多阈值 Kaniadakis 熵的简化表达和递推方法

何 熠[1,2]
HE Yi

摘 要 Kaniadakis 熵具有描述图像中非扩展性信息的能力，曾被用于构建单阈值 Kaniadakis 熵方法。为进一步拓展 Kaniadakis 熵在图像阈值化中的应用，文章在现有单阈值 Kaniadakis 熵方法的基础上提出了多阈值 Kaniadakis 熵方法。然而，多阈值 Kaniadakis 熵的表达式相对复杂，不利于多个阈值的计算。通过分析推导出多阈值 Kaniadakis 熵的简化表达式，并用数学归纳法证明了简化表达式的正确性。为更快速地确定阈值，进一步提出了一种基于简化 Kaniadakis 熵的多阈值递推方法。在基于穷举搜索法求解最优阈值的对比实验中，使用传统 Kaniadakis 熵、简化 Kaniadakis 熵及递推简化 Kaniadakis 熵三种表达式所得最优阈值保持一致，但简化 Kaniadakis 熵所耗运行时间约为传统 Kaniadakis 熵所耗运行时间的 50%，递推简化 Kaniadakis 熵所耗运行时间仅为传统 Kaniadakis 熵所耗运行时间的 1%。

关键词 多阈值化；Kaniadakis 熵；递推方法

doi：10.3969/j.issn.1672-9528.2025.03.012

0 引言

阈值化方法是一种低层次图像处理技术，其实现简单、性能稳定，被广泛应用于医学图像分析 [1]、工业缺陷检测 [2]、农业图像分析 [3]、红外图像处理 [4] 等领域。

在众多阈值化方法中，应用信息论中熵相关概念和理论的熵阈值法颇具代表性。特别是基于 Kaniadakis 熵的图像阈值化方法在处理具有长尾分布直方图的图像时表现出了显著的优势，广泛适用于工业无损检测、红外图像分割等应用场景。然而，现有的 Kaniadakis 熵方法主要是单阈值方法，对包含多个对象的复杂真实世界图像，单阈值无法有效分离各个对象，这限制了现有方法的适用范围。

为拓展现有 Kaniadakis 熵阈值方法，同时也为对图像进行 Kaniadakis 熵多阈值化处理，在现有单阈值 Kaniadakis 熵方法 [5] 的基础上，根据 Kaniadakis 熵的广义可加性 [6]，提出了多阈值 Kaniadakis 熵方法。由于 Kaniadakis 熵的广义可加性，多阈值 Kaniadakis 熵方法的目标函数表达式较为复杂，既不利于多个阈值的计算，且计算效率较低。针对这一问题，本文提出了多阈值 Kaniadakis 熵的简化表达式，并用数学归纳法从理论上证明了该简化表达式的正确性。相比传统表达式，简化表达式形式简洁，在大幅降低计算时间复杂度的同时更加易于实现。为进一步降低时间成本，提出了基于简化 Kaniadakis 熵的多阈值递推方法，在确保与非递推 Kaniadakis 熵方法选取阈值一致的基础上，进一步提高了计算效率。

1 本文方法

1.1 多阈值 Kaniadakis 熵方法

首先将单阈值 Kaniadakis 熵方法拓展至多阈值 Kaniadakis 熵方法。对于一幅大小为 $M\times N$，灰度级为 L 的灰度图像 I，灰度级为 i 的像素点出现的次数记为 $h(i)$，则每个灰度值出现的概率可表示为：

$$p(i)=\frac{h(i)}{M\times N},i=0,1,\cdots,L-1 \tag{1}$$

在多阈值情况下，令阈值个数为 n，则图像 I 被阈值 $\{t_1,t_2,\cdots,t_n\}$ 分为 n+1 类 $C_1,C_2,\cdots,C_{n+1}$，对应的 n+1 个概率分布为：

$$\begin{aligned}
&C_1:\ \frac{p(0)}{P_1},\ \frac{p(1)}{P_1},\ \cdots,\ \frac{p(t_1)}{P_1}\\
&C_2:\ \frac{p(t_1+1)}{P_2},\ \frac{p(t_1+2)}{P_2},\ \cdots,\ \frac{p(t_2)}{P_2}\\
&\cdots\\
&C_n:\ \frac{p(t_{n-1}+1)}{P_n},\ \frac{p(t_{n-1}+2)}{P_n},\ \cdots,\ \frac{p(t_n)}{P_n}\\
&C_{n+1}:\ \frac{p(t_n+1)}{P_{n+1}},\ \frac{p(t_n+2)}{P_{n+1}},\ \cdots,\ \frac{p(L-1)}{P_{n+1}}
\end{aligned} \tag{2}$$

式中：$P_1=\sum_{i=0}^{t_1}p(i)$，$P_2=\sum_{i=t_1+1}^{t_2}p(i)$，…，$P_{n+1}=\sum_{i=t_n+1}^{L-1}p(i)$。

运用 Kaniadakis 熵进行阈值化，则：

1. 水电工程智能视觉监测湖北省重点实验室 湖北宜昌 443002
2. 三峡大学计算机与信息学院 湖北宜昌 443002

$$
\begin{aligned}
S_\kappa(C_1)&=-\frac{1}{2\kappa}\left\{\sum_{i=0}^{t_1}\left(\left(\frac{p(i)}{P_1}\right)^{1+\kappa}-\left(\frac{p(i)}{P_1}\right)^{1-\kappa}\right)\right\}\\
S_\kappa(C_2)&=-\frac{1}{2\kappa}\left\{\sum_{i=t_1+1}^{t_2}\left(\left(\frac{p(i)}{P_2}\right)^{1+\kappa}-\left(\frac{p(i)}{P_2}\right)^{1-\kappa}\right)\right\}\\
&\cdots\\
S_\kappa(C_{n+1})&=-\frac{1}{2\kappa}\left\{\sum_{i=t_n+1}^{L-1}\left(\left(\frac{p(i)}{P_{n+1}}\right)^{1+\kappa}-\left(\frac{p(i)}{P_{n+1}}\right)^{1-\kappa}\right)\right\}
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
K(C_1)&=\frac{1}{2}\left\{\sum_{i=0}^{t_1}\left(\left(\frac{p(i)}{P_1}\right)^{1+\kappa}+\left(\frac{p(i)}{P_1}\right)^{1-\kappa}\right)\right\}\\
K(C_2)&=\frac{1}{2}\left\{\sum_{i=t_1+1}^{t_2}\left(\left(\frac{p(i)}{P_2}\right)^{1+\kappa}+\left(\frac{p(i)}{P_2}\right)^{1-\kappa}\right)\right\}\\
&\cdots\\
K(C_{n+1})&=\frac{1}{2}\left\{\sum_{i=t_n+1}^{L-1}\left(\left(\frac{p(i)}{P_{n+1}}\right)^{1+\kappa}+\left(\frac{p(i)}{P_{n+1}}\right)^{1-\kappa}\right)\right\}
\end{aligned}
\tag{4}
$$

多阈值 Kaniadakis 熵阈值化方法的目标函数为：

$$
\begin{aligned}
&S_\kappa(C_1\cup C_2\cup\cdots\cup C_{n+1})\\
&=\sum_{r=0,\text{step}=2}^{n}\kappa^r\cdot\sum_{\Omega\subseteq U,|\Omega|=r+1}\left(\prod_{i\in\Omega}S_\kappa(C_i)\prod_{j\in U-\Omega}K(C_j)\right)
\end{aligned}
\tag{5}
$$

式中：$r=2j$，$j\in\mathbb{Z}$，$0\le j\le\lfloor n/2\rfloor$；令 U 表示全集，$U=\{1,2,\cdots,n+1\}$；$\Omega\subseteq U$，$|\Omega|=r+1$ 表示集合 Ω 为全集 U 的子集，子集的大小为 $r+1$，即对于每个不同的 r 的取值，有 C_{n+1}^{r+1} 种情况；$\prod_{i\in\Omega}S_\kappa(C_i)\prod_{j\in U-\Omega}K(C_j)$ 表示选择集合 Ω 中所对应的几个分类的 Kaniadakis 熵与 Ω 的补集中所对应的其余分类的 Kaniadakis 熵的伴生式相乘。

当 Kaniadakis 熵之和 $S_\kappa(C_1\cup C_2\cup\cdots\cup C_{n+1})$ 达到最大时，其最优阈值 $\{t_1^*,t_2^*,\cdots,t_n^*\}$ 选取准则为：

$$
\{t_1^*,t_2^*,\cdots,t_n^*\}=\arg\max_{0<t_1<\cdots<t_n<L-1}\{S_\kappa(C_1\cup C_2\cup\cdots\cup C_{n+1})\}
\tag{6}
$$

1.2 多阈值 Kaniadakis 熵的简化表达式

多阈值 Kaniadakis 熵的表达式（6）具有较高的计算复杂度。以阈值数为 6 为例，运用 Kaniadakis 熵进行阈值化时需要计算 64 个多项式。随着阈值数 n 的增加，所需计算的多项式数量呈指数级增长，具体为 2^n。因此，显著增加了多阈值 Kaniadakis 熵方法在阈值计算上的难度。为解决这一问题，提出了多阈值 Kaniadakis 熵的简化表达式。该表达式通过优化计算过程，减少了所需计算的多项式数量，从而有效降低了阈值化的计算难度。多阈值 Kaniadakis 熵的简化表达式推导过程为：

对于一组给定的阈值 $\{t_1,t_2,\cdots,t_n\}$，图像被划分为 $n+1$ 类，令：

$$
Q_1=\sum_{i=0}^{t_1}\left(\frac{p(i)}{P_1}\right)^{1+\kappa},\ Q_2=\sum_{i=t_1+1}^{t_2}\left(\frac{p(i)}{P_2}\right)^{1+\kappa},\ \cdots,\ Q_{n+1}=\sum_{i=t_n+1}^{L-1}\left(\frac{p(i)}{P_{n+1}}\right)^{1+\kappa}
\tag{7}
$$

$$
R_1=\sum_{i=0}^{t_1}\left(\frac{p(i)}{P_1}\right)^{1-\kappa},\ R_2=\sum_{i=t_1+1}^{t_2}\left(\frac{p(i)}{P_2}\right)^{1-\kappa},\ \cdots,\ R_{n+1}=\sum_{i=t_n+1}^{L-1}\left(\frac{p(i)}{P_{n+1}}\right)^{1-\kappa}
\tag{8}
$$

则每个分类的 Kaniadakis 熵式（3）和 Kaniadakis 熵伴生式（4）均可通过 Q_i 和 R_i 表示：

$$
S_\kappa(C_i)=-\frac{1}{2\kappa}(Q_i-R_i),\ K(C_i)=\frac{1}{2}(Q_i+R_i)
\tag{9}
$$

据此，给出两个性质。

性质 1：当 $n\ge1$ 时，多阈值 Kaniadakis 熵可以简化为：

$$
S_\kappa(C_1\cup C_2\cup\cdots\cup C_{n+1})=-\frac{1}{2\kappa}(Q_1Q_2\cdots Q_{n+1}-R_1R_2\cdots R_{n+1})
\tag{10}
$$

性质 2：当 $n\ge1$ 时，多阈值 Kaniadakis 熵伴生式可以简化为：

$$
K(C_1\cup C_2\cup\cdots\cup C_{n+1})=\frac{1}{2}(Q_1Q_2\cdots Q_{n+1}+R_1R_2\cdots R_{n+1})
\tag{11}
$$

证明：

（1）当 $n=1$ 时，灰度图像 I 被阈值 t 划分为两类 C_1 和 C_2。此时表示为：

$$
Q_1=\sum_{i=0}^{t}\left(\frac{p(i)}{P_1}\right)^{1+\kappa},\ Q_2=\sum_{i=t+1}^{L-1}\left(\frac{p(i)}{P_2}\right)^{1+\kappa}
\tag{12}
$$

$$
R_1=\sum_{i=0}^{t}\left(\frac{p(i)}{P_1}\right)^{1-\kappa},\ R_2=\sum_{i=t+1}^{L-1}\left(\frac{p(i)}{P_2}\right)^{1-\kappa}
\tag{13}
$$

则二分类 Kaniadakis 熵为：

$$
\begin{aligned}
&S_\kappa(C_1\cup C_2)\\
&=S_\kappa(C_1)K(C_2)+S_\kappa(C_2)K(C_1)\\
&=-\frac{1}{2\kappa}(Q_1-R_1)\cdot\frac{1}{2}(Q_2+R_2)-\frac{1}{2\kappa}(Q_2-R_2)\cdot\frac{1}{2}(Q_1+R_1)\\
&=-\frac{1}{4\kappa}(Q_1Q_2+Q_1R_2-Q_2R_1-R_1R_2)-\frac{1}{4\kappa}(Q_1Q_2-Q_2R_1-Q_1R_2-R_1R_2)\\
&=-\frac{1}{2\kappa}(Q_1Q_2-R_1R_2)
\end{aligned}
\tag{14}
$$

二分类 Kaniadakis 熵伴生式为：

$$
\begin{aligned}
&K(C_1\cup C_2)\\
&=K(C_1)K(C_2)+\kappa^2S_\kappa(C_1)S_\kappa(C_2)\\
&=\frac{1}{2}(Q_1+R_1)\cdot\frac{1}{2}(Q_2+R_2)+\kappa^2\cdot-\frac{1}{2\kappa}(Q_1-R_1)\cdot-\frac{1}{2\kappa}(Q_2-R_2)\\
&=\frac{1}{4}(Q_1Q_2+Q_1R_2+Q_2R_1+R_1R_2)+\frac{1}{4}(Q_1Q_2-Q_1R_2-Q_2R_1+R_1R_2)\\
&=\frac{1}{2}(Q_1Q_2+R_1R_2)
\end{aligned}
\tag{15}
$$

（2）假设 $n=k$ 时，结论成立，则：

$$
S_\kappa(C_1\cup C_2\cup\cdots\cup C_{k+1})=-\frac{1}{2\kappa}(Q_1Q_2\cdots Q_{k+1}-R_1R_2\cdots R_{k+1})
\tag{16}
$$

$$
K(C_1\cup C_2\cup\cdots\cup C_{k+1})=\frac{1}{2}(Q_1Q_2\cdots Q_{k+1}+R_1R_2\cdots R_{k+1})
\tag{17}
$$

当 $n=k+1$ 时，记 $C=C_1\cup C_2\cup\cdots\cup C_{k+1}$，则 $C_1\cup C_2\cup\cdots\cup C_{k+1}\cup C_{k+2}=C\cup C_{k+2}$。因为 C 和 C_{k+2} 为两个相互独立的子系统，则：

$$\begin{aligned}&S_\kappa(C_1\cup C_2\cup\cdots\cup C_{k+1}\cup C_{k+2})\\&=S_\kappa(C\cup C_{k+2})\\&=S_\kappa(C)K(C_{k+2})+S_\kappa(C_{k+2})K(C)\\&=\frac{1}{2}S_\kappa(C)\left(Q_{k+2}+R_{k+2}\right)-\frac{1}{2\kappa}\left(Q_{k+2}-R_{k+2}\right)K(C)\end{aligned}\tag{18}$$

$$\begin{aligned}&K(C_1\cup C_2\cup\cdots\cup C_{k+1}\cup C_{k+2})\\&=K(C\cup C_{k+2})\\&=K(C)K(C_{k+2})+\kappa^2S_\kappa(C)S_\kappa(C_{k+2})\\&=\frac{1}{2}K(C)\left(Q_{k+2}+R_{k+2}\right)-\frac{\kappa}{2}\left(Q_{k+2}-R_{k+2}\right)S_\kappa(C)\end{aligned}\tag{19}$$

应用 $n=k$ 时的结论，则式（18）和式（19）可以进一步表示为：

$$\begin{aligned}&S_\kappa(C_1\cup C_2\cup\cdots\cup C_{k+1}\cup C_{k+2})\\&=\frac{1}{2}S_\kappa(C)\left(Q_{k+2}+R_{k+2}\right)-\frac{1}{2\kappa}\left(Q_{k+2}-R_{k+2}\right)K(C)\\&=-\frac{1}{4\kappa}\left(Q_1Q_2\cdots Q_{k+1}-R_1R_2\cdots R_{k+1}\right)\left(Q_{k+2}+R_{k+2}\right)\\&\quad-\frac{1}{4\kappa}\left(Q_{k+2}-R_{k+2}\right)\left(Q_1Q_2\cdots Q_{k+1}+R_1R_2\cdots R_{k+1}\right)\\&=-\frac{1}{2\kappa}\left(Q_1Q_2\cdots Q_{k+1}Q_{k+2}-R_1R_2\cdots R_{k+1}R_{k+2}\right)\end{aligned}\tag{20}$$

$$\begin{aligned}&K(C_1\cup C_2\cup\cdots\cup C_{k+1}\cup C_{k+2})\\&=\frac{1}{2}K(C)\left(Q_{k+2}+R_{k+2}\right)-\frac{\kappa}{2}\left(Q_{k+2}-R_{k+2}\right)S_\kappa(C)\\&=\frac{1}{4}\left(Q_1Q_2\cdots Q_{k+1}+R_1R_2\cdots R_{k+1}\right)\left(Q_{k+2}+R_{k+2}\right)\\&\quad+\frac{1}{4}\left(Q_{k+2}-R_{k+2}\right)\left(Q_1Q_2\cdots Q_{k+1}-R_1R_2\cdots R_{k+1}\right)\\&=\frac{1}{2}\left(Q_1Q_2\cdots Q_{k+1}Q_{k+2}+R_1R_2\cdots R_{k+1}R_{k+2}\right)\end{aligned}\tag{21}$$

基于性质 1，多阈值 Kaniadakis 熵的最优阈值选取准则可以简化为：

$$\begin{aligned}\left\{t_1^*,t_2^*,\cdots,t_n^*\right\}&=\arg\max_{0<t_1<\cdots<t_n<L-1}\left\{S_\kappa(C_1\cup C_2\cup\cdots\cup C_{n+1})\right\}\\&=\arg\max_{0<t_1<\cdots<t_n<L-1}\left\{-\frac{1}{2\kappa}\left(Q_1Q_2\cdots Q_{n+1}-R_1R_2\cdots R_{n+1}\right)\right\}\end{aligned}\tag{22}$$

1.3 基于简化 Kaniadakis 熵的多阈值递推方法

本文提出的简化表达式使得实现多阈值 Kaniadakis 熵方法相对更容易，但其时间复杂度仍然较高。注意到多阈值 Kaniadakis 熵的简化表达式（22）的计算代价主要集中在 Q_i 和 R_i，即式（7）（8），接下来将给出基于简化 Kaniadakis 熵的多阈值递推方法。

创建 3 个长度为 L 的一维数组 S、V 和 T。其中 L 为图像的灰度等级；数组 S 用于记录 $p(i)$ 的累积概率分布，数组 V 用于记录 $[p(i)]^{1+\kappa}$ 的累积概率分布；数组 T 用于记录 $[p(i)]^{1-\kappa}$ 的累积概率分布。初始条件为 $S(0)=p(0)$，$V(0)=[p(0)]^{1+\kappa}$，$T(0)=[p(0)]^{1-\kappa}$。对于数组的后续元素，进行递推运算，表示为：

$$\begin{aligned}S(i)&=S(i-1)+p(i)\\V(i)&=V(i-1)+\left[p(i)\right]^{1+\kappa}\\T(i)&=T(i-1)+\left[p(i)\right]^{1-\kappa}\end{aligned}\tag{23}$$

式中：$p(i)$ 为灰度概率，$i=1,2,\cdots,L\text{-}1$；κ 为 Kaniadakis 熵参数。

在求出数组 S、V 和 T 之后，P_i、Q_i 和 R_i 均可以通过 S、V 和 T 表示为：

$$P_1=S(t_1),\ P_2=S(t_2)-S(t_1),\ \cdots,\ P_{n+1}=S(L-1)-S(t_n)\tag{24}$$

$$Q_1=\frac{V(t_1)}{\left[P_1\right]^{1+\kappa}},\ Q_2=\frac{V(t_2)-V(t_1)}{\left[P_2\right]^{1+\kappa}},\ \cdots,\ Q_{n+1}=\frac{V(L-1)-V(t_n)}{\left[P_{n+1}\right]^{1+\kappa}}\tag{25}$$

$$R_1=\frac{T(t_1)}{\left[P_1\right]^{1-\kappa}},\ R_2=\frac{T(t_2)-T(t_1)}{\left[P_2\right]^{1-\kappa}},\ \cdots,\ R_{n+1}=\frac{V(L-1)-V(t_n)}{\left[P_{n+1}\right]^{1-\kappa}}\tag{26}$$

在计算出 Q_i 和 R_i 后，应用式（22）来选取最优阈值。在使用简化 Kaniadakis 熵计算一幅输入图像的多个最优阈值时，选择式（22）作为目标函数。然而，在使用穷举搜索法来更新最优阈值解集时，计算 Q_i 和 R_i，会存在重复的幂运算和加法运算，这无疑会增加大量的时间成本。比较而言，在提出的基于简化 Kaniadakis 熵的多阈值递推方法中，数组 S、V 和 T 会存储已经计算的结果，当遇到重复计算时，程序会直接从数组 S、V 和 T 中调用相关数据，而不是重新进行计算，这可以有效的降低时间复杂度，并显著提升效率。

2 实验结果与讨论

首先在固定参数下，对传统 Kaniadakis 熵表达式（6）、简化 Kaniadakis 熵表达式（22）和递推简化 Kaniadakis 熵表达式的计算复杂度进行比较和分析。

假设输入图像的类别数为 n+1，传统 Kaniadakis 熵需要计算每个分类的 Kaniadakis 熵式（3）和 Kaniadakis 熵伴生式（4）以及不同分类之间的组合运算，尽管 Kaniadakis 熵和 Kaniadakis 熵伴生式仅在乘积系数和加减符号上有所不同，但计算的值是重复的，这表示在使用传统 Kaniadakis 熵时，计算量是双倍的，即 Q_i 和 R_i 部分重复计算了一次。与传统 Kaniadakis 熵中涉及的乘法运算相比，简化 Kaniadakis 熵的乘法运算次数由原来的 2^n 减少到 2 次，且 Q_i 和 R_i 部分仅需计算一次。由于总体计算代价主要集中在 Q_i 和 R_i 部分，因此简化 Kaniadakis 熵的时间成本约为传统 Kaniadakis 熵的一半，降低了约 50%。然而，无论是传统 Kaniadakis 熵还是简化 Kaniadakis 熵，时间复杂度还是相对较高。这内在的原因在于 Q_i 和 R_i 项中累积概率的重复计算，该计算中涉及多次幂运算和多次加法运算。递推简化 Kaniadakis 熵通过递推方式计算出所需的所有累积概率，在每次计算 Q_i 和 R_i 时，调用这些预先计算好的累积概率，然后完成一次幂运算和一次

除法运算，从而实现对 Q_i 和 R_i 的求解。递推简化 Kaniadakis 熵避免了重复执行多次幂运算和多次加法运算，从而大大降低了时间复杂度。

为验证本文内容的有效性，将 3 种表达式结合穷举搜索法进行对比实验。从 Berkeley 数据集中选取 6 幅不同复杂程度的图像作为测试图像，如图 1 所示，在 Berkeley 数据集中的图像编号分别为 #24 063、#310 007、#42 049、#61 060、#71 046 和 #8 049，测试图像的大小均为 481 px×321 px。对于 Kaniadakis 熵参数 κ 的取值，文献 [7] 研究表明，当 κ=0.7 时，相关的 Kaniadakis 熵阈值方法能获得更佳的阈值化结果图像，因此在实验中设置 κ=0.7。实验所用的主要软硬件参数如下：Intel Core i7-8550U 1.8 GHz CPU，16 GB DDR4 内存，Windows 10 64 位操作系统，MATLAB R2021b 64 位。

表 1 显示了 6 幅测试图像在传统 Kaniadakis 熵（traditional kaniadakis entropy, TKE）、简化 Kaniadakis 熵（simplified kaniadakis entropy, SKE）和递推简化 Kaniadakis 熵（recursive simplified kaniadakis entropy, RSKE）3 种表达式下，通过穷举搜索法求解得到的阈值、适应度值和 CPU 运行时间。

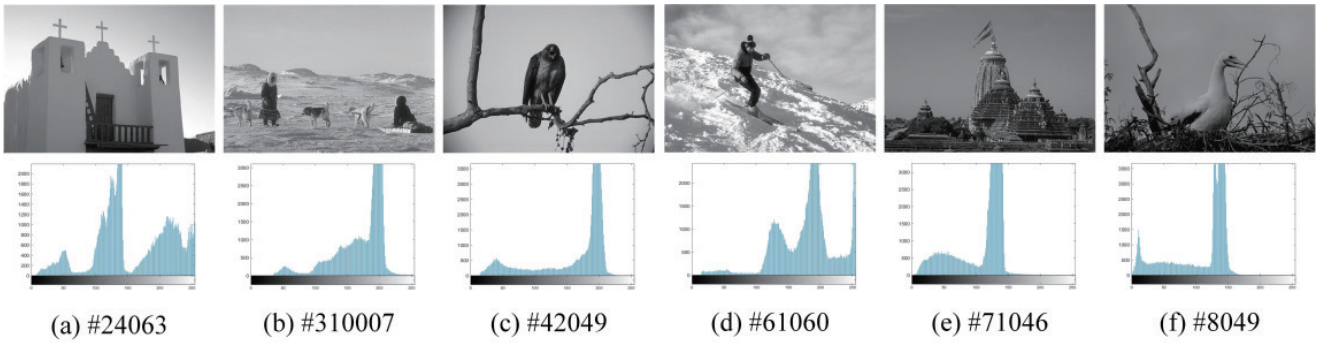

图 1 测试图像及其直方图

表 1 三种表达式下的阈值、适应度值和 CPU 运行时间

Image	n	TKE			SKE			RSKE		
		Threshold	Fitness	Runtime/s	Threshold	Fitness	Runtime/s	Threshold	Fitness	Runtime/s
#24063	1	103	5 248 061	0.140 528	103	524.806 1	0.019 783	103	524.806 1	**0.003 648**
	2	97 181	6 436.752	4.407 802	97 181	6 436.752	2.192 396	97 181	6 436.752	**0.030 824**
	3	89 144 198	6.07E+04	412.957 370	89 144 198	6.07E+04	202.410 038	89 144 198	6.07E+04	**1.550 453**
	4	58 98 144 198	5.20E+05	25 569.825 618	5 898 144 198	5.20E+05	1 269 895 3 92	5 898 144 198	5.20E+05	**161.810 304**
#310007	1	130	399.268 2	0.031 765	130	399.268 2	0.016 387	130	399.268 2	**0.001 796**
	2	123 211	4 335.529	3.944 142	123 211	4 335.529	1.966 350	123 211	4 335.529	**0.023 871**
	3	98 154 211	4.31E+04	355.551 607	98 154 211	4.31E+04	176.008 13	98 154 211	4.31E+04	**1.542 795**
	4	84 127 169 212	3.23E+05	23 182.461 478	84 127 169 212	3.23E+05	11 448.744 019	84 127 169 212	3.23E+05	**144.554 556**
#42049	1	123	411.729	0.031 266	123	411.729	0.015 450	123	411.729	**0.000 533**
	2	86 158	4 717.115	4.052 002	86 158	4 717.115	2.025 543	86 158	4 717.115	**0.022 594**
	3	67 119 171	4.14E+04	363.560 651	67 119 171	4.14E+04	181.319 403	67 119 171	4.14E+04	**1.380 933**
	4	64 114 164 212	3.17E+05	24 052.430 988	64 114 164 212	3.17E+05	12 027.498 905	64 114 164 212	3.17E+05	**164.090 639**
#61060	1	112	541.764 2	0.032 852	112	541.764 2	0.017 156	112	541.764 2	**0.001 203**
	2	107 177	6 464.957	4.271 455	107 177	6 464.957	2.120 519	107 177	6 464.957	**0.024 684**
	3	61 110 177	6.04E+04	383.035 343	61 110 177	6.04E+04	190.486 511	61 110 177	6.04E+04	**1.446 908**
	4	61 109 160 206	5.19E+05	25 375.229 72	61 109 160 206	5.19E+05	12 550.490 26	61 109 160 206	5.19E+05	**172.497 226**
#71046	1	149	417.018 5	0.034 493	149	417.018 5	0.018 717	149	417.018 5	**0.003 165**
	2	78 151	4 956.581	4.077 83	78 151	4 956.581	2.033 258	78 151	4 956.581	**0.023 855**
	3	59 111 160	4.42E+04	364.821 305	59 111 160	4.42E+04	182.213 247	59 111 160	4.42E+04	**1.721 432**
	4	58 109 146 190	3.82E+05	24 214.418 83	58 109 146 190	3.82E+05	12 085.037 92	58 109 146 190	3.82E+05	**175.136 05**
#8 049	1	85	294.7865	0.030 109	85	294.786 5	0.014 911	85	294.786 5	**0.000 289**
	2	62 121	2854.163	3.523 415	62 121	2 854.163	1.737 264	62 121	2 854.163	**0.019 860**
	3	57 111 153	2.28E+04	319.883 424	57 111 153	2.28E+04	157.296 888	57 111 153	2.28E+04	**1.854 198**
	4	43 83 124 153	1.67E+05	21 206.881 26	4 383 124 153	1.67E+05	10 525.570 79	4 383 124 153	1.67E+05	**148.514 416**

表 1 表明，对于阈值数分别为 1、2、3、4 的情况下，3 种表达式所得阈值是一致的，同时适应度值也是一致的。在 CPU 运行时间方面，使用 SKE 作为目标函数的运行时间比使用 TKE 作为目标函数的运行时间更短，总体来看，使用 SKE 作为目标函数的运行时间约为使用 TKE 作为目标函数的运行时间的 50%，使用 RSKE 作为目标函数的运行时间约为使用 TKE 作为目标函数的运行时间的 1%，显著降低了时间成本。（注：随着阈值个数的增加，穷举搜索法的计算代价也急剧增大。比如，当阈值个数为 5 时，TKE 计算时间已超过 48 h。故此处不列出 5 个及以上阈值情形下的运行时间比较结果。）

图像多阈值化质量通过峰值信噪比（peak signal-to-noise ratio, PSNR)[8]、结构相似度（structural similarity index measure, SSIM）[9] 和特征相似度（feature similarity index measure, FSIM）[10]3 个量化指标进行评价。表 2 展示了使用穷举搜索法对 6 幅测试图像进行阈值化时的 PSNR、FSIM 和 SSIM 值。

表 2 RSKE 表达式求解的 PSNR、SSIM 和 FSIM 值

Image	*n*	PSNR	SSIM	FSIM
#24 063	1	10.005 167 2	0.671 426 1	0.758 170 8
	2	15.792 825 6	0.802 594 1	0.841 277 4
	3	16.619 420 5	0.804 274 9	0.850 079 7
	4	17.798 002 6	0.828 197 5	0.859 397 9
#310 007	1	11.830 981 7	0.583 278 5	0.575 694 6
	2	11.637 384 5	0.587 430 0	0.581 138 3
	3	16.130 040 1	0.761 198 9	0.742 644 4
	4	18.947 122 7	0.823 101 9	0.832 306 3
#42 049	1	11.322 445 5	0.731 442 8	0.783 402 2
	2	16.056 414 0	0.822 081 0	0.824 967 7
	3	18.812 299 9	0.850 864 9	0.851 421 8
	4	17.914 711 1	0.852 660 8	0.852 946 1
#61 060	1	10.128 828 7	0.485 485 1	0.538 061 4
	2	16.213 118 2	0.726 025 8	0.705 531 9
	3	16.671 069 8	0.736 796 8	0.711 572 6
	4	18.562 433 8	0.826 061 6	0.805 748 0
#71 046	1	7.020 322 2	0.015 976 6	0.613 967 6
	2	14.088 698 1	0.694 862 3	0.753 038 4
	3	19.845 180 3	0.832 193 5	0.805 071 8
	4	19.667 034 8	0.836 863 9	0.812 836 7
#8 049	1	14.175 096 6	0.678 141 8	0.775 324 5
	2	20.643 969 4	0.805 285 1	0.833 517 8
	3	19.252 307 2	0.812 155 4	0.845 161 2
	4	23.203 406 9	0.866 064 6	0.888 304 4

这些结果是使用 RSKE 表达式通过穷举搜索法得到的。随着阈值数量的增加 PSNR、SSIM 和 FSIM 的数值随之上升，表明多阈值化结果图像质量得到了改善。由此可以推断，多阈值 Kaniadakis 熵方法更适合处理复杂图像。

3 结论

针对灰度图像的多阈值化问题，本文在现有单阈值 Kaniadakis 熵方法的基础上，利用 Kaniadakis 熵的广义可加性，将其拓展至多阈值情况。考虑到传统 Kaniadakis 熵在多阈值情况下表达式复杂的问题，分析推导出了 Kaniadakis 熵的简化表达式。为了更快速地确定阈值，进一步提出了基于简化 Kaniadakis 熵的递推算法。将穷举搜索法和 3 种表达式进行结合，并进行了对应的阈值确定，CPU 运行耗时等实验。实验结果表明，在不同的阈值数量下，3 种表达式在求解最优阈值时结果一致。简化 Kaniadakis 熵在求解最优阈值时的计算代价约为传统 Kaniadakis 熵的一半，而递推简化 Kaniadakis 熵显著提高了求解的效率，能够满足图像多阈值化的应用需求。在未来的工作中，将会考虑通过自适应熵参数选取策略，为不同类型的图像选择合适的熵参数，提高方法的适应性和灵活性。

参考文献：

[1] HOUSSEIN E H, ABDALKARIM N, HUSSAIN K, et al. Accurate multilevel thresholding image segmentation via oppositional snake optimization algorithm: real cases with liver disease[J]. Computers in biology and medicine, 2024, 169(2): 107922.

[2] ZHU R H, XIN B J, DENG N, et al. Fabric defect detection using ACS-based thresholding and GA-based optimal gabor filter[J]. The journal of the textile institute, 2024, 115(9): 1432-1446.

[3] SONG H H, WANG J Q, JIN L B, et al. Modified snake optimizer based multi-level thresholding for color image segmentation of agricultural diseases[J]. Expert systems with applications, 2024, 255: 124624.

[4] LEI B, FAN J L. Infrared pedestrian segmentation algorithm based on the two-dimensional kaniadakis entropy thresholding[J]. Knowledge-based systems, 2021, 225(8): 107089.

[5] SPARAVIGNA A C. Shannon, tsallis and kaniadakis entropies in bi-level image thresholding[DB/OL]. (2015-02-23)[2024-06-26].https://doi.org/10.48550/arXiv.1502.06556.

可动态扩展的机载网络服务数据通信方法

加达宇[1] 张杨阳[1] 孙豪杰[1]
JIA Dayu ZHANG Yangyang SUN Haojie

摘 要 航空电子工程委员会（AEEC）所颁布的 ARINC 834-3 标准虽然支持基于订阅的数据交换机制，然而该标准在数据源的扩展性方面存在局限，这在一定程度上制约了多数据源间的交互能力，使得在实际应用中难以满足日益增长的数据需求。针对此问题，文章提出了一种创新的、可动态扩展的机载网络服务数据通信方法。该方法基于航空数据广播协议（ADBP），采用订阅-发布机制，并通过引入中间节点，实现了多数据源与多客户端之间动态可扩展的通信。该方法有效解决了在机载环境下，不同网络域以及不同网络协议之间动态可扩展的数据订阅与发布所面临的挑战。

关键词 机载网络；飞机参数；航空数据广播协议；数据通信；数据服务

doi：10.3969/j.issn.1672-9528.2025.03.013

0 引言

2012 年，航空电子工程委员会（Airlines Electronic Engineering Committee，简称 AEEC）正式颁布了 ARINC 834-3 标准，这一标准详细规定了一整套协议体系。这些协议包括通用航空参数服务（GAPS）、简单文本航空电子协议（STAP）以及航空数据广播协议（ADBP）。特别是 ADBP 协议中规定了一种基于订阅的服务模式，通过这种模式，数据源（服务器）与客户端之间能够以 XML 对象的形式高效地交换飞机参数数据。这种机制确保了机载系统内部应用能够顺利访问特定的航空电子网络数据。

美国航空运输协会（Air Transport Association of America，简称 ATA）在其《ATA 需求 100—制造商技术数据需求》的 ATA46 章节中，对机载信息系统进行了明确的定义。机载信息系统在航空电子领域扮演着至关重要的角色，在国产大飞机 C919 和空客 A350 等先进机型上均有装备[1]，它负责信息的转换和传输等关键任务[2]。因此现有机载信息系统中，存在机载网络服务系统航电接口应用软件通过采用 ARINC 834-3 标准实现对飞机参数的高效获取[3]。在这种架构中，数据源应用充当服务器角色，而航空电子应用则作为客户端。这种服务器与客户端的模型使得机载系统内部的飞机参数获取成为可能。

然而，ARINC 834-3 标准在实际应用中也暴露出一些问题。首先，该标准并不支持数据源（服务器）的扩展性，这限制了多个客户端与多个数据源进行交互的灵活性。因为在这种架构下，服务器与客户端之间是一对多的关系，导致了数据交互的局限性。此外，机载信息系统肩负着网络安保的职责[4]，因此数据交换还要有能力处理不同网络域与不同网

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710076

[6] SPARAVIGNA A C. On the generalized additivity of Kaniadakis entropy[J]. International journal of sciences, 2015(6): 44-48.
[7] 聂方彦，李建奇，张平凤，等．复杂图像的 Kaniadakis 熵阈值分割方法 [J]. 激光与红外，2017, 47(8): 1040-1045.
[8] HORÉ A, ZIOU D. Image quality metrics: PSNR vs. SSIM[C]//2010 20th International Conference on Pattern Recognition. Piscataway: IEEE, 2010: 2366-2369.
[9] WANG Z, BOVIK A C, SHEIKH H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE transactions on image processing, 2004, 13(4): 600-612.
[10] ZHANG L, ZHANG L, MOU X Q, et al. FSIM: a feature similarity index for image quality assessment[J]. IEEE transactions on image processing, 2011,20(8): 2378-2386.

【作者简介】
何熠（1999—），男，湖北黄冈人，硕士研究生，研究方向：数字图像处理。

（收稿日期：2025-02-16）