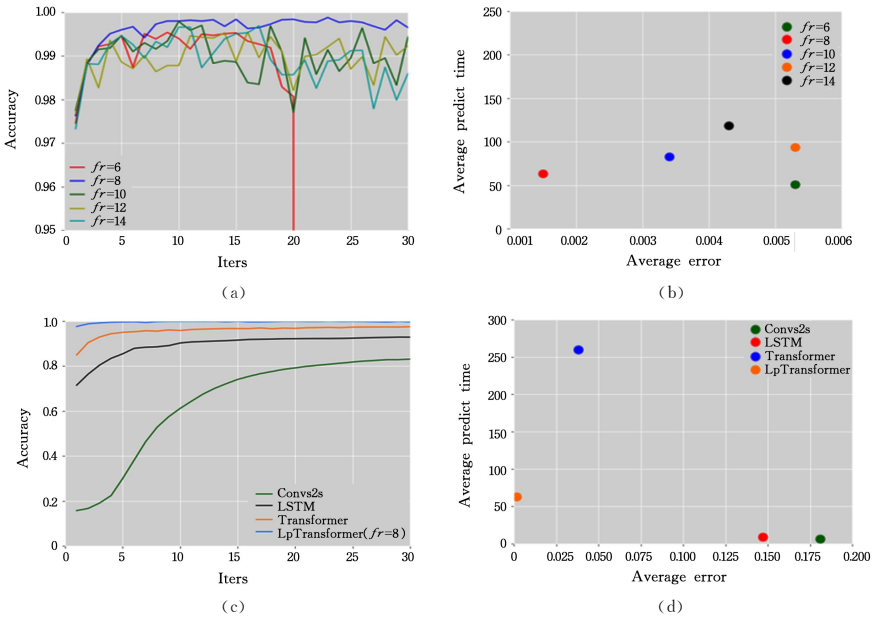


图 4 对比实验结果

Fig. 4 Comparative experimental results

如表 1 所列，ConvS2S，LSTM 和 LpTransformer 模型在 BLEU-1 上取得高分，这意味着模型在预测单个数据时效果很好，然而其 BLEU-2/3/4 急剧下降；而 LpTransformer 保持稳定，这意味着 LpTransformer 在长序列的预测情况下优于其他 3 种模型。ConvS2S 的 PTA 为 0.251，这意味着该模型几乎预测不到拼接位置。LSTM 和 Transformer 的 PTA 接近 1，这表明其几乎预测出了所有的拼接位置。然而，它们的 PAA 低至 0.784 和 0.601，这证明了大多数预测的拼接位置是错误的。LpTransformer 的所有指标均优于其他 3 种模型，证明了模型的有效性。

表 1 对比实验各个模型的指标结果

Table 1 Metrics of each model in contrast experiment

Metrics	ConvS2S	LSTM	Transformer	LpTransformer
BELU-1	0.744	0.881	0.735	**0.894**
BELU-2	0.649	0.771	0.652	**0.869**
BELU-3	0.599	0.677	0.615	**0.848**
BELU-4	0.563	0.610	0.615	**0.824**
PTA	0.251	0.999	0.996	**0.999**
PAA	0.999	0.784	0.601	**0.999**

4.4 消融实验

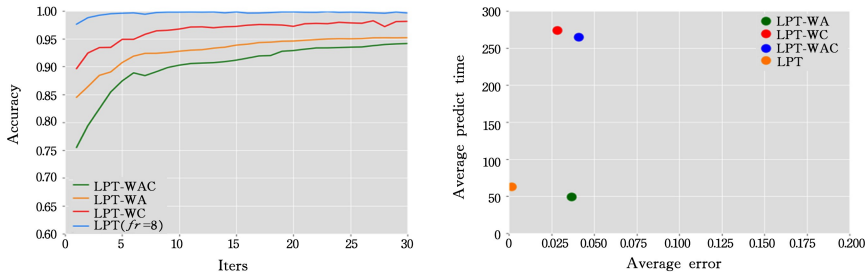
消融实验的具体网络结构如表 2 所列，其中 LPT 表示 LpTransformer 模型，LPT-WA 表示 SPSN 不具有注意力层，LPT-WC 表示 TFGN 不具有选择层，LPT-WAC 表示二者均无。

表 2 消融实验模型表

Table 2 Ablation rxperiment models

Model	Attention	Culling-Layer
LPT-WA	×	√
LPT-WC	√	×
LPT-WAC	×	×
LPT	√	√

首先，图 5(a)中给出了 4 个模型训练过程中模型迭代次数与精度的变化，可以观察出在训练结束时各个模型均已收敛，且 LPT 模型具有最高的精度。其次，图 5(b)表示 4 个模型的平均误差和平均预测时间之间的关系，由于 LPT-WA 模型缺少注意力层，因此平均误差大于 LPT，而 LPT-WC 与 LPT-WAC 均缺少选择层，导致其预测的平均时间远长于 LPT。表 3 给出各个模型的评价指标结果，由于 LPT-WA 和 LPT-WAC 缺少注意力层，因此模型的 PAA 较低，同时 LPT-WC 和 LPT-WAC 缺少选择层，导致模型的 BELU-4 评分较低，这表明模型在长序列的手势预测上存在缺陷。



(a) (b)

图 5 消融实验结果

Fig. 5 Ablation experimental results

表 3 消融实验各个模型的指标结果

Table 3 Metrics of each model in ablation experiment

Metrics	LPT-WA	LPT-WC	LPT-WAC	LPT
BELU-1	0.854	0.823	0.844	**0.894**
BELU-2	0.785	0.774	0.733	**0.869**
BELU-3	0.731	0.697	0.688	**0.848**
BELU-4	0.704	0.642	0.621	**0.824**
PTA	0.999	0.999	0.984	**0.999**
PAA	0.829	0.999	0.801	**0.999**

4.5 可视化实验

本小节展示过渡帧的可视化图像。由于 Conv-S2S 的预测效果极差，因此仅展示 LSTM，Transformer 和 LpTransformer 的可视化结果。如图 6 所示，3 个模型的前 2 帧预测过渡帧没有表现出较大差异，均与实际过渡帧一致。然而，随着预测帧数逐渐增加，LSTM 和 Transformer 出现了明显的手势扭曲现象。

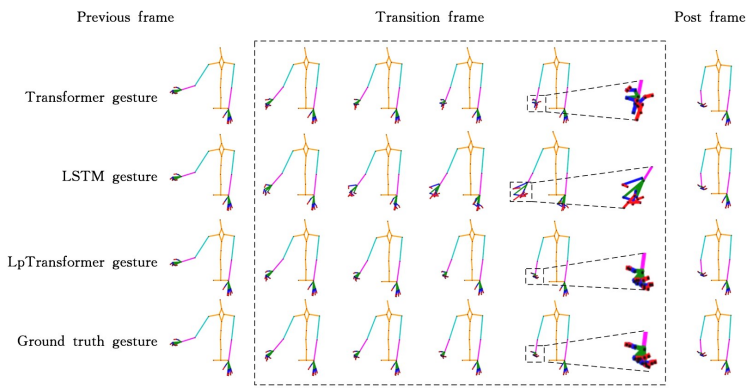
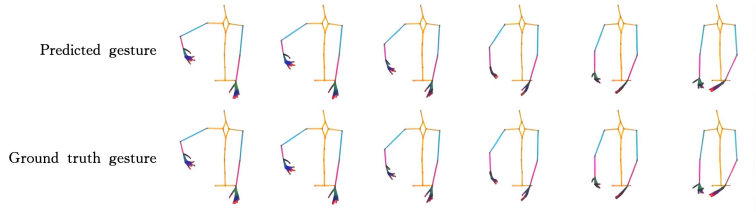


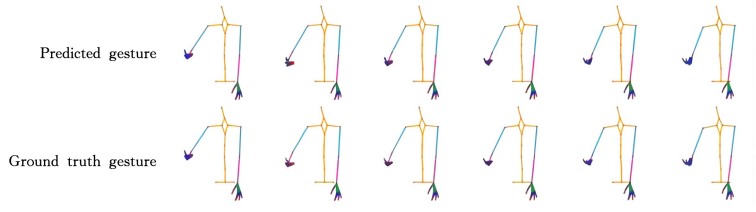
图 6 模型生成的过渡帧可视化图像

Fig. 6 Visualization of transition frames generated by the model

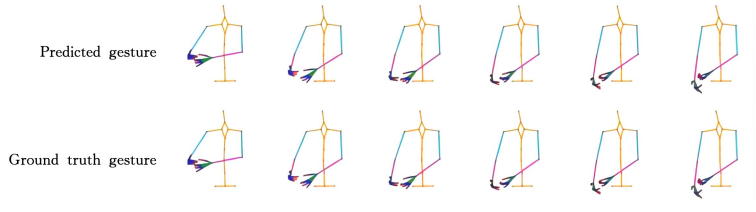
实验表明，LpTransformer 的精度优于 LSTM 和 Transformer。LpTransformer 预测的拼接位置和生成的过渡帧与实际一致，并且随着预测帧数量的增加，预测的过渡帧手势保持稳定。更多手语动画过渡帧的实验结果如图 7 所示。



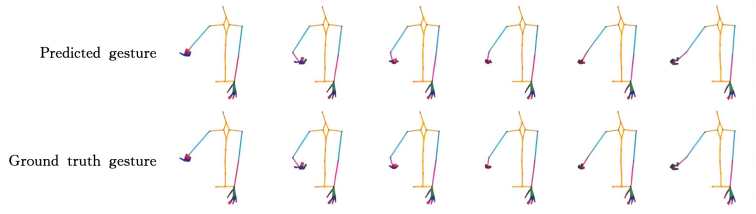
(a) Right and left hands closed



(b) Rotate left hand and fix right hand



(c) Drop left hand and clench right hand



(d) Move left hand and fix right hand

图 7 不同手语动作的视觉图像

Fig. 7 Visual images of different sign language movements

结束语 手语动画拼接是生成完整手语语句的一项重要任务。在现有的工作中，一般采用传统算法[1-3]寻找拼接点并生成过渡帧。本文提出了一种基于 LSTM 的拼接点探测模型和基于 Transformer 的并发过渡帧生成模型。实验结果表明，本文提出的 LpTransformer 在手语动画拼接工作上的效率比 Transfor-mer 快 5 倍，在消费级显卡上可以实现实时拼接，且生成的手势稳定不变形，保证了手语意义解读的正确性。在骨骼的动画片段动作前后段差异较大的情况下，LpTransformer 能够生成更加自然的过渡帧，主要体现在过渡帧中的人体各关节速度和方向更接近于真实场景。因此相较于传统方法或者其他模型，本文提出的 LpTransformer 有更好的表现。

参 考 文 献

[1] ZHU T T. The research of chinese sign language video synthesis aided by 3D information [D]. Beijing: Beijing University of Technology, 2014.

[2] CHEN J X. Study on key technologies of the chinese sign language synthesis based on the video stitching [D]. Hefei: University of Science and Technology of China, 2017.

[3] ZHAO H N. Chinese sign language news broadcasting system based on virtual human technology [D]. Harbin: Harbin Institute of Technology, 2008.

[4] DUARTE A C. Cross-modal neural sign language translation [C]// Proceedings of the 27th ACM International Conference on Multimedia. Nice: ACM, 2019: 1650-1654.

[5] KAPOOR P, MUKHOPADHYAY R, HEGDE S B, et al. Towards Automatic Speech to Sign Language Generation[C]// Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association. Brno: ISCA, 2021: 3700-3704.

[6] XIAO Q, QIN M, YIN Y. Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people[J]. Neural networks, 2020, 125: 41-55.

[7] SAUNDERS B, CAMGOZ N C, BOWDEN R. Progressive transformers for end-to-end sign language production[C]// European Conference on Computer Vision. Glasgow: Springer, 2020: 687-705.

[8] ZELINKA J, KANIS J. Neural sign language synthesis: Words are our glosses[C]// Proceedings of the IEEE/CVF Winter Con-

ference on Applications of Computer Vision. Snowmass Village: IEEE, 2020: 3395-3403.

[9] SUNDERS B, CAMGOZ N C, BOWDEN R. Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks[J]. International Journal of Computer Vision, 2021, 129(7): 2113-2135.

[10] HUANG W, PAN W, ZHAO Z, et al. Towards Fast and High-Quality Sign Language Production[C]// Proceedings of the 29th ACM International Conference on Multimedia. China: ACM, 2021: 3172-3181.

[11] ZHOU C, LAI Z, WANG S, et al. Learning a deep motion interpolation network for human skeleton animations[J]. Computer Animation and Virtual Worlds, 2021, 32(3/4): e2003.

[12] SAUNDERS B, CAMGOZ N C, BOWDEN R. Skeletal Graph Self-Attention: Embedding a Skeleton Inductive Bias into Sign Language Production[J]. arXiv: 2112. 05277, 2021.

[13] ZHANG Z, XUE W, HUANG W, et al. Effective Video Frame Acquisition for Image Stitching [J]. IEEE access, 2020, 8: 217086-217097.

[14] LIU Q, SU X, ZHANG L, et al. Panoramic video stitching of dual cameras based on spatio-temporal seam optimization[J]. Multimedia Tools and Applications, 2020, 79(5): 3107-3124.

[15] VASUHI S, SAMYDURAI A, VIJAYAKUMAR M. Multicamera Video Stitching for Multiple Human Tracking[J]. International Journalof Computer Vision and Image Processing (IJCVIP), 2021, 11(1): 17-38.

[16] CAO W. Applying image registration algorithm combined with CNN model to video image stitching[J]. The Journal of Supercomputing, 2021, 77(12): 13879-13896.

[17] DAS A, RAUN E S K, KJARGAARD M B. Cam-stitch: Trajectory cavity stitching method for stereo vision cameras in a public building[C]// Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things. New York: Association for Computing Machinery, 2019: 8-14.

[18] GEHRING J, AULI M, GRANGIE D, et al. Convolutional sequence to sequence learning[C]// International Conference on Machine Learning. Sydney: PMLR, 2017: 1243-1252.

[19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv: 1706. 03762, 2017.

HUANG Hanqiang, born in 1998, postgraduate. His main research interests include graphic image processing and sign language processing.

XING Yunbing, born in 1982, master, senior engineer. His main research interests include sign language and human-computer interaction.

（责任编辑：柯颖）