浙江大學

硕 士 研 究 生 读 书 报 告

题 目 Word-As-Image for Semantic Typography

作者姓名 龙咏柳

作者学号 22351137

指导教师 李启雷

学科专业 软件工程

所在学院 软件学院

提交日期 二○二四年 1 月 9 日

Word-As-Image for Semantic Typography

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Qilei Li

By

Yongliu Long

Zhejiang University, P.R. China

2023

摘要

词汇即图像（word-as-image）是一种语义排版技术，通过单词插图在保持可读性的同时呈现单词含义的视觉表达。该论文提出了一种自动创建词汇即图像插图的方法。这一任务充满挑战，要求对单词的语义有深刻理解，并在视觉上以令人愉悦和易读的方式描绘这些语义。该方法依赖于最近大规模预训练的语言-视觉模型在视觉上提炼文本概念的卓越能力。

研究的目标是设计简单、简洁、以黑白为主的图案，以有效传达语义。字母的颜色和纹理保持不变，不使用修饰。方法通过预训练的 Stable Diffusion 模型引导，优化每个字母的轮廓以传达所需的概念。引入额外的损失项以确保文本的可读性，并保留字体风格。该研究展示了在众多示例中产生的高质量和引人入胜的结果，并与其他技术进行了比较。

关键词：词汇即图像，语义排版技术，语义理解

Abstract

A word-as-image constitutes a semantic typography technique, wherein a word illustration visually represents the meaning of the word while maintaining readability. The paper introduces a method for the automatic creation of word-as-image illustrations. This task poses significant challenges, demanding a semantic grasp of the word and a creative approach to visually depict these semantics in an aesthetically pleasing and legible manner. The approach relies on the notable capabilities of recent large pretrained language-vision models to visually distill textual concepts.

The research targets simple, concise, black-and-white designs that effectively convey semantics. The color and texture of the letters remain unchanged, and embellishments are deliberately avoided. The method optimizes the outline of each letter to communicate the desired concept, guided by a pretrained Stable Diffusion model. Additional loss terms are incorporated to ensure text legibility and preserve the font's style. The findings showcase high-quality and engaging results across numerous examples, with comparisons made to alternative techniques.

Keywords： word-as-image, semantic typography technique, semantic understanding

1 引言

本次读书报告解读的论文为《Word-As-Image for Semantic Typography》[1]，改论文收录为 SIGGRAPH 2023 中，并且获得了 Honorable Mention Award。因此本人认为文章具有一定的阅读价值，在此对论文进行解读。

这篇文章的主要目标是通过使用形状骨架化和识别离散局部对称性的方法，针对语义排版任务，将单词转化为语义图像。作者提出了一种基于预训练的 Stable Diffusion[2]模型的方法，通过优化控制点的坐标来生成字形图示。

语义排版是使用排版在视觉上强化文本含义的实践。这可以通过选择字体、字体大小、字体样式和其他排版元素来实现。一种更精细、更吸引人的语义排版技术是单词作为图像的插图，在这种插图中，给定单词的语义仅使用其字母的图形元素来说明。这样的插图提供了一个词的意思的视觉表现，同时也保留了这个词作为一个整体的可读性。

下面图表 1 是一些手动创建的 word-as-image 示例：

图表 1 手动创建的文字图像插图

例如，为了创造“爵士”的描述，设计师必须首先选择最适合文本语义的视觉概念（萨克斯风），考虑所需的字体特征，然后选择最合适的字母来替换。寻找合适的视觉元素来说明一个概念是不明确的，因为有无数的方法来说明任何给定的概念。此外，人们不能简单地将选定的视觉元素复制到单词上——需要找到字母形状的细微修改。

在这篇文章中，作者们基于深度学习的最新进展和结合语言和视觉理解的巨大基础模型的可用性，定义了一种用于自动创建单词图像插图的算法。得到的插图（见图表 2）可以用于标志设计、标志、贺卡和邀请，或者只是为了好玩。它们可以按原样使用，也可以作为进一步改进设计的灵感。

文中的文字图像插图专注于改变字母的几何形状来传达意思，不改变颜色或质地，也不使用装饰。这允许简单，简洁，黑白分明的设计，清晰地传达语义。此外，由于作者保留了字母的基于矢量的表示，这允许在任何大小的平滑光栅化，以及在需要时使用颜色和纹理对插图应用额外的样式操作。

图表 2 文中文字图像插图的几个示例

给定一个输入单词，文章中的方法分别应用于每个字母，允许用户稍后选择最喜欢的组合进行替换。每个字母会表示为一个封闭的矢量形状，并优化其参数以反映单词的含义，同时仍然保留其原始风格和设计。

文章中依靠预训练的 Stable Diffusion 模型来连接文本和图像，并利用分数蒸馏抽样[3]方法来鼓励字母的外观反映所提供的文本概念。由于 Stable Diffusion 模型是在光栅图像上训练的，因此作者们使用了可微光栅器[4]，该光栅器允许从基于光栅的损失向形状参数反向传播梯度。

为了保持原始字母的形状并确保单词的易读性，文中使用了两个额外的损失函数。第一个损失通过限制字母形状的三角形上的形变尽可能地保形来调节形状的修改。第二种损失通过比较得到的栅格化字母的低通滤波器与原始滤波器来保留字母的局部音调和结构。

文中比较了几个基线，并展示了使用各种字体和大量概念的许多结果。生成的文字图像插图传达了预期的概念，同时保持易读性和保留字体的外观，展示了视觉创造力。

2 背景

2.1 字体和矢量表示

现代字体格式，如 TrueType 和 PostScript，使用矢量图形表示其轮廓来表示字形。具体来说，轮廓通常由一组直线和 B 样条曲线表示。这种表示允许按比例缩放字母，并将其光栅化成任何所需的大小，类似于其他矢量表示。文中的方法保留了这个属性，因为输出保留了字母的矢量化表示。

2.2 Latent Diffusion Models

扩散模型是生成模型，通过对从高斯分布中采样的变量进行逐渐去噪来训练以学习数据分布。文中的工作使用了公开可用的文本到图像的 Stable Diffusion 模型。Stable Diffusion 是一种潜在扩散模型(Latent Diffusion Model)，其扩散过程是在预训练图像自编码器的潜在空间上进行的。编码器$\mathcal{E}$的任务是将输入图像 x 映射到一个潜在向量 z 上，并且训练解码器$\mathcal{D}$进行解码，从而使$\mathcal{D}(z) = x$。

作为第二阶段，一个去噪扩散概率模型(DDPM)[5]会被训练在学习到的潜在空间内生成代码。在训练的每一步，标量$t \in \{1,2, \ldots T\}$被均匀采样，并用于定义一个带噪声的潜在码$z_t = \alpha_t z + \sigma_t \epsilon$，其中$\epsilon \sim N(0, I)$，$\alpha_t$，$\sigma_t$是控制噪声调度的项，并且是扩散过程时间$t$的函数。

基于 UNet 架构[6]的去噪网络ϵ_θ，将噪声代码z_t、时间步长t和可选条件向量$c(y)$作为输入，并负责预测添加的噪声ϵ。LDM（Latent Diffusion Model）损失定义为:

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathrm{z} \sim \mathcal{E}(\mathrm{x}), \mathrm{y}, \epsilon \sim N(0,I), \mathrm{t}}\left[||\epsilon - \epsilon_\theta(\mathrm{z_t}, \mathrm{t}, \mathrm{c(y)})||_2^2\right].$$

在 Stable Diffusion 中，对于文本到图像的生成，条件向量是由预训练的 CLIP 文本编码器[7]生成的文本嵌入。在推理时，对随机潜码$z_T \sim N(0, I)$进行采样，并通过训练后ϵ_θ进行迭代去噪，直到得到一个干净的z_0潜码，该潜码通过解码器$\mathcal{D}$生成图像x。

2.3 分数蒸馏

预训练的大型文本模型的强先验需要被利用来生成超越光栅化图像的模态。在 Stable Diffusion 中，文本调节是通过 UNet 网络中以不同分辨率定义的跨注意

层来执行的。因此，用条件扩散模型来指导优化过程并非易事。

DreamFusion[8]提出了一种使用扩散损失来优化 NeRF 模型参数的方法，用于文本到 3d 的生成。在每次迭代中，从随机的角度渲染辐射场，形成图像x，然后对图像进行降噪处理，形成$x_t = \alpha_t x + \sigma_t \epsilon$。然后将噪声图像传递给预训练 UNet 模型 Imagen[9]的，该模型输出对噪声ϵ的预测。分数蒸馏损失由原始扩散损失的梯度定义:

$$\nabla_{\phi} \mathcal{L}_{SDS} = \left[w(t)(\epsilon_{\theta}(x_t, t, y) - \epsilon) \frac{\partial x}{\partial \phi} \right]$$

其中，y是条件文本提示符，是 NeRF 的参数，$w(t)$是依赖于α_t的常数乘子。在训练过程中，梯度被反向传播到 NeRF 参数，以逐渐改变 3D 对象以适应文本提示。UNet 的梯度被跳过，修改 Nerf 参数的梯度直接来自 LDM 损失。

2.4 VectorFusion

最近，VectorFusion[10]利用 SDS 损失来完成文本到 SVG 生成的任务。提出的生成流程包括两个阶段。首先，给定一个文本提示，使用 Stable Diffusion(在提示后添加后缀)生成图像，然后使用 LIVE[11]自动矢量化。这定义了一组初始参数，将在使用 SDS 损失的第二阶段进行优化。在每次迭代中，使用可微光栅器[4]生成 600 × 600 的图像，然后按照 CLIPDraw[12]中的建议对其进行增强，以获得 512 × 512 的图像x_{aug}。然后将x_{aug}输入到 Stable Diffusion 预训练的编码器$\mathcal{E}$中，得到相应的潜码$z = \mathcal{E}(x_{aug})$。然后将 SDS 损失应用于该潜在空间，与 DreamFusion 中定义的方法类似:

$$\nabla_{\theta} \mathcal{L}_{\text{LSDS}} = \mathbb{E}_{t,\epsilon} \left[w(t) \left(\hat{\epsilon}_{\phi}(\alpha_t z_t + \sigma_t \epsilon, y) - \epsilon \right) \frac{\partial z}{\partial z_{aug}} \frac{\partial x_{aug}}{\partial \theta} \right]$$

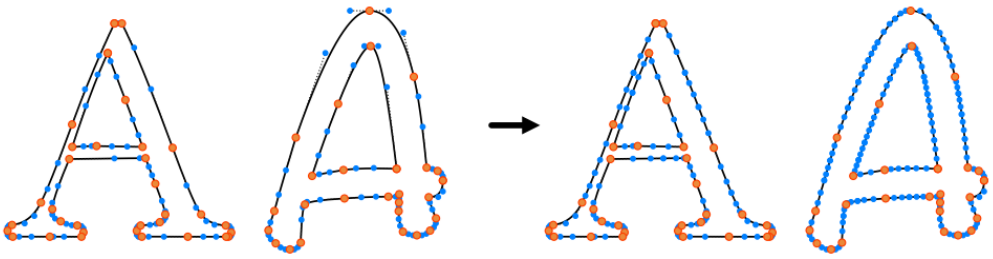
作者发现 SDS 方法对生成语义符号的任务很有用，因此遵循了 VectorFusion 中提出的技术步骤(例如增强和添加后缀)。

3 方法

给定一个单词W，单词有n个字母l_1，…l_n，文中的方法分别应用于每个字母l_i，以产生字母的语义视觉描述。然后，用户可以选择替换哪些字母，保留哪些字母的原始形式。

3.1 字母表示

首先需要定义W中字母的参数表示。文中使用 FreeType 字体库来提取每个字母的轮廓。然后，将每个轮廓转换为一组立方贝塞尔曲线，以便在不同字体和字母之间具有一致的表示，并便于使用 diffvg [4]的可微光栅化。



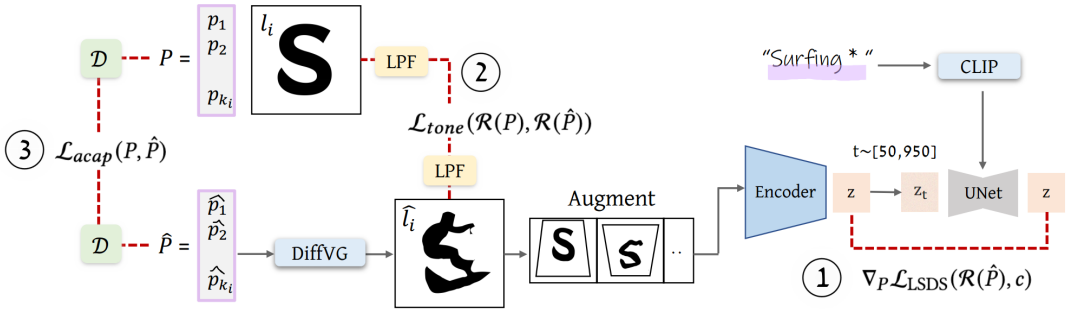
图表 3 字母的轮廓和控制点之前(左)和之后(右)的细分过程

根据字母的复杂性和字体的样式，提取的轮廓由不同数量的控制点定义。作者们发现，初始控制点的数量对最终的外观有显著的影响：随着控制点数量的增加，视觉变化发生的自由度也越大。因此，还需要对包含少量控制点的字母应用细分程序。文中为字母表中的每个字母定义所需数量的控制点(在不同的字体之间共享)，然后迭代地细分贝塞尔段，直到达到这个目标数量。在每次迭代中，作者们计算所有贝塞尔段之间的最大弧长，并将此长度的每个段分成两个(见图表 3)。具体在第 5.3 节中会分析控制点数量的影响。

这个过程定义了一组k_i控制点$P_i = \{p_j\}_{j=1}^{k_i}$，来代表字母$l_i$的形状。

3.2 优化

具体方法的流程如图表 4 所示。由于分别优化了每个字母l_i，因此为了简洁起见，文中将在下面的文本中省略字母索引 i，并将输入字母的控制点集定义为 P。



图表 4 方法概述

给定了 P 和期望的文本概念c(都在图表 4 中以紫色标记)，目标是生成一组新的控制点，$\hat{P}$，和一个调整后的字母$\hat{l}$，该字母既能传达给定的概念，又能保持原始字母l的整体结构和特征。

文中将学习到的控制点集合$\hat{P}$初始化，并将其传递给一个可微栅格化器$\mathcal{R}$[4](用蓝色标记)，该栅格化器输出栅格化的字母$\mathcal{R}(\hat{P})$。然后，栅格化的字母被随机增强并传递到预训练的 Stable Diffusion 模型中，该模型以 CLIP 嵌入给定文本c为条件。然后，按照第 2 节的描述，使用 SDS 损失$\nabla_{\hat{P}}\mathcal{L}_{\text{LSDS}}$来鼓励$\mathcal{R}(\hat{P})$传达给定的文本提示符。

为了保持每个字母的形状并确保整个单词的易读性，文中使用了两个额外的损失函数来指导优化过程。第一种损失通过定义形状变形的尽可能共形约束来限制整体形状变化。第二种损失通过限制修改后的字母的色调(即形状局部的暗区与亮区数量)，使其与原始字母不偏离太多，从而保留了字体的整体形状和风格(参见第 3.3 节)。

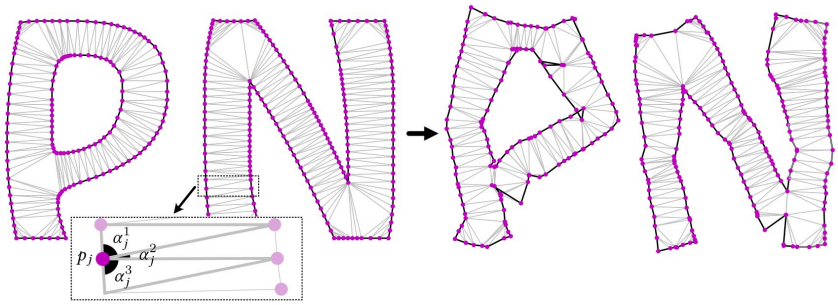
然后反向传播从所有损失中获得的梯度，以更新参数。实验中重复此过程 500 步，大约需要 5 分钟才能在 RTX2080 GPU 上生成单个字母插图。

3.3 损失函数

文中鼓励的主要目标产生的形状来传达意图的语义概念,是利用$\nabla_{\hat{P}}\mathcal{L}_{\text{LSDS}}$损失(在第 2 节中描述)。作者们观察到只使用$\nabla_{\hat{P}}\mathcal{L}_{\text{LSDS}}$会导致结果偏离最初字母的外观,这是不符合意图的。因此，训练的额外目标是保持字母$\mathcal{R}(\hat{P})$的形状和易读性，以及保持原始字体的特征。为此，文中使用两个附加损失。

3.3.1 尽可能保形变形损失

为了防止最终的字母形状偏离初始形状太多，文中对字母的内部进行三角测量，并将字母的变形约束为尽可能的保形(ACAP)[13]。文中使用约束 Delaunay 三角剖分在控制点集合上定义字形。而 Delaunay 三角剖分可以用来产生轮廓的骨架，因此 ACAP 丢失也隐含地捕获了字母形式的骨架表示。



图表 5 单词“pants”的初始形状(左)和结果形状(右)的约束 Delaunay 三角剖分的可视化说明

Delaunay 三角剖分$\mathcal{D}(P)$将以P为代表的字形分割成一组三角形。这为每个控制点p_j定义了一组大小为m_j的对应角度(见图 5)。这组角度表示为$\{\alpha_j^i\}_{i=1}^{m_j}$。ACAP 损失使优化后形状$\hat{P}$的诱导角不要偏离原始形状$P$的角度太多，定义为对应角度之间的 L2 距离:

$$\mathcal{L}_{acap}\left(P,\hat{P}\right)=\frac{1}{k}\sum_{j=1}^{k}\left(\sum_{i=1}^{m_j}\left(\alpha_j^i-\hat{\alpha}_j^i\right)^2\right),$$

其中$k=|P|$，$\hat{\alpha}$是由$\mathcal{D}\left(\hat{P}\right)$诱导产生的角度。

3.3.2 色调保存损失

为了保留字体的风格和字母的结构，文中增加了一个局部色调保存损失项。这个项限制调整后的字母的色调(形状中所有区域的黑色和白色的数量)，使其与原字体的字母的色调不偏离太多。为此，文中对光栅化的字母(变形之前和之后)应用低通滤波器(LPF)，并计算得到的模糊字母之间的 L2 距离:

$$\mathcal{L}_{tone}=\|LPF(\mathcal{R}(P))-LPF(\mathcal{R}(\hat{P}))\|_2^2.$$

一个字母模糊的例子如图 6 所示，可以看出，在模糊核中使用一个高的标准差来模糊掉小的细节，比如熊的耳朵。

图表 6 比较变形前(左)和变形后(右)的字母图像的低通滤波器，色调保存损失保留了字体的局部色调

最终的损失是用三个项的加权平均和来定义：

$$\min_{\hat{P}} \nabla_{\hat{P}} \mathcal{L}_{\text{LSDS}}(\mathcal{R}(\hat{P}), c) + \alpha \cdot \mathcal{L}_{\text{acap}}(P, \hat{P}) + \beta_t \cdot \mathcal{L}_{\text{tone}}(\mathcal{R}(P), \mathcal{R}(\hat{P}))$$

3.4 加权

选择上述三种损失的相对权重对于最终字母的外观至关重要。$\nabla_{\hat{P}}\mathcal{L}_{\text{LSDS}}$损失鼓励形状偏离原来的外观更好地符合语义概念,而$\mathcal{L}_{\text{tone}}$和$\mathcal{L}_{\text{acap}}$两个项负责维持原来的形状。因此，作者设置在公式中有两个相互竞争的部分，并希望在它们之间找到一个平衡，以保持字母的易读性，同时允许所需的语义形状发生变化。

而事实上，$\mathcal{L}_{\text{tone}}$是非常占优势的。在某些情况下，如果从一开始就使用它，则不会执行语义变形。因此，将$\mathcal{L}_{\text{tone}}$的权重调整为只在发生一些语义变形之后才加入。具体定义如下:

$$\beta_t = a \cdot \exp\left(-\frac{(t-b)^2}{2c^2}\right),$$

其中a= 100，b= 300，c= 30。相同的超参数选择适用于各种单词、字母和字体。

4 实验

文中的方法能够处理广泛的输入概念以及支持不同的字体设计。实验中的结果表明，文中的方法可以处理来自许多不同类别和各种字体的输入，并且生成的结果是易读的和有创意的。图 7 演示了用文中的方法为同一个单词创建的插图如何遵循不同字体的特征。虽然文字图像插图的感知美学可能是主观的，但作者为有效的结果定义了三个目标:(1)它应该在视觉上捕捉给定的语义概念，(2)它应该保持可读性，(3)它应该保留原始字体的特征。

图表 7 文中的方法为单词“YOGA”生成图像，使用了 8 种不同的字体

文中在随机选择的一组输入上评估方法的性能。作者选择了五个常见的概念类——动物、水果、植物、运动和职业。使用 ChatGPT，文中为每个类随机抽取 10 个实例，总共得到 50 个单词。接下来，文中选择了四种具有鲜明视觉特征的字体，分别是 Quicksand、Bell MT、Noteworthy-Bold 和 HobeauxRococeauxSherman。对于每个单词，随机抽取四种字体中的一种，并将文中的方法应用于每个字母。对于每个含有n字母的单词，可以生成2^n可能的 word-as-images，它们都是替换插图字母的可能组合，一些示例如图 8 中所示。

图表 8 文中的方法为随机选择的单词创建单词图像插图

文中进行了一项感性研究，以定量地评估实验中所得到的词作为图像的三个目标。对于上面描述的五个类，作者从每个产生的 word-as-image 的插图中随机选择两个实例，并在视觉上从每个单词中选择一个字母，结果总共有 10 个字母。在每个问题中，作者们展示了一个孤立的字母插图，没有单词的上下文。为了评估文中的方法在视觉上描述所需概念的能力，作者们提供了来自同一类的四个标签选项，并要求参与者选择最能描述字母插图的一个。为了评估结果的易读性，作者们要求参与者从四个字母的随机列表中选择最合适的字母。为了评估字体风格的保留，作者们展示了四种字体，并要求参与者选择最适合插图的字体。文中收集了 40 位参与者的答案，结果见表 1。可以看出，概念的可识别性和字母的易读性水平是非常高的，字母插图与原字体 51%的风格匹配度远远高于随机的 25%。文中还测试了其中的算法，在相同的单词和字母上没有两个额外的结构和风格保留损失($\mathcal{L}_{\text{tone}}$和$\mathcal{L}_{\text{acap}}$)(表中“Only SDS”)。正如预期的那样，如果没有额外的约束，字母会明显变形，从而提高概念的可识别性，但降低易读性和字体样式保存。

表格 1 感性研究结果

Method	Semantics	Legibility	Font
Ours	0.8	0.9	0.51
Only SDS	0.88	0.53	0.33

5 总结

文中提出了一种自动创建矢量格式的文字图像插图的方法。文中的方法可以处理大量的语义概念，并使用任何字体，同时保持文本的易读性和字体的样式。文中的文字图像插图展示了视觉创造力，并为使用大型视觉语言模型进行语义排版打开了可能性，以达到机器学习模型和人类更协同的设计方法。

不过文章中提出的方法还有一定的局限性。首先，文中的方法是一个字母，一个字母地工作，因此，它不会改变整个单词的形状。将来的工作可以尝试优化几个字母的形状。其次，这种方法在具体的视觉概念上效果最好，而在更抽象的概念上可能会失败。这可以通过使用不同的概念而不是单词本身来优化字母的形状来缓解。第三，字母的布局也可以自动化，可以参考其他文献中的方法[14]。

参考文献

[1] Iluz S, Vinker Y, Hertz A, et al. Word-as-image for semantic typography[J]. arXiv preprint arXiv:2303.01818, 2023.

[2] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10684-10695.

[3] Poole B, Jain A, Barron J T, et al. Dreamfusion: Text-to-3d using 2d diffusion[J]. arXiv preprint arXiv:2209.14988, 2022.

[4] Li T M, Lukáč M, Gharbi M, et al. Differentiable vector graphics rasterization for editing and learning[J]. ACM Transactions on Graphics (TOG), 2020, 39(6): 1-15.

[5] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.

[6] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015: 234-241.

[7] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.

[8] Poole B, Jain A, Barron J T, et al. Dreamfusion: Text-to-3d using 2d diffusion[J]. arXiv preprint arXiv:2209.14988, 2022.

[9] Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding[J]. Advances in Neural Information Processing Systems, 2022, 35: 36479-36494.

[10] Jain A, Xie A, Abbeel P. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 1911-1920.

[11] Ma X, Zhou Y, Xu X, et al. Towards layer-wise image

vectorization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 16314-16323.

[12] Frans K, Soros L, Witkowski O. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders[J]. Advances in Neural Information Processing Systems, 2022, 35: 5207-5218.

[13] Hormann K, Greiner G. MIPS: An efficient global parametrization method[J]. Curve and Surface Design: Saint-Malo 1999, 2000: 153-162.

[14] Wang Y, Pu G, Luo W, et al. Aesthetic text logo synthesis via content-aware layout inferring[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 2436-2445.