

# MOVING OBJECT DETECTION USING DELAYED-CELLULAR NEURAL NETWORK (\*)

**Osman N. Uçan**

*İstanbul University Electrical Engineering Department*

**Lokman Ağırman**

*Türk Ticaret Bankası, Software Specialist*

**Abstract:** In this paper, we have studied moving objects in 2-D images using Delayed Cellular Neural Network (DCNN). DCNN was first introduced in 1993. It is shown that for a network whose cells are specified, complete asymptotic stability providing the delay is less than a bound which depends on only the cell parameters. Especially nowadays, only moving part of the whole image is getting more important according to the practical cases such as estimation of biomedical issues which is enlarging due to the cancer property. We have used Java language for our synthetic examples and satisfactory results were obtained.

**Keywords:** *Delayed Cellular Neural Networks, Moving Object Detection*

**Özet:** Bu makalede, 2 boyutlu görüntüler geciktirilmiş hücresel yapay sinir ağı (GHYSA) ile incelenmiştir. GHYSA ilk defa 1993 yılında tanıtılmıştır. Gecikme miktarı belirli bir değerden küçük seçilmesi halinde, asimptotik kararlılık sağlanır. Özellikle son yıllarda, görüntünün hareketli olan kısmı diğer bölgelere göre daha önemli olabilmektedir. Tıp biliminde, kanserli hücrelerin önemini GHYSA ile belirlemek çok önem taşımaktadır. Burada Java dilinde yazılım gerçekleştirilmiş ve yapay örnekler için iyi sonuçlar elde edilmiştir.

**Anahtar kelimeler:** *Geciktirilmiş Hücresel Yapay Sinir Ağ (GHYSA), Hareketli Cisim Saptama*

---

(\*) Bu çalışma İstanbul Üniversitesi araştırma Fonunca desteklenmiştir. Proje No: 1409/05052000

## INTRODUCTION

Moving object detection is among the most appealing tasks in the field of the image processing.[1] Conventional digital computation methods have same drawbacks due to their serial nature. To overcome this problem, a realatively novel class of information processing system, called Neural Networks is proposed. This new computational model is based on some aspects of neurobiology and adapted to integrated circuits. In this paper we are interested in image processing(first without considering motion and after that detection of moving objects), our attention will be focused on  $M \times N$  Cellular Neural Network, having  $M \times N$  cells arranged in  $M$  rows and  $N$  Columns in pixels.

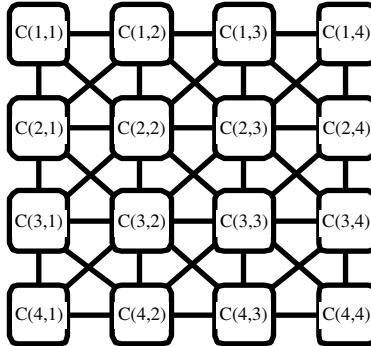
Cellular Neural Network is a parallel computing paradigm defined in  $M \times N$  space and characterized by locality of connections between processing elements (cells or pixels in our Cellular Neural Network is a parallel computing paradigm defined in  $M \times N$  space and characterized by locality of connections between processing elements (cells or pixels, in our example) [2],[3]. In this new computational model, the key features are asynchronous parallel processing, continues time dynamics and global interaction of network elements. The main difference between Cellular Neural Network and other Neural Network paradigms is the fact that information is only exchanged between neighboring neurons.

Besides Cellular Neural Network, processing of moving images requires the introduction of delay in the signals transmitted among the cells. In this paper we show how a Cellular Neural Network with delay detects moving objects in images.

### **Cellular Neural Network Definition**

A Cellular Neural Network is a system of cells defined on a normalized space. In this system, cell is the basic circuit unit containing linear and nonlinear circuit elements, which are linear capacitors, linear resistors, linear and nonlinear controlled sources and independent sources. The main idea is that connections are only allowed between adjacent cells. Any cell in a cellular neural network is connected to only its neighbor cells. But cells can affect each other indirectly. The propagation effects of the continuous time dynamics of the Cellular Neural Network provide this interaction between cells in space.

Theoretically, we can define a Cellular Neural Network of any dimension, but due the fact that we are interested in images, we will concentrate on the two-dimensional case. Fig 1 shows an example of Cellular Neural Network.



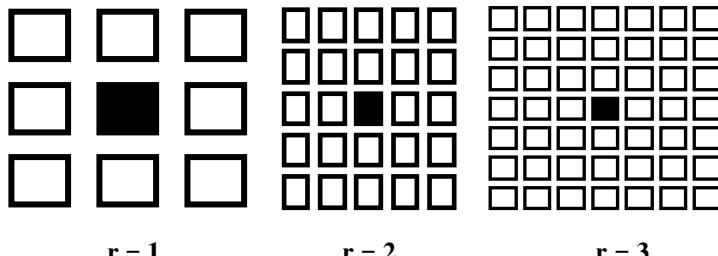
**Figure 1.** A two-dimensional cellular neural network of  $4 \times 4$  size. Squares are electrical circuit elements and represent pixels for image

The restriction of connections which are allowed to only neighboring cells requires the definition of neighborhood. Let us consider a cellular neural network with  $M \times N$  cells arranged in  $M$  rows and  $N$  columns. A cell in this space is represented with  $(i,j)$  location,  $r$ -neighborhood and denoted by  $C(i,j)$  as in Figure 1.

The  $r$ -neighborhood of a cell  $C(i,j)$ , in a cellular neural network is defined by

$$N_r(i,j) = \{C(k,l) | \max\{|k-i|, |l-j|\} \leq r, 1 \leq k \leq M; 1 \leq l \leq N\} \quad (1)$$

$r$  is a positive integer number. Following example shows  $r = 1$ ,  $r = 2$  and  $r = 3$  neighborhood of cells, respectively.



**Figure 2.** The neighborhood of cell  $C(i,j)$  for  $r = 1$ ,  $r = 2$  and  $r = 3$ , respectively

We call the  $r = 1$  neighborhood a "3 x 3 neighborhood", the  $r = 2$  neighborhood a "5 x 5 neighborhood" and so on. But  $r = 1$  is the most common neighborhood using image processing. Because if the neighborhood size were as large as image itself, we might obtain a fully connected network and in this case we shall not call such a network cellular. Generally the neighborhood shall have small size.

We call cellular neural network as a dynamical system operating in continuous or discrete time. A general form of the equations may be stated as follows:

State equation:

$$\begin{aligned}
 C \frac{dV_{xijt}(t)}{dt} = & -\frac{1}{R_x} V_{xijt}(t) \\
 & + \sum_{\substack{C(k,l) \in Nr(i,j)}} A(i,j;k,l) V_{ykl}(t) \\
 & + \sum_{\substack{C(k,l) \in Nr(i,j)}} B(i,j;k,l) V_{ukl}(t) + I, \\
 & 1 \leq k \leq M; 1 \leq l \leq N
 \end{aligned} \tag{2}$$

Output equation:

$$\begin{aligned}
 V_{yijt}(t) = & \frac{1}{2} (|V_{xij}(t) + 1| - |V_{xij}(t) - 1|) \\
 & 1 \leq k \leq M; 1 \leq l \leq N
 \end{aligned} \tag{3}$$

Input equation:

$$V_{uji} = E_{ij}, \quad 1 \leq k \leq M; 1 \leq l \leq N \tag{4}$$

A cellular neural network is completely characterized by set of equations as above, associated with the cells in the circuit. The state equation of a cellular neural network, composed by  $M \times N$  cells after having ordered the cells in some way (e.g. by rows or columns) can be rewritten in continuous time as follows:

$$\frac{dx(t)}{dt} = -x(t) + Ay(t) + Bu + I \tag{5}$$

The equation (5) then can be rewritten in discrete form as,

$$x(n+1) = -x(n) + Ay(n) + Bu + I \tag{6}$$

In Equation(6),  $x(n+1)$ ,  $y$ ,  $u$ ,  $I$  denote respectively cell state, output, input and bias.  $X$  is a local instantaneous feedback function,  $A$  and  $B$  are arrays of parameters.

### **Delay Cellular Neural Network Definition:**

Cellular neural networks with delay  $\tau$  was first introduced in 1993 [4], [5]. By assuming that the input of each cell is constant, they are described by state equations of the form:

$$\begin{aligned}
 C \frac{dV_{xijt}(t)}{dt} = & -\frac{1}{R_x} V_{xijt}(t) \\
 & + \sum_{\substack{C(k,l) \in N, (i,j)}} A(i,j;k,l) V_{ykl}(t) \\
 & + \sum_{\substack{C(k,l) \in N, (i,j)}} A^T(i,j;k,l) V_{ykl}(t - \tau)
 \end{aligned} \tag{7}$$

$$+ \sum_{C(k,l) \in N, (i,j)} B(i,j;k,l) V_{ukl}(t) + I,$$

by output equations:

$$V_{yijt}(t) = \frac{1}{2} (|V_{xij}(t) + 1| - |V_{xij}(t) - 1|) \quad (8)$$

and input equation:

$$V_{uij} = E_{ij}, = const \quad (9)$$

$N, (i,j)$  represents the neighborhood of order  $r$  of the cell  $C(i,j)$ . For delay cellular neural network, the space-invariance property is expressed by

$$A(i,j;k,l) = A(i-k, j-l)$$

$$B(i,j;k,l) = A(i-k, j-l) \quad (10)$$

$$A^T(i,j;k,l) = A^T(i-k, j-l) \quad (11)$$

State equation (7), by ordering the cells and assuming  $R_x = C = I$ , then can be rewritten in a more compact form:

$$\frac{dx(t)}{dt} = -x(t) + A_0 y(t) + A_1 y(t-\tau) + Bu + I \quad (12)$$

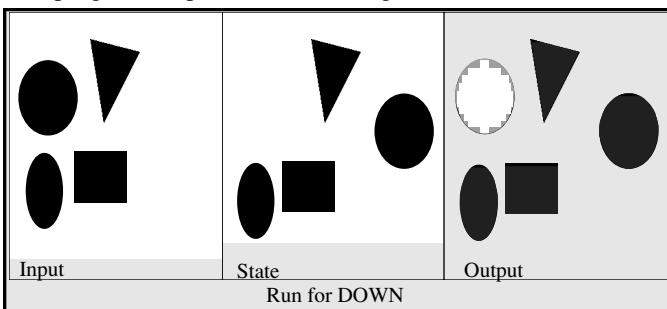
The Equation (12) can be rewritten in discrete form as,

$$x(n+1) = -x(n) + A_0 y(n) + A_1 y(n-\tau) + Bu + I \quad (13)$$

As we see in the state Equation (13) of delay cellular neural network, delay parameter  $\tau$  adds an extra  $A_1 y(n-\tau)$  (in discrete form) operand to the ordinary cellular neural networks state equation.  $A_0$  and  $A_1$  can be easily calculated from the cloning templates and delay cloning template.  $y(n-\tau)$  can also be calculated easily by using output equation with previous state of delay cellular neural network. In our Java application, we managed this using following:program.

```
...
c = getBU_Plus_I(getConvSum(pu,B), z); // B*U + I
for (int i = 0; i <= 10; i++) {
    n=i+1;
    x = AY_Plus_TY_Plus_C(getConvSum(y,A), getConvSum(y_d,T),c);
    if (i >= to) y_d = new_Y(xx[MOD(i,to)]);
    setXX (x,xx, MOD(i,to));
    y = new_Y(x);}
```

The program output is shown in Figure 3.



**Figure 3.** Delay CNN output of a synthetic example.

In the output pane we see the circle which is a moving object was detected by drawing the edge of itself after some iteration.

### **Conclusion:**

In this paper, we try to explain a cellular neural network with delay for a synthetic example. If the signals exchanged among cells are delayed, the network is called delay cellular neural network, then state and response can exhibit oscillations. In our example, we show how we can use delay cellular neural network by modifying ordinary cellular neural network and model delay cellular neural network with a Java programming language.

### **REFERENCE**

- [1] CIVALLERI, P. P., GILLI M. and PANDOLFI, L. (1993), "On Stability of Cellular Neural Networks with Delay" IEEE Trans. Circuit System.Vol. 40, No.3.
- [2] CHUA, L. O. . and YANG, L. (1988), "Cellular Neural Networks: Theory", IEEE Trans. Circuit System.Vol. 33, No.15.
- [3] LEON O.C. and LIN Y. (1988), "Cellular Neural Networks: Applications", IEEE Trans. Circuit System.Vol. 35, No.10.
- [4] GELGON, M., BOUTHEMY, P. (1996), "A Region-level Graph Labeling Approach To Motion-Based Segmentation", Institut National de Recherche en Informatique et en Automatique.
- [5] CIMAGALLI, V. and BALSI, M. (1993), "Cellular Neural Networks : A review" Proceedings of Sixth Italian Workshop on Parallel Architectures an Neural Networks. Vietri sul Mare, Italy, May 12-14, World Scientific.

# AFİNOR ALANININ YATAY LİFTİNİN TAŞINMASI HAKKINDA

**Ekrem Kadıoğlu, Muhammet Kamali, Arif A. Salimov**

*Atatürk Üniversitesi, Matematik Bölümü*

**Özet:**  $V_n$  Riemann manifoldu,  $T_q^p(V_n)$  ve  $T_{q+1}^{p-1}(V_n)$  ise bu Riemann Manifoldu üzerinde sırasıyla  $(p,q)$  ve  $(p-1, q+1)$  tipli tensör demetleri olsun.

$$f : T_q^p(V_n) \rightarrow T_{q+1}^{p-1}(V_n), (y^I = y^I(x^K); I, K = 1, 2, 3, \dots, n + n^{p+q})$$

diffeomorfizmi verilsin. Bu çalışmada,  $\nabla$  Riemann konneksiyonu ise,  ${}^H\varphi_2$  tensörünün,  $f$  altında,  ${}^H\varphi_1$  tensörünün dönüşümü olarak elde edildiği gösterildi.

**Anahtar Kelimeler:** *Tensör demeti, lift, yatay lift.*

**Abstract:** Let  $V_n$  be the Riemann manifold. Let  $T_q^p(V_n)$  and  $T_{q+1}^{p-1}(V_n)$  be the tensor bundle of type  $(p,q)$  and  $(p-1, q+1)$  over the Riemann Manifold  $V_n$ , respectively. It is given a diffeomorphism

$$f : T_q^p(V_n) \rightarrow T_{q+1}^{p-1}(V_n), (y^I = y^I(x^K); I, K = 1, 2, 3, \dots, n + n^{p+q}).$$

In this work, if  $\nabla$  is the Riemann connection, it is shown that  ${}^H\varphi_1$  is transformed to  ${}^H\varphi_2$  by the diffeomorphism  $f$ .

**Key words:** *Tensor bundle, lift, horizontal lift.*

## 1. GİRİŞ

$M_n$ ,  $C^\infty$  sınıfından bir manifold,  $A_m$  birimli komütatif ve birleşmeli bir cebir ve

$\Pi = \{ \quad \} (\quad = 1, 2, \dots, m)$ ,  $A_m$  cebirine izomorf olan  $M_n$  üzerinde poliafinor  $\Pi$  -

yapısı olsun. Burada  $\varphi$ , cebirin e baz elemanlarına karşı gelen (1,1) tipli tensörlerdir.

$\omega \in T_2^0(M_n)$ , (0,2) tipli tensör alanı  $\Pi$  - yapısına göre

$$\omega(\varphi Z_1, Z_2) = \omega(Z_1, \varphi Z_2), \quad (\quad = 1, 2, \dots, m; Z_1, Z_2 \in T_0^l(M_n)) \quad (1)$$

koşulunu sağlarsa  $\omega$  tensörüne pür tensör alanı denir. (1) koşulu  $\{\partial_i\}$  doğal çatısında

$$\omega_{mj_2} \varphi_{j_1}^m = \omega_{j_1 m} \varphi_{j_2}^m$$

şeklinde olur. Bu  $\omega_{\alpha j_1 j_2}^* = \omega_{mj_2} \varphi_{j_1}^m = \omega_{j_1 m} \varphi_{j_2}^m$  ile gösterilir.

(0,2) tipli pür tensör alanlarının oluşturduğu  $T_2^0(M_n)$  modülünde Tachibana operatörünün

$$\begin{aligned} (\varphi_\alpha \omega)(X, Z_1, Z_2) &= \varphi_\alpha(X) (\omega(Z_1, Z_2)) - X (\omega(\varphi_\alpha Z_1, Z_2)) \\ &\quad + \omega((L_{Z_1} \varphi_\alpha) X, Z_2) + \omega(Z_1, (L_{Z_2} \varphi_\alpha) X) \end{aligned} \quad (2)$$

şeklinde olduğu bilinmektedir (bkz.[3] ve [5]). Burada  $L_Z$ ,  $Z$  boyunca Lie türev operatörünü göstermektedir.

(2) operatörü doğal çatıda.

$$\Phi_k \omega_{j_1 j_2} = \varphi_k^m \partial_m \omega_{j_1 j_2} - \partial_k \omega_{\alpha j_1 j_2}^* + (\partial_{j_1} \varphi_k^r) \omega_{r j_2} + (\partial_{j_2} \varphi_k^r) \omega_{j_1 r} \quad (2')$$

olarak yazılır.

(2) veya (2') operatörünün özelliği, (0,2) tipli pür tensörü (0,3) tipli tensöre dönüştürmesidir.

İntegralenemeyen  $\Pi$  - yapısına göre

$$\partial_k \omega_{j_1 j_2} = 0$$

şartını sağlayan  $\omega_{j_1 j_2}$  pür tensör alanına almost (hemen hemen)  $A$ -holomorf tensör alanı denir. (bkz.[3] ve [5]).

## 2. Vishnevski Operatörü

Keyfi  $t \in T_q^p(M_n)$  tensör alanı için Vishnevski operatörünü

$$\begin{aligned} (\underset{\alpha}{\varphi} t)(X, Z_1, \dots, Z_q, \underset{\alpha}{\overset{1}{\cdots}}, \underset{\alpha}{\overset{p}{\cdots}}) &= (\nabla_{\alpha} t)(Z_1, \dots, Z_q, \underset{\alpha}{\overset{1}{\cdots}}, \underset{\alpha}{\overset{p}{\cdots}}) \\ &= \begin{cases} (\nabla_x t)(\underset{\alpha}{\varphi} Z_1, \dots, Z_q, \underset{\alpha}{\overset{1}{\cdots}}, \underset{\alpha}{\overset{p}{\cdots}}), & p \geq 0, q \geq 1 \\ (\nabla_x t)(Z_1, \dots, Z_q, \underset{\alpha}{\varphi}, \underset{\alpha}{\overset{1}{\cdots}}, \underset{\alpha}{\overset{p}{\cdots}}), & p \geq 1, q \geq 0 \end{cases} \end{aligned} \quad (4)$$

şeklinde yazabiliriz (bkz.[4], s. 194). Burada  $\nabla_x M_n$  üzerinde tanımlanmış afin konneksiyonunda kovaryant diferensiyel operatörü ve  $\underset{\alpha}{\varphi}$  ve  $\underset{\alpha}{\varphi}$  afinorunun eşlenik afinorudur. (4) operatörü doğal şartda

$$\sim k^{i_1 \dots i_p}_{j_1 \dots j_q} = \underset{\alpha}{\varphi} \underset{k}{\nabla} \underset{m}{\nabla} t^{i_1 \dots i_p}_{j_1 \dots j_q} - \begin{cases} \underset{\alpha}{\varphi} \underset{m}{\overset{i_1}{\vee}} \underset{k}{t} \underset{j_1 \dots j_q}{\overset{m i_2 \dots i_p}{\cdots}}, & p \geq 0, q \geq 0 \\ \underset{\alpha}{\varphi} \underset{j_1}{\overset{m}{\nabla}} \underset{k}{t} \underset{m j_2 \dots j_q}{\overset{i_1 \dots i_p}{\cdots}}, & p \geq 0, q \geq 1 \end{cases} \quad (4')$$

şeklinde yazılır.

**2.1. Lemma:** Eğer  $\Pi$ -yapısı almost integrallenebilir (yani  $\nabla_{\alpha} \underset{\alpha}{\varphi} = 0$ ,  $T(X, Y) = \nabla_X Y - \nabla_Y X - T(X, Y) = 0$ ) ise bu durumda  $T_2^0(M_n)$  pür tensör modülü üzerinde Tachibana ve Vishnevski operatörü çakışır.

**İspat:**  $T$  torsiyon tensörü formülünden

$$L_X Y = [X, Y] = \nabla_X Y - \nabla_Y X - T(X, Y) \quad (5)$$

yazılır. (1), (5) ve

$$(L_X \omega)(Y_1, Y_2) = X(\omega(Y_1, Y_2)) - \omega([X, Y_1], Y_2) - \omega(Y_1, [X, Y_2])$$

formülünü kullanarak

$$\begin{aligned} & (\underset{\alpha}{\varphi} \omega)(X, Z_1, Z_2) \\ &= \underset{\alpha}{\varphi}(X)(\omega(Z_1, Z_2)) - X(\omega(\underset{\alpha}{\varphi} Z_1, Z_2)) - \omega(\nabla_{\varphi X} Z_1 - \nabla_{Z_1} \underset{\alpha}{\varphi}(X) - T(\underset{\alpha}{\varphi} X, Z_1), Z_2) \\ &\quad + \omega(\underset{\alpha}{\varphi}(\nabla_X Z_1 - \nabla_{Z_1} X - T(X, Z_1), Z_2) - \omega(Z_1, \nabla_{\varphi X} Z_2 - \nabla_{Z_2} \underset{\alpha}{\varphi} X - T(\underset{\alpha}{\varphi} X, Z_2)) \\ &\quad + \omega(Z_1, \underset{\alpha}{\varphi}(\nabla_X Z_2 - \nabla_{Z_2} X - T(X, Z_2))) \\ &= \underset{\alpha}{\varphi}(X)(\omega(Z_1, Z_2) - X(\omega(\underset{\alpha}{\varphi} Z_1, Z_2)) - \omega(\nabla_{\varphi X} Z_1, Z_2) + \omega(\nabla_{Z_1} \underset{\alpha}{\varphi} X, Z_2) \end{aligned}$$

$$\begin{aligned}
& + \omega_{\alpha}(T(\varphi X, Z_1), Z_2) + \omega_{\alpha}(\nabla_{\varphi X} Z_1, Z_2) - \omega_{\alpha}(\varphi(\nabla_{Z_1} X), Z_2) - \omega_{\alpha}(\varphi(T(X, Z_1)), Z_2) \\
& - \omega(Z_1, \nabla_{\varphi X} Z_2) + \omega(Z_1, \nabla_{Z_2} \varphi X) + \omega(Z_1, T(\varphi X, Z_2)) + \omega_{\alpha}(\varphi Z_1, \nabla_X Z_2) \\
& - \omega(Z_1, \varphi(\nabla_{Z_2} X)) - \omega(Z_1, \varphi(T(X, Z_2)))
\end{aligned}$$

yazılır (bkz. [1], s. 37). Şimdi [1], s. 124'teki

$$\begin{aligned}
(\vee K)(X_1, \dots, X_S, X) &= (\vee_X K)(X_1, \dots, X_S) \\
&= \nabla_X(K(X_1, \dots, X_S)) - \sum_{i=1}^S K(X_i, \dots, \nabla_X X_i, \dots, X_S)
\end{aligned} \tag{6}$$

formülünü kullanarak

$$\begin{aligned}
(\nabla_{Z_1} X, Z_2) &= ((\nabla_{Z_1} X), Z_2) + (Z_1, \nabla_{Z_2} X) - \\
(Z_1, \nabla_{Z_2} X) &= ((\nabla_{Z_2} X), Z_1) + (Z_1, (\nabla_{Z_2} X))
\end{aligned} \tag{7}$$

elde edilir. (5) ve (7) eşitliklerinden

$$\begin{aligned}
(\varphi \omega)(X, Z_1, Z_2) &= X(\omega(Z_1, Z_2)) - X(\omega(-Z_1, Z_2)) - \omega(\nabla_{\varphi X} Z_1, Z_2) + \omega((\nabla_{\varphi X} Z_1) Z_2) \\
&+ \omega(Z_1, (\nabla_{\varphi X} Z_2)) + \omega(T(-X, Z_1), Z_2) - \omega(Z_1, \nabla_{\varphi X} Z_2) + \omega(Z_1, T(-X, Z_2)) \\
&+ \omega((\nabla_X Z_1), Z_2) - \omega((T(X, Z_1)), Z_2) + \omega(-Z_1, \nabla_X Z_2) - \omega(Z_1, (T(X, Z_2)))
\end{aligned}$$

bulunur. Burada yapının almost integrallenebilir (yani  $\vee = 0$ ,  $T = 0$ ) olduğunu göze alarak

$$\begin{aligned}
(\varphi \omega)(X, Z_1, Z_2) &= X(\omega(Z_1, Z_2)) - X(\omega(\varphi Z_1, Z_2)) - \omega(\nabla_{\varphi X} Z_1, Z_2) \\
&- \omega(Z_1, \nabla_{\varphi X} Z_2) + \omega(\varphi(\nabla_X Z_1), Z_2) + \omega(\varphi Z_1, \nabla_X Z_2)
\end{aligned} \tag{8}$$

yazılır. (8) eşitliğinde (6) ifadesi kullanır ve  $\vee_X f = Xf$  olduğunu dikkate alırsak

$$\begin{aligned}
(\varphi \omega)(X, Z_1, Z_2) &= (\nabla_{\varphi X} \omega)(Z_1, Z_2), \\
(\vee_X(\omega \circ \varphi))(Z_1, Z_2) &= (\nabla_{\varphi X} \omega)(Z_1, Z_2) - (\nabla_X \omega)(Z_1, Z_2) = (\varphi \omega)(X, Z_1, Z_2)
\end{aligned} \tag{9}$$

elde edilir.

Vishnevski operatörü  $g$  Riemann metrik tensörüne uygulanırsa ve  $\nabla$  konneksiyonu olarak Riemann konneksiyonu alınırsa her zaman  $(\tilde{\Phi}_{\varphi} \omega)(X, Z_1, Z_2) = 0$  olduğu görülür. Eğer  $g$  pür tensör ve Riemann konneksiyonunda  $\Pi$  - yapı almost integrallenebilir ise 2.1. Lemma ve (9) ifadesine göre  $g$  her zaman almost  $A$  - holomorf tensör olur.

### 3. Horizontal Liftin Taşınması

Şimdi kabul edelimki  $V_n$  Riemann manifoldudur.  $T_q^p(V_n)$  ve  $T_{q+l}^{p-l}(V_n)$  ise bu Riemann Manifoldu üzerinde sırasıyla  $(p,q)$  ve  $(p-1, q+1)$  tipli tensör demetleri olsun.

$$f : T_q^p(V_n) \rightarrow T_{q+l}^{p-l}(V_n), (y^I = y^I(x^K); I, K = 1, 2, 3, \dots, n+n^{p+q})$$

diffeomorfizmi

$$\left\{ \begin{array}{l} y^i = \delta_k^i x^k \\ y^{\bar{i}} = t^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} \\ = g_{im} t^{m i_2 \dots i_p}_{j_1 j_2 \dots j_q} \\ = g_{il_I} t^{l_I l_2 \dots l_p}_{k_I \dots k_q} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} \\ = g_{il_I} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} x^{\bar{k}} \end{array} \right.$$

şeklinde tanımlansın. Burada “.” indeksin indirilmesini gösterir ve  $x^{\bar{k}} = t^{l_I l_2 \dots l_p}_{k_I \dots k_q}$  alınmıştır.  $f$  dönüşümünün Jakobian matrisi

$$A = \begin{pmatrix} \frac{\partial y^I}{\partial x^K} \end{pmatrix} = \begin{pmatrix} \delta_k^i & 0 \\ 0 & g_{il_I} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} \end{pmatrix}$$

şeklindedir.  $f^{-1}$  ters dönüşümü

$$\left\{ \begin{array}{l} x^k = y^k = \delta_i^k y^i \\ x^{\bar{k}} = t^{l_I l_2 \dots l_p}_{k_I \dots k_q} \\ = g^{l_m m} t^{l_2 \dots l_p}_{m k_I \dots k_q} \\ = g^{l_I i} t^{i_2 \dots i_p}_{ij_1 \dots j_q} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} \\ = g^{l_I i} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} y^{\bar{i}} \end{array} \right.$$

olarak yazılır. Bu dönüşümün Jakobian matrisi

$$A^{-1} = \begin{pmatrix} \frac{\partial x^K}{\partial y^I} \end{pmatrix} = \begin{pmatrix} \delta_i^k & 0 \\ 0 & g^{l_I i} \delta_{i_2}^{i_2} \dots \delta_{i_p}^{i_p} \delta_{j_1}^{k_I} \dots \delta_{j_q}^{k_q} \end{pmatrix}$$

biçimindedir.

$\varphi \in T_l^I(V_n)$  afinor alanının  $T_q^P(V_n)$  ve  $T_{q+1}^{P-I}(V_n)$  tensör demetlerine, bunlar arasında  $f$  diffeomorfizminde karşılık gelen kesitler boyunca  ${}^H\varphi_1$  ve  ${}^H\varphi_2$  yatay liftleri

$${}^H\varphi_l^k = \varphi_l^k, \quad {}^H\varphi_l^{\frac{k}{l}} = 0, \quad {}^H\varphi_l^{\frac{k}{l}} = - \underset{l}{\sim} \xi_{k_l \dots k_q}^{l_l \dots l_p}$$

$${}^H\varphi_l^{\frac{k}{l}} = \begin{cases} \varphi_{s_l}^{l_l} \delta_{s_2}^{l_2} \dots \delta_{s_p}^{l_p} \delta_{k_l}^{r_l} \dots \delta_{k_q}^{r_q}, & p \geq 1, q \geq 0 \\ \delta_{s_l}^{l_l} \dots \delta_{s_p}^{l_p} \delta_{k_l}^{r_l} \dots \delta_{k_q}^{r_q}, & p \geq 0, q \geq 1 \end{cases}$$

formülü kullanılarak tanımlanır.

**3.1. Teorem :** Kabul edelim ki  ${}^H\varphi_1$  ve  ${}^H\varphi_2$ ,  $\varphi \in T_l^I(V_n)$  afinor alanının uygun olarak  $T_q^P(V_n)$  ve  $T_{q+1}^{P-I}(V_n)$  tensör demetlerine  $f$  diffeomorfizminde karşılık gelen

kesitler boyunca yatay liftleri olsun. Eğer  $\underset{\varphi}{\sim}(g) = 0$  ( $\underset{\varphi}{\sim}$  Vishnevski operatörü,  $g$  Riemann metrik tensörü) ise bu durumda  ${}^H\varphi_2$ ,  $f$  diffeomorfizmi yardımıyla,  ${}^H\varphi_1$  liftinin taşınmasıdır.

**İspat:** Gerçekten de

$${}^H\varphi_2 = \left( \underset{j}{\sim} \xi_{ij_l \dots j_q}^{i_2 \dots i_p} \varphi_i^l \delta_{j_l}^{l_l} \dots \delta_{j_q}^{l_q} \delta_{k_2}^{i_2} \dots \delta_{k_p}^{i_p} \right) \quad (10)$$

şeklindedir (bkz.[2]). Burada  $\underset{l}{\sim}$ ,

$$\underset{j}{\sim} \xi_{ij_l \dots j_q}^{i_2 \dots i_p} = \varphi_j^m \nabla_m \xi_{ij_l \dots j_q}^{i_2 \dots i_p} - \varphi_i^m \nabla_j \xi_{mj_l \dots j_q}^{i_2 \dots i_p}$$

olarak tanımlanan Vishnevski operatördür. Açık olarak

$$\underset{j}{\sim} \xi_{ij_l \dots j_q}^{i_2 \dots i_p} = \underset{j}{\sim} (g_{im} \xi_{ij_l \dots j_q}^{mi_2 \dots i_p}) = g_{im} \underset{j}{\sim} \xi_{j_l \dots j_q}^{mi_2 \dots i_p} + (\underset{j}{\sim} g_{im}) \xi_{j_l \dots j_q}^{mi_2 \dots i_p}$$

olur. Bunu (10) formülünde yerine yazıp ve  $A, A^{-1}, {}^H\varphi_1$  kullanılırsa  $\underset{j}{\sim} g_{im} = 0$

koşulu altında

$${}^H\varphi_2 = \left( g_{im} \underset{j}{\sim} \xi_{j_l \dots j_q}^{mi_2 \dots i_p} - (\underset{j}{\sim} g_{im}) \xi_{j_l \dots j_q}^{mi_2 \dots i_p} \varphi_i^l \delta_{j_l}^{l_l} \dots \delta_{j_q}^{l_q} \delta_{k_2}^{i_2} \dots \delta_{k_p}^{i_p} \right)$$

$$\begin{aligned}
&= \begin{pmatrix} \varphi_j^i & 0 \\ -g_{im} & \tilde{\xi}_{j_1 \dots j_q}^{m i_2 \dots i_p} \varphi_i^l \delta_{j_1}^{i_1} \dots \delta_{j_q}^{i_q} \delta_{k_1}^{i_2} \dots \delta_{k_p}^{i_p} \end{pmatrix} \\
&= \begin{pmatrix} \delta_k^i & 0 \\ 0 & g_{il_1} \delta_{l_2}^{i_2} \dots \delta_{l_p}^{i_p} \delta_{j_1}^{k_1} \dots \delta_{j_q}^{k_q} \end{pmatrix} \\
&\times \begin{pmatrix} \varphi_l^k \\ -\tilde{\xi}_{k_1 \dots k_q}^{l_1 \dots l_p} \varphi_{s_1}^{l_1} \delta_{s_2}^{l_2} \dots \delta_{s_p}^{l_p} \delta_{k_1}^{r_1} \dots \delta_{k_q}^{r_q} \end{pmatrix} \\
&= A^H \varphi_1 A^{-l}
\end{aligned}$$

elde edilir. Burada

$$\begin{cases} x^{\bar{l}} = t_{r_1 \dots r_q}^{s_1 \dots s_p}, & x^{\bar{k}} = t_{k_1 \dots k_q}^{l_1 \dots l_p} \\ y^{\bar{i}} = t_{ij_1 \dots j_q}^{i_2 \dots i_p}, & y^{\bar{j}} = t_{il_1 \dots l_q}^{k_1 \dots k_p} \end{cases}$$

biçimindedir.

Bu teoremden, çok önemli olan, şu sonuç çıkarılır:

**3.2 Sonuç:** Eğer  $\nabla$  Riemann konneksiyonu ise  ${}^H\varphi_2$ ,  $f$  diffeomorfizmi

yardımıyla,  ${}^H\varphi_1$  yatay liftinin taşınmasıdır.

Eğer  $g$ ,  $\Pi$ - yapısına göre pür tensör ve  $\Pi$ - yapısı  $\nabla$  Riemann konneksiyonunda almost integrallenebilir ise bu durumda (2. Başlıkta verilenlere göre)  $\tilde{{}^H\varphi}_2(g) = 0$  almost holomorfluk koşulu olacaktır.

## KAYNAKLAR

- [1]. KOBAYASHÍ, S., NOMÍZU, K. (1963), Foundation of Differential Geometry I, New York: Interscience Publishers.
- [2]. MAĞDEN, A., SALIMOV, A.A. (1997), "Afinorun tensör demete kesit boyunca tam lifti" Sakarya Üniv. Matematik Semp.
- [3]. TACHÍBANA, S. (1960), "Analytic tensor and generalization" Tohoku Math. J., 201-221.
- [4]. VISHNEVSKI, V.V., SHÍROKOV, A.P., SHURYGİN, V.V. (1985), "Spaces over algebras" Kazan Gos. Univ., Kazan, (Russian).
- [5]. YANO, K., AKO, M. (1968), "On certain operators associate with tensor fields" Kodai Math. Sem. Rep., 20(4), 414-436.